Enhanced 3D Kinematic Modeling of Wheeled Mobile Robots

Neal Seegmiller and Alonzo Kelly
Robotics Institute, Carnegie Mellon University

***Abstract*—Most fielded wheeled mobile robots (WMRs) today use basic 2D kinematic motion models in their planning, control, and estimation systems. On uneven or low traction terrain, or during aggressive maneuvers, higher fidelity models are required which account for suspension articulations, wheel slip, and liftoff. In this paper we present a simple, algorithmic method to construct 3D kinematic models for any WMR configuration. We also present a novel enhancement to predict the effects of slip on body-level motion. Extensive experimental results are presented to validate our model formulation. We show odometry improvement by calibrating to data logs and modeling 3D articulations. We also show comparable predictive accuracy of our enhanced kinematic model to a full dynamic model, at much lower computational cost.**

I. Introduction

Motion models predict changes in state given a sequence of inputs. All wheeled mobile robot (WMR) navigation systems rely critically on such models. Estimation systems rely on them for odometry, especially in the absence of GPS or other localization sensors. Model predictive planning/control systems repeatedly forward simulate these models to find and follow optimal trajectories.

Motion models vary widely in complexity. Models can be kinematic, such that body frame velocity is computed from joint rate inputs and integrated once to predict pose. Models can also be dynamic, such that acceleration is computed from force/torque inputs and integrated twice. Models can account for 3D suspension deflections, or neglect them in a planar representation. For several reasons, most fielded WMRs today make do with basic 2D kinematic models.

First, 2D kinematic models are easy to derive in closed form. Deriving the full 3D dynamics of a WMR from scratch is laborious. One could model the WMR in a general-purpose physics simulator (e.g. Open Dynamics Engine, Bullet), but most only enforce Coulomb's law of friction at contact points, whereas real wheel-terrain interaction is much more complex.

Second, 2D kinematic models are computationally cheap. Receding-horizon model predictive planning requires that models be simulated *many orders of magnitude* faster than real time, as numerous candidate trajectories are evaluated at each time step. Often, only a fraction of a processor is allocated for such planning.

Finally, 2D kinematic models are often sufficient, but only because WMRs are used conservatively. Most WMRs are restricted to driving slowly on flat ground; steep or rough patches are avoided as obstacles. Greater mobility is possible with wheeled locomotion, for example: traversing rubble at disaster sites, climbing steep mountainsides, or racing while staying just within the tire friction envelope. However, to safely push the limits of mobility, WMR navigation systems require fast, high-fidelity motion models. In this paper we present an *enhanced* formulation of 3D WMR kinematics, that provides near dynamic model fidelity at less computational cost.

II. Related Work

The earliest *general* approach to deriving WMR kinematics was published by Muir and Neuman [19]; they assign frames in a transformation graph following the Sheth-Uicker convention, cascade transforms relating wheel to body positions, then symbolically differentiate them to obtain Jacobians relating wheel to body velocities. More recently, Kim et al. [13] and Fu and Krovi [7] extended methods for parallel manipulators (transfer method, twist-based) to derive the kinematics of planar WMRs.

Some recent work concerns the 3D kinematics of articulated WMRs on uneven terrain. Tarokh and McDermott [25] present a 3D version of Muir and Neuman's transform-based approach, and demonstrate it on a rocker-bogie rover. Le Menn et al. [16] extend a reciprocal screw-based approach to derive the 3D kinematics of multi-monocycle robots. Tarokh et al. (in later papers) [26][27], Chang et al. [4], and Kelly and Seegmiller [12] use a velocity propagation-based approach, similar to the one used here. Choi and Sreenivasan [5] and Chakraborty and Ghosal [3] do not provide general methods for kinematics derivation, but simulate 3D WMR kinematics to validate mechanism designs. Lamon and Siegwart [14] provide rare experimental 3D odometry results for their articulated SOLERO platform.

Despite much progress, the literature still lacks models proven to be accurate and efficient enough for model predictive planning/control for high mobility WMRs. Importantly, none of the above publications includes a *predictive* model of wheel slip, even though real WMRs slip significantly, especially in low traction, steep, or high-speed conditions. Most just estimate slip as it occurs (e.g. [1],[23]) then rely on feedback control to correct for it ([8],[17]), but unpredicted wheel slip can cause a collision before the feedback controller can react, or entrapment [20]. There are many wheel-level models of slip in the literature but few publications incorporate these into full WMR models. Tian et al. [28] incorporate an empirical tire model into a planar dynamics model of a differential drive robot. Ishigami et al. [10] incorporate their

own terramechanics-based model for rigid wheels in loose soil into an Open Dynamics Engine model of an articulated rover. However, computation times were excessive when using this model for motion planning [11].

Most wheel-ground contact models relate slip and force, so they are not directly compatible with kinematic models which require velocity (vs. force) inputs. Our prior work [22] included slip in a simplistic kinematic model. We required that WMR kinematics be reduced to a "unicycle" model, which moves instantaneously in the tangent plane to the terrain surface (ignoring suspension deflections). Body-level slip was parametrized over an ad hoc polynomial of velocity and gravity terms.

III. Overview of Contribution

In this paper, we present a general method to construct 3D kinematic models for *any* articulated WMR configuration (Sections IV-A through IV-C). While the vector algebra basis of our approach exists in prior work, our exact algorithm is unique and straightforward to implement.

Our most significant theoretical contribution is *enhancement* of the kinematics to predict the effects of wheel slip on body frame motion (IV-D). We do so in a principled way, parametrizing slip over inertial force (due to gravity, acceleration) and dissipative forces. Calculating the inertial force also enables rollover risk assessment (IV-E).

In comparison to related work on 3D WMR kinematics, we provide an unprecedented level of experimental validation of our formulation. In all experiments we calibrate model parameters to data logs, optimizing predictive accuracy over extended horizons (IV-F). This makes the models suitable for model predictive control vs. just instantaneous feedback control.

We show improved odometry results for an articulated rover traversing obstacles by modeling 3D kinematics (V-A). Our experiments are of longer duration than in prior work [14]. We also show that our enhanced kinematic models are comparably accurate to more computationally expensive dynamic models (V-B). We know of no other experimental comparison of kinematic vs. dynamic WMR models in the literature.

By replacing the basic models used in most WMR planning/control/estimation systems with higher-fidelity models, autonomous mobility can be improved on uneven, slippery terrain and during aggressive maneuvers. Some tradeoff between fidelity and computation speed is unavoidable. Ideally, model complexity should vary per application for optimal performance. We provide a unique model of intermediate complexity suitable for many applications.

