Received: 15 December 2018 | Revised: 17 June 2019 | Accepted: 2 July 2019

DOI: 10.1002/rob.21903

REGULAR ARTICLE

Terrain-adaptive wheel speed control on the Curiosity Mars rover: Algorithm and flight results

Olivier Toupet | Jeffrey Biesiadecki | Arturo Rankin | Amanda Steffy | Gareth Meirion-Griffith | Dan Levine | Maximilian Schadegg | Mark Maimone

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California

Correspondence
Olivier Toupet, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109.
Email: Olivier.Toupet@jpl.nasa.gov

Funding information
Mars Science Laboratory Program

Abstract

NASA's Mars Science Laboratory Curiosity rover landed in August 2012 and began experiencing higher rates of wheel damage beginning in October 2013. While the wheels were designed to accumulate considerable damage, the unexpected damage rate raised concerns regarding wheel lifetime. In response, the Jet Propulsion Laboratory developed and deployed mobility flight software on Curiosity that reduces the forces on the wheels. The new algorithm adapts each wheel's speed to fit the terrain topography in real time, by leveraging the rover's measured attitude rates and rocker/bogie suspension angles and rates. Together with a rigid-body kinematics model, it estimates the real-time wheel-terrain contact angles and commands idealized, no-slip wheel angular rates. In addition, free-floating "wheelies" are detected and autonomously corrected. Ground test data indicate that the forces on the wheels are reduced by 19% for leading wheels and 11% for middle leading wheels. On the ground, the required data volume increased by up to 129%, and drive duration increased by up to 25%. In flight, data collected over 3.6 km and 149 drives confirmed a reduction in wheel current, correlated with wheel torque, of 18.7%. The new algorithm proved to use fewer resources in flight than ground estimates suggested, as only a 10% increase in drive duration and double the drive data volume were experienced. These data indicate the promise of the new algorithm to extend the life of the wheels for the Curiosity rover. This paper describes the algorithm, its ground testing campaign and associated challenges, and its validation, implementation, and performance in flight.

KEYWORDS
control, planetary robotics, wheeled robots

1 | INTRODUCTION

The Mars Science Laboratory (MSL) Curiosity rover has traveled more than 20 km over the course of its prime and extended missions. On Sol 490, Mars Hand Lens Imager (MAHLI) images revealed an unexpectedly high rate of damage to the rover wheels. Because the rover was only 4.6 km into the prime mission, the MSL project launched an investigation into the causes of wheel wear. After a detailed investigation (Arvidson et al., 2017), the MSL project began efforts to reduce further wheel damage by altering the way the vehicle drives over obstacles. Along with careful terrain classification and drive planning, periodic trending of the wheel state, and extensive characterization of the mechanisms of wheel failure per terrain type, the MSL project began the development of new software to slow the damage rate.

Our goal was to reduce the rate of wear on the skin of the rover wheels. Our vehicle does not include force or torque sensors on the mobility subsystem, nor can it measure slip with high enough frequency to be able to react to it. Visual odometry (VO) slip measurements occur relatively infrequently (typically once every meter or just over 1 minute apart) due to the limited computational resources of the rover and its impact on traverse rate (the rover needs to stop to acquire images). Therefore we developed an approach that aims to reduce forces on the wheels and improve traction when driving on uneven surfaces by commanding the idealized, no-slip wheel rates based on rover inverse kinematics and real-time attitude, suspension, and steering angle data. For this reason, we call our method Traction Control (TRCTL), although the term is not meant to imply “direct computation of desired traction forces and application of wheel motor torques to achieve these” (Iagnemma & Dubowsky, 2004).

Without the TRCTL algorithm, the rover drives under the assumption of flat terrain. Based on the Ackermann steering model, it moves along a commanded arc by turning each wheel at a constant speed. If it drives along its commanded path and encounters a single rock in front of the left wheel of the rover, that front wheel travels a further path length over the rock relative to the other wheels on flat ground in the same amount of time. This method of commanding the wheels results in slip, as the five flat wheels push the front left wheel forward. In addition, the constant wheel speed contributes to the puncture damage to the rover wheels, especially in terrain where rocks are embedded and sharp, with a tip that can fit between the wheel treads and puncture the wheel skin. Punctures to the wheel skin sections initiate the damage rate acceleration seen in flight (see Figure 1) and in terrestrial Mars Yard testing.

TRCTL does not make this assumption of flat terrain, since the Martian terrain is not flat and contains a variety of local obstacle types. Instead, the wheel traversing the obstacle can be sped up relative to the other wheels, to reduce overall the wheel slip. Rover wheel speeds are capped, so to better adapt to the terrain the five wheels on flat ground are slowed, with the left front wheel continuing over the rock at the maximum speed. By preventing the five wheels from pushing the leading wheel into the obstacle, TRCTL aims to reduce the initiation and exacerbation of puncture damage.

There are a number of challenges to modeling the unknown terrain topography of Mars to modulate the rover wheel speeds. Rover-generated height maps (Goldberg, Maimone, & Matthies, 2002; Maimone, Johnson, Cheng, Willson, & Matthies, 2006) of the nearby terrain are resource-intensive and significantly decrease the rover traverse rate. Additionally, due to noise in the maps and uncertainty in rover position, these are not enough to calculate rover wheel speeds. Instead, the TRCTL approach only relies on the rover’s measured attitude rates and suspension angles (from the rover’s gyros and rocker/bogies resolvers), and leverages the rigid-body kinematic model of the rover to calculate the optimal wheel speeds as the rover drives.

Validation of the TRCTL algorithm involved extensive ground testing on MSL’s mobility test vehicle and the Vehicle System Testbed (VSTB). Engineering the test terrains, with science team input, provided a variety of stressing cases to ensure effective behavior of the algorithm. One of these cases, a high-friction terrain consisting of cement tiles with varying sizes of embedded rocks, caused the vehicle’s middle wheel to “wheelie”—to rise off the ground. Repeatability testing determined that the rock configuration, a large rock ahead of the front wheel with the middle wheel cresting a rock, and the tension in the suspension system, contributed to this event. To protect against this case in flight, a wheelie suppression behavior was added to the TRCTL algorithm.

The algorithm was implemented as a hot patch to the rover flight software, meaning that it is loaded after each boot of the flight system computer. Additional flight considerations included changes to resource models of data volume and drive duration, as well as updates to ground tools to assess the drive performance. After a three-stage checkout to ensure vehicle safety, the flight performance of the vehicle was closely monitored. Trending results and preliminary flight data confirm the ground performance of the algorithm.

This paper first describes the TRCTL algorithm and test performance, then discusses the details of its implementation, including the integration into mission operations, test results, and assessment tools. It concludes with the results of flight performance and plans for long-term trending.

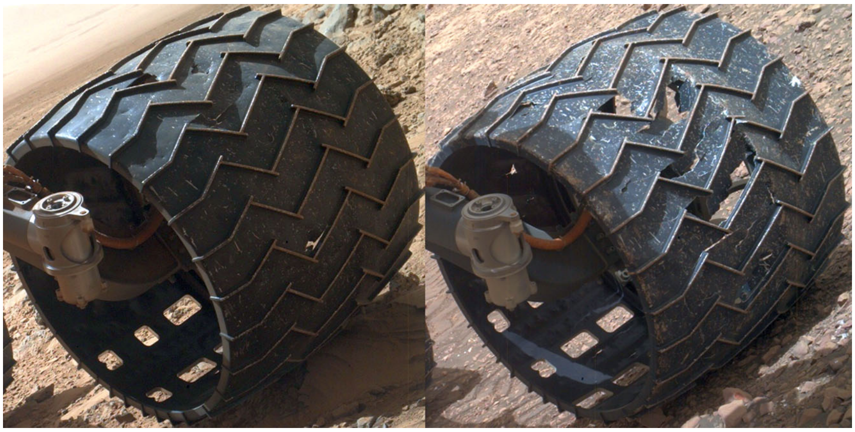


FIGURE 1 MAHLI image of the Left Middle (LM) wheel on Sol 490 (left) and Sol 2030 (right). In the Sol 2030 image, one of the two LM grouser breaks is visible. MAHLI, Mars Hand Lens Imager [Color figure can be viewed at wileyonlinelibrary.com]

1.1 | Related work

Several techniques have been proposed in the literature for traction control of a rover with a rocker-bogie suspension system. In Sreenivasan and Wilcox (1994), a traction control approach was developed which takes into account the effects of the terrain, but assumes knowledge of the terrain geometry and real-time measurements of the wheel-ground contact forces. Those contact forces cannot be measured on Curiosity, and neither can we assume knowledge of the terrain geometry, which is only generated for very few drives due to the computational expense of running stereo vision to generate a mesh of the surrounding terrain and its substantial impact on the rover's traverse rate (this is only done for AutoNav drives where the rover needs to create a height map to choose obstacle-free paths). Even for those few drives where the terrain geometry is known, the location of the rover relative to the terrain would not necessarily be well known, as the rover slip is measured infrequently, if at all.

Yoshida et al. developed an approach in Yoshida and Hamano (2002) and Yoshida, Hamano, and Watanabe (2003) which rely on a physical model of the terramechanics representing the tire–soil interaction to estimate the slip ratio of each wheel and keep it low by adapting the commanded wheel speed to limit excessive tire force and prevent the wheels from getting stuck in loose soil. Their rover testbed does not have any force/torque sensors but a video-tracking system is used to develop the terramechanics model for a given type of terrain. Martian soil is extremely challenging to characterize and model accurately, and it also varies a lot, even over the course of a single drive, as Curiosity traverses various types of terrains. Therefore we decided against relying on any type of terramechanics model.

In Ishigami, Nagatani, and Yoshida (2009), two techniques are developed: one that uses feedforward control based again on a terramechanics model to predict slip, and another which uses feedback control based on slip measurements provided by VO to minimize slip and path deviations. While the feedback control technique does not rely on an a priori soil–tire interaction model, it does rely on being able to run VO at 30 Hz to quickly measure and compensate for slip. Due to the limited computational resources of the Curiosity rover, VO is not always used, as it slows down the drive, and when it is used, it is only run at a very low rate (typically once every 90 s) as the rover needs to stop to image, process the images, and then run VO, all of which takes substantial time. Therefore, we decided not to rely on VO for our high-rate TRCTL approach.

Iagnemma et al. proposed an extended Kalman filter (EKF) formulation in Iagnemma and Dubowsky (2000a), Iagnemma and Dubowsky (2000b), and Iagnemma and Dubowsky (2004) to estimate the wheel-terrain contact angles, a key variable for traction control algorithms, by fusing the noisy onboard sensor signals. The approach is only demonstrated on a 2D rover model however, and while the idea of taking into account the noise of the various sensors to produce a more accurate estimate of wheel-terrain contact angles is attractive, the additional complexity and risk associated with an EKF implementation (e.g., filter divergence), especially given the high nonlinearity of the 3D inverse rover kinematics, was a practical concern.

Our work was inspired by the approach described in Tarokh, McDermott, Hayati, and Hung (1999), which uses the full 3D vehicle kinematics to predict wheel-terrain contact angles. The work described in this paper was first presented with preliminary results in Toupet et al. (2018).

2 | ALGORITHM

Our approach consists in using rigid-body kinematics to relate the velocities of each moving part of the rover. The centers of the two front wheels rotate in opposite directions relative to the main body of the rover around the rocker joint. The centers of the middle and rear wheels rotate relative to the rocker body around the left and right bogie joints. Those rotation angles and rates are measured by encoders and enable us to express the wheel velocities as a function of the geometry of the rover, and the measured attitude and suspension rates, and contact angles between the wheels and the terrain.

We first introduce the mathematical framework, including the symbols, coordinate frames, and kinematic formulas. We then describe how we estimate the wheels' contact angles, and conclude with the calculation of the ideal wheel speeds.

2.1 | Rigid-body rover model

A description of all the symbols used in this section can be found in the Appendix A.

We simplify the rover geometry to avoid unnecessary parameters by placing the rocker D and bogies B_1 and B_2 in the x-z plane of the rover's body frame (no lateral offset from the rover's origin O), as illustrated by Figure 2. The rover origin O is between the middle wheels on the surface, when on flat ground.

2.2 | Frames

We define the following frames and rotation matrices:

- Body frame bd: follows the aerospace convention, with the x axis along the rover's body length, pointing forward, the y axis pointing to the right of the x axis, and the z axis pointing down.
- Rocker frames rk_1 and rk_2: body frame rotated by the rocker angle ($+\beta$ for rk_1 and $-\beta$ for rk_2). We define the following rotation matrices (from body to rocker frames):

$$^{rk_1}R_{bd} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}, \tag{1}$$

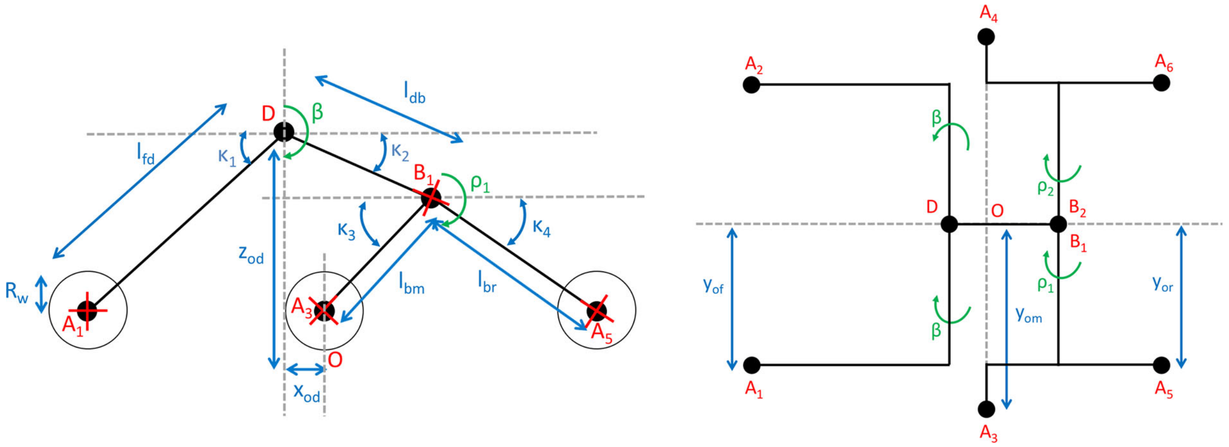


FIGURE 2 Rover model on flat ground, viewed from the left side (left) and above (right). Note that the rover's forward direction is to the left, but the rover can drive both forward and backward [Color figure can be viewed at wileyonlinelibrary.com]

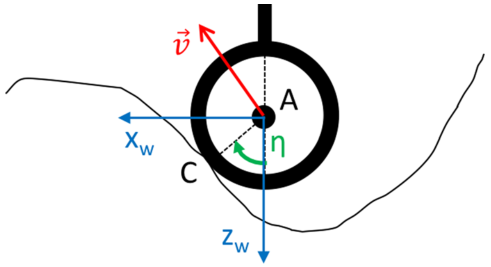


FIGURE 3 Definition of the contact angle η. $\vec{v}$ is the wheel velocity, and $\vec{x}_w$ and $\vec{z}_w$ are the x and z axis of the wheel frame (see Section 2.2 for a definition of that frame, which does not depend on gravity). Note that the rover's forward direction is to the left, but the rover can drive both forward and backward [Color figure can be viewed at wileyonlinelibrary.com]

$$^{rk_2}R_{bd} = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix}. \tag{2}$$

- Bogie frames bg_1 and bg_2: rocker frames rotated by the bogie angles (ρ_1 for bg_1 and ρ_2 for bg_2). We define the following rotation matrices (from rocker to bogie frames):

$$^{bg_i}R_{rk_i} = \begin{bmatrix} \cos(\rho_i) & 0 & \sin(\rho_i) \\ 0 & 1 & 0 \\ -\sin(\rho_i) & 0 & \cos(\rho_i) \end{bmatrix} \text{for all } i \in \{1,\ 2\}. \tag{3}$$

- Wheel frames w_i, $i \in [1, 6]$: rocker frames (for front wheels) or bogie frames (for middle/rear wheels) rotated by the wheel's steering angle ψ_i. We define the following rotation matrices (from rocker/bogie to wheel frames):

$$^{w_i}R_{rk_i} = \begin{bmatrix} \cos(\psi_i) & -\sin(\psi_i) & 0 \\ \sin(\psi_i) & \cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{for all } i \in \{1,\ 2\}, \tag{4}$$

$$^{w_i}R_{bg_1} = \begin{bmatrix} \cos(\psi_i) & -\sin(\psi_i) & 0 \\ \sin(\psi_i) & \cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{for all } i \in \{3,\ 5\}, \tag{5}$$

$$^{w_i}R_{bg_2} = \begin{bmatrix} \cos(\psi_i) & -\sin(\psi_i) & 0 \\ \sin(\psi_i) & \cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{for all } i \in \{4,\ 6\}. \tag{6}$$

- Contact angle frames η_i, $i \in [1, 6]$: wheel frames rotated by the wheel's contact angle η_i. We define the following rotation matrices (from wheel to contact angle frames):

$$^{\eta_i}R_{w_i} = \begin{bmatrix} \cos(\eta_i) & 0 & \sin(\eta_i) \\ 0 & 1 & 0 \\ -\sin(\eta_i) & 0 & \cos(\eta_i) \end{bmatrix} \text{for all } i \in [1,\ 6]. \tag{7}$$

2.3 | Contact angle definition

We define the contact angle η between a wheel and the ground as the angle between the steering actuator axis and the contact point of the wheel and the ground, as illustrated on Figure 3.

