# Control of a Passively Steered Rover using 3-D Kinematics

Neal Seegmiller and David Wettergreen

***Abstract*— This paper describes and evaluates a 3-D kinematic controller for passively-steered rovers. Passively-steered rovers have no steering motors, but rely on differential wheel velocities to change the axle steer angles. This passive steering design is reliable and efficient but more challenging to control than powered steering designs, especially when driving on rough terrain. A controller based on 2-D kinematics fails to accurately maintain the desired trajectory when traversing obstacles. The presented 3-D kinematic controller uses inertial and proprioceptive sensing to modify commanded steer angles and wheel velocities, greatly improving steering accuracy. Validation in simulation and physical experiments is presented. These results are significant because they establish the viability of the passive-steering configuration for precise navigation.**

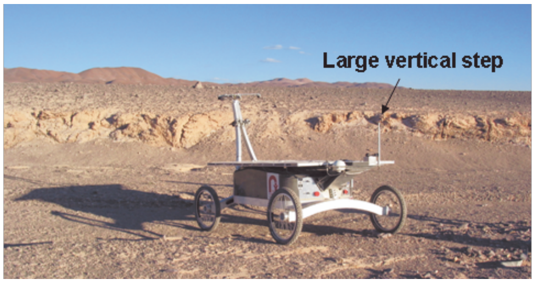

Fig. 1. Photograph of Zoë approaching a large vertical step in a dried river bed in the Atacama desert [2]. The presented 3-D kinematic controller is designed to traverse obstacles such as this, that are too large to circumvent.

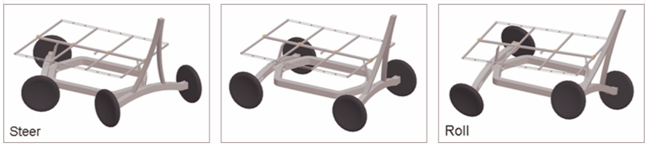

Fig. 2. Zoë's mechanical design [2]. Each axle can passively articulate in two degrees of freedom, yaw (or steer) and roll.

## I. Introduction

The objective of the presented controller is to enable precise steering of passively-steered vehicles on rough terrain. The mobility, efficiency, and endurance of the passive-steering configuration were first demonstrated by the Zoë rover, which surveyed the distribution of microscopic life in Chile's Atacama Desert [1] (see Fig. 1). Zoë autonomously traversed over 250 km of rough desert terrain. The presented controller enables passively-steered rovers like Zoë to not only traverse rough terrain, but to follow exact trajectories while doing so.

Zoë has four independently driven wheels on two passively articulated axles. Each axle is free to rotate in two degrees-of-freedom: yaw and roll (see Fig. 2). Zoë is not skid-steered; steering is achieved through closed-loop control of the axle yaw angles. Specifically, each wheel velocity must be coordinated not only to propel the rover forward but also to control the steer angle of the passively-articulated axles. The axle roll motion allows the wheels to follow the contours of uneven terrain. A linkage between the axles averages the roll angles with respect to the robot body, providing smooth motion to the payload.

There are key differences between the passive-steering design and other common rover configurations such as the rocker-bogie design (used on the Mars Exploration Rovers [3]), actively steered four-wheel designs (such as [4]), and skid-steered designs. First, passively-steered rovers require fewer parts (linkages, motors, wheel modules) than the rocker-bogie or actively-steered four-wheel designs. Eliminating parts reduces mass which improves energy efficiency. In addition, fewer parts means mechanical simplicity which can improve reliability. Second, passive-steering minimizes wheel slip, especially compared to skid-steering. When wheels slip energy is expended on soil work instead of moving the vehicle, and so the elimination of wheel slip contributes to energy efficiency. In addition, minimized wheel slip enables far more accurate position estimation. Tradeoffs with competing designs include the inability to turn in place and a lack of redundancy. The passive-steering design is better suited for long-range exploration than for short-range missions requiring tight maneuverability.

A challenge (but not disadvantage) of passive-steering is added complexity to the controller. In prior work, a basic controller was developed based on a 2-D kinematic model that assumes the rover drives in a plane [2]. This report describes a controller based on a 3-D kinematic model, capable of accurately steering the rover on uneven terrain.

Prior research has shown the usefulness of 3-D kinematic models for other rover designs. Multiple 3-D kinematic models have been developed for the rocker-bogie design [5] [6]. Lamon and Siegwart demonstrated how a 3-D model improves position estimation on rough terrain, and is the basis for a controller that minimizes wheel slip [7]. Howard demonstrated the necessity of modeling 3-D kinematic constraints for motion planning on complex terrain [8].

Manuscript received March 28, 2011.