IV. Model Construction

This section explains our systematic approach to model construction and calibration. The kinematics presentation in Sections IV-A through IV-C is similar to that in [21] (on WMR dynamics); however, we provide more insight and implementation detail here.

Table I summarizes the mathematical notation used in this paper. Regarding Cartesian coordinate vectors, the variables $\vec{r}$, $\vec{v}$, and $\vec{\omega}$ will denote position, linear velocity, and angular velocity respectively. Sometimes the left superscript is omitted leaving the coordinate system unspecified or to be inferred from context; when two or more quantities are operated on they must be expressed in the same coordinates. The skew-symmetric matrix notation $[\,]_\times$ explicitly means:

$$[\vec{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \tag{1}$$

Finally, in this document, vector and matrix indexing is one-based (vs. zero-based).

TABLE I. Mathematical Notation

Symbol	Meaning
$\mathbf{u}$	(bold lowercase) column vector
$\vec{u}$	3-element Cartesian coordinate vector
$\overrightarrow{u}$	6-element Plücker coordinate (or spatial) vector
$[\vec{u}]_\times$	skew-symmetric matrix formed from the vector $\vec{u}$. Used for cross products: $\vec{a} \times \vec{b} = [\vec{a}]_\times \vec{b}$
${}^c u_a^b$	the quantity u is of frame a with respect to frame b, expressed in the coordinates associated with frame c
R_a^b	the rotation matrix of frame a relative to frame b. Used to transform coordinates: ${}^b\vec{u} = R_a^b({}^a\vec{u})$
$\mathbf{u}(i)$	the i^{th} element of vector $\mathbf{u}$, also called u_i
$\mathbf{u}(i...i+n)$	the subvector of elements i through $i+n$ of $\mathbf{u}$
$M(i,j)$	the element of M at row i, column j.
$M(i,...)$	row i of M
$M(...,j)$	column j of M

A. Specification of the Kinematic Tree

The first step in model construction is the specification of frame information for the kinematic tree. The root is the *body* frame, which has 6 degrees of freedom (DOF) with respect to the ground-fixed *world* frame. Additional frames are defined for each movable rigid body (including the wheels). Each frame is attached to a single parent frame by a revolute or prismatic joint. Unlike some prior work [26][4] our method does not require the Denavit-Hartenberg convention, which allows more intuitive frame definitions; joint axes may be aligned with any one of the x, y, or z axes. Wheel frames are attached via revolute joints about their y axes.

Contact frames are automatically appended for each wheel. They are defined with their origin at the wheel-terrain contact point. In reality contact occurs over one or more patches on the wheel surface, but we approximate to a single point. Their z-axis is aligned with the terrain normal, and their x and y axes are aligned with the longitudinal and lateral slip directions.

Frame information is stored in an ordered list such that every frame's index is greater than its parent's: $i > p(i)$. As an example, Table II lists frame information for the Zoë rover, which is used in the experiments in Section V-A. The Type column specifies whether the associated joint is revolute (R) or prismatic (P) and about the x, y, or z axis. The Act. column indicates if the joint is actuated (Y/N for yes/no). The last six columns specify the frame's position and orientation

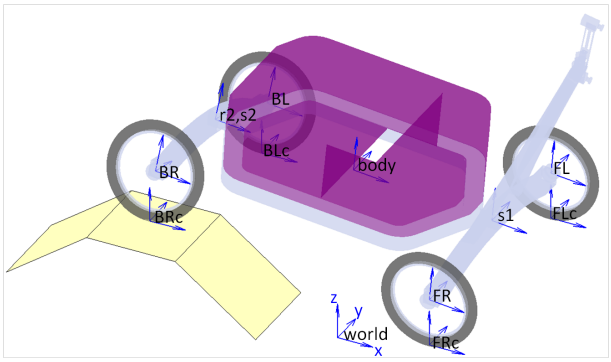


Fig. 1. Zoë rover frame diagram. FLc, FRc, BLc, BRc are the contact frames.

TABLE II. Zoë rover frame information [29]

i	Frame	Parent	Type	Act.	x	y	z	θ_x	θ_y	θ_z
1	body	world								
2	s1	body	RZ	N	l	0	0	0	0	0
3	FL	s1	RY	Y	0	w	-d	0	0	0
4	FR	s1	RY	Y	0	-w	-d	0	0	0
5	r2	body	RX	N	-l	0	0	0	0	0
6	s2	r2	RZ	N	0	0	0	0	0	0
7	BL	s2	RY	Y	0	w	-d	0	0	0
8	BR	s2	RY	Y	0	-w	-d	0	0	0

in meters: l=.955, w=.820, d=.119, wheel radius=.325, total mass = 198 kg

with respect to its parent when joint displacement is zero. The contact frames appended for each wheel would be 9-12 in the list. Fig. 1 displays the frame locations.

The vehicle state is compactly expressed by the vector $\mathbf{q}$:

$$\mathbf{q} = \begin{bmatrix} \boldsymbol{\rho}_b^w \\ \boldsymbol{\theta} \end{bmatrix}, \quad \boldsymbol{\rho}_b^w = \begin{bmatrix} \mathbf{o}_b^w \\ {}^w\vec{r}_b^{\,w} \end{bmatrix} \tag{2}$$

The first subvector is the pose of the body frame ($\boldsymbol{\rho}$) with respect to the world frame, which comprises position ($\vec{r}$) and orientation ($\mathbf{o}$, expressed using either Euler angles or a quaternion). The remaining elements are the revolute/prismatic joint displacements ($\boldsymbol{\theta}$).

Given $\mathbf{q}$ and the frame information, we can compute the homogeneous transform between each frame i and its parent frame $p(i)$:

$$T_i^{p(i)} = \begin{bmatrix} R_i^{p(i)} & {}^{p(i)}\vec{r}_i^{\,p(i)} \\ [0] & 1 \end{bmatrix} \tag{3}$$

Rotation (R) and translation ($\vec{r}$) are functions of the joint displacement (θ_i) for revolute and prismatic joints respectively. We also recursively compute the homogeneous transform of each frame with respect to the world frame: $T_i^w = T_{p(i)}^w T_i^{p(i)}$.

B. Recursive Velocity Kinematics

The symbolic differentiation of homogeneous transforms in [19][25] can be tedious and expensive. Instead we compute the velocity kinematics directly using a recursive Newton-Euler technique, originally developed for manipulators by Luh et al. [18]. Later papers on WMR kinematics have used similar velocity propagation techniques [26][4][12].