Note that when the rover is on flat ground, the contact angle for each wheel is zero. If the front wheel climbs over a rock while the other wheels remain on flat ground, the contact angle would become positive as the wheel climbs and negative as the wheel descends.

In reality, the wheels of the rover do not make contact with the ground at a single point. This is fine, however, as we can still model the motion of the wheel as if there was a single contact point, whose location on the wheel is defined such that the vector from the center of the wheel to the contact point ($\vec{AC}$ in Figure 3) is orthogonal to the velocity vector of the wheel ($\vec{v}$ in Figure 3).

2.4 | Kinematic equations

With the parameters of our rover model defined, we can relate the velocities of the wheels to the velocity of any point on the rover using rigid-body kinematics.

The linear velocities of any two points A and B, attached to the same rigid body $\mathcal{R}$, relative to some frame $\mathscr{I}$, and expressed in some arbitrary frame $\mathcal{F}$, are related according to the following key equation:

$$^{\mathcal{F}}\vec{v}_{A/\mathscr{I}} = {}^{\mathcal{F}}\vec{v}_{B/\mathscr{I}} + {}^{\mathcal{F}}\vec{\omega}_{\mathcal{R}/\mathscr{I}} \times {}^{\mathcal{F}}\overrightarrow{BA}, \tag{8}$$

where $^{\mathcal{F}}\vec{\omega}_{\mathcal{R}/\mathscr{I}}$ is the angular velocity vector of the rigid body relative to the frame $\mathscr{I}$, expressed in frame $\mathcal{F}$.

For our purposes, all velocities will be relative to the inertial frame and the angular velocity vector will always be the one of the rover's body relative to the inertial frame, so we'll simplify the notations as follows:

$$^{\mathcal{F}}\vec{v}_{A} = {}^{\mathcal{F}}\vec{v}_{B} + {}^{\mathcal{F}}\vec{\omega} \times {}^{\mathcal{F}}\overrightarrow{BA}. \tag{9}$$

This kinematic relationship can be used to relate the velocities of any two points on the articulated rover body. To illustrate our approach, described in details in the following subsections, we show how we can express the velocity of the left front wheel (wheel 1) as a function of the velocity of the rover's origin:

$$\begin{aligned}
^{bd}\vec{v}_{O} &= \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \\
^{bd}\vec{v}_{D} &= {}^{bd}\vec{v}_{O} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{OD} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \\
&\quad + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{OD}, \\
^{bd}\vec{v}_{A_1} &= {}^{bd}\vec{v}_{D} + \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{DA_1} \\
&= \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{OD} \\
&\quad + \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{DA_1}.
\end{aligned}$$

However, since we want to calculate the ideal (no-slip) wheel rates, we need to express the wheel velocity in the contact angle frame, since we know that the x component, $^{\eta_1}v^{x}_{A_1}$, will be proportional to the wheel rate $\dot{\theta}_1$:

$$^{\eta_1}v^{x}_{A_1} = R_w \, \omega^{y}_{1},$$

with ω^{y}_{1} the angular rate of the wheel. However, ω^{y}_{1} is not exactly the same as the angular rate delivered by the drive motor $\dot{\theta}_1$ (what we want to solve for), since that angular rate is relative to the drive actuator, which itself rotates relative to the inertial frame due to the rover's body and suspension rates:

$$\omega^{y}_{1} = \dot{\theta}_1 + \zeta^{y}_{1}, \text{ with } \vec{\zeta_1} = {}^{\eta_1}R_{w_1} \, {}^{w_1}R_{rk_1} \, {}^{rk_1}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix}.$$

To express the wheel velocity in the contact angle frame, we can use the rotation matrices defined earlier:

$$\begin{aligned}
^{\eta_1}\vec{v}_{A_1} &= {}^{\eta_1}R_{w_1} \, {}^{w_1}R_{rk_1} \, {}^{rk_1}R_{bd} \, {}^{bd}\vec{v}_{A_1} \\
&= \begin{bmatrix} \cos(\eta_1) & 0 & -\sin(\eta_1) \\ 0 & 1 & 0 \\ \sin(\eta_1) & 0 & \cos(\eta_1) \end{bmatrix} \\
&\quad \times \begin{bmatrix} \cos(\psi_1) & \sin(\psi_1) & 0 \\ -\sin(\psi_1) & \cos(\psi_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&\quad \times \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \\
&\quad \times \left(\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{OD} + \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{DA_1} \right).
\end{aligned}$$

Those equations show that the wheel rate is a function of the attitude and suspension rates, suspension angles, steering angles (all of which are measured), and the wheel's contact angle and rover's linear velocity (not measured).

2.5 | Algorithm overview

The functional diagram depicting our technical approach is shown in Figure 4.

2.6 | Calculation of the contact angle estimates

There are several ways to estimate the contact angles (Iagnemma & Dubowsky, 2004). We chose to trade off some accuracy in favor of robustness by choosing an approach that does not depend on the commanded nor measured wheel velocities. This way, the output of the algorithm, namely, the wheel rate commands, do not affect the input of the algorithm at the next time step, thus avoiding any feedback loops that might have caused errors and jeopardized the stability of the algorithm.

A simplifying assumption was made: We approximated the rover's linear velocity to its value on flat ground when calculating the contact angle estimates.

2.6.1 | Estimating the rover's linear velocity

The flat ground approximation of the rover's linear velocity in the body frame is

$$^{bd}\vec{v}_{O} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ 0 \\ 0 \end{bmatrix}. \tag{10}$$

- For straight driving, all points on the rover have the same speed, capped by the maximum wheel rate $\dot{\theta}_{\max}$:

$$\dot{x}_0 = \text{dir } R_w \, \dot{\theta}_{\max}, \tag{11}$$

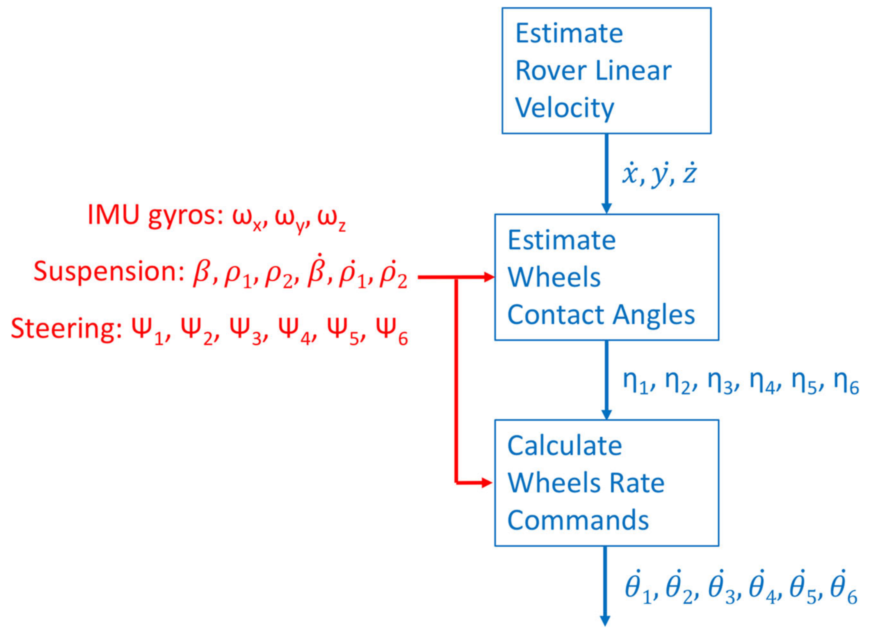

Estimate Rover Linear Velocity
$\dot{x}, \dot{y}, \dot{z}$
IMU gyros: $\omega_x, \omega_y, \omega_z$
Suspension: $\beta, \rho_1, \rho_2, \dot{\beta}, \dot{\rho}_1, \dot{\rho}_2$
Steering: $\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Psi_6$
Estimate Wheels Contact Angles
$\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6$
Calculate Wheels Rate Commands
$\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6$

FIGURE 4 Functional diagram of the TRCTL algorithm. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

where dir is the drive direction (+1 for forward; −1 for backward), and R_w is the wheel radius.

- For turns, we are also limited by $\dot{\theta}_{\max}$, but $\dot{x}_0$ also depends on the turn radius r of the commanded arc:

$$\dot{x}_0 = \text{dir}\ R_w\ \dot{\theta}_{\max}\ \frac{|r|}{\max(r_f, r_m, r_r)}, \tag{12}$$

where r_f, r_m, and r_r are the turn radius of the front, middle, and rear wheels, respectively, located to the outside of the turn (away from the center of rotation—those are the wheels with the largest turn radius and therefore going the fastest). Note that r_f, r_m, and r_r are greater than r since the distance from the center of rotation to the rover's origin is always shorter than the distance to the outside wheels, and hence, the velocity of the rover's origin is decreased during turns compared to straight driving. These turn radii of the wheels can be computed as follows:

$$r_f = \sqrt{x_{fm}^2 + (y_{of} + |r|)^2}, \tag{13}$$

$$r_m = y_{om} + |r|, \tag{14}$$

$$r_r = \sqrt{x_{mr}^2 + (y_{or} + |r|)^2}. \tag{15}$$

2.6.2 | Estimating the contact angles

The contact angle for each wheel is computed by carrying out the following steps:

- Calculate the wheel's linear velocity vector expressed in the wheel's frame.
- Compute the contact angle as the angle between the x and z components of that vector.

We can compute the wheels' linear velocities using the kinematics equation (9) and our approximation of the linear velocity of the rover's origin equation (10). First we compute the linear velocities of the wheels in the body frame:

$${}^{bd}v_D = \begin{bmatrix}\dot{x}_0\\ 0\\ 0\end{bmatrix} + \begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{OD}, \tag{16}$$

$${}^{bd}v_{A_1} = {}^{bd}v_D - \begin{bmatrix}\omega_x\\ \omega_y + \dot{\beta}\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{A_1D}, \tag{17}$$

$${}^{bd}v_{A_2} = {}^{bd}v_D - \begin{bmatrix}\omega_x\\ \omega_y - \dot{\beta}\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{A_2D}, \tag{18}$$

$${}^{bd}v_{B_1} = {}^{bd}v_D - \begin{bmatrix}\omega_x\\ \omega_y + \dot{\beta}\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{B_1D}, \tag{19}$$

$${}^{bd}v_{B_2} = {}^{bd}v_D - \begin{bmatrix}\omega_x\\ \omega_y - \dot{\beta}\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{B_2D}, \tag{20}$$

$${}^{bd}v_{A_3} = {}^{bd}v_{B_1} - \begin{bmatrix}\omega_x\\ \omega_y + \dot{\beta} + \dot{\rho}_1\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{A_3B_1}, \tag{21}$$

$${}^{bd}v_{A_4} = {}^{bd}v_{B_2} - \begin{bmatrix}\omega_x\\ \omega_y - \dot{\beta} + \dot{\rho}_2\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{A_4B_2}, \tag{22}$$

$${}^{bd}v_{A_5} = {}^{bd}v_{B_1} - \begin{bmatrix}\omega_x\\ \omega_y + \dot{\beta} + \dot{\rho}_1\\ \omega_z\end{bmatrix} \times {}^{bd}\overrightarrow{A_5B_1}, \tag{23}$$

$$^{bd}v_{A_6} = {}^{bd}v_{B_2} - \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} + \dot{\rho_2} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_6B_2}, \tag{24}$$

with the following parameters based on the rover geometry shown in Figure 2:

$$^{bd}\overrightarrow{OD} = \begin{bmatrix} x_{od} \\ 0 \\ z_{od} \end{bmatrix}, \tag{25}$$

$$^{bd}\overrightarrow{A_1D} = {}^{bd}R_{rk_1} \begin{bmatrix} -\,l_{fd}\,\cos(\kappa_1) \\ y_{of} \\ -\,l_{fd}\,\sin(\kappa_1) \end{bmatrix}, \tag{26}$$

$$^{bd}\overrightarrow{A_2D} = {}^{bd}R_{rk_2} \begin{bmatrix} -\,l_{fd}\,\cos(\kappa_1) \\ -\,y_{of} \\ -\,l_{fd}\,\sin(\kappa_1) \end{bmatrix}, \tag{27}$$

$$^{bd}\overrightarrow{B_1D} = {}^{bd}R_{rk_1} \begin{bmatrix} l_{db}\,\cos(\kappa_2) \\ 0 \\ -\,l_{db}\,\sin(\kappa_2) \end{bmatrix}, \tag{28}$$

$$^{bd}\overrightarrow{B_2D} = {}^{bd}R_{rk_2} \begin{bmatrix} l_{db}\,\cos(\kappa_2) \\ 0 \\ -\,l_{db}\,\sin(\kappa_2) \end{bmatrix}, \tag{29}$$

$$^{bd}\overrightarrow{A_3B_1} = {}^{bd}R_{rk_1}\,{}^{rk_1}R_{bg_1} \begin{bmatrix} -\,l_{bm}\,\cos(\kappa_3) \\ y_{om} \\ -\,l_{bm}\,\sin(\kappa_3) \end{bmatrix}, \tag{30}$$

$$^{bd}\overrightarrow{A_4B_2} = {}^{bd}R_{rk_2}\,{}^{rk_2}R_{bg_2} \begin{bmatrix} -\,l_{bm}\,\cos(\kappa_3) \\ -\,y_{om} \\ -\,l_{bm}\,\sin(\kappa_3) \end{bmatrix}, \tag{31}$$

$$^{bd}\overrightarrow{A_5B_1} = {}^{bd}R_{rk_1}\,{}^{rk_1}R_{bg_1} \begin{bmatrix} l_{br}\,\cos(\kappa_4) \\ y_{or} \\ -\,l_{br}\,\sin(\kappa_4) \end{bmatrix}, \tag{32}$$

$$^{bd}\overrightarrow{A_6B_2} = {}^{bd}R_{rk_2}\,{}^{rk_2}R_{bg_2} \begin{bmatrix} l_{br}\,\cos(\kappa_4) \\ -\,y_{or} \\ -\,l_{br}\,\sin(\kappa_4) \end{bmatrix}. \tag{33}$$

Now we can compute the wheels' linear velocities in the wheels' frames:

$$^{w_1}v_{A_1} = {}^{w_1}R_{rk_1}\,{}^{rk_1}R_{bd}\,{}^{bd}v_{A_1}, \tag{34}$$

$$^{w_2}v_{A_2} = {}^{w_2}R_{rk_2}\,{}^{rk_2}R_{bd}\,{}^{bd}v_{A_2}, \tag{35}$$

$$^{w_3}v_{A_3} = {}^{w_3}R_{bg_1}\,{}^{bg_1}R_{rk_1}\,{}^{rk_1}R_{bd}\,{}^{bd}v_{A_3}, \tag{36}$$

$$^{w_4}v_{A_4} = {}^{w_4}R_{bg_2}\,{}^{bg_2}R_{rk_2}\,{}^{rk_2}R_{bd}\,{}^{bd}v_{A_4}, \tag{37}$$

$$^{w_5}v_{A_5} = {}^{w_5}R_{bg_1}\,{}^{bg_1}R_{rk_1}\,{}^{rk_1}R_{bd}\,{}^{bd}v_{A_5}, \tag{38}$$

$$^{w_6}v_{A_6} = {}^{w_6}R_{bg_2}\,{}^{bg_2}R_{rk_2}\,{}^{rk_2}R_{bd}\,{}^{bd}v_{A_6}. \tag{39}$$

Finally, we can compute the contact angles as follows:

$$\forall\, i \in [1, 6], \quad \eta_i = -\arctan\left(\frac{^{w_i}v^z_{A_i}}{^{w_i}v^x_{A_i}}\right). \tag{40}$$

2.7 | Calculation of the wheel rate commands

As described in Section 2.4, it is possible to relate the wheel angular rates to the linear velocity of the rover origin, contact angles, attitude rates, and suspension angles and rates. Let us derive those equations for all six wheels.