N. Seegmiller is a Ph.D. candidate at the Carnegie Mellon University Robotics Institute `nseegmiller@cmu.edu`

D. Wettergreen is a Research Professor at the Carnegie Mellon University Robotics Institute `dsw@ri.cmu.edu`

TABLE I

TABLE OF COMMON VARIABLES

| Name | Description | Name | Description |
|---|---|---|---|
| $\theta$ | Axle steer angle | $B$ | Axle Width (1.64 m)[a] |
| $\phi$ | Axle roll angle | $R_w$ | Wheel Radius (.325 m) |
| $\gamma$ | Robot roll angle | $L$ | Robot Length, distance between steering joints (1.91 m) |
| $\beta$ | Robot pitch angle | $D$ | Axle Drop, height of roll/steer joints above wheel centers (.119 m) |
| $\alpha$ | Robot yaw angle | $\tilde{\theta}^{comp}$ | Commanded steer angle (axle roll compensation) |
| $\psi$ | Robot heading angle | $s^{comp}$ | Feedforward velocity scalar (axle roll compensation) |

[a] dimensions in parentheses are for the Zoë rover

## II. THE 2-D AND 3-D CONTROL EQUATIONS

The basic 2-D controller assumes that all wheels drive in a plane. Below is the control equation for a *single* axle:

$$\begin{bmatrix} \tilde{v}_l \\ \tilde{v}_r \end{bmatrix} = \begin{bmatrix} \frac{1}{cos(\tilde{\theta})} & \frac{-B}{2} \\ \frac{1}{cos(\tilde{\theta})} & \frac{B}{2} \end{bmatrix} \begin{bmatrix} \tilde{V} \\ \tilde{\omega} \end{bmatrix} + K_p \begin{bmatrix} -(\tilde{\theta} - \theta) \\ \tilde{\theta} - \theta \end{bmatrix} \tag{1}$$

Where:

$$\tilde{\theta}_{front} = atan\left(\frac{L/2}{\tilde{R}}\right), \quad \tilde{\omega} = \frac{\tilde{V}}{\tilde{R}} \tag{2}$$

In all equations, the tilde symbol $\tilde{}$ is used to denote commanded variables. In (1) $\tilde{v}_l$ and $\tilde{v}_r$ are the commanded left and right wheel velocities for a single axle. In (2) $\tilde{R}$ denotes the commanded turn radius (positive for a counter-clockwise arc, negative for clockwise). Symmetric front and rear steer angles ($\tilde{\theta}$) are commanded to allow the tightest possible turn radii for the steering range of motion. $K_p$ is the proportional gain. Refer to Table I for the meaning of common variables and dimensions not identified in the text.

Driving over obstacles causes non-zero axle roll angles, breaking the planar assumption. This affects steering in two ways:

First, the yaw of the robot changes with respect to heading ($|\alpha - \psi| > 0$, see Fig. 3). Steering angles are measured with respect to the robot frame, and so the commanded steer angles must be changed in order to align the axles with the desired heading. $\tilde{\theta}^{comp}$ in (3) denotes the new commanded steer angle (*comp* is an abbreviation for axle roll compensation).

Second, the 3-D wheel velocity vector is no longer parallel to the ground plane. When projected onto the ground plane, wheels that are traversing obstacles appear to move slower, which if uncorrected causes deviations in steer angle. This is fixed by scaling the feedforward velocity by $s^{comp}$.

The new 3-D control equation with the added roll compensation terms ($\tilde{\theta}^{comp}$ and $s^{comp}$) is:

$$\begin{bmatrix} \tilde{v}_l \\ \tilde{v}_r \end{bmatrix} = \begin{bmatrix} s_l^{comp} & 0 \\ 0 & s_r^{comp} \end{bmatrix} \begin{bmatrix} \frac{1}{cos(\tilde{\theta}^{comp})} & \frac{-B}{2} \\ \frac{1}{cos(\tilde{\theta}^{comp})} & \frac{B}{2} \end{bmatrix} \begin{bmatrix} \tilde{V} \\ \tilde{\omega} \end{bmatrix} + K_p \begin{bmatrix} -(\tilde{\theta}^{comp} - \theta) \\ \tilde{\theta}^{comp} - \theta \end{bmatrix} \tag{3}$$

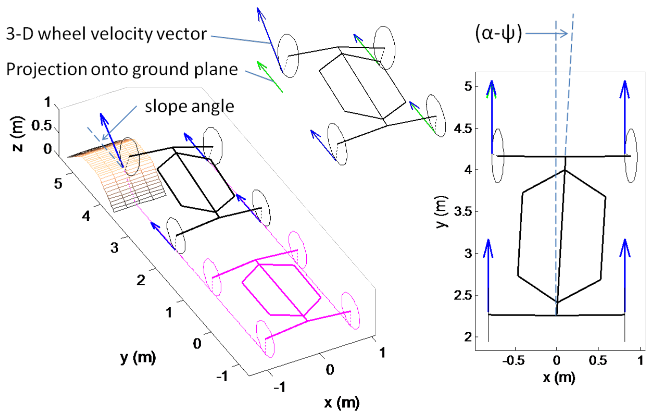

Fig. 3. Depiction of the difference between yaw and heading ($\alpha - \psi$) as Zoë traverses an obstacle. Note also that the 3-D front-left wheel velocity vector is not parallel to the ground plane.