A chain of frames can be extracted from the kinematic tree for each wheel by starting at the contact frame (c) and adding successive parent frames. For example, the chain for the Zoë rover's back right wheel is:

$$w \leftarrow b \leftarrow r2 \leftarrow s2 \leftarrow BR \leftarrow BRc \tag{4}$$

Then position, linear velocity, and angular velocity can be recursively propagated as follows:

$$\begin{aligned} \vec{r}_{i+1}^{\,w} &= \vec{r}_i^{\,w} + \vec{r}_{i+1}^{\,i} \\ \vec{v}_{i+1}^{\,w} &= \vec{v}_i^{\,w} + \vec{w}_i^{\,w} \times \vec{r}_{i+1}^{\,i} + \vec{v}_{i+1}^{\,i} \\ \vec{\omega}_{i+1}^{\,w} &= \vec{\omega}_i^{\,w} + \vec{\omega}_{i+1}^{\,i} \end{aligned} \tag{5}$$

where i is the index *in the chain*, not in the frame list for the entire WMR. Note how the propagation of linear velocity depends on angular velocity.

Let 0 and n be the indices of the world and contact frames respectively. Then, based on (5), the velocity of the contact point with respect to the ground-fixed world frame can be expressed as the summation:

$$\vec{v}_n^{\,0} = \sum_{i=0}^{n-1} \vec{v}_{i+1}^{\,i} + \sum_{i=0}^{n-2} \vec{\omega}_{i+1}^{\,i} \times \vec{r}_n^{\,i+1} \tag{6}$$

This expression makes the contribution of each joint clear. $\vec{\omega}_{i+1}^{\,i}$ is nonzero for revolute joints, $\vec{v}_{i+1}^{\,i}$ is nonzero for prismatic joints, and both are nonzero for the 6-DOF "joint" between the world and body frames (0 and 1). Equation (6) can be evaluated in any coordinate system.

C. Wheel-Ground Contact Constraints

We solve for WMR motion, i.e. the time derivative of state ($\dot{\mathbf{q}}$), through constraints on the contact point velocities. To simplify the constraint calculations, we define the *joint space velocity*:

$$\dot{\mathbf{q}}' = \begin{bmatrix} {}^b\vec{v}_b^{\,w} \\ \dot{\boldsymbol{\theta}} \end{bmatrix}, \quad {}^b\vec{v}_b^{\,w} = \begin{bmatrix} {}^b\vec{\omega}_b^{\,w} \\ {}^b\vec{v}_b^{\,w} \end{bmatrix} \tag{7}$$

$\dot{\mathbf{q}}'$ is equivalent to $\dot{\mathbf{q}}$, except that the time derivative of pose is converted to the *spatial velocity* of the body frame in body coordinates. The constraint equation for a single wheel is:

$$A\dot{\mathbf{q}}' = \vec{v}_c \tag{8}$$

where $\vec{v}_c$ is short for ${}^c\vec{v}_c^{\,w}$, the velocity of the contact point with respect to the world frame in local coordinates. The Jacobian A is calculated by Algorithm 1.

In Algorithm 1, i is the index in the frame list for the entire WMR (not just a particular chain). All rotation matrices (R) and position vectors ($\vec{r}$) are obtained directly from precomputed homogeneous transforms between each frame and the world frame (T_i^w). $a(i)$ returns the axis number for joint i (x=1, y=2, z=3).

This algorithm is based on (6); it loops through the kinematic chain from contact to body frame accounting for each joint. On lines 5 and 7, the columns of the rotation matrices R_i^w are the axes along which the revolute and prismatic joint velocities are directed (i.e. the unit vectors $\hat{\omega}_i^{p(i)}$, $\hat{v}_i^{p(i)}$). Line 11 accounts for the angular velocity of the body frame, and uses the identity: $\vec{\omega} \times \vec{r} = -\vec{r} \times \vec{\omega} = [\vec{r}]_\times^T \vec{\omega}$. Line 13

Algorithm 1 Jacobian Calculation for a Single Wheel

$A \leftarrow [0]$
$c \leftarrow$ contact frame index, $i \leftarrow p(c)$
while $i > 1$ **do**
  if joint i type = revolute **then**
    $A(..., i+5) \leftarrow R_i^w(..., a(i)) \times (\vec{r}_c^{\,w} - \vec{r}_i^{\,w})$
  else if joint i type = prismatic **then**
    $A(..., i+5) \leftarrow R_i^w(..., a(i))$
  end if
  $i \leftarrow p(i)$
end while
$A(..., 1...3) \leftarrow [\vec{r}_c^{\,w} - \vec{r}_1^{\,w}]_\times^T R_1^w$
$A(..., 4...6) \leftarrow R_1^w$
$A \leftarrow (R_c^w)^T A$

converts from world to contact frame coordinates such that nonholonomic and holonomic constraints are separated.

The first two rows of (8) constrain the x and y components of the contact point velocity; they enforce nonholonomic constraints on longitudinal/lateral wheel slip. The third row of (8) constrains the z component of contact point velocity (in the terrain normal direction); it enforces the holonomic constraint that the wheel not penetrate or lift off of the terrain surface *to first-order*. Violation of the holonomic constraint due to numerical drift can be eliminated by using Baumgarte's stabilization method (Yun and Sarkar [32]). Set $\vec{v}_c = [0\ 0\ -\Delta z/\tau]^T$, where Δz is the contact height error and τ is a time constant larger than the integration time step.