First we can set the *y* component of the rover's linear velocity to zero since we do not want the rover to move sideways:

$$\dot{y} = 0. \tag{41}$$

This results in

$$^{bd}\overrightarrow{v_O} = \begin{bmatrix} \dot{x} \\ 0 \\ \dot{z} \end{bmatrix}. \tag{42}$$

Then we compute the wheel linear velocities in the body frame, by applying our kinematics formula equation (9) down the chain of linked rigid bodies, starting from the rover's body (at origin O), and moving down to the wheels, passing through the rocker and bogies points:

$$^{bd}v_D = \begin{bmatrix} \dot{x} \\ 0 \\ \dot{z} \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{OD}, \tag{43}$$

$$^{bd}v_{A_1} = {}^{bd}v_D - \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_1D}, \tag{44}$$

$$^{bd}v_{A_2} = {}^{bd}v_D - \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_2D}, \tag{45}$$

$$^{bd}v_{B_1} = {}^{bd}v_D - \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{B_1D}, \tag{46}$$

$$^{bd}v_{B_2} = {}^{bd}v_D - \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{B_2D}, \tag{47}$$

$$^{bd}v_{A_3} = {}^{bd}v_{B_1} - \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} + \dot{\rho_1} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_3B_1}, \tag{48}$$

$$^{bd}v_{A_4} = {}^{bd}v_{B_2} - \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} + \dot{\rho_2} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_4B_2}, \tag{49}$$

$$^{bd}v_{A_5} = {}^{bd}v_{B_1} - \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} + \dot{\rho_1} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_5B_1}, \tag{50}$$

$$ {}^{bd}v_{A_6} = {}^{bd}v_{B_2} - \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} + \dot{\rho_2} \\ \omega_z \end{bmatrix} \times {}^{bd}\overrightarrow{A_6B_2}. \tag{51} $$

Using the rotation matrices defined in Section 2.2, we express the body-frame linear velocities of the wheels in the contact angle frames:

$$ {}^{\eta_1}v_{A_1} = {}^{\eta_1}R_{w_1} \; {}^{w_1}R_{rk_1} \; {}^{rk_1}R_{bd} \; {}^{bd}v_{A_1}, \tag{52} $$

$$ {}^{\eta_2}v_{A_2} = {}^{\eta_2}R_{w_2} \; {}^{w_2}R_{rk_2} \; {}^{rk_2}R_{bd} \; {}^{bd}v_{A_2}, \tag{53} $$

$$ {}^{\eta_3}v_{A_3} = {}^{\eta_3}R_{w_3} \; {}^{w_3}R_{bg_1} \; {}^{bg_1}R_{rk_1} \; {}^{rk_1}R_{bd} \; {}^{bd}v_{A_3}, \tag{54} $$

$$ {}^{\eta_4}v_{A_4} = {}^{\eta_4}R_{w_4} \; {}^{w_4}R_{bg_2} \; {}^{bg_2}R_{rk_2} \; {}^{rk_2}R_{bd} \; {}^{bd}v_{A_4}, \tag{55} $$

$$ {}^{\eta_5}v_{A_5} = {}^{\eta_5}R_{w_5} \; {}^{w_5}R_{bg_1} \; {}^{bg_1}R_{rk_1} \; {}^{rk_1}R_{bd} \; {}^{bd}v_{A_5}, \tag{56} $$

$$ {}^{\eta_6}v_{A_6} = {}^{\eta_6}R_{w_6} \; {}^{w_6}R_{bg_2} \; {}^{bg_2}R_{rk_2} \; {}^{rk_2}R_{bd} \; {}^{bd}v_{A_6}. \tag{57} $$

We can then relate the wheel rate commands to the x component of the wheels' linear velocities in the contact angle frames:

$$ {}^{\eta_1}v^x_{A_1} = R_w(\dot{\theta}_1 + \zeta^y_1), \tag{58} $$

$$ {}^{\eta_2}v^x_{A_2} = R_w(\dot{\theta}_2 + \zeta^y_2), \tag{59} $$

$$ {}^{\eta_3}v^x_{A_3} = R_w(\dot{\theta}_3 + \zeta^y_3), \tag{60} $$

$$ {}^{\eta_4}v^x_{A_4} = R_w(\dot{\theta}_4 + \zeta^y_4), \tag{61} $$

$$ {}^{\eta_5}v^x_{A_5} = R_w(\dot{\theta}_5 + \zeta^y_5), \tag{62} $$

$$ {}^{\eta_6}v^x_{A_6} = R_w(\dot{\theta}_6 + \zeta^y_6), \tag{63} $$

with

$$ \overrightarrow{\zeta_1} = {}^{\eta_1}R_{w_1} \; {}^{w_1}R_{rk_1} \; {}^{rk_1}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} \\ \omega_z \end{bmatrix}, \tag{64} $$

$$ \overrightarrow{\zeta_2} = {}^{\eta_2}R_{w_2} \; {}^{w_2}R_{rk_2} \; {}^{rk_1}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} \\ \omega_z \end{bmatrix}, \tag{65} $$

$$ \overrightarrow{\zeta_3} = {}^{\eta_3}R_{w_3} \; {}^{w_3}R_{bg_1} \; {}^{bg_1}R_{rk_1} \; {}^{rk_1}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} + \dot{\rho}_1 \\ \omega_z \end{bmatrix}, \tag{66} $$

$$ \overrightarrow{\zeta_4} = {}^{\eta_4}R_{w_4} \; {}^{w_4}R_{bg_2} \; {}^{bg_2}R_{rk_2} \; {}^{rk_2}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} + \dot{\rho}_2 \\ \omega_z \end{bmatrix}, \tag{67} $$

$$ \overrightarrow{\zeta_5} = {}^{\eta_5}R_{w_5} \; {}^{w_5}R_{bg_1} \; {}^{bg_1}R_{rk_1} \; {}^{rk_1}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y + \dot{\beta} + \dot{\rho}_1 \\ \omega_z \end{bmatrix}, \tag{68} $$

$$ \overrightarrow{\zeta_6} = {}^{\eta_6}R_{w_6} \; {}^{w_6}R_{bg_2} \; {}^{bg_2}R_{rk_2} \; {}^{rk_2}R_{bd} \begin{bmatrix} \omega_x \\ \omega_y - \dot{\beta} + \dot{\rho}_2 \\ \omega_z \end{bmatrix}, \tag{69} $$

which, after leveraging Equations (1)–(7) provides:

$$ \zeta^y_1 = (\omega_y + \dot{\beta})\cos(\psi_1) - (\omega_x\cos(\beta)) - \omega_z\sin(\beta))\sin(\psi_1), \tag{70} $$

$$ \zeta^y_2 = (\omega_y - \dot{\beta})\cos(\psi_2) - (\omega_x\cos(\beta)) + \omega_z\sin(\beta))\sin(\psi_2), \tag{71} $$

$$ \zeta^y_3 = \omega_x + \dot{\beta} + \dot{\rho_1}, \tag{72} $$

$$ \zeta^y_4 = \omega_x - \dot{\beta} + \dot{\rho_2}, \tag{73} $$

$$ \begin{aligned} \zeta^y_5 = {} & (\omega_x + \dot{\beta} + \dot{\rho_1})\cos(\psi_5) - (\omega_x\cos(\beta + \rho_1) \\ & - \omega_z\sin(\beta + \rho_1))\sin(\psi_5), \end{aligned} \tag{74} $$

$$ \begin{aligned} \zeta^y_6 = {} & (\omega_x - \dot{\beta} + \dot{\rho_2})\cos(\psi_6) - (\omega_x\cos(\rho_2 - \beta) \\ & - \omega_z\sin(\rho_2 - \beta))\sin(\psi_6). \end{aligned} \tag{75} $$

Substituting those variables into the right sides of Equations (58)–(63) and expanding the left side using the prior equations enables us to obtain the desired wheel rate equations, whose expressions are provided in Appendix B for conciseness.

Hence we can compute the wheel rate commands to achieve a desired rover motion (body linear velocity and heading rate). However, the mobility system of the Mars rover is not designed to achieve a predetermined rover velocity, nor heading rate. Instead, it drives the wheels as fast as possible, that is setting at least one wheel to its maximum rate $\dot{\theta}_{\max}$, based on the arc being driven which defines the proportion of longitudinal motion and heading change (i.e., the ratio of $\dot{x}$ and ω_z).

Our approach is to turn the wheel rate equations around to express the rover's linear velocity (and heading rate for turns) as a function of the wheel rate commands, then set all wheel rate commands to the max value $\dot{\theta}_{\max}$, and determine the minimum absolute value of the computed rover velocity (or heading rate for turns) over all wheels. This technique allows us to determine which wheel gets to its max angular rate first and deduce the associated maximum rover velocity and heading rate.

To do this we first need to relate $\dot{z}$, ω_z, and $\dot{x}$ since a single (wheel rate) equation can only solve for a single unknown.

- For straight driving, our desired heading rate is zero:

$$ \omega_z = 0. \tag{76} $$

Thus we only need to express $\dot{z}$ as a function of $\dot{x}$.

- For turns, we chose to solve for the heading rate ω_z rather than $\dot{x}$, since the latter is zero for the special case of turns in place (where the turn radius is zero). We can use the known turn radius r to relate $\dot{x}$ to ω_z:

$$ \dot{x} = r\ \omega_z. \tag{77} $$

Then, if we can express $\dot{z}$ as a function of $\dot{x}$, like in the straight driving case, we can also express $\dot{z}$ as a function of ω_z (using Equation (77)).

To express $\dot{z}$ as a function of $\dot{x}$, we compute the linear velocity of each wheel in its contact angle frame, and leverage the fact that the z component, which depends on both $\dot{x}$ and $\dot{z}$, is zero (since by definition the contact angle frame is rotated so that the wheel velocity vector in the x-z plane is along the x axis only):

$$\forall\ i \in [1,\ 6],\ \ {}^{\eta_i}v^{z}_{A_i} = 0. \tag{78}$$

Expanding Equations (52)–(57) and applying Equation (78) above to their z components, we get

$$\forall\ i \in [1,\ 6],\ \ a_{i1}\dot{x} + a_{i2}\dot{z} + a_{i3}\omega_z = b_i, \tag{79}$$

with the intermediary variables a_{i1}, a_{i2}, a_{i3}, and b_i defined in Appendix C for conciseness.

Based on the above, we define:

- For straight driving ($\omega_z = 0$), for wheel i:

$$\dot{z} = c_i - d_i\dot{x}, \tag{80}$$

with

$$c_i = \frac{b_i}{a_{i2}}, \tag{81}$$

$$d_i = \frac{a_{i1}}{a_{i2}}. \tag{82}$$

- For turns of turn radius r ($\dot{x} = r\ \omega_z$), for wheel i:

$$\dot{z} = c_i - e_i\omega_z, \tag{83}$$

with

$$e_i = \frac{a_{i3} + ra_{i1}}{a_{i2}}. \tag{84}$$

Note that we really only need one of the equations relating $\dot{z}$ to $\dot{x}$ (from Equation (80)) and $\dot{z}$ to ω_z (from Equation (83)). However, using the equation associated with the same wheel for which we are trying to compute $\dot{x}$ and ω_z does simplify the formulas. In particular it removes some denominators, and hence the risk of dividing by zero, and makes the expressions of the remaining denominators easier to interpret (they become zero if and only if the associated wheel is perpendicular to the direction of motion).

We then use Equations (76)–(84) to substitute ω_z and $\dot{z}$ in Equations (B1)–(B5) and compute the $\dot{x}$ for straight driving, and ω_z for turns, associated with each wheel turning at its max rate dir $\dot{\theta}_{\max}$ (where dir represents the direction of rotation of the wheel).

The algorithm to calculate the commanded wheel rates can be summarized as follows:

(1) Compute the desired $\dot{x}$, $\dot{z}$, and ω_z that brings at least one wheel to its maximum rate.

1. For straight driving,

$$\begin{cases} i_{\min} = \underset{i\in[1,6]}{\arg\min}\ |\dot{x}^i|, \\ \dot{x} = \dot{x}^{i_{\min}}, \\ \dot{z} = c_{i_{\min}} - d_{i_{\min}}\dot{x},\ \text{using Equations (82)-(90),} \\ \omega_z = 0. \end{cases}$$

2. For turns,

$$\begin{cases} i_{\min} = \underset{i\in[1,6]}{\arg\min}\ |\omega_z^i|, \\ \omega_z = \omega_z^{i_{\min}}, \\ \dot{x} = r\ \omega_z, \\ \dot{z} = c_{i_{\min}} - e_{i_{\min}}\omega_z,\ \text{using Equations (84)-(90).} \end{cases}$$

(2) Calculate the wheel rate commands using Equations (B1)–(B5).

Figure 5 shows the functional block diagram associated with the implementation of the entire TRCTL algorithm.

2.8 | Simulation testing

Both 2D and 3D motion simulators were developed to verify the output of the algorithm (see Figure 6). For the 2D case, the ideal "no-slip" wheel rates were calculated from the equation of the terrain curve. For the 3D case, we used a straight-line distance approximation by settling the rover on the terrain at very short spatial intervals. In both cases, we verified that the simulated, ideal wheel rates match our algorithm's output when we remove the approximation on the linear velocity of the rover origin (see Section 2.6) and instead use the correct rover velocity, which can be computed in simulation.

Figure 7 shows a comparison of the ideal wheel rates in simulation (left), the wheel rates generated by our algorithm in simulation (middle), and the wheel rates measured on our rover testbed when commanded by our algorithm (right), for a straight arc test case where the right front wheel of the rover drives over a rock while the others remain on flat ground.

3 | PERFORMANCE IN TEST

Development and validation and verification (V&V) testing of our TRCTL algorithm was performed at Jet Propulsion Laboratory's (JPL's) Mars Yard on the VSTB and Scarecrow testbed rovers. The VSTB rover, shown in Figure 8, is the flight-like testbed that runs the same flight software as Curiosity. The mass of Curiosity and the VSTB is approximately 899 kg. The Scarecrow rover, shown in Figure 9, was developed in support of MSL mobility system

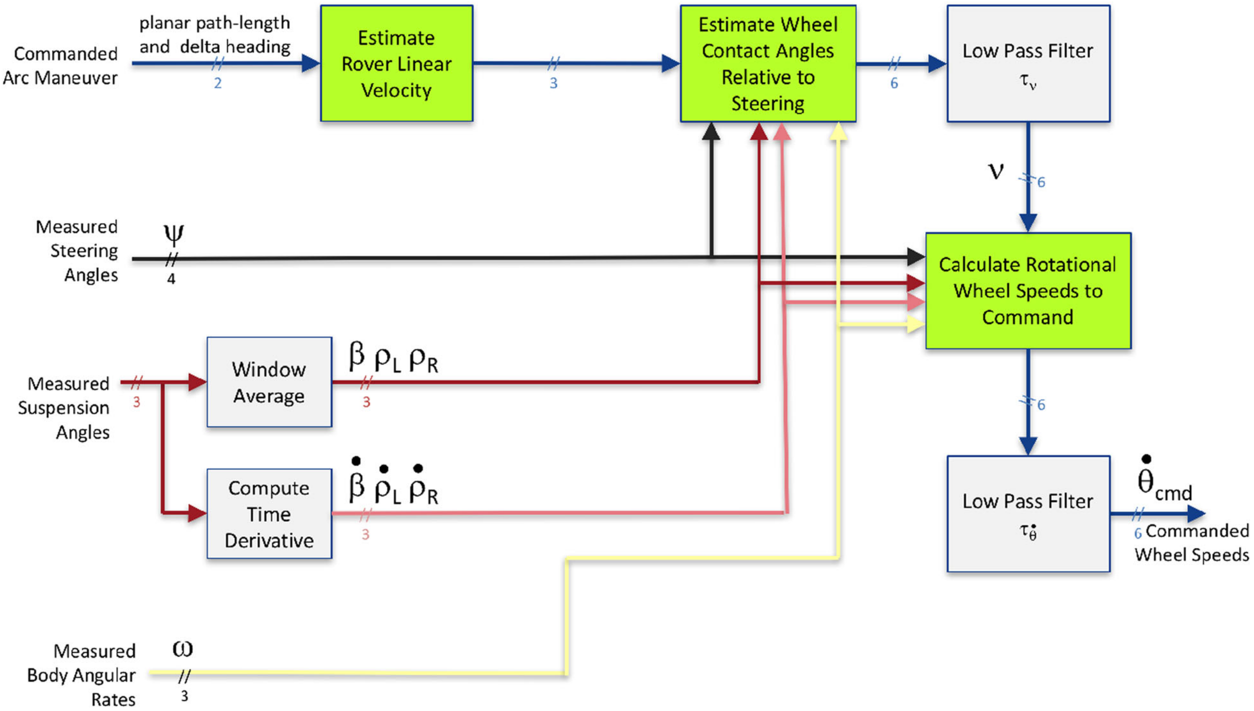
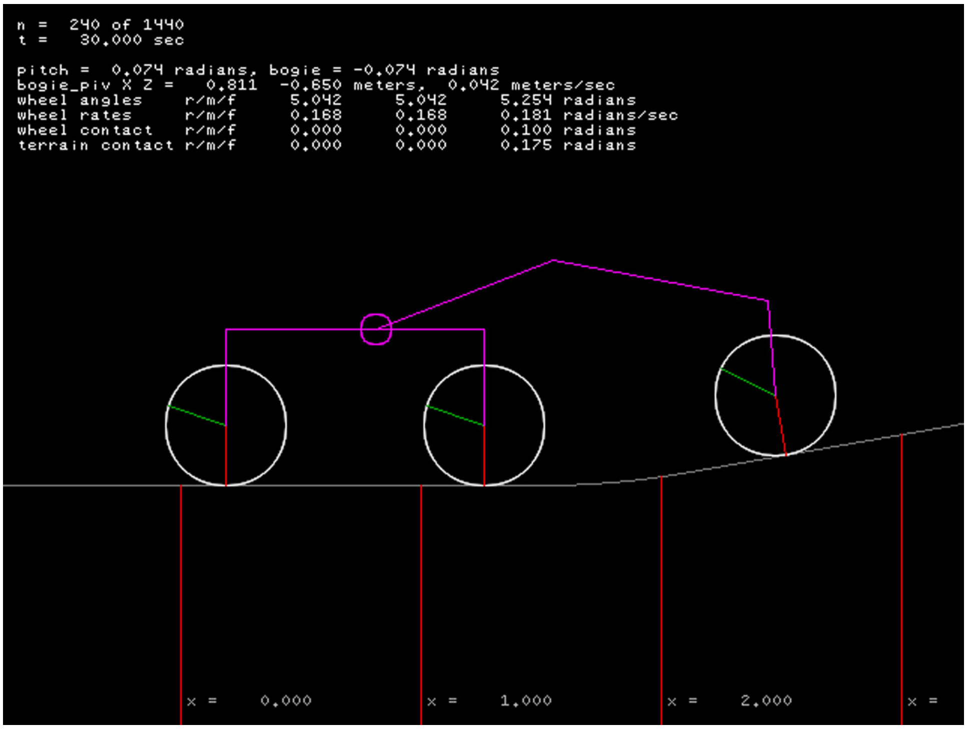


FIGURE 5 Data flow for the TRCTL algorithm implementation. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

performance evaluation. The Scarecrow rover mobility system (wheels, rocker, bogies, and differential) is kinematically identical to Curiosity and the VSTB and has a mass of 318 kg. This reduced mass yields a system that weighs approximately the same on Earth as Curiosity does on Mars, enabling realistic mobility tests to be performed at JPL's Mars Yard, inside test facilities, and at remote locations that have some Mars-like terrain (e.g., sand dunes at Dumont Dunes, California). Most of the TRCTL testing related to the physical response to terrain was performed on Scarecrow. The VSTB was primarily used to validate the TRCTL flight software.