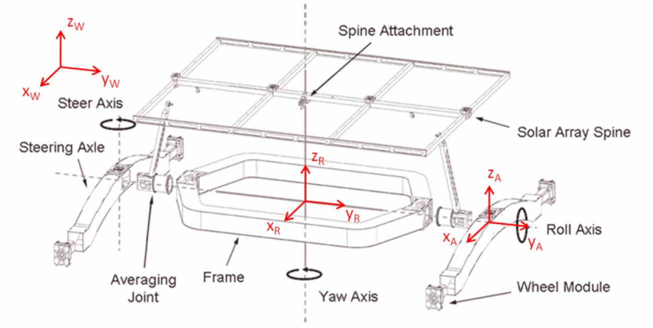

Fig. 4. Depiction of the axle (A), robot (R), and world (W) coordinate frames for a passively steered rover.

The effect of the 3-D controller is to make the *projection* of the 3-D wheel velocity vector onto the ground plane match the wheel velocity vector commanded by the 2-D controller in both direction and magnitude: $\text{proj}\left(\underline{\tilde{v}}^{3D}\right) = \underline{\tilde{v}}^{2D}$

## III. 3-D KINEMATIC MODEL

A 3-D kinematic model of the passive steering design is required to calculate the axle roll compensation terms described in Section II. The following equations are used to transform wheel/terrain contact point coordinates from the axle frame to the world frame. These transformations were derived by following Denavit-Hartenberg conventions [9], then combining intermediate transforms into two intuitive homogeneous transforms: the transform from axle to robot frame (${}^{R}_{A}T$) and the transform from robot to world frame (${}^{W}_{R}T$). See Fig. 4 for the location of coordinate frames.

$${}^{W}P = {}^{W}_{R}T\,{}^{R}_{A}T\,{}^{A}P \tag{4}$$

In (4), ${}^{W}P$ denotes the coordinates of the wheel/terrain contact point in the world frame. The coordinates of each wheel/terrain contact point in the axle frame (${}^{A}P$) are fixed:

$${}^{A}P = \begin{bmatrix} (\pm B/2) & 0 & (-D - R_w) & 1 \end{bmatrix}^T \tag{5}$$

In (5) $B/2$ is positive for right wheels and negative for left wheels.

$$
{}^{R}_{A}T(\theta,\phi)=\begin{bmatrix} c\phi c\theta & -c\phi s\theta & s\phi & 0 \\ s\theta & c\theta & 0 & \pm L/2 \\ -s\phi c\theta & s\phi s\theta & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}
$$

${}^{R}_{A}T$ in (6) is the transform from axle to robot frame. $L/2$ is positive for front axle wheels and negative for rear wheels. Sine and cosine are abbreviated by $s$ and $c$. Because of the roll averaging mechanism, the front and rear axle roll angles are always symmetric ($\phi_{front} = -\phi_{rear}$).

$$
{}^{W}_{R}T(\gamma,\beta,\alpha,x,y,z)=
$$

$$
\begin{bmatrix} c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta & c\alpha s\gamma + s\alpha s\beta c\gamma & x \\ s\alpha c\gamma + c\alpha s\beta s\gamma & c\alpha c\beta & s\alpha s\gamma - c\alpha s\beta c\gamma & y \\ -c\beta s\gamma & s\beta & c\beta c\gamma & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}
$$

${}^{W}_{R}T$ in (7) is the transform from robot to world frame. The Euler angle convention used is ${}^{W}_{R}T = Trans(x,y,z)Rot_z(\alpha)Rot_x(\beta)Rot_y(\gamma)$.

The control equation requires that a reference ground plane be defined. The commanded velocity and arc radius are defined with respect to this plane. In this paper the "world" frame is aligned with the ground plane, meaning the world $x$ and $y$ axes span the ground plane.

## IV. CALCULATION OF AXLE ROLL COMPENSATION TERMS

This section explains the calculation of both roll compensation terms: $\tilde{\theta}^{comp}$ and $s^{comp}$. These terms modify both the commanded steer angle and wheel velocity when driving on complex terrain so that the passively-steered rover maintains the desired trajectory.