Constraint equations for all n_w wheels are stacked to obtain an equation with $3n_w$ rows:

$$A\dot{\mathbf{q}}' = \mathbf{v}_c \tag{9}$$

Hereafter let A refer to the stacked matrix. Note that additional constraints can be appended, for example, to constrain two suspension joint angles to be symmetric. The degrees of freedom in $\dot{\mathbf{q}}'$ can be partitioned into *free* and *fixed*:

$$A(..., \text{free})\dot{\mathbf{q}}'(\text{free}) = \mathbf{v}_c + A(..., \text{fixed})\dot{\mathbf{q}}'(\text{fixed}) \tag{10}$$

In a simulation/prediction context, the fixed DOFs include actuated joint rates, and the free DOFs include passive joint rates and the body frame velocity. Most WMR configurations are overconstrained, and so (10) must be solved using the pseudoinverse:

$$\dot{\mathbf{q}}'(\text{free}) = A(..., \text{free})^+ \left(\mathbf{v}_c + A(..., \text{fixed})\dot{\mathbf{q}}'(\text{fixed})\right) \tag{11}$$

To integrate WMR motion (i.e. update $\mathbf{q}$), $\dot{\mathbf{q}}'$ must first be converted to $\dot{\mathbf{q}}$, which requires the conversion of body frame spatial velocity into the time derivative of pose:

$$\frac{d}{dt}\begin{bmatrix} \mathbf{o}_b^w \\ {}^w\vec{r}_b^{\,w} \end{bmatrix} = \begin{bmatrix} \Omega(\mathbf{o}_b^w) & [0] \\ [0] & R_b^w \end{bmatrix} \begin{bmatrix} {}^b\vec{\omega}_b^{\,w} \\ {}^b\vec{v}_b^{\,w} \end{bmatrix} \tag{12}$$

where Ω converts angular velocity to Euler angle or quaternion rates.

D. Body-Level Slip Prediction

Related work on WMR kinematics seldom accounts for nonzero wheel slip, even though real WMRs can slip significantly. The obvious way to include slip in our model is to set the x,y components of $\vec{v}_c$ in (8) to nonzero values, but to what values? Canonical wheel-ground contact models relate force to slip, but kinematic models provide no information about force at the wheels. Besides being unknown, wheel slip values may be unnecessary; in many applications only the motion of the vehicle body matters. Accordingly, we enhance the kinematics to predict *body-level* slip.

Empirically, wheel slip ratio (s) and angle (α) are linearly proportional to normalized longitudinal and lateral force, up to a limit [2]. *Normalized* means the ratio with respect to normal force. Let v_x,v_y denote the x,y components of $\vec{v}_c$, or the slip velocities. Let f_x,f_y,f_z denote the components of contact force ($\vec{f}_c$). Finally let V_x,V_y denote the velocities of the wheel frame with respect to the ground. All variables are in contact frame coordinates (Figure 2). Then:

$$s = \frac{r\omega - V_x}{V_x} = \frac{-v_x}{V_x} \propto \frac{f_x}{f_z}, \qquad v_x \propto \frac{f_x}{f_z}V_x \tag{13}$$

$$\alpha = \operatorname{atan}\left(\frac{v_y}{V_x}\right) \approx \frac{v_y}{V_x} \propto \frac{f_y}{f_z}, \qquad v_y \propto \frac{f_y}{f_z}V_x \tag{14}$$

Up to a limit, slip velocity is linearly proportional to the product of normalized force and longitudinal velocity. We extend this wheel-level observation to model slip at the body-level.

Forces at the wheels are reactions to inertial effects (due to gravity, acceleration) and dissipative effects (such as rolling resistance). We compute inertial force at the body frame using spatial vector algebra:

$$\vec{f}_b = I_b^c \vec{g} - \vec{v}_b \times I_b^c \vec{v}_b \tag{15}$$

Due to space limitations this equation cannot be fully explained here, but it is in [6][21]. $\vec{g}$ is the acceleration of gravity, and $\vec{v}_b$ is the spatial velocity of the body frame (obtained from the least-squares slip solution for $\dot{\mathbf{q}}'$). I_b^c is the composite spatial inertia of the vehicle. Our kinematic model is first-order, but (15) still accounts for the Coriolis and centripetal components of acceleration.

The magnitudes of $\vec{v}_b$ and $\vec{f}_b$ depend on where the user chooses to define the body frame, but physical slip does not. For consistency, we transform these values to a new frame, defined with its origin at the center of gravity (cg):

$$\vec{v}_{cg} = (X_b^{cg})\vec{v}_b, \quad \vec{f}_{cg} = (X_b^{cg})^{-T}\vec{f}_b \tag{16}$$

The cg position is required to compute the Plücker transform X_b^{cg}) and the spatial inertia I_b^c. Like all spatial vectors, $\vec{v}_{cg}$ and $\vec{f}_{cg}$ have linear and angular components:

$$\vec{v}_{cg} = \begin{bmatrix} \vec{\omega}_{cg} \\ \vec{v}_{cg} \end{bmatrix}, \quad \vec{f}_{cg} = \begin{bmatrix} \vec{\tau}_{cg} = \mathbf{0} \\ \vec{f}_{cg} \end{bmatrix} \tag{17}$$

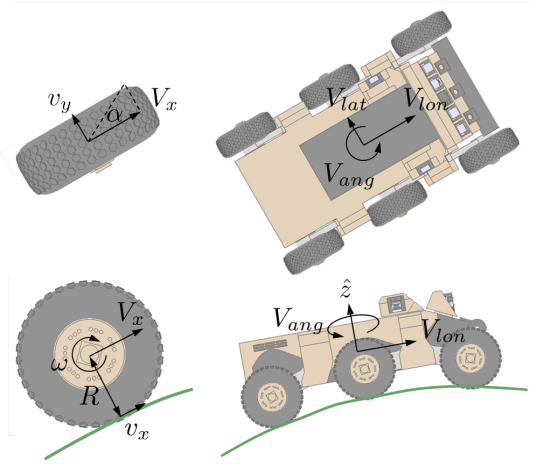


Fig. 2. Diagram of wheel and body-level slip variables

We also define unit vectors in the longitudinal ($\hat{lon}$), lateral ($\hat{lat}$), and angular slip ($\hat{ang}$) directions. These are commonly aligned with the body frame x, y, and z axis but do not have to be (See Figure 2). Body slip velocity is parametrized over these values as follows:

$$\vec{v}_{cg,s} = \left(p_1 \frac{f_{lon}}{f_z} V_{lon} + p_2 V_{lon}\right) \hat{lon} + \left(p_3 \frac{f_{lat}}{f_z} V_{lon}\right) \hat{lat} \quad (18)$$

$$\vec{\omega}_{cg,s} = \left(p_4 \frac{f_{lat}}{f_z} V_{lon} + p_5 V_{lon} + p_6 V_{ang}\right) \hat{ang} \quad (19)$$

where the following are abbreviations for dot products with unit vectors:

$$f_{lon} = \vec{f}_{cg} \cdot \hat{lon} \qquad V_{lon} = \vec{v}_{cg} \cdot \hat{lon}$$
$$f_{lat} = \vec{f}_{cg} \cdot \hat{lat} \qquad V_{lat} = \vec{v}_{cg} \cdot \hat{lat}$$
$$f_z = \vec{f}_{cg} \cdot \hat{z}$$

where $\hat{z}$ is the normal force direction.