Two load measurement systems were used throughout TRCTL testing; a ground load cell attached to the bottom of a hemispherical


```
n =  240 of 1440
t =   30.000 sec

pitch =  0.074 radians, bogie = -0.074 radians
bogie_piv X Z =   0.811  -0.650 meters,  0.042 meters/sec
wheel angles    r/m/f      5.042      5.042      5.254 radians
wheel rates     r/m/f      0.168      0.168      0.181 radians/sec
wheel contact   r/m/f      0.000      0.000      0.100 radians
terrain contact r/m/f      0.000      0.000      0.175 radians
```

FIGURE 6 Two-dimensional simulator used to validate our algorithm. The rover's forward direction is to the right [Color figure can be viewed at wileyonlinelibrary.com]

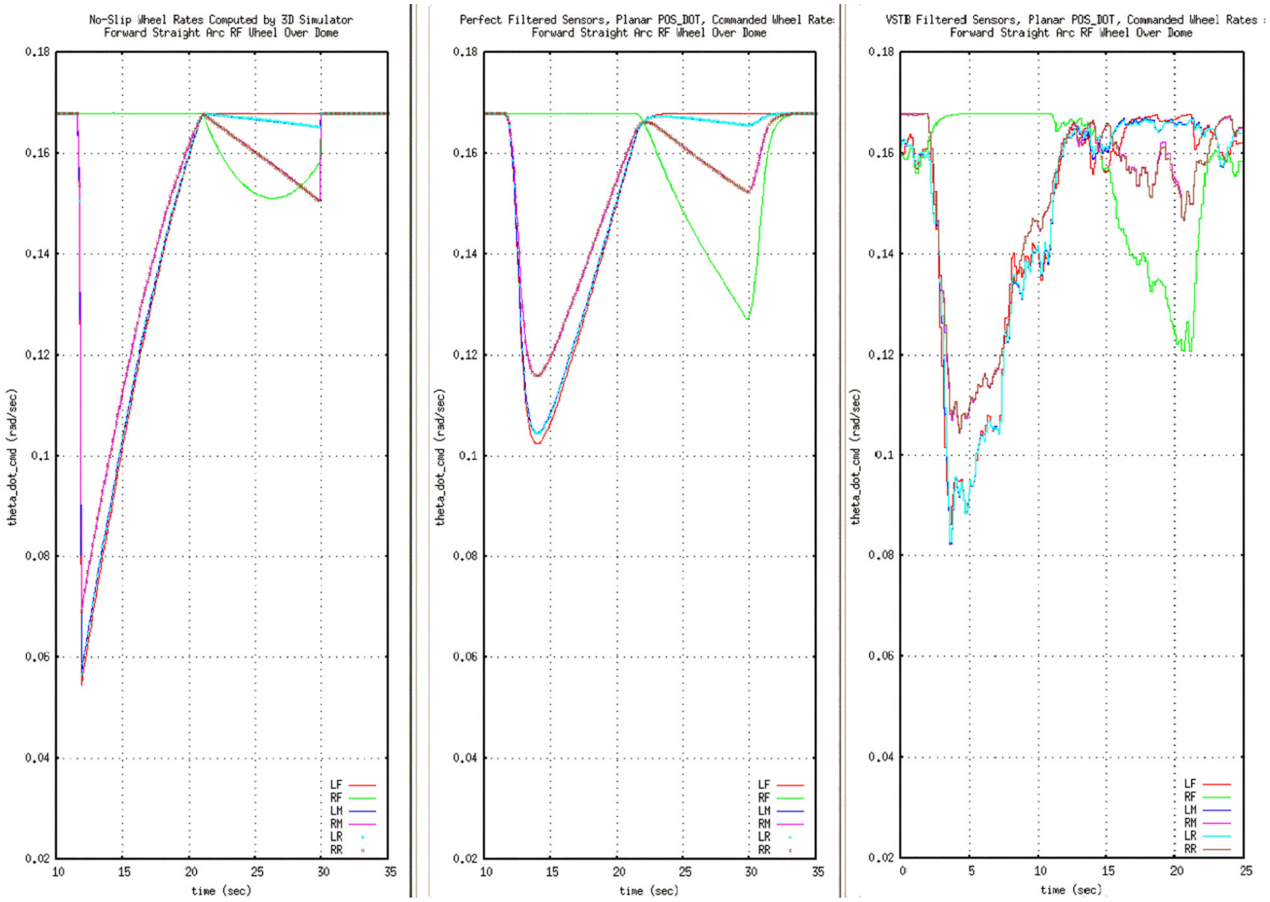


FIGURE 7 Comparison of the wheel rates obtained in simulation with perfect knowledge (left), output by our algorithm in simulation (middle), and measured in the rover testbed (VSTB) when using our algorithm (right) [Color figure can be viewed at wileyonlinelibrary.com]

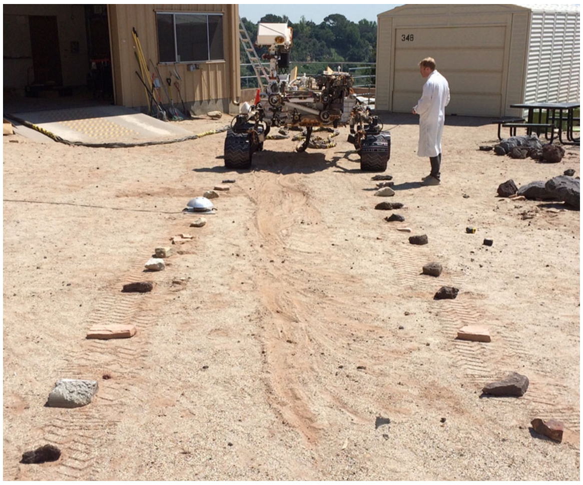


FIGURE 8 TRCTL testing in the JPL Mars Yard on the VSTB testbed. One of the objects driven over on the left side is a ground anchored FTS with an aluminum hemisphere dome mounted on top of it. FTS, force-torque sensor; JPL, Jet Propulsion Laboratory; TRCTL, our algorithm inspired by traction control; VSTB, Vehicle System Testbed [Color figure can be viewed at wileyonlinelibrary.com]

dome, and a hub-mounted load cell attached to the center of each of Scarecrow's starboard side wheels. The hemispherical dome functioned as part of the ground load measurement system and a terrain feature. When a wheel ascends and descends the dome, it follows a continuous arc instead of a step function. The hemisphere was constructed of aluminum and has a radius of 15 cm. An aluminum plate attached to the bottom of the load cell was anchored to the ground by four 30.5 cm stakes, yielding a rigid structure with negligible flexure when a wheel traverses the dome. The ATI Omega 160 six-axis force-torque sensor was used for both the ground and hub-mounted load cell.

The majority of load data gathered during TRCTL development testing were collected to enable the drawing of qualitative conclusions regarding the efficacy of TRCTL. That is, due to the complexity and range of all possible terrain types, terrain geometry, drive arc curvature, drive commands, and drive actuator failure conditions Curiosity may encounter over the course of its remaining life, we did not attempt to quantify the precise load reduction on one, or a small subset of scenarios. Rather, the test philosophy adopted here was to perform a small number of trials on a wide array of terrains under various driving conditions such that the expected average load reduction could be approximated.

Test scenarios for TRCTL enabled and disabled included:

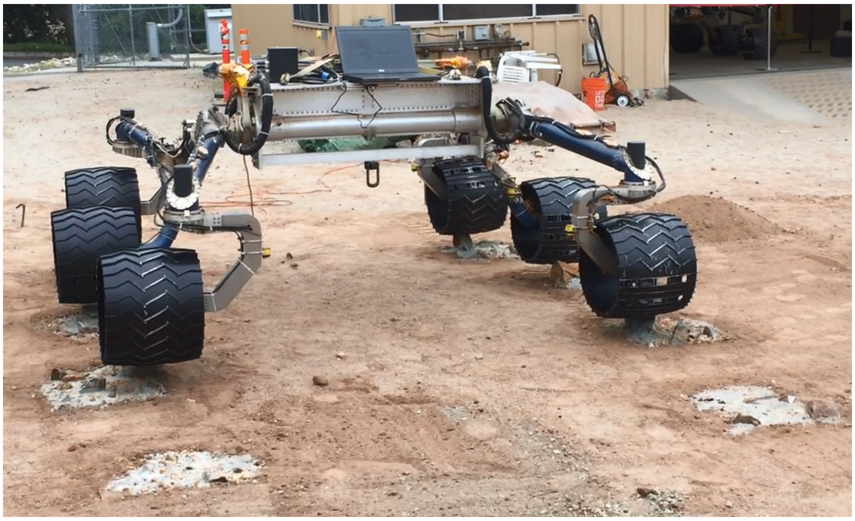


FIGURE 9 The Scarecrow testbed rover on a complex-terrain test course in the JPL Mars Yard. Eighteen cement tiles containing embedded rocks were installed at 1 m intervals along the paths of the wheels. The tile spacing on the left and right sides was the same to prevent wheels from missing some tiles due to yawing off course. JPL, Jet Propulsion Laboratory [Color figure can be viewed at wileyonlinelibrary.com]

1. straight, shallow, medium, and turn-in-place arcs on benign terrain, and over the dome,
2. driving over sand dunes with all wheels and a subset of wheels,
3. driving up and down a sand incline,
4. driving over a speed bump constructed with sandbags,
5. driving up, down, and across slopes of up to 20°,
6. driving up and down a 9° rigid ramp,
7. driving using high-level commands to go to a waypoint,
8. autonomous navigation on benign terrain and in the presence of non-traversable obstacles,
9. driving with a disabled wheel, and
10. driving on complex terrain.

The term “complex terrain” is used to describe a series of decimeter-scale rocks, which when traversed, yield complex vehicle-terrain interaction in which measurements made by the inertial measurement unit (IMU) and suspension resolvers are a function of multiple ground interactions. A complex terrain test course was constructed with rigid, high-friction cement tiles containing rocks of various sizes. The cement tiles were submerged so that their top surface was level with the ground. The Scarecrow rover was commanded to perform a 10 m arc across the tiles, which were spaced at approximately 1 m intervals. The tiles were placed such that the rocks being traversed by either the right or left sides of the vehicle were approximately symmetric (Figure 8). This course was driven three times with TRCTL enabled and three times with TRCTL disabled. Wheels with a larger skin thickness were used to prevent wheel punctures, so that the condition of each wheel was the same for every run.

Example RR wheel results are shown in Figures 10 and 11. These figures show the cumulative frequency distribution of loads and drive torques during TRCTL enabled and disabled runs. Green represents TRCTL enabled, while red represents disabled. The leftward shift in

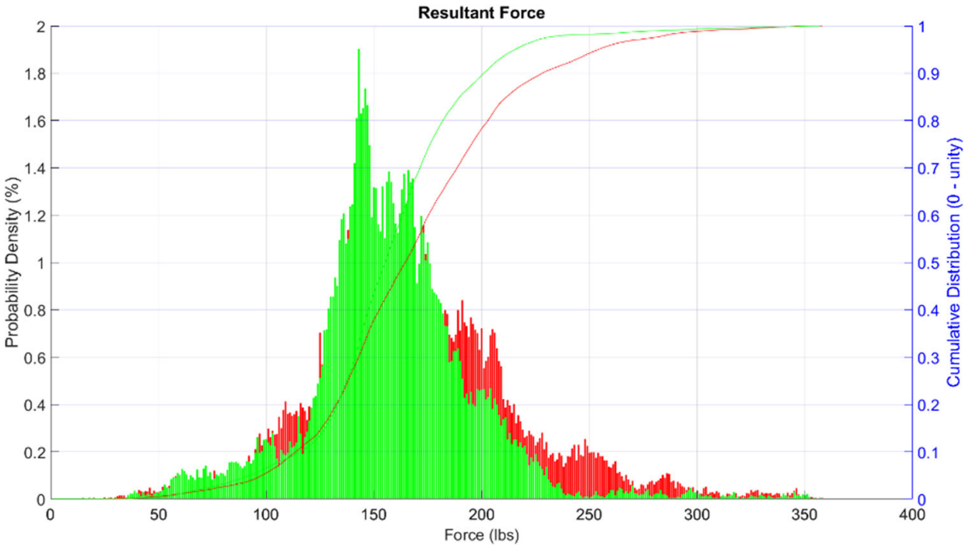


FIGURE 10 Cumulative frequency distribution of resultant loads on Scarecrow RR wheel during traverse of high friction, embedded rock tiles. Green and red correspond to TRCTL enabled and disabled, respectively. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

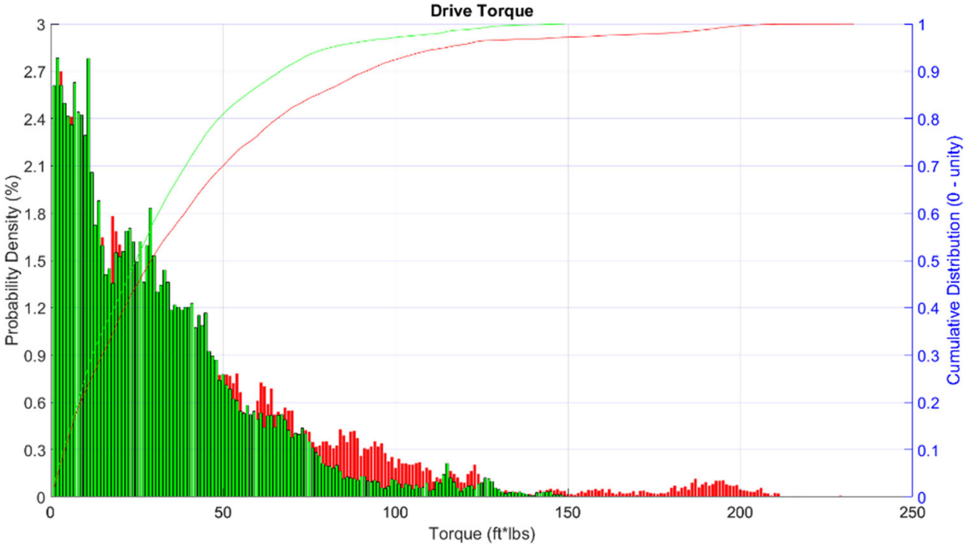
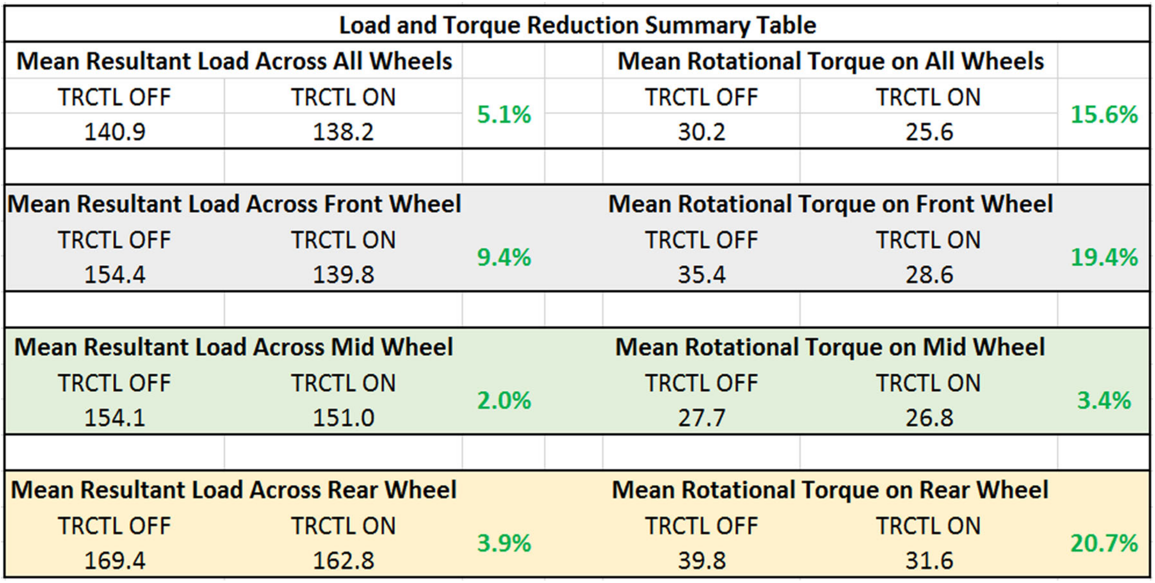


FIGURE 11 Cumulative distribution of drive torques experienced by Scarecrow RR wheel during traverse of high friction, embedded rock tiles. Green and red correspond to TRCTL enabled and disabled, respectively. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

the cumulative distribution function curve clearly identifies a reduction in the integral of both histograms, demonstrating a lowering of the forces and required drive torques when using TRCTL.