### A. Commanded Steer Angle

To compute a unique solution for the compensated commanded steer angles ($\tilde{\theta}^{comp}$) one additional constraint is required; In the desired steering configuration the vectors from rear-left to front-left and from rear-right to front-right wheel/terrain contact points are aligned with the vehicle heading. (see Fig. 5)

$\tilde{\theta}^{comp}$ for both the front and rear axle are computed iteratively using Newton's method. This algorithm has been verified to converge within 4 iterations for all values of $\theta$ and $\phi$ within Zoë's range of motion limits. The calculation of $\tilde{\theta}^{comp}$ is summarized in Algorithm 1. All underlined variables in Algorithm 1 are $2{\times}1$ vectors, with the first element corresponding to the front axle and the second element corresponding to the rear. A detailed explanation each major step is provided below.

*Line 4*. The locations of *all four* wheel/terrain contact points are calculated in world coordinates using the transformations in (6) and (7). Note that yaw ($\alpha$) and translation are *not* included in the transformation.

*Line 5*. In Algorithm 1, the desired heading direction is the "world" $y$ axis. As stated above, we will constrain vectors from rear to front wheel/terrain contact points to align with heading. To do so we rotate about the world $z$ axis by $(\alpha-\psi)$, the difference between the yaw and heading angle, which is estimated as follows:

$$
\alpha-\psi=\frac{\Delta h_{left}(\alpha-\psi)_{right}+\Delta h_{right}(\alpha-\psi)_{left}}{\Delta h_{left}+\Delta h_{right}} \tag{8}
$$

Where:

$$
(\alpha-\psi)_{left/right}=atan2\big(P_{front}(y)-P_{rear}(y),\; P_{front}(x)-P_{rear}(x)\big) \tag{9}
$$

$$
\Delta h_{left/right}=|P_{front}(z)-P_{rear}(z)| \tag{10}
$$

where $P$ is an abbreviation for ${}^{W}P$. $(\alpha-\psi)_{left}$ and $(\alpha-\psi)_{right}$ are the angles of the rear-to-front vectors in the ground plane (for left and right wheels respectively), and $\alpha-\psi$ is a weighted average of these two angles. More weight is given to the side (left or right) with a smaller height difference between the wheels (i.e. the side driving on flatter ground). If $\Delta h_{left}+\Delta h_{right}$ is near zero then $\alpha-\psi$ is instead equal to $\big((\alpha-\psi)_{left}+(\alpha-\psi)_{right})\big)/2$.

*Line 6*. The axle heading error ($\underline{error}$) is then calculated for both the front and rear axles. Error is defined as the difference between the nominal commanded steer angle on flat terrain ($\tilde{\theta}$) and the axle heading ($\psi_{axle}$), measured by projecting the left and right wheel/terrain contact points onto the ground plane:

$$
\psi_{axle}=atan2\big(P_{right}(y)-P_{left}(y),\; P_{right}(x)-P_{left}(x)\big) \tag{11}
$$

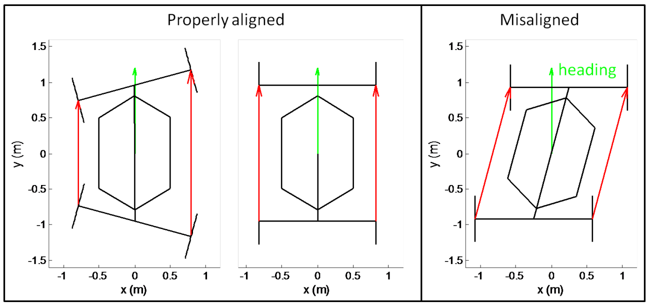

Fig. 5. Both the left and right side vectors from rear to front wheels (red) should be aligned with heading (green).

**Algorithm 1** Calculation of roll compensated commanded steer angle, $\tilde{\theta}^{comp}$

```
1: $\underline{\tilde{\theta}}^{comp} \leftarrow \underline{\tilde{\theta}}$
2: $\underline{error} \leftarrow threshold$
3: while $\underline{error} \geq threshold$ do
4:     ${}^{W}P \leftarrow {}^{W}_{R}T(\gamma,\beta,0,0,0,0)\,{}^{R}_{A}T(\tilde{\theta}^{comp},\phi)\,{}^{A}P$ (all wheels)
5:     ${}^{W}P \leftarrow R_z(\alpha-\psi)\,{}^{W}P$   (all wheels)
6:     $\underline{error} \leftarrow \underline{\tilde{\theta}} - \underline{\psi}^{axle}$
7:     $\underline{\tilde{\theta}}^{comp} \leftarrow \underline{\tilde{\theta}}^{comp} - J^{-1}\underline{error}$
8: end while
```