The p_1 and p_3 terms in (18) are direct analogues of the wheel-level relationship between slip and force in (13) and (14). The p_2 term accounts for rolling resistance, which prevents the vehicle from coasting at constant speed. Longitudinal rolling resistance is proportional to normal force [30], so the f_z terms cancel out.

While only longitudinal and lateral slip are considered at the wheel level, angular slip must also be considered at the body level. The p_4 term in (19) accounts for oversteer/understeer behavior, which is proportional to lateral acceleration [30]. The p_5 term accounts for left/right asymmetry in rolling resistance, for example due to a flat tire. The p_6 term accounts for angular slip as a result of skidding. Skid-steer vehicles can't turn without dragging some wheels along the ground. The pseudoinverse solution assumes isotropic friction, but real friction forces may be anisotropic.

The slip velocity at the cg ($\vec{v}_{cg,s}$) must be transformed back to the body frame and added to $\vec{v}_b$ in the least-squares slip solution for $\dot{\mathbf{q}}'$, prior to integration. Alternatively, body level slip can be converted to nonzero x,y components of the contact point velocity $\vec{v}_c$ in (8). The z component is left unchanged so that holonomic constraints are unaffected. After updating all slip velocities in $\mathbf{v}_c$ with nonzero values, $\dot{\mathbf{q}}'$ is re-solved for using the pseudoinverse per (11).

This model does not capture some second-order inertial effects, such as skidding with locked brakes, but it works under the near steady-state conditions of normal operation. As seen in Section V-B, this can be sufficient to predict motion as accurately as a full dynamic model.

E. Liftoff Prediction

Just as wheel slip is significant in steep and high-speed conditions, so is the risk of liftoff and rollover. Carrying heavy loads can raise a vehicle's cg, making it particularly susceptible. While not the focus of this paper, we note that the calculation of inertial force in (15) enables rollover risk assessment.

According to [24] and others since, if the vector $\vec{f}_{cg}$, originating at the cg, intersects the ground plane outside of the support polgyon (whose vertices are the wheel-ground contact points) then liftoff is imminent. While our first-order model cannot account for all types of acceleration, it does account for centripetal acceleration and gravity, major contributors to sideways rollover.

F. Parameter Calibration

A more detailed model is only more accurate if its parameters are correctly identified. In Section V we demonstrate the calibration of enhanced 3D kinematic model to data logs. Any observable combination of kinematic, sensor, and slip parameters may be calibrated.

WMR motion models are typically expressed as a nonlinear differential equation of the form:

$$\dot{\mathbf{q}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) \quad (20)$$

where $\mathbf{u}$ comprises the inputs (i.e. actuated/sensed joint rates) and $\mathbf{p}$ the parameters to be identified.

The simplest calibration approach is to minimize *velocity residuals*, or the differences between measured and predicted velocities. Instead, we calibrate to *pose residuals*, in which the predictions require forward simulation of the model (or solution of the differential equation) over the interval $(t - t_0)$:

$$\mathbf{r} = \mathbf{q}_{meas}(t) - \mathbf{q}_{pred}(t) \quad (21)$$

$$\mathbf{q}_{pred}(t) = \mathbf{q}_{meas}(t_0) + \int_{t_0}^{t} \mathbf{f}(\mathbf{q}(\tau), \mathbf{u}(\tau), \mathbf{p}) d\tau \quad (22)$$

The subscripts *meas* and *pred* denote measured and predicted values respectively. The residual can be a subset of $\mathbf{r}$ in (21), such as only the (x,y,yaw) components of the body frame pose.

This calibration method is an extension of our prior work on Integrated Prediction Error Minimization (IPEM) [22]. It optimizes for simulation accuracy over extended horizons (vs. instantaneous accuracy), which is desirable for model predictive planning/control and estimation applications. Importantly, this method also enables characterization of stochasticity, such that the *uncertainty* of model predictions can be quantified (see [22] for details).

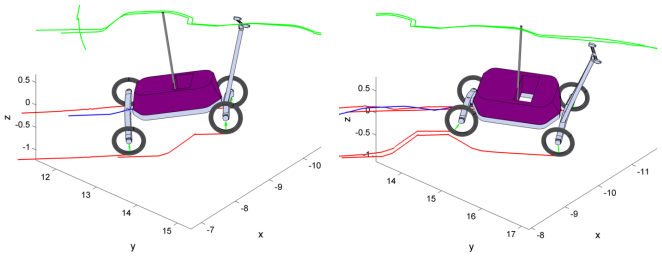


Fig. 3. Animated 3D odometry as the rover's right wheels traverse a ramp. The blue and red lines trace the estimated paths of the body and contact frames (using only wheel speed, axle angle, and attitude inputs). The green line is the path of the pole-mounted prism tracked by the total station.

Fig. 4. Photograph of the test rover and four ramp obstacles at the Robot City test location.

V. Experimental Results

This paper is distinguished from prior work on WMR kinematics not only by our novel enhancement to account for wheel slip, but also by extensive experimental results. Our results show how calibrating to data logs over extended horizons, modeling 3D articulations, and accounting for wheel slip improve predictive accuracy.

A. 3D Odometry

In this section we show improved odometry results by calibrating to data logs and modeling 3D articulations. We conducted experiments using the Zoë rover [29]. Zoë's front and rear axles are passively steered by varying the four independently actuated wheel speeds. The rear axle's roll degree of freedom keeps all wheels in contact on uneven terrain (Fig. 1, Table II).

The wheel speed, axle angle, and attitude (i.e. roll, pitch) inputs were sensed by encoders, potentiometers, and inclinometers respectively. Ground truth position was measured using a Leica robotic total station. The prism tracked by the total station had to be mounted high above the rover's camera mast to maintain line of sight (Fig. 4). Ground truth orientation was measured using an IMU.

In each test, the rover executed two laps around a course. Speeds were kept below 0.5 m/s to facilitate tracking. During each lap the rover traversed two ramp obstacles, 41cm high by 179cm long with a 61cm long flat middle section, as seen in Fig. 1. Though commanded to drive straight over the ramps, each perturbed the rover slightly off course. When a wheel climbs the ramp on one side, the body sways to the other affecting both the yaw and steer angles. The proportional steering controller, having no 3D kinematic model, drives the sensed steer angles back to zero which actually causes a turn. Fig. 3 depicts the estimated trajectory of the rover over one ramp.