The summary in Figure 12 details the average reduction of loads and drive torques during the Scarecrow rover testing on the complex-terrain test course. As seen throughout development, TRCTL provides a modest yet consistent reduction in wheel loading. Drive actuator torques were also significantly reduced. Other secondary benefits observed during TRCTL development testing were a modest reduction in unintended yaw and wheel slip in some terrain types. From testing on the VSTB, the costs of driving with TRCTL enabled were up to a 23.7% longer traverse rate and an increase in telemetry volume of between 80% on flat ground and 129.0% on complex terrain where the suspension was constantly changing (Toupet et al., 2018).

During development testing, it was observed that contact angles become noisy when the instantaneous center of rotation about which the rover is turning is close to a middle wheel. A software modification was made to the TRCTL algorithm to use non-TRCTL command angular rates while the distance between the instantaneous center of rotation and the vertical axis through the center of a middle wheel is less than a threshold that represents the smallest turning radius at middle wheels for which TRCTL is allowed. This threshold is a changeable parameter; thus far, it has always been set to 0.5 m in flight. The net effect is very sharp arcs, which are rarely used, are executed with non-TRCTL



Load and Torque Reduction Summary Table					
Mean Resultant Load Across All Wheels			Mean Rotational Torque on All Wheels		
TRCTL OFF	TRCTL ON	5.1%	TRCTL OFF	TRCTL ON	15.6%
140.9	138.2		30.2	25.6	
Mean Resultant Load Across Front Wheel			Mean Rotational Torque on Front Wheel		
TRCTL OFF	TRCTL ON	9.4%	TRCTL OFF	TRCTL ON	19.4%
154.4	139.8		35.4	28.6	
Mean Resultant Load Across Mid Wheel			Mean Rotational Torque on Mid Wheel		
TRCTL OFF	TRCTL ON	2.0%	TRCTL OFF	TRCTL ON	3.4%
154.1	151.0		27.7	26.8	
Mean Resultant Load Across Rear Wheel			Mean Rotational Torque on Rear Wheel		
TRCTL OFF	TRCTL ON	3.9%	TRCTL OFF	TRCTL ON	20.7%
169.4	162.8		39.8	31.6	

FIGURE 12 Summary of the reduction on average load and drive torques over high-friction tiles. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

command angular rates. This has no impact on turn-in-place arcs, since the middle wheels are more than 0.5 m from the rover center.

In certain terrain scenarios, a wheelie can occur. For example, on Sol 313, a LM wheelie occurred when straightening the wheels. When a wheelie occurs, there is an increase in the loads on the wheels that remain in contact with the terrain. In addition, wheelies could potentially grow until a suspension limit is exceeded, ending the drive. During development testing on an indoor complex-terrain test course, it was observed that the original TRCTL algorithm could exacerbate wheelies in uneven and unmodifiable terrain. A wheelie detection/correction routine (called Wheelie Suppression) was added to the TRCTL algorithm for middle and rear wheels.

A wheelie is detected when the angle and rate of a bogie arm exceed thresholds, while a wheel motor current magnitude on that bogie arm is below a threshold. To release a wheelie, the speed of the other wheel on that bogie arm is adjusted. As illustrated in Figure 13, in all three trials with TRCTL enabled and Wheelie Suppression disabled, a particular section of the test course caused a LM wheelie that did not release. The LM wheelie occurred when the LM wheel had just climbed a rock while the LF wheel was pushed up against an embedded rock and the LR wheel was on fairly flat terrain. In three trials with TRCTL and Wheelie Suppression enabled, the wheelie released. No wheelies were observed during testing on the test courses in Figures 8 and 9 with TRCTL and Wheelie Suppression enabled.

4 | INTEGRATION INTO MISSION OPERATIONS

The development of TRCTL software was completed long after the most recent update to Curiosity's flight software. Thus it was incorporated into the MSL flight system as a hot patch applied to the current (R12) version of flight software. We assessed the impact the new behavior would have on the general planning model (each day's plan predicts power, data, and duration of all activities), as well as the changes imposed on the creation and validation of sequences of drive commands.

4.1 | Flight software integration

MSL has fully updated its onboard flight software (FSW) four times since launch. The most recent full update was the R12 release, which was built in August 2014 (before completion of this algorithm) and deployed in January 2015. The R12 release incorporated updates that simplified the later integration of TRCTL. Those updates included: Software hooks added to invoke a function pointer to evaluate drive wheel rates at 8 Hz; extra placeholder integer and floating point parameters; a new Data Product Object downlink record type with placeholders for a small number of integer and floating point downlink values; and new commands to enable and disable the future capability.

Once the new software was completed, it was compiled into a single object file following standard protocols for MSL FSW hot patches (Benowitz & Maimone, 2015). In each bootup period that requires it, this file is loaded into the VxWorks operating system and assigned to a function pointer global variable via a shell command script. TRCTL then becomes available for use during the remainder of the current boot cycle.

Incorporating this software as a patch allowed us to take advantage of this capability relatively quickly. MSL project procedures require thorough regression testing and planning across all FSW when updating the whole FSW image, whereas a patch only requires regression test of mobility capabilities. The patch is also hundreds of times smaller than a full FSW load would be, requiring fewer days to uplink to Mars and reducing the impact to the operations schedule.

4.2 | Mission planning integration

Several aspects of operations planning were impacted by the ongoing use of this new software. The MSL ground operations

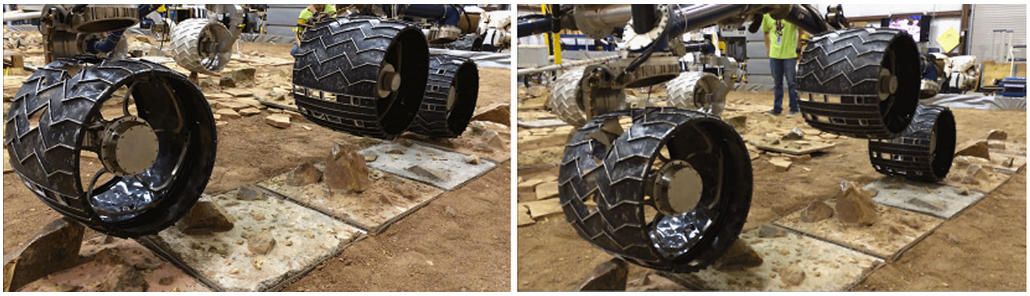


FIGURE 13 A scenario on a complex-terrain test course where a LM wheelie was repeatable. The left image shows the highest LM wheel height during TRCTL with Wheelie Suppression enabled. The right image shows the LM wheel height at the comparable rover position during TRCTL with Wheelie Suppression disabled. The wheelie continued to grow until the operator stopped the drive. This test was performed with wheels having the same skin thickness as Curiosity. Some of the wheel damage was from previous experimentation. Each run on this test course resulted in additional wheel damage. After this test, wheels were purchased with thicker skin; the repeatability test in Figure 9 was subsequently performed with those wheels. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

team models the duration, power, and data volume usage of all onboard activities. TRCTL software is active during nearly all drive modes, so the resource models for all drive activities were updated.

Based on the testing we performed on the Earth, we predicted that TRCTL would slow down drives by no more than 25%. Our initial drives on Mars were planned taking this worst-case model into account. But while the actual slowdown is terrain-dependent (the more uneven the terrain, the greater the overall slowdown), in practice we have observed slowdowns only on the order of 10% and have since updated our models to presume no more than 15%. Power modeling was not changed, except for the implication that longer drives would require more CPU and IMU energy overall: This implementation adds less than 3% to the total CPU usage while driving. But in terms of data volume, the extra data generated by the initial implementation was significant; it typically nearly doubles the amount of motion history telemetry collected due to the records of individual wheel speed commands being collected at 8 Hz.

All MSL drive command sequences are constructed by the Rover Planner team, so their environment was updated to support nominal use of this new capability. Standard startup command sequences were updated to load and enable the software patch, removing the need to explicitly remember to turn it on for each drive. The command sequence static analyzer RP-check (Maimone, Maxwell, Biesiadecki, & Algermissen, 2018) was updated to issue a warning for any drive commanded without TRCTL being enabled. The rover sequencing and visualization program (RSVP; Wright et al., 2006) surface simulation component was updated to accept an average expected TRCTL speed ratio (defined below) as input and adjust planned drive durations accordingly. And strategic downlink analysis tools were updated to enable mobility downlink and Rover Planner team members to quickly measure the actual slowdown seen over any number of recent drives, so they could apply it to the duration estimate needed for the current drive.

4.3 | Downlink inputs to tactical planning

TRCTL software generates downlink telemetry to enable ground understanding of its performance. Section 5 will cover our detailed downlink assessment capabilities, and Table 1 shows a list of all the values available for daily and long-term trending queries. These fields are populated within the MSL Strategic Mobility database, an update to the comparable system used for the Mars Exploration Rover Spirit

TABLE 1 Strategic database names for high-rate TRCTL downlink data (collected at 8 Hz)

Field	Description	Units
f_trctl_active	Active	Boolean
f_trctl_algorithm	Algorithm	0_ADAPTIVE, 1_PLANAR
f_trctl_control_mode	Mode	0_RATE, 1_POS
f_trctl_telem_mode	Orientation Telemetry (otlm) Mode	0_WORLD, 1_ROVER
f_trctl_speed_ratio	Ratio of Commanded to Ackermann Reference Speed	Float between [0, 1]
f_differential_rate	Differential Angular Velocity	rad/s
f_bogie_l_rate	Bogie L Angular Velocity	rad/s
f_bogie_r_rate	Bogie R Angular Velocity	rad/s
f_wheelie_lm	Left Middle (LM) Wheelie Correction Active	Boolean
f_wheelie_lr	Left Rear (LR) Wheelie Correction Active	Boolean
f_wheelie_rm	Right Middle (RM) Wheelie Correction Active	Boolean
f_wheelie_rr	Right Rear (RR) Wheelie Correction Active	Boolean
f_world_x_rate	Local Level Frame X Rate	rad/s
f_world_y_rate	Local Level Frame Y Rate	rad/s
f_world_z_rate	Local Level Frame Z Rate	rad/s
f_rover_x_rate	Rover Frame X Rate	rad/s
f_rover_y_rate	Rover Frame Y Rate	rad/s
f_rover_z_rate	Rover Frame Z Rate	rad/s
Fields that are repeated for each wheel: left/right, front/middle/rear		
f_contact_angle_(lf, lm, lr, rf, rm, rr)	Wheel Contact Angle	rad
f_theta_dot_cmd_(lf, lm, lr, rf, rm, rr)	Wheel Commanded Rotational Speed	rad/s
f_wheel_rate_cmd_(lf, lm, lr, rf, rm, rr)	Wheel Commanded Linear Speed	cm/s
f_wheel_rate_ack_(lf, lm, lr, rf, rm, rr)	Ackermann Wheel Reference Linear Speed	cm/s
f_contact_fail_(lf, lm, lr, rf, rm, rr)	Wheel Contact Angle Computation Failed	Boolean
f_rate_fail_(lf, lm, lr, rf, rm, rr)	Wheel Rate Computation Failed	Boolean

Abbreviation: TRCTL, our algorithm inspired by traction control.

and Opportunity missions (Biesiadecki, Liebersbach, & Maimone, 2008).

For planning purposes, the primary number of interest is the TRCTL speed ratio, defined as follows. For any given 8 Hz sample, we know the speed at which each wheel is commanded to drive. We can also compute the equivalent speed that would have been commanded by the planar (double Ackermann) style non-TRCTL driving algorithm. To estimate the overall progress of the vehicle, we compute the ratio of commanded speed/Ackermann speed for each wheel, disregard the slowest wheel (which might be commanded at 0 m/s if it rests on the center of the turning circle) then pick the median of the five remaining values. Figure 14 shows this TRCTL speed ratio evaluated at 8 Hz during drive operations in 2,545 samples from Sol 1814.

Rover Planners use the average speed ratio over any given range of sols to inform their duration estimate for the next drive. Overall, the average speed ratio during the first 6 months of operations from sol 1646 through 1822 is 0.899 from 351,992 samples. That is, TRCTL tends to drive 10% slower than planar drives in the current terrain.

However, the net impact on the whole system has been smaller. MSL typically uses onboard image processing to refine its position estimate, using a VO algorithm (Johnson, Goldberg, Cheng, & Matthies, 2008; Maimone, Cheng, & Matthies, 2007). MSL typically drives long distances in 1 m steps, stopping after each step to acquire new VO images and process them. The time needed to stop and process images is nearly double the time needed to drive the 1 m step, which means the time spent physically moving is less than 40% of the total drive activity; hence, the slowdown impact of TRCTL on VO-enabled drives is less than 4% on average.

5 | ASSESSMENT TOOLS

Motion history data products are generated whenever rover mobility is commanded. Since R12, these data products record a subset of the estimates and control signals computed by the TRCTL algorithm, when it is enabled. A suite of analysis tools was developed to process mobility data products and assess the performance of TRCTL in both the ground testing and flight phases. As per Figure 4, of particular interest are time-series plots of the following quantities: rover attitude angles and rates, suspension (bogie and differential) angles and calculated rates, estimated contact angles for each wheel, and commanded angular velocity for each drive actuator. Where possible, the magnitude axes were pinned for informative comparison across all drives.

During ground testing, the quality of the contact angle estimates was assessed relative to both the configured obstacle environment and the commanded motion. For example, driving on flat terrain should result in small and zero-mean estimated contact angles, whereas the front left wheel driving over a hemisphere should (all else being equal) result in a clear signature for that wheel only. In flight, a comparison is possible only with respect to a computed mesh, which has its own error sources; however, any unrealistic contact angle estimates that differ significantly from terrain models would be observable. The commanded angular velocities, which are the TRCTL outputs, can be assessed for consistency with the joint ego-motion estimate, including contact angles, attitude angles and rates, and suspension angles and rates. The juxtaposition of the contact angle and angular rate plots was crucial for algorithm assessment.