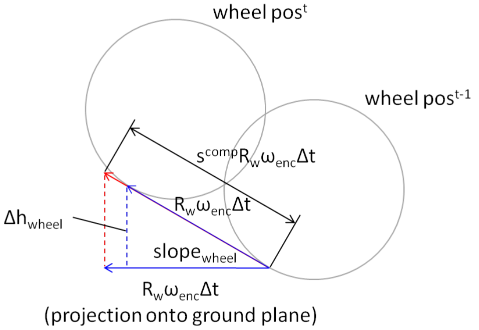
wheel $\text{pos}^t$
wheel $\text{pos}^{t-1}$
$s^{comp} R_w \omega_{enc} \Delta t$
$R_w \omega_{enc} \Delta t$
$\Delta h_{wheel}$
$slope_{wheel}$
$R_w \omega_{enc} \Delta t$
(projection onto ground plane)

Fig. 6. Diagram of the calculation of $s^{comp}$. The circles depict the wheel position at two adjacent timesteps as the wheel climbs a slope. The ground plane is horizontal in the figure.

*Line 7*. According to Newton's method, the update to $\underline{\tilde{\theta}}^{comp}$ is calculated using the 2×2 Jacobian of the $\underline{error}$ vector with respect to $\underline{\tilde{\theta}}^{comp}$. This Jacobian ($J$) is calculated numerically.

### B. *Feedforward Velocity Scaling*

The feedforward velocity scaling terms ($s^{comp}$) scale the commanded wheel velocities when driving on rough terrain. Fig. 6 illustrates the geometry of the calculation. As an intermediate step the terrain slope angle is calculated for each wheel ($slope_{wheel}$ in Fig. 6). After scaling by $s^{comp}$ the projection of the 3-D wheel velocity vector onto the ground plane is equivalent in magnitude to the 2-D velocity vector on flat terrain.

The calculation of $s^{comp}$ for a single wheel is summarized in Algorithm 2. A detailed explanation of each step is provided below.

**Algorithm 2** Calculation of feedforward velocity scalar $s^{comp}$ *for a single wheel*

```
1: loop
2:     $^{W}P^{t} \leftarrow {}^{W}_{R}T(\gamma,\beta,0,0,0,0)\,{}^{R}_{A}T(\theta,\phi)\,{}^{A}P^{t}$
3:     $^{W}P^{t} \leftarrow {}^{W}P^{t} + \Delta h_{robot}$
4:     $\Delta h_{wheel} \leftarrow {}^{W}P^{t}(z) - {}^{W}P^{t-1}(z)$
5:     $slope_{wheel} = asin(\Delta h_{wheel}/(R_w \omega_{enc} \Delta t))$
6:     $s^{comp} \leftarrow 1/cos(slope_{wheel})$
7:     $^{W}P^{t-1} \leftarrow {}^{W}P^{t}$
8: end loop
```

*Line 2*. The location of the wheel/terrain contact point is calculated in world coordinates. Note that yaw ($\alpha$) and translation are not included in the transformation.

*Line 3*. The offset $\Delta h_{robot}$ is the difference in height of the robot frame origin between the current and previous timestep (measured with respect to the ground plane). This value could be difficult to measure in practice; an inertial measurement is prone to drift and the update rate for visual motion estimation may be too slow (as observed by Lamon and Siegwart [7]).

*Line 4*. $\Delta h_{wheel}$ is the difference in height of the wheel/terrain contact point between the current and previous timestep. This noisy differential value may require smoothing.

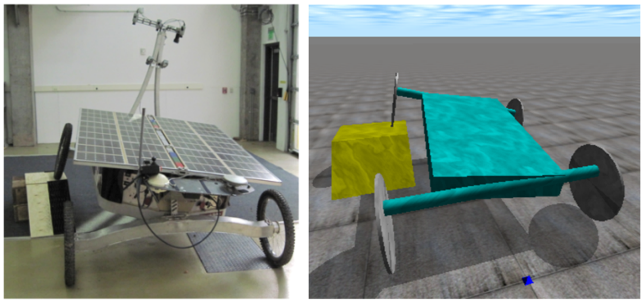

Fig. 7. *Left*. Photo of Zoë as its left wheels traverse a ramp obstacle. *Right*. Graphical display of the Open Dynamics Engine simulation.

*Line 5*. A slope angle is calculated for each wheel using the differential change in height and the encoder measurement of wheel velocity. (Fig. 6)

*Line 6*. Finally, this slope angle is used to calculate the feedforward velocity scale factor ($s^{comp}$) for each wheel. Note that feedforward velocity increases with the steepness of the slope. The $s^{comp}$ term is capped at 1.5 to prevent wheel velocities from exceeding hardware limits.