Fig. 5 presents paths and estimation errors for three 3D odometry tests. The first two tests took place on a paved lot, while the second took place on a larger lot with paved and grassy sections (Fig. 4). In the second test the rover traversed four total ramps per lap; the additional two ramps were 42cm high by 134cm long with no flat section.

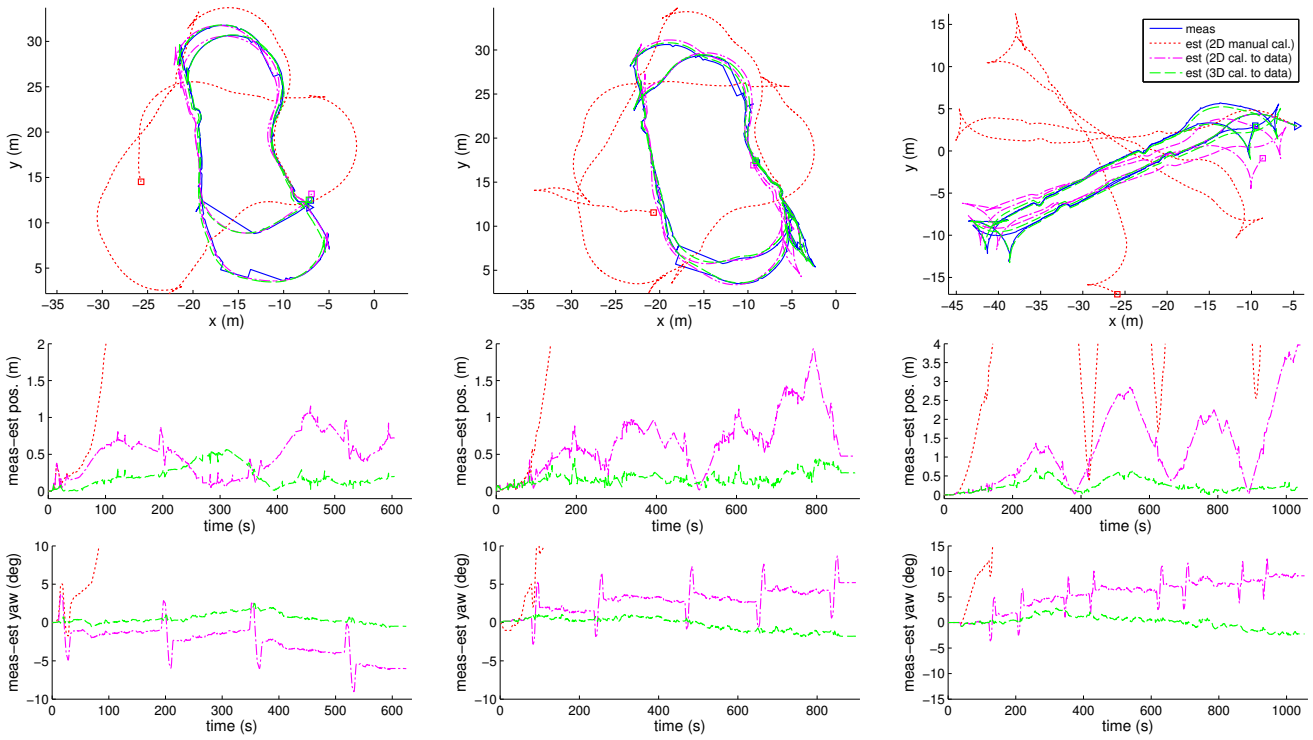
Results are shown for three different kinematic models. For the manually calibrated model, wheel radius was measured with a meterstick, and parameters for the linear conversion from potentiometer voltages to steer angles (slope, intercepts) were calculated by manually setting the steer angles to $+20°$ and $-20°$ using precision machined blocks. In the other models, wheel radius and pot. conversion parameters were calibrated to data logs as explained in Section IV-F. The 2D models assume axle roll and all contact angles are zero. Contact angles (γ) specify where along the wheel's circumference contact occurs; $\gamma = 0$ corresponds to the bottom with respect to the unrotated wheel frame.

Calibration to data logs results in a major accuracy improvement over manual calibration, and accounting for 3D articulations results in a significant additional improvement. When using 2D kinematics, spikes in yaw error occur when traversing ramps, but these are completely eliminated by using 3D kinematics. 3D odometry is very accurate: less than .25m and 2.3° error after traveling up to 201.5m on uneven terrain, without using a gyroscope to aid yaw estimation.

Interestingly, we noticed that 3D odometry assuming $\gamma = 0$ is no better than 2D odometry. In our experiment, we used prior knowledge of the obstacle geometry to help determine γ. Iagnemma and Dubowsky [9] propose a method to estimate contact angles from wheel speeds and pitch rate, but it assumes no wheel slip. Slip is unavoidable for our rover configuration on uneven terrain. In general, contact angle estimation may require additional sensing such as an instrumented wheel [15][31]. Given a perception-generated terrain elevation map, contact points my be solved for by geometric collision detection. This is a necessity for simulation/planning contexts, but is also possible in estimation. Unfortunately perception was unavailable for our tests.

B. Enhanced Kinematic vs. Dynamic Model Comparison

In this section we show improved predictive accuracy of a kinematic motion model by accounting for slip as presented in Section IV-D. Whereas the Zoë rover used in Section V-A was designed for minimal slip, the skidsteer LandTamer platform used here cannot turn without slipping (Fig. 6). We obtained vehicle pose ground truth via Real Time Kinematic (RTK) GPS and a tactical grade IMU.



(a) CMU paved lot, right wheels on ramps (b) CMU paved lot, left wheels on ramps (c) Robot City, left wheels on ramps

Fig. 5. *Top row*: Paths of the test rover: (solid line) measured by the total station, (short-dash) estimated using a manually calibrated 2D kinematic model, (dash-dot) estimated using a 2D kinematic model calibrated to the data log, (long-dash) estimated using a 3D kinematic model calibrated to the data log, (triangle) path start, (square) path end. *Middle row*: Position error (Euclidean distance) vs. time for trajectories estimated using the above models. *Bottom row*: Yaw error vs. time. For the 2D models, spikes in yaw error occur when traversing ramps.

Fig. 6. The LandTamer (made by PFM Manufacturing Inc.) is a six-wheeled skidsteer platform.