Also analyzed were time-series plots for the current estimated by the motor controller flight software to be supplied to each drive actuator in response to the wheel angular rates commanded by TRCTL. These plots were used to verify that TRCTL did not significantly increase the current demanded by the drive actuators during mobility activities. For each drive actuator, the current, voltage, and wheel contact angle estimates were also used to plot modeled torque on each wheel. Because torque is indirectly commanded via wheel rate, correspondence between the shape of the contact angle estimate curves and the drive torque curves would

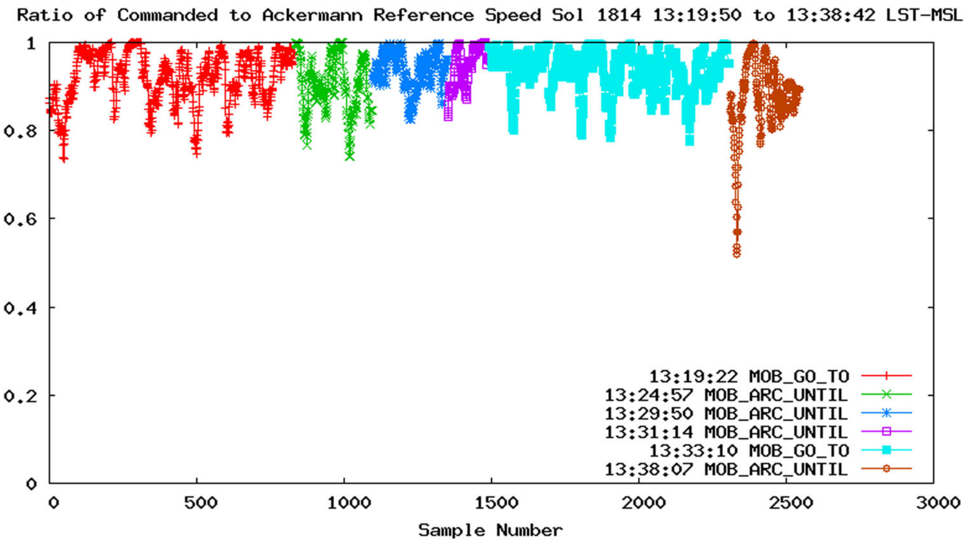


FIGURE 14 The average TRCTL speed ratio for Sol 1814 is 0.924 from 2,545 samples. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

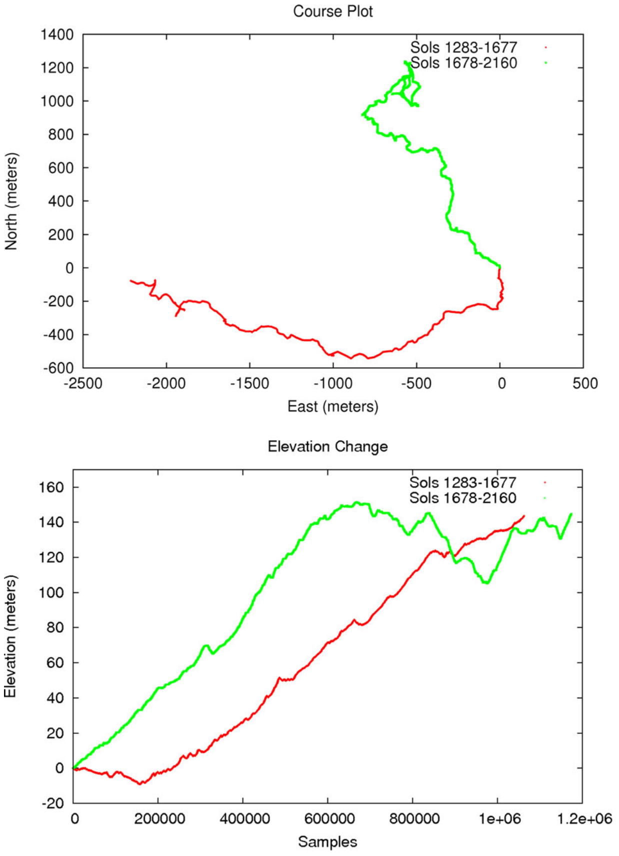


FIGURE 15 The path of the rover in the XY plane (upper plot) and the change in rover elevation (lower plot) for the TRCTL disabled and enabled periods, relative to the initial rover position for both periods. In this and subsequent figures, red indicates TRCTL disabled and green indicates enabled. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

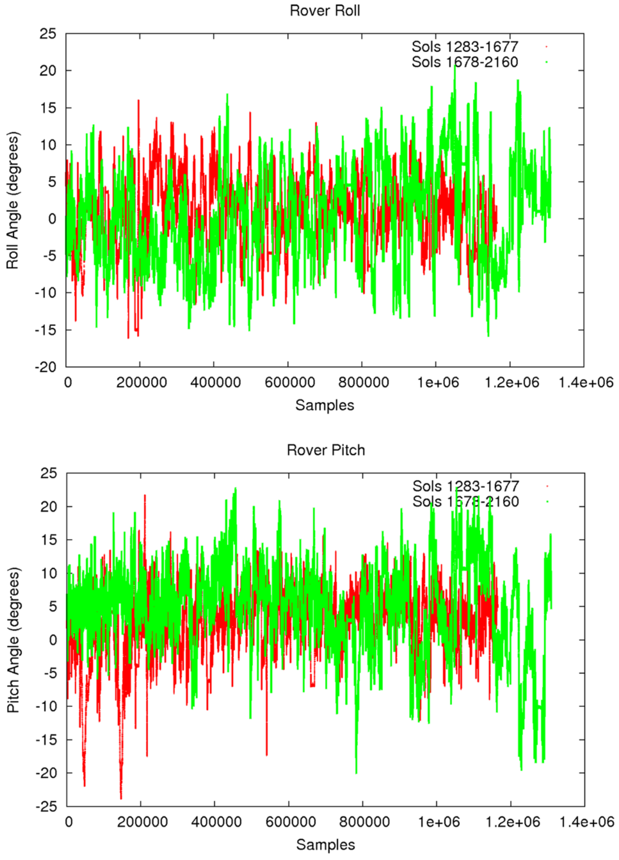


FIGURE 16 Rover roll and pitch angles for the 3.6 km TRCTL disabled and enabled periods. To reduce the data for plotting purposes, data points within 0.0005° of the last plotted data point were discarded. This accounts for the differences in number of samples. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

be a measure of the efficiency of the algorithm when scaling obstacles and is the subject of ongoing investigation.

After free-floating wheelies on the middle wheels were induced in some test configurations, wheelie detection and suppression logic was added, with indicator variables for the wheelie statuses being exposed in telemetry. The combination of these time-series plots was helpful in formulating and verifying the wheelie detection and suppression strategies.

6 | PERFORMANCE IN FLIGHT

Three checkout tests were performed on Mars after software delivery was approved in March 2017. For checkout test 1, performed on Sol 1644, the software patch was uploaded to Curiosity and initial parameters were set, saved in nonvolatile memory, and recorded in a drive module data product. After the drive module data product was downlinked, the operations team verified the parameters were correctly saved onboard the rover. For checkout test 2, performed on Sol 1646, a short (5 m) drive was executed with TRCTL enabled, logging drive telemetry at 64 Hz. After assessment of the telemetry by the operations team, the team proceeded to checkout test 3. In this test, performed on Sol 1662, a 20-m drive was executed with TRCTL enabled, again logging drive telemetry at 64 Hz. Following a review of checkout test telemetry with MSL management, the new capability was approved for nominal use on Curiosity in April 2017.

Since its first nominal use on Sol 1678, Curiosity has driven 3.587 km in 149 drives, as of September 3, 2018, 99.38% of it with TRCTL enabled. Initial flight results were reported in Toupet et al. (2018). The exceptions are seven approximately 1.3-m drive segments on Sols 1682, 1730, 1798, 1887, 1989, 2030, and 2115 to perform full MAHLI wheel imaging (FMWI), 2.5 m at the beginning of the Sol 1787 drive when recovering from a drive fault on the previous sol that left the right bogie close to its soft limit, a 10.09 m drive leg on Sol 1800 that was similar to terrain where a drive fault had previously occurred due to a TRCTL timeout, 7.5 cm on Sol 1846 during the recovery from a right rear steer actuator stall on Sol 1843, and 1.3 m on Sol 2030 to undo a FMWI drive before turning for a TRCTL enabled drive. FMWI is currently performed every 500 m to assess changes in damage to the wheels. TRCTL is disabled during FMWI to ensure that each wheel will rotate the same amount, so that the full wheel surface can be imaged in the fewest number of steps. Since there are no sensors on Curiosity to measure the loads on wheels,

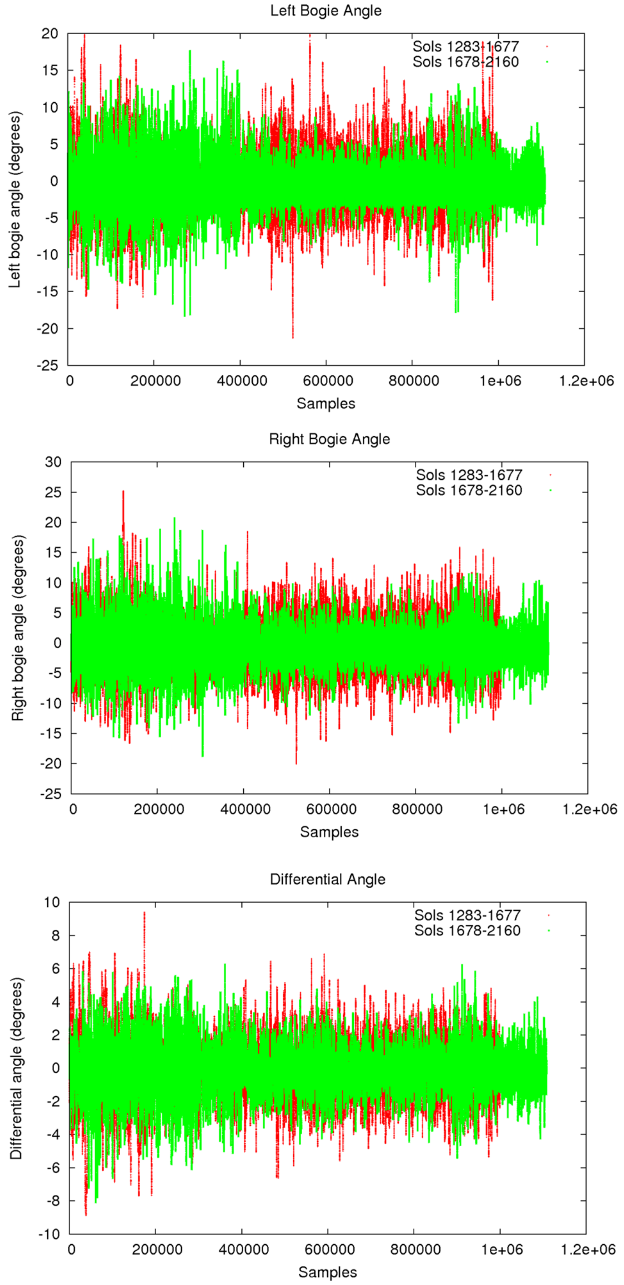


FIGURE 17 Suspension angles for the 3.6 km TRCTL disabled and enabled periods. To reduce the data for plotting purposes, data points within 0.0005° of the last plotted data point were discarded. This accounts for the differences in number of samples. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

the operations team monitors wheel currents as an indicator of how much work each wheel is doing. During Earth-based TRCTL testing, wheel currents were compared for TRCTL enabled and disabled trials by returning to a start position and traversing the same terrain. However, the high priority on achieving science objectives on Mars precludes performing such a test on Martian terrain. Instead, we compare drive currents from before and after starting nominal use of TRCTL, recognizing there may be differences in terrain type.

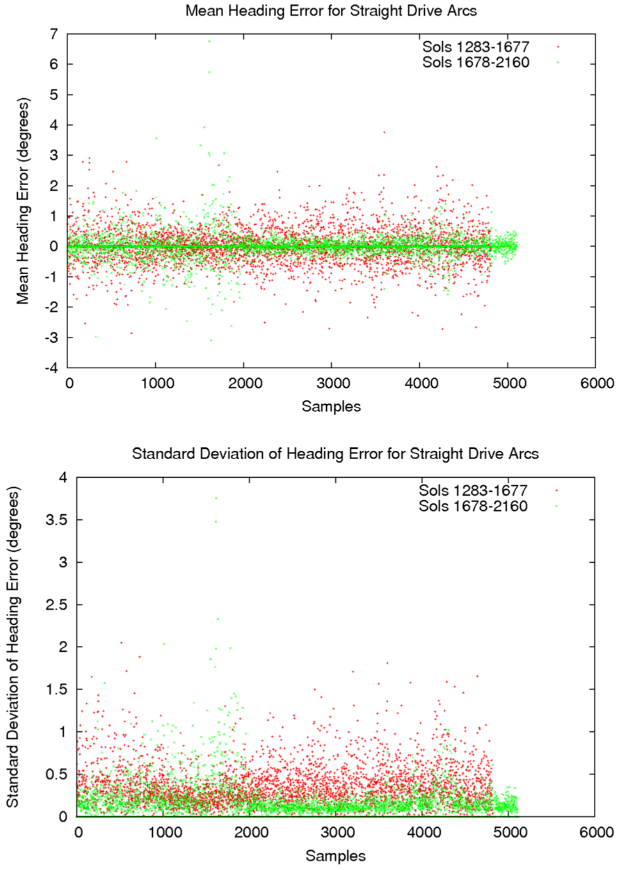


FIGURE 18 Mean and standard deviation of heading error for each straight arc commanded during the 3.6 km TRCTL disabled and enabled periods. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

A comparison has been performed of approximately 3.6 km of driving before and after TRCTL nominal use started. The northeast course plot in Figure 15 indicates Curiosity mostly drove westward during the TRCTL disabled period and northward during the TRCTL enabled period. Near the end of the TRCTL enabled period, the cluster of points indicates driving was largely performed in a local area of interest for science activities. A terrain unit map manually generated from HiRISE data (1 m resolution) and images acquired by Curiosity provided the differences in terrain type between the two 3.6 km periods. For the TRCTL disabled period, the percentages of odometry Curiosity traversed over smooth, ridged, fractured, sand pit and ripple fields, and sand dune terrain was 0.5%, 31.6%, 60.0%, 4.3%, and 3.6%, respectively. For the TRCTL enabled period, the percentages of odometry Curiosity traversed over pitted, fractured, and sandy pits and ripple fields terrain was 49.1%, 49.6%, and 1.3%, respectively (Condus & Arvidson, 2018).

Rover driving can be performed in the forward direction, in the backward direction, or by turning in place. Turning in place is commonly used on benign terrain (a) after the rover reaches a waypoint, to align itself with the next waypoint, or (b) after the rover has reached the drive goal position, to turn to a heading optimal for transmitting telemetry to a Mars orbiter or Earth. Turning in place does not use TRCTL, regardless of

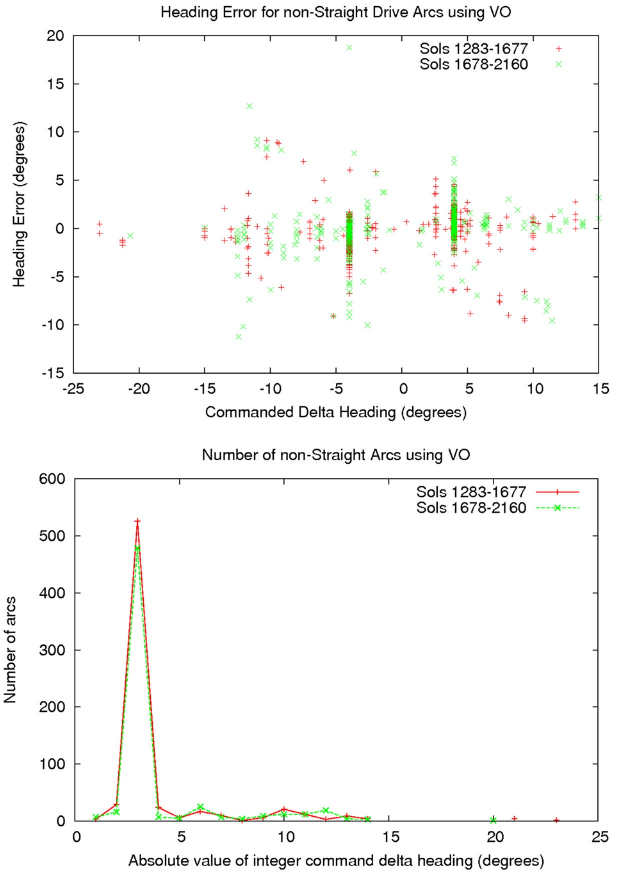


FIGURE 19 For nonstraight arcs using VO that were executed during the TRCTL enabled and disabled periods, here is the heading error (top) and number of arcs (bottom) versus command delta heading. Heading error is not correlated with command delta heading. The nonstraight arc most commonly commanded had a command delta heading of +/−3°, when reduced to an integer. TRCTL, our algorithm inspired by traction control; VO, visual odometry [Color figure can be viewed at wileyonlinelibrary.com]

whether TRCTL is disabled or enabled. Steering on harsh terrain can cause more wheel damage than driving on harsh terrain. Since turning in place requires steering the outer four wheels, turning in place is limited to mostly benign terrain.

Although backwards driving has been shown to reduce the forces on the middle and front wheels, forward driving is much more common than backward driving for two primary reasons related to the desire to limit turning in place. First, transitioning between a forward drive and a backward drive in the most efficient manner typically requires turning in place. Secondly, a turn in place is typically needed at the end of a backward drive to perform postdrive imaging in the direction of the next drive. Postdrive imaging is performed to update the ground-based terrain mesh used to plan the next drive. A turn in place is typically needed after a backward drive because the rover body occludes nearby terrain when imaging backwards with the engineering cameras on the mast.

During the TRCTL disabled period, the percentage of forward, backward, and turn-in-place driving was 88.3%, 3.8%, and 7.9%, respectively. During the TRCTL enabled period, the percentage of forward, backward, and turn-in-place driving was 79.1%, 14.5%, and 6.4% respectively. These percentages were derived by processing the angular rate data for a left and right wheel. Positive angular rates for a left and right wheel correspond to forward driving, negative angular rates for a left and right wheel correspond to backward driving, and left and right wheel angular rates with a different sign correspond to turning in place.

Figure 15 contains a plot of the change in elevation for the two periods, relative to the initial elevation for each period. Figure 16 contains plots of rover roll and pitch angles for the two periods. Although the increase in rover elevation over the two periods was identical, the average rover pitch was 2.2° higher for the TRCTL enabled period, where positive pitch corresponds to uphill driving. During forward and backward driving, the average rover pitch was 1.8° and 4.1° higher for the TRCTL enabled period, respectively. That is, the terrain had a slightly steeper uphill incline during forward driving and a slightly steeper downhill incline during backward driving. The average rover roll angle for the TRCTL disabled period was 0.5° higher than for the TRCTL enabled period.