With properly calculated $\tilde{\theta}^{comp}$ and $s^{comp}$ terms, the controller will command the ideal steering angles and wheel velocities to accurately steer the rover on rough terrain.

## V. DYNAMIC SIMULATION RESULTS

Evaluation of the controller was performed on an Open Dynamics Engine (ODE) simulation of the Zoë rover. Care was taken to accurately model Zoë's mass, dimensions, joint configuration (roll averaging), and controller (frequency, gains, and acceleration limits). The friction coefficient and force-dependent-slip values were tuned to match the low-slip wheel/terrain contact observed experimentally. Two tests are presented; in both tests Zoë is commanded to drive straight but the controller is equally effective when an arc is commanded. Both $\tilde{\theta}^{comp}$ and $s^{comp}$ were tested.

### A. *Ramp Obstacle Test*

In this test Zoë's left wheels drive over a ramp obstacle at 0.15 m/s. The dimensions of the ramp are 1.71 m length × 0.41 m height with a 36° slope angle (see Fig. 7, *Right*). With roll compensation off, the controller behaves as when Zoë is driving on flat terrain and forces both the front and rear steer angles to zero, causing a final heading error of 2.4°. With roll compensation on, the controller commands non-zero steering angles and the error is eliminated (see the dashed black lines in Fig. 8). Note that after the obstacle is traversed and Zoë resumes driving on the flat ground plane, the yaw and heading angles are aligned ($\alpha = \psi$).

In Fig. 9, note that the perceived slope angles ($slope_{wheel}$ in Algorithm 2) are nearly correct when driving up the ramp, but are overestimated by 5-10° when driving down the ramp. This is because insufficient traction causes the wheels to slide down the ramp. Wheel slip for the front left wheel

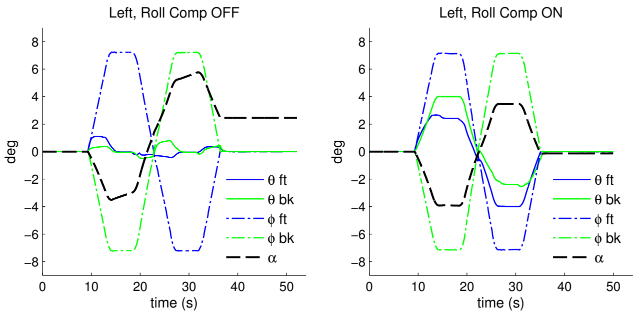

Fig. 8. Simulation results. Plot of yaw ($\alpha$), steering ($\theta$), and axle roll angles ($\phi$) as Zoë's left wheels drive over a ramp obstacle. *Left.* With no roll compensation, a final heading error of 2.4° occurs (see the dashed black line). *Right.* With roll compensation enabled the heading error is eliminated.

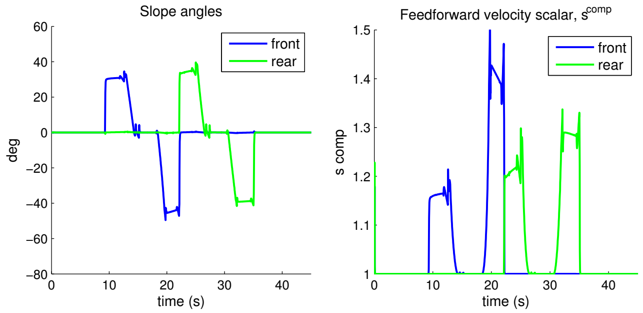

Fig. 9. Simulation results. Plot of slope angle and feedforward velocity scalar ($s^{comp}$) as Zoë's left wheels drive over a ramp obstacle with roll compensation enabled.

is approximately -17% when descending the ramp, where wheel slip is defined as:

$$slip = 1 - \frac{V}{R_w \omega_{enc}} \times 100\% \qquad (12)$$

where $V$ is the linear velocity of the wheel center, $R_w$ is the wheel radius, and $\omega_{enc}$ is the wheel angular velocity (as measured by the encoder) [10]. Fortunately, the PID controller maintains the desired steer angle despite the minor disturbance caused by wheel slip.

### *B. Random Rough Terrain Test*

In this test Zoë drives on random rough terrain (RMS height = 0.15 m, correlation length = 0.75 m, see Fig. 10) at a speed of 0.5 m/s. With roll compensation enabled, a final heading error of -4.4° is eliminated (see Fig. 11).

## VI. EXPERIMENTAL RESULTS

The controller was also evaluated in physical experiments on the Zoë rover. This section presents experimental results replicating the first simulation test in which Zoë's left wheels are driven over a ramp obstacle at 0.15 m/s (see photo in Fig. 7). Experimental setup is identical including ramp dimensions and controller gain ($K_p$ = 2.0). Both $\tilde{\theta}^{comp}$ and $s^{comp}$ were tested. The experiment was repeated with Zoë's right wheels driving over the obstacle for comparison.