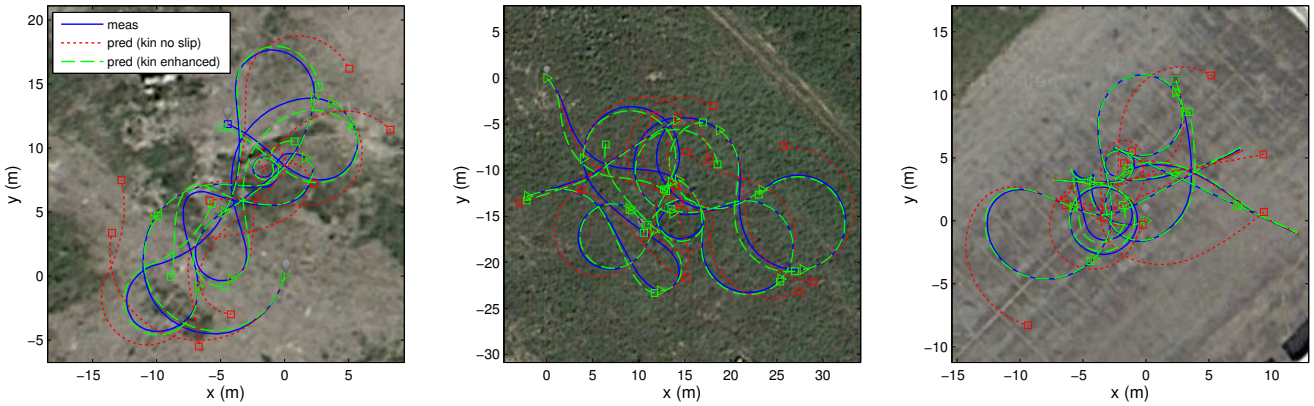
Although these results were generated by post-processing data logs, we claim they are indicative of *predictive* performance, as wheel speeds measured by encoders can be replaced with planned commands. Likewise, attitude measured by an IMU can instead be predicted by enforcing holonomic contact constraints with respect to a terrain elevation map.

We drove the LandTamer at speeds up to 2.5 m/s and 0.5 rad/s for 150 seconds on three terrain types: dirt, grass, and a paved parking lot. Fig. 7 shows the vehicle paths overlaid on aerial images. We evaluated three model types: a no slip kinematic model, an enhanced kinematic model, and a full dynamic model.

The "no slip" (or non-enhanced) kinematic model does account for slip to some extent by minimizing it in a least-squares sense, per equation (11). Even this is more accurate than common technique reducing skid-steer kinematics to differential drive kinematics (which greatly overestimates yaw rate). The full dynamic model is constructed as presented in [21], and uses the Pacejka "magic formula" tire model [2]. Parameters of the enhanced kinematic model (i.e. coefficients in (18) and (19)) and the dynamic model (tire model coefficients, cg location) were calibrated to *separate* training data logs collected on the same day as the logs used for evaluation.

Table III displays position and yaw prediction errors for all three motion models, for all three terrain types, for horizons of 10 and 20m. Table IV displays the percent reduction in errors for the enhanced kinematic and dynamic models relative to the non-enhanced kinematic model. On pavement, error reductions of 83-96% were achieved for the 10m horizon. Smaller but still significant reductions of 64-76% were acheived on the rough, inhomogeneous dirt and grass.

While the predictive accuracy of the enhanced kinematic model and dynamic model are comparable (kin.+ did slightly better on pavement, dyn. did slightly better on dirt and grass), the enhanced kinematic model is computationally cheaper. Relative to the non-enhanced kinematic model, forward simulation of the dynamic model took 4.5-5.5$\times$ more computation time (depending on inputs), whereas the enhanced kinematic model took only 1.3$\times$ more. The enhanced kinematic model also required fewer parameters.



(a) dirt (b) grass (c) paved parking lot

Fig. 7. Paths of the LandTamer overlaid on aerial images, 20m prediction horizon: (solid) measured by RTK GPS, (short-dash) predicted by a non-enhanced kinematic model which assumes no wheel slip, (long-dash) predicted by an enhanced kinematic model, (triangle) path start, (square) path end

TABLE III. LandTamer position & yaw prediction errors

		Dirt (24 trials)		Grass (36)		Pavement (33)	
Dist.	Model	pos.	yaw	pos.	yaw	pos.	yaw
10m	kin.	1.717	0.291	1.465	0.277	1.774	0.493
	kin.+	0.480	0.104	0.445	0.096	0.146	0.021
	dyn.	0.420	0.076	0.379	0.081	0.294	0.039
20m	kin.	4.354	0.402	3.972	0.447	4.295	0.827
	kin.+	1.222	0.192	1.273	0.176	0.249	0.034
	dyn.	0.979	0.150	0.978	0.103	0.445	0.061

(pos.) mean position error (Euclidean distance) in meters, (yaw) mean absolute yaw error in radians, (kin.) non-enhanced kinematic model, (kin.+) enhanced kinematic model, (dyn.) dynamic model

TABLE IV. Percent reduction in prediction errors*

		Dirt		Grass		Pavement	
Dist.	Model	pos.	yaw	pos.	yaw	pos.	yaw
10m	kin.+	72%	64%	70%	65%	92%	96%
	dyn.	76%	74%	74%	71%	83%	92%
20m	kin.+	72%	52%	68%	61%	94%	96%
	dyn.	78%	63%	75%	77%	90%	93%

*relative to the non-enhanced kinematic LandTamer model.

VI. Conclusions and Future Work

We have presented a straightforward, algorithmic method to construct 3D kinematic models for any articulated WMR configuration, and a novel enhancement to predict the effects of wheel slip on body-level motion. In addition, we provided unprecedented experimental validation of our model formulation. We showed major improvements in odometry accuracy by calibrating to data logs and modeling 3D articulations. We also showed that our enhanced kinematic models can provide comparable accuracy to full dynamic models at a fraction of the computation cost.

We plan to further analyze the tradeoff between fidelity and computation speed in future work. We aim to derive formulas for how performance varies with model complexity and with control/terrain inputs. Given this, we could select the optimal level of model complexity per application. We could even dynamically switch between models within an application based on situation (e.g. soil type, steepness, speed).

We have implemented a modular software library that supports dynamic, kinematic, and enhanced kinematic WMR simulations. We are in the process of converting this software from MATLAB to C++ so that we can fairly benchmark computation speeds against alternatives (e.g. Open Dynamics Engine, CarSim). In preliminary tests, the dynamics and enhanced kinematics of the Zoë rover can be simulated 1200× and 5200× faster than real time on a 3.4 GHz processor, respectively. Regardless of speed, our method provides functionality/flexibility that is lacking in alternative resources.