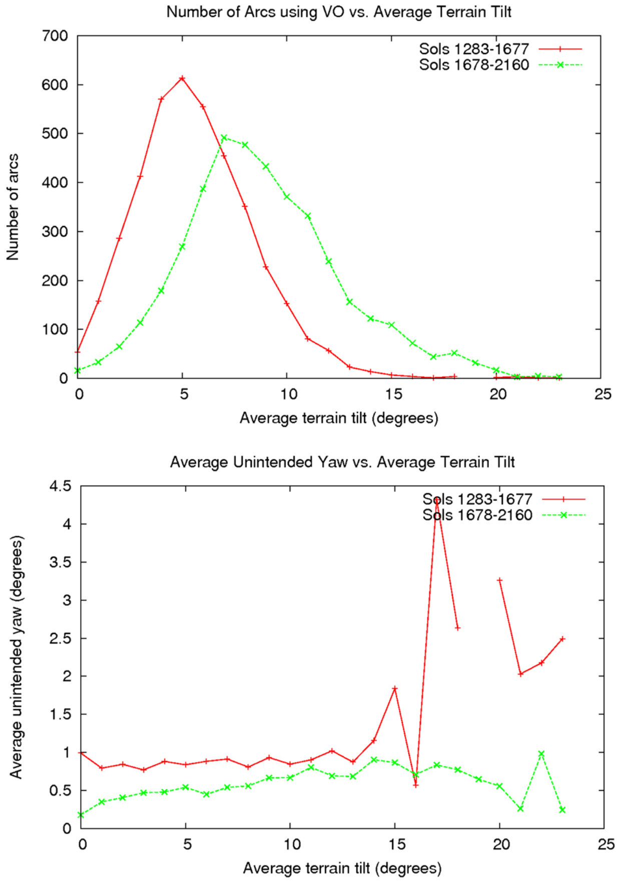


FIGURE 20 Number of arcs using VO and average unintended yaw versus average terrain tilt for non-turn-in-place arcs executed during the TRCTL disabled and enabled periods. The increase in unintended yaw at higher terrain tilt for the TRCTL disabled period may be due to single-event outliers. More data are needed to draw conclusions about unintended yaw differences between TRCTL disabled and enabled at average rover tilts of larger than 15°. TRCTL, our algorithm inspired by traction control; VO, visual odometry [Color figure can be viewed at wileyonlinelibrary.com]

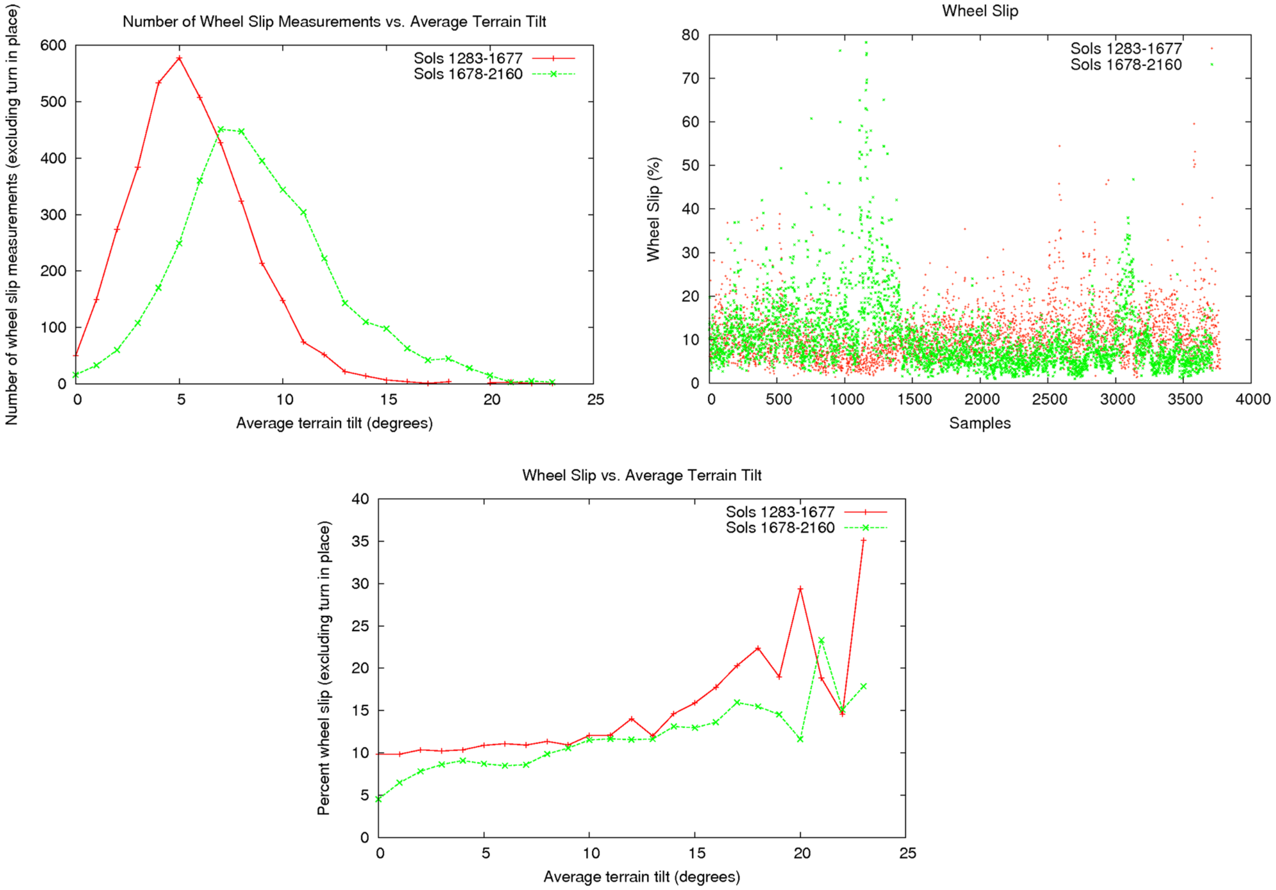


FIGURE 21 Number of wheel slip measurements versus average terrain tilt, raw wheel slip measurements, and average wheel slip versus average terrain tilt for non-turn-in-place arcs greater than 0.35 m, executed during the TRCTL disabled and enabled periods. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

Figure 17 contains rover suspension angles for the 3.6 km TRCTL disabled and enabled periods. Resolvers measure the bogie and differential angles. These suspension angles are recorded in motion history telemetry. A differential angle increases when the corresponding front wheel is climbing and decreases when it is descending. A bogie angle increases when the corresponding middle wheel is climbing and/or the rear wheel is descending. There is no direct way to compare the height and frequency of rocks the wheels traverse during the TRCTL disabled and enabled periods. However, comparing the suspension angles is an indirect way to infer if one period had more difficult terrain with respect to climbing rocks. The average absolute value of the differential angle was 0.1° higher for the TRCTL disabled period, which corresponds to an average higher wheel height of approximately 2 mm. The average absolute value of the left and right bogie angles were 0.1° and 0.3° higher for the TRCTL disabled period, which corresponds to an average higher middle wheel height of 2 and 6 mm. Given these are small wheel heights, it implies that on average, the terrain difficulty with respect to climbing rocks was similar for the TRCTL disabled and enabled periods.

Curiosity executes drive steps primarily using two drive primitives: an arc and a turn-in-place maneuver. Arcs are specified with an arc length and a desired change in heading, or delta heading. Straight arcs can be specified with a delta heading of zero. Turn-in-place arcs can be specified with an arc length of zero. Arcs can be commanded as single drive-step commands, or as a high-level command to drive to a waypoint. The high-level command to drive to a waypoint treats each drive step as an arc, where the arc length and delta heading are generated by the navigation flight software. During the execution of an arc, the front and rear wheels are steered to appropriate steer angles, the steering actuator brakes are engaged, and the drive step is executed along a constant curvature arc.

Some heading error (i.e., unintended yaw) inevitably occurs during constant curvature arc driving, the extent of which depends on rover tilt and the wheel/terrain interaction. Most of Curiosity's arcs are commanded straight. Figure 18 contains a plot of the mean and standard deviation of heading error for each straight arc commanded during the 3.6 km TRCTL disabled and enabled periods. For those periods, the percentage of successful arcs that were commanded straight were 83.2% and 84.8%, respectively. For each straight arc, Curiosity calculates the mean heading error onboard and records it in motion summary telemetry. For the TRCTL enabled period, the reduction in both the mean of the absolute value of the mean and standard deviation of the heading error for straight arcs was 44%.

During both the 3.6 km TRCTL disabled and enabled periods, nonstraight arcs were commanded much less frequently than straight arcs. For those periods, the percentage of successful arcs that were

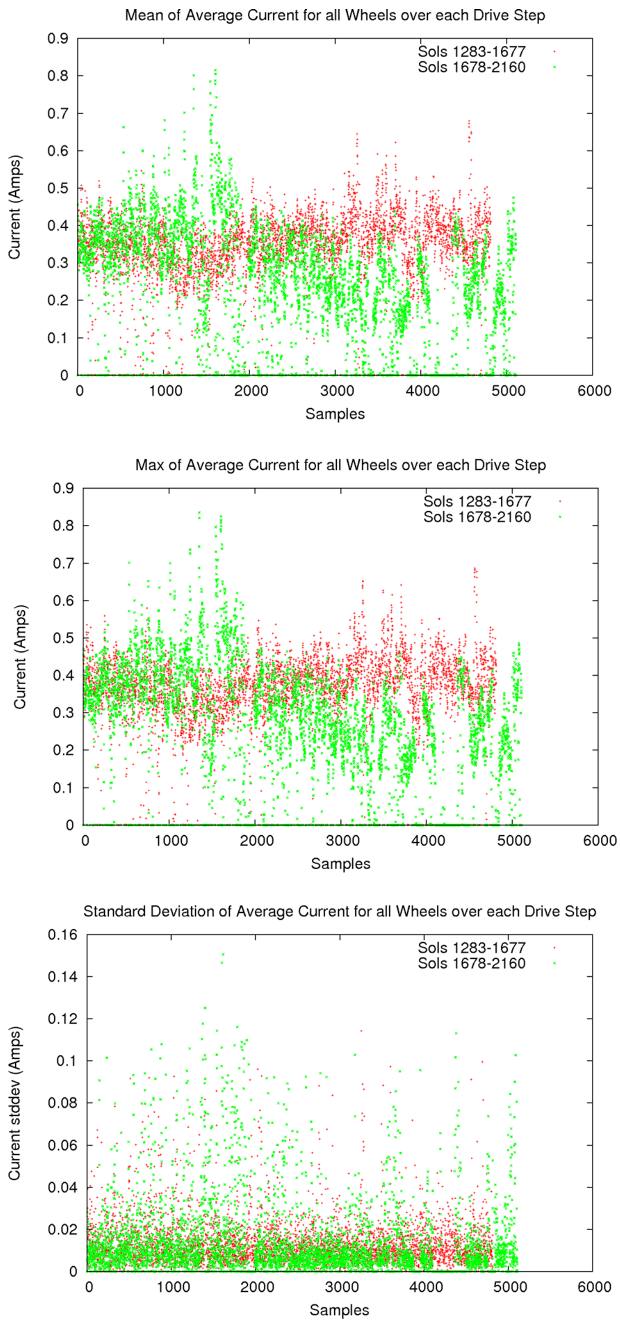


FIGURE 22 Mean, max, and standard deviation of average current for all six wheels over each drive step during the 3.6 km TRCTL disabled and enabled periods. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

commanded nonstraight were 16.8% and 15.2%, respectively. Arcs that failed to complete (due system fault protection) were excluded from heading error analysis since they result in heading error unrelated to TRCTL. An analysis was performed to determine if the magnitude of the heading error is a function of the magnitude of the commanded delta heading. Figure 19 contains a plot of heading error for all nonstraight drive arcs using VO, commanded during the 3.6 km TRCTL disabled and enabled periods. All of the command delta heading angles for these two periods fall within the range of −25° to 15°. It is clear from the plot there is no direct correlation between command delta heading and heading error, that is, small command delta heading does no always result in low heading error and large command delta heading does not always result in high heading error. Figure 19 also contains a plot of the number of nonstraight arcs using VO, commanded during the 3.6 km TRCTL disabled and enabled periods. By far, the nonstraight arc most commonly commanded had a command delta heading of +/−3.99°. During the 3.6 km TRCTL disabled and enabled periods, it was commanded 518 and 475 times, respectively. The average absolute value of the heading error for those arcs was 1.0° and 0.87°, respectively, a 13% reduction for the TRCTL enabled period.

For the 3.6 km TRCTL disabled and enabled periods, the absolute value of the heading error for all straight and nonstraight arcs (excluding those where the arc length was zero) were binned into average rover tilt bins having a 1° width. In this context, rover tilt is synonymous with terrain tilt. In each average rover tilt bin, the number of arcs, the sum of the absolute value of heading error, and the average heading error were calculated. As illustrated in Figure 20, more time was spent executing arcs with nonzero arc length on higher terrain tilts during the TRCTL enabled period than during the TRCTL disabled period. For all average rover tilt bins up to 23° (except for the 16° bin), the average unintended yaw was lower for the TRCTL enabled period. Beyond an average terrain tilt of 15°, average unintended yaw for the TRCTL disabled curve diverged from the TRCTL enabled curve. However, more data are needed to draw conclusions about unintended yaw differences between TRCTL disabled and enabled at average rover tilts of larger than 15°.

Wheel slip is dependent on terrain tilt and wheel/terrain interaction. Each time VO is performed, FSW generates a message that contains both the wheel slip fraction and average rover tilt over the drive step. Wheel slip fraction is calculated as the sum of the six wheel slips divided by the sum of the six wheel path lengths. FSW does not calculate a wheel slip fraction for arcs that are less than 0.35 m, since wheel slip fraction can be large for small wheel slip amounts during short arcs, which could result in an excessive slip drive fault. For the TRCTL disabled and enabled periods, wheel slip fraction data for non-turn-in-place arcs were binned into average rover tilt bins having a 1° width. In each average rover tilt bin, the number, sum, and average of wheel slip measurements were calculated. As illustrated in Figure 21, more time was spent executing non-turn-in-place arcs on higher terrain tilts during the TRCTL enabled period than during the TRCTL disabled period. Note that there is a period of high wheel slip between samples 1000 and 1500 for the TRCTL enabled period. During this period, Curiosity encountered sandy terrain. On Sol 1754, the max wheel slip reached 76.4% in sand. On Sol 1789, the max wheel slip reached 78.3% in sand. Curiosity terminated that drive early when the excessive wheel slip caused the yaw limit to be exceeded. For all average rover tilt bins up to 20°, the average percent wheel slip was lower for the TRCTL enabled period. More data is needed to draw conclusions about wheel slip differences between TRCTL disabled and enabled at average rover tilts of larger than 20°. Overall, the average wheel slip during the TRCTL enabled period was 0.8% lower than the TRCTL disabled period. The max wheel slip that occurred during the 3.6 km TRCTL disabled and enabled periods was 59.6% (on Sol 1648) and 98.7% (on Sol 2087). The

TABLE 2 Comparison of approximately 3.6 km of driving before and after TRCTL was approved for nominal use

	Sols 1283–1677 (before TRCTL)	Sols 1678–2160 (after TRCTL approved)
Odometry (m)	3,574.4	3,565.4 TRCTL (99.4%), 22.2 non-TRCTL (0.6%), 3587.7 total
Number of drives	131	146 TRCTL 3 non-TRCTL (FMWI)
Number of commanded drive steps (excluding turn in place)	4,690	5,015
Elevation change (m)	+145.0	+144.9
Percent of driving executed forward, backward, and turning in place	88.3% (forward), 3.8% (backward), 7.9% (turn in place)	79.1% (forward), 14.5% (backward), 6.4% (turn in place)
Average rover pitch (deg)	2.8 (forward), 1.9 (backward), 2.3 (turn in place), 2.7 (all)	4.6 (forward), 6.0 (backward), 5.2 (turn in place), 4.9 (all)
Average rover roll (deg)	1.0	0.5
Average absolute value of left bogie angle (deg)	2.6	2.5
Average absolute value of right bogie angle (deg)	2.9	2.6
Average absolute value of differential angle (deg)	1.2	1.1
Mean of |mean heading error| for straight arcs (deg)	0.45 (*N* = 3,439)	0.25 (*N* = 3505)
Mean of standard deviation of heading error for straight arcs (deg)	0.36 (*N* = 3,439)	0.20 (*N* = 3505)
Average |heading error| for +/−3.99° arcs (deg)	1.00 (*N* = 518)	0.87 (*N* = 475)
Max wheel slip	59.6% (Sol 1648)	98.7% (Sol 2087)
Average wheel slip	9.9%	9.1% 8.9% (when slip <60%)
Average traverse rate (m/hr)	35.0 (*N* = 113)	33.3 (*N* = 127)
Average compressed MobMotionHistory DP size (Mbit/m)	0.7	1.2 (1.8x)
Broken grousers	2	0
Average wheelie detection duration (s)	N/A	2.7
Average distance between wheelie detections (m)	N/A	4.6

Note: Data from Sol 1646 and 1662 (TRCTL checkouts) were excluded because motion history telemetry was recorded at 64 Hz, eight times the rate for other sols. TRCTL is not used for full MAHLI wheel imaging (FWMI) drive steps. The data in the Sols 1678–2160 column were generated for the combined TRCTL and non-TRCTL driving during that period; the exceptions are that FMWI data were excluded from the average traverse rate calculation, and non-TRCTL driving was excluded from the average telemetry size calculation.
Abbreviation: TRCTL, our algorithm inspired by traction control.

98.7% wheel slip is a mission record that was caused by a combination of the right rear wheel attempting to traverse a steep rock while the rover was driving up terrain with a relatively moderate tilt (18.6°).