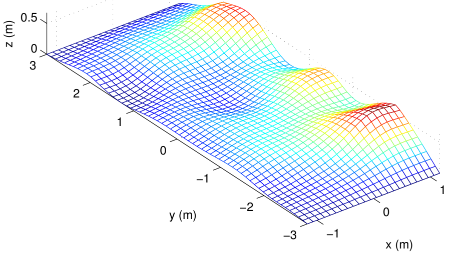

Fig. 10. Mesh of random rough surface used in simulation.

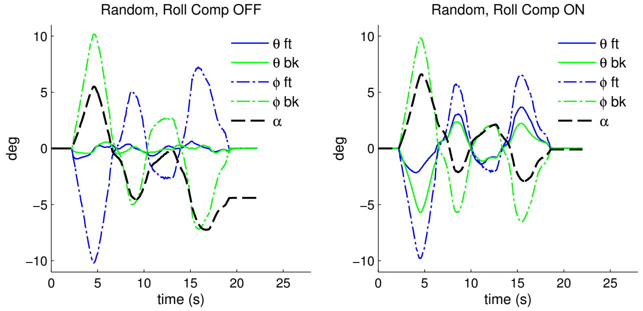

Fig. 11. Simulation results. Plot of yaw, steering, and axle roll angles as Zoë drives over random rough terrain. Note that with roll compensation enabled, a heading error of -4.4° is eliminated (see the dashed black lines)

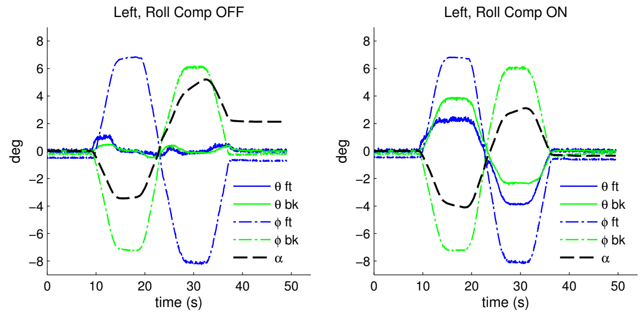

Fig. 12. Experimental results. Plot of yaw, steering, and axle roll angles as Zoë's *left* wheels drive over a ramp obstacle. Yaw (black) is measured with respect to a reference yaw obtained by driving the same arc with the obstacle removed. Roll compensation reduces a final heading error of 2.1° to approximately 0.3°. Compare to simulation results in Fig. 8

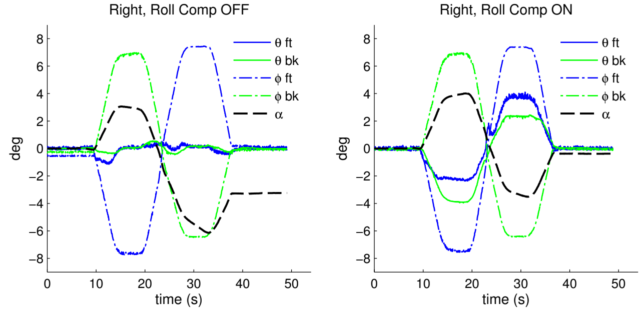

Fig. 13. Experimental results. Plot of yaw, steering, and axle roll angles as Zoë's *right* wheels drive over a ramp obstacle. Roll compensation reduces a final heading error of 3.3° to approximately 0.3° (see the dashed black lines)

Without roll compensation, a final yaw error of 2.1°-3.3° is produced by driving over the obstacle. With roll compensation this error is reduced to approximately 0.3° in both tests (see Fig. 12 and 13). The asymmetry between the left and right wheels traversing the obstacle is likely due to inexactness in the hardware fabrication and calibration. For example, note that the magnitude of the front axle roll is consistently larger than the rear axle roll.

In Fig. 14, note that the slope angle is overestimated when descending the ramp, just as it was in simulation (Fig. 9). Due to sensing limitations, in this test the change in height of the robot ($\Delta h_{robot}$ in Algorithm 2) was inferred from the axle roll angles by assuming only positive obstacles. This assumption was only necessary to compute the $s^{comp}$ terms which are less critical at low speeds, because the PID controller can easily maintain the desired steer angles. No assumptions were necessary to compute the $\tilde{\theta}^{comp}$ terms.

In Fig. 15, note that with roll compensation enabled, wheels that do not traverse the obstacle (the right wheels) deviate less from the nominal velocity. This indicates that the scaled feedforward velocities are maintaining the desired steer angles with little error for the PID controller to correct.