While modeling WMR motion is not a new problem, we have shown that major improvements in model accuracy and speed can still be achieved. Further research on this topic is warranted, as model improvements directly benefit the planning, control, and estimation systems that rely on them.

Acknowledgments

This research was made with U.S. Government support by the Army Research Laboratory (W911NF-10-2-0016) and by the National Science Foundation Graduate Research Fellowship (0946825). Our colleagues Forrest Rogers-Marcovitz, Ammar Husain, and David Kohanbash assisted with data collection.

References

[1] J. Bob Balaram. Kinematic observers for articulated rovers. In *International Conference on Robotics and Automation*. IEEE, 2000.

[2] Raymond M. Brach and R. Matthew Brach. Tire models for vehicle dynamic simulation and accident reconstruction. *SAE Technical Paper*, (2009-01-0102), 2009.

[3] Nilanjan Chakraborty and Ashitava Ghosal. Kinematics of wheeled mobile robots on uneven terrain. *Mechanism and Machine Theory*, 39(12):1273–1287, December 2004.

[4] Yong Chang, Shugen Ma, Hongguang Wang, and Dalong Tan. A kinematic modeling method for a wheeled mobile robot. In *International Conference on Mechatronics and Automation*. IEEE, 2009.

[5] B. J. Choi and S. V. Sreenivasan. Gross motion characteristics of articulated mobile robots with pure rolling capability on smooth uneven surfaces. *Transactions on Robotics*, 15(2):340–343, 1999.

[6] Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, Boston/Dordrecht/Lancaster, 1987.

[7] Qiushi Fu and Venkat Krovi. Articulated wheeled robots: Exploiting reconfigurability and redundancy. In *Dynamic Systems and Control Conference*. ASME, 2008.

[8] Daniel M. Helmick, Stergios I. Roumeliotis, Yang Cheng, Daniel S. Clouse, Max Bajracharya, and Larry H. Matthies. Slip-compensated path following for planetary exploration rovers. *Advanced Robotics*, 20(11):1257–1280, November 2006.

[9] Karl Iagnemma and Steven Dubowsky. Vehicle wheel-ground contact angle estimation: with application to mobile robot traction control. In *International Symposium on Advances in Robot Kinematics*, 2000.

[10] Genya Ishigami, Akiko Miwa, Keiji Nagatani, and Kazuya Yoshida. Terramechanics-based model for steering maneuver of planetary exploration rovers on loose soil. *Journal of Field Robotics*, 24(3):233–250, 2007.

[11] Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida. Path planning and evaluation for planetary rovers based on dynamic mobility index. In *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 2011.

[12] A. Kelly and N. Seegmiller. A vector algebra formulation of mobile robot velocity kinematics. In *Field and Service Robotics*, July 2012.

[13] Wheekuk Kim, Byung-Ju Yi, and Dong Jin Lim. Kinematic modeling of mobile robots by transfer method of augmented generalized coordinates. *Journal of Robotic Systems*, 21(6):301–322, 2004.

[14] P. Lamon and R. Siegwart. 3D position tracking in challenging terrain. *International Journal of Robotics Research*, 26(2):167–186, February 2007.

[15] Pierre Lamon. *3D-Position Tracking and Control for All-Terrain Robots*, volume 43 of *Tracts in Advanced Robotics*. Springer, 2008.

[16] Frederic Le Menn, Philippe Bidaud, and F. Ben Amar. Generic differential kinematic modeling of articulated multi-monocycle mobile robots. In *International Conference on Robotics and Automation*. IEEE, 2006.

[17] E. Lucet, C. Grand, D. Sallé, and P. Bidaud. Dynamic sliding mode control of a four-wheel skid-steering vehicle in presence of sliding. In *RoManSy*, Tokyo, Japan, 2008.

[18] John Y. S. Luh, Michael W. Walker, and Richard P. C. Paul. On-line computational scheme for mechanical manipulators. *J. Dyn. Sys., Meas., Control*, 102(2):69–76, June 1980.

[19] Patrick F. Muir and Charles P. Neuman. Kinematic modeling of wheeled mobile robots. *Tech. Report CMU-RI-TR-86-12*, June 1986.

[20] Katharine Sanderson. Mars rover Spirit (2003-10). *Nature*, 463(7281):600, February 2010.

[21] N. Seegmiller and A. Kelly. Modular dynamic simulation of wheeled mobile robots. In *Field and Service Robotics*, December 2013.

[22] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly. Vehicle model identification using integrated prediction error minimization. *International Journal of Robotics Research*, 32(8):912–931, July 2013.

[23] Xiaojing Song, Zibin Song, Lakmal D Seneviratne, and Kaspar Althoefer. Optical flow-based slip and velocity estimation technique for unmanned skid-steered vehicles. In *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 2008.

[24] S. Sugano, Q. Huang, and I. Kato. Stability criteria in controlling mobile robotic systems. In *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, July 1993.

[25] M. Tarokh and G. J. McDermott. Kinematics modeling and analyses of articulated rovers. *Transactions on Robotics*, 21(4):539–553, August 2005.

[26] Mahmoud Tarokh and Gregory McDermott. A systematic approach to kinematics modeling of high mobility wheeled rovers. In *International Conference on Robotics and Automation*. IEEE, 2007.

[27] Mahmoud Tarokh, Huy Dang Ho, and Antonios Bouloubasis. Systematic kinematics analysis and balance control of high mobility rovers over rough terrain. *Robotics and Autonomous Systems*, 61(1):13–24, January 2013.

[28] Yu Tian, Naim Sidek, and Nilanjan Sarkar. Modeling and control of a nonholonomic wheeled mobile robot with wheel slip dynamics. In *Symposium on Computational Intelligence in Control and Automation*. IEEE, 2009.

[29] Michael Wagner, Stuart Heys, David Wettergreen, James Teza, Dimitrios Apostolopoulos, and George Kantor. Design and control of a passively steered, dual axle vehicle. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005.

[30] J. Y. Wong. *Theory of Ground Vehicles*. Wiley, New York, third edition, 2001.

[31] He Xu, Xing Liu, Hu Fu, Bagus Bhirawa Putra, and Long He. Visual contact angle estimation and traction control for mobile robot in rough-terrain. *Journal of Intelligent & Robotic Systems*, July 2013.

[32] Xiaoping Yun and Nilanjan Sarkar. Unified formulation of robotic systems with holonomic and nonholonomic constraints. *Transactions on Robotics*, 14(4):640–650, 1998.