Figure 22 contains plots of the mean, max, and standard deviation of average current for all six wheels over each drive step during the 3.6 km TRCTL disabled and enabled periods. The average of data in the mean of the average current plot for all wheels over each drive step for the two periods was 0.353 and 0.287 A, respectively. The average of the data in the max of the average current plot over all wheels for each drive step for the two periods was 0.375 and 0.312 A, respectively. These values correspond to reductions of 18.7% and 16.8%.

The mean of mean current for each drive step was calculated for each wheel. As seen in the Earth-based testing, the reduction in mean current for TRCTL enabled was the highest for the leading (front) wheels. The reduction in the mean of the mean current for the left and right front wheels for TRCTL enabled was 9.7% and 12.4%. The mean of the mean current and mean of the current standard deviation was lower for all wheels with TRCTL enabled. The TRCTL Sol 1646 and 1662 checkout data were excluded when generating these results because motion history telemetry was recorded at 64 Hz, eight times the rate for other sols. The lower TRCTL enabled peak and average drive currents observed in these results are consistent with the development and V&V test results. A summary of the 3.6 km comparison is provided in Tables 2 and 3. The per-wheel current reduction is statistically significant ($\alpha = .05$) for all wheels except the left middle one.

Since there were differences in terrain type between the 3.6 km TRCTL disabled and enabled periods, a comparison was also performed of approximately 1 km of driving before and after TRCTL nominal use started. The terrain unit map indicated the odometry Curiosity traversed during both 1 km periods was solely on fractured terrain (Condus & Arvidson, 2018). Figure 23 illustrates this and several other terrain types encountered during these drives. As shown in Figure 24, the difference in terrain steepness between the TRCTL enabled and disabled periods is larger than in the previous comparison. During the TRCTL enabled period, the average rover pitch was 3.6° (2.1x) higher and the change in elevation was 26.88 m larger than the

TABLE 3 Comparison of wheel actuator current data of approximately 3.6 km of driving before and after TRCTL was approved for nominal use.

	Sols 1283–1677 (before TRCTL)	Sols 1678–2160 (after TRCTL approved)	Reduction (%)
LF mean of mean current for each drive step (A)	0.354	0.320	9.7
LM mean of mean current for each drive step (A)	0.316	0.312	1.4
LR mean of mean current for each drive step (A)	0.412	0.385	6.5
RF mean of mean current for each drive step (A)	0.370	0.324	12.4
RM mean of mean current for each drive step (A)	0.300	0.289	3.6
RR mean of mean current for each drive step (A)	0.433	0.399	9.7
LF mean of current *SD* for each drive step (A)	0.215	0.166	23.0
LM mean of current *SD* for each drive step (A)	0.184	0.155	15.5
LR mean of current *SD* for each drive step (A)	0.238	0.177	25.7
RF mean of current *SD* for each drive step (A)	0.224	0.164	26.8
RM mean of current *SD* for each drive step (A)	0.181	0.149	18.0
RR mean of current *SD* for each drive step (A)	0.245	0.177	27.7
Average the data for the mean of average current for all wheels over each drive step (A)	0.353	0.287	18.7
Average the data for the max of average current for all wheels over each drive step (A)	0.375	0.312	16.8

Note: Data from Sol 1646 and 1662 (TRCTL checkouts) were excluded because motion history telemetry was recorded at 64 Hz, eight times the rate for other sols. The data in the Sol 1678–2160 column were generated for the combined TRCTL and non-TRCTL driving during that period.
Abbreviation: TRCTL, our algorithm inspired by traction control.

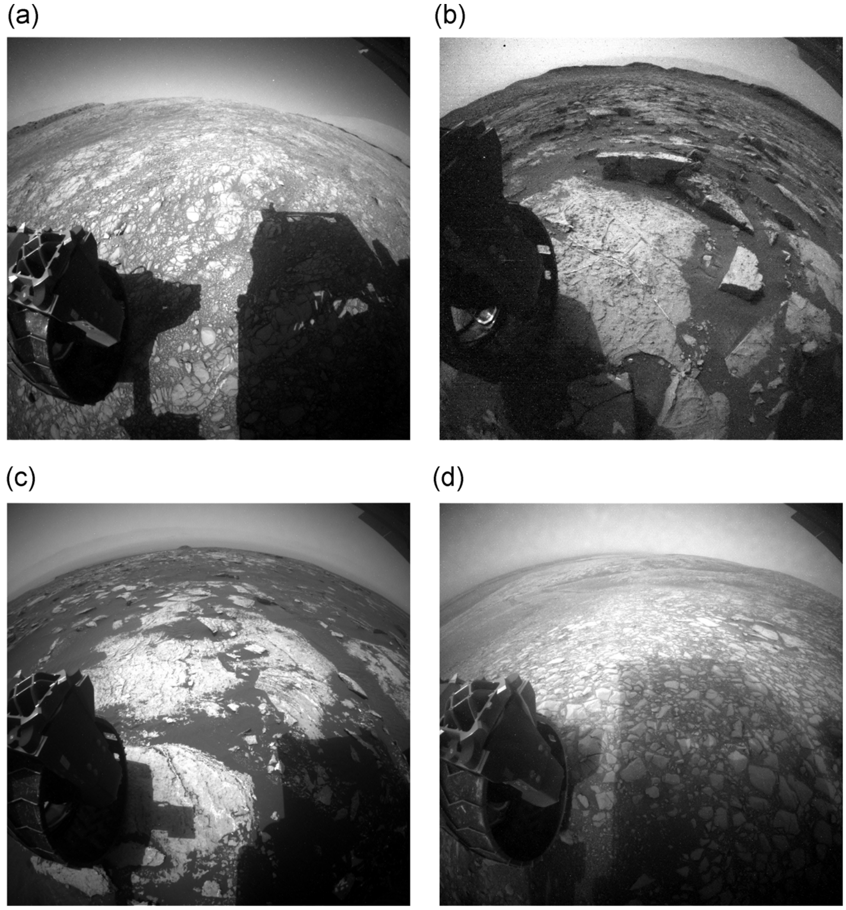


FIGURE 23 Illustration of terrain types encountered during TRCTL disabled and enabled periods. Images taken by Curiosity's Rear Hazcams, available online at https://mars.nasa.gov/msl/multimedia/raw/. (a) Sol 1399 Ridged terrain with fractured terrain between ridges, sand in lows. (b) Sol 1468 ridged terrain, sand in lows. (c) Sol 1676 fractured terrain, sand in lows. (d) Sol 2157 pitted terrain, sand in lows, densely fractured rocks without much relief as compared to fractured terrain (Condus & Arvidson, 2018). TRCTL, our algorithm inspired by traction control

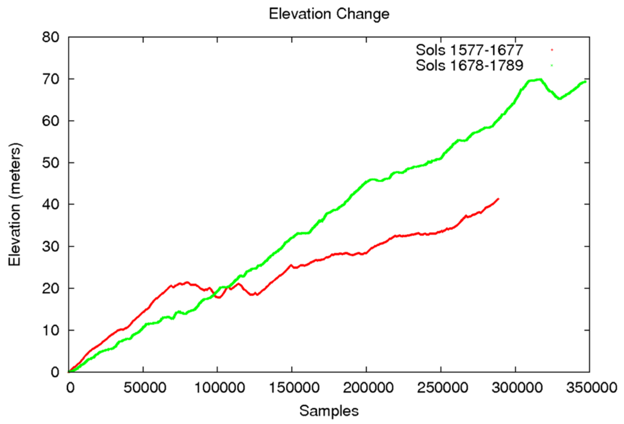


FIGURE 24 The change in rover elevation for the 1 km TRCTL disabled and enabled periods, relative to the initial rover position for both periods. TRCTL, traction control [Color figure can be viewed at wileyonlinelibrary.com]

TRCTL disabled period. A summary of the 1 km comparison is provided in Table 4.

The average of the data for the mean of the average current over all wheels for each drive step for the 1 km TRCTL disabled and enabled periods was 0.350 and 0.321 A. The average of the data for the max of the average current over all wheels for each drive step for the two periods was 0.368 and 0.345 A. These values correspond to reductions of 8.3% and 6.6%, occurring despite the higher average uphill slope with TRCTL enabled.

For the 1 km TRCTL disabled and enabled periods, the percentage of successful arcs that were commanded straight were 84.0% and 80.5%, respectively. During the TRCTL enabled period, the reduction on average heading error for successful straight arcs was 30%, in spite of the higher average uphill slope and excessive wheel slip (78.3% peak) on Sol 1789 in cohesionless sand. The average wheel slip during the TRCTL enabled period was 0.7% higher than during the TRCTL disabled period.

When TRCTL is enabled, a timeout is calculated for each drive step. Exceeding this timeout is the only new fault type introduced. Of the 146 nominal-use drives thus far, only one has ended early due to such a timeout fault; the Sol 1786 drive faulted after 15.86 m of the planned 27.9 m when a 32.77 s timeout was exceeded while the right rear wheel was driving over a large rock. Incidentally, the right bogie suspension angle was 0.3° away from exceeding its 18° limit; the drive was seconds away from being stopped with a suspension fault.

During development testing, a middle wheel wheelie behavior was observed on high-friction terrain. To prevent propagation of a middle or rear wheelie with TRCTL enabled, a wheelie suppression behavior, which adjusts the speed of the other bogie wheel to lower the elevated wheel, was added to the TRCTL software. The detection of a wheelie event occurs when the suspension rate and bogie angle exceed a threshold, and the motor current magnitude is below a threshold. The amount a bogie wheel is adjusted is proportional to the bogie angle and bogie angle rate. These values are parameterized and set conservatively to err on the side of wheelie suppression. When driving over complex terrain with TRCTL enabled, it is not unusual for the wheelie detector to be triggered for short durations. The average distance between wheelie detections in the 146 TRCTL drives was 4.6 m (with

TABLE 4 Comparison of approximately 1 km of driving before and after TRCTL was approved for nominal use

	Sols 1577–1677 (before TRCTL)	Sols 1678–1789 (after TRCTL approved)
Odometry (m)	1,004	1,005 TRCTL (99.5%), 5 non-TRCTL (0.5%), 1,010 total
Elevation change (m)	+42.6	+69.5
Percent of driving executed forward, backward, and turning in place	87.7% (forward), 3.1% (backward), 9.2% (turn in place)	92.1% (forward), 2.6% (backward), 5.3% (turn in place)
Average rover pitch (deg)	3.5 (forward), 2.8 (backward), 3.4 (all)	7.1 (forward), 6.4 (backward), 7.0 (all)
Average traverse rate (m/hr)	35.2 (N = 37)	31.5 (N = 39)
Average compressed MobMotionHistory DP size (Mbit/m)	0.7	1.3 (1.9x)
Mean of |mean heading error| for straight arcs (deg)	0.5 (N = 964)	0.35 (N = 1,052)
Max wheel slip	59.6% (Sol 1648)	78.3% (Sol 1789)
Average wheel slip	11.3%	12.0% 11.5% (when slip <60%)
Average the data for the mean of average current for all wheels over each drive step (A)	0.350	0.321
Average the data for the max of average current for all wheels over each drive step (A)	0.368	0.345

Note: Data from Sol 1646 and 1662 (TRCTL checkouts) were excluded because motion history telemetry was recorded at 64 Hz, eight times the rate for other sols. The data in the Sol 1678–1789 column were generated for the combined TRCTL and non-TRCTL driving during that period.
Abbreviation: TRCTL, our algorithm inspired by traction control.

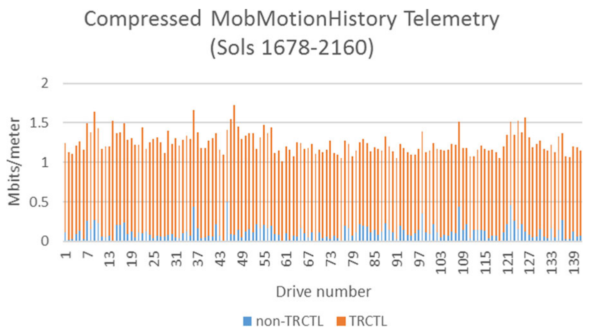


FIGURE 25 The size of compressed mobility motion history data products during the 3.6 km TRCTL enabled period. Note that a small amount of non-TRCTL driving is typical on TRCTL enabled sols, since TRCTL is not used for turn-in-place motions and commanded arcs of less than 10 cm. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

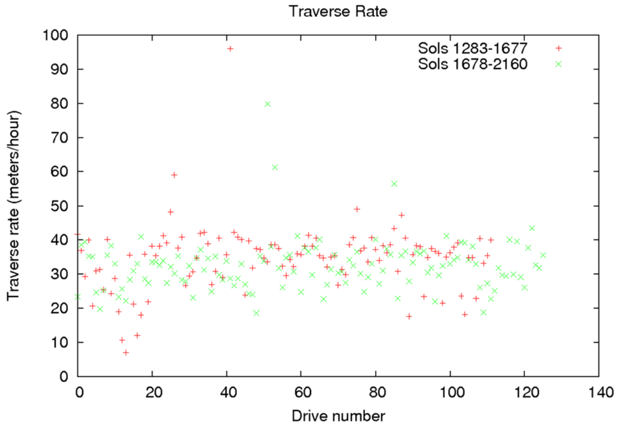


FIGURE 27 Rover traverse rate for the 3.6 km TRCTL disabled and enabled periods. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

average duration of 2.7 s), 2.7 times more than the 1.7 m average distance between wheelie detections observed during the Earth-based testing on the complex terrain in Figure 9.

Using TRCTL requires larger mobility motion history data product size and longer traverse times. One motion history data product is generated for every drive step. As shown in Figure 25, there is usually at least a small portion of non-TRCTL motion history data products, even on drives where TRCTL was enabled. This is because turn-in-place motions and commanded arcs of less than 10 cm are not performed using the TRCTL algorithm. Figure 26 contains a plot of average compressed motion history data product size for the 3.6 km TRCTL disabled and enabled periods. The compressed motion history data products were on average 1.81× larger for the TRCTL enabled drive steps. This is lower than the worst-case predict from VSTB measurements on complex terrain (2.29×) and close to the best-case predict from the VSTB measurements on flat terrain (1.8×).

The time between the dispatch of first motion command and completion of last motion command for each drive can be obtained from the drive telemetry. For each drive, the total VO corrected odometry divided by this time was used to generate the drive traverse rates. For the 3.6 km period before and after TRCTL nominal use, the TRCTL enabled drive steps took on average 5.0% longer. Figure 27 contains a plot of the traverse rate for the 3.6 km TRCTL disabled and enabled periods.

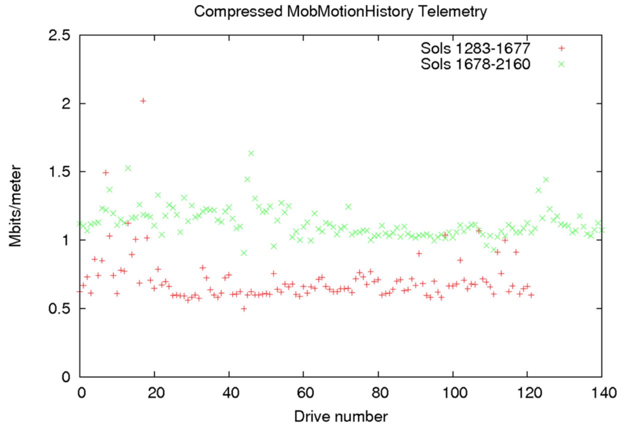


FIGURE 26 The size of compressed MobMotionHistory data products for the 3.6 km TRCTL disabled and enabled periods. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

There are 19 grousers, or cleats, on each wheel. Each wheel has an interior stiffener ring that helps provide support for the grousers. The distance from the inner edge of a wheel to the stiffener ring is two-thirds the width of the wheel. Because grouser breaks on the outer one-third of a wheel are rare, the outer one-third of a wheel is considered a rim. On the inner two-thirds of a wheel, however, grousers become at risk of breaking after the average of the total crack length of the skin sections on both side of a grouser exceeds 70% of the inner two-thirds wheel width. Grouser breaks on the inner two-thirds of a wheel nearly always occur at the stiffener ring. If all the grousers and odometry features on the inner two-thirds of a wheel break and the inner two-thirds of the wheel is shed, Curiosity could continue to drive indefinitely on the rimmed wheel.

After each FMWI, the MSL Wheel Wear team inspects the grousers on each wheel in the downlinked wheel images. In addition, the length of each crack is measured in pixels and converted to millimeters, using known wheel dimensions. The current damage on each wheel is assessed after approximately 500 m of driving. The most heavily damaged wheels on Curiosity are the LF, LM, and RM. As of Sol 2115, the number of cracks on the LF, LM, LR, RF, RM, and RR (Left/Right, Front/Middle/Rear) wheel is 65, 41, 2, 30, 61, and 3, respectively, and the total crack length on the LF, LM, RF, and RM wheel is 285.1, 275.6, 58.2, and 284.3 cm, respectively. (Because the rear wheels have so few cracks, the cracks on those wheels are not currently measured.)

The start and end sols of our 3.6 km TRCTL disabled and enabled periods do not coincide with FMWI sols. To compare the wheel damage rates, we shift the start and end sols of the two periods. Figure 28 contains a plot of the change in total crack length over the inner two-thirds of the front and middle wheels for a TRCTL disabled period of Sols 1260–1641, and a TRCTL