## VII. CONCLUSIONS AND FUTURE WORK

Passively steered vehicles are ideal for many planetary exploration applications due to their reliability and efficiency. The Zoë rover demonstrated this by autonomously traversing over 250 km of rough desert terrain. With the presented 3-D kinematic controller, passively-steered vehicles can not only traverse rough terrain, but follow exact trajectories while doing so. This is accomplished by compensating for axle roll in both the commanded steer angles and the feedforward wheel velocities. In both simulation and physical experiments, trajectory following for the Zoë rover was greatly improved using the new controller. This enhanced controller establishes the viability of the passive-steering rover design for precise navigation.

Concurrent research includes the high-frequency estimation of velocity using optical flow. This system could be used to detect loss of traction, an important task as the passive-steering design requires traction to control the steer angles. Future work includes integration of the controller with a higher-level planner that uses terrain map data. The planner could use terrain map data to define the ground plane. In addition, the planner could determine whether or not an obstacle is traversable (based on shape, height, etc.) then reduce the vehicle speed before reaching the obstacle.

## ACKNOWLEDGMENT

This research was made with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. In addition, the authors would like to thank Dominic Jonak and Michael Wagner for their prior software development on Zoë and assistance in conducting experiments. The NASA funded Life in the Atacama project (NAG5-12890) supported development of the Zoë rover.

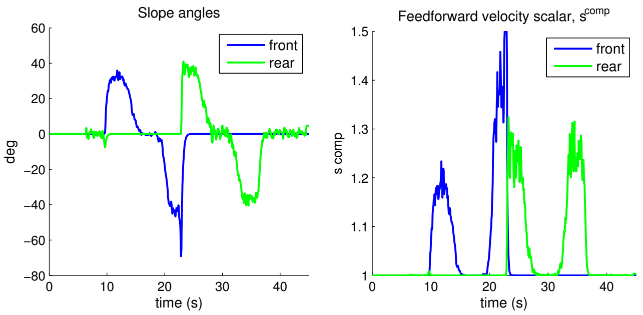

Fig. 14. Experimental results. Plot of slope angle and $s^{comp}$ as Zoë's left wheels drive over a ramp obstacle with roll compensation enabled. Compare to simulation results in Fig. 9

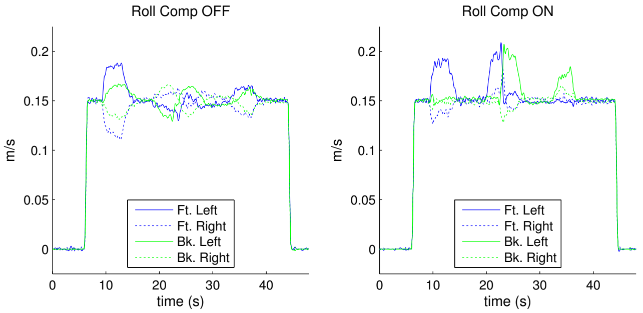

Fig. 15. Experimental results. Plot of all four wheel velocities (measured by encoders) as Zoë's left wheels drive over a ramp obstacle. With roll compensation enabled, the right wheels which are driving on flat ground deviate less from the nominal velocity.

## REFERENCES

[1] D. Wettergreen *et al.*, "Second experiments in the robotic investigation of life in the Atacama desert of Chile," in *Proc. 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005.

[2] M. Wagner *et al.*, "Design and control of a passively steered, dual axle vehicle," in *Proc. 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005.

[3] R. Lindemann and C. Voorhees, "Mars exploration rover mobility assembly design, test and performance," in *Proc. International Conference on Systems, Man, and Cybernetics*. IEEE, 2005.

[4] L. Pedersen, M. Allan, V. To, H. Utz, W. Wojcikiewicz, and C. Chautems, "High speed lunar navigation for crewed and remotely piloted vehicles," in *Proc. International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Aug. 2010.

[5] P. Ge and R. Wang, "Velocity kinematics modeling and simulation for a lunar rover," in *Proc. International Conference on Electrical and Control Engineering*, 2010.

[6] M. Tarokh and G. McDermott, "Kinematics modeling and analyses of articulated rovers," *Transactions on Robotics*, 2005.

[7] P. Lamon and R. Siegwart, "3D position tracking in challenging terrain," in *Proc. Field and Service Robotics*, 2005.

[8] T. Howard, "Adaptive model-predictive motion planning for navigation in complex environments," Ph.D. dissertation, Rob. Inst., Carnegie Mellon Univ., 2009.

[9] J. Craig, *Introduction to robotics: mechanics and control*, 2nd ed. Addison-Wesley, 1989.

[10] J. Wong, *Theory of ground vehicles*, 3rd ed. New York: Wiley, 2001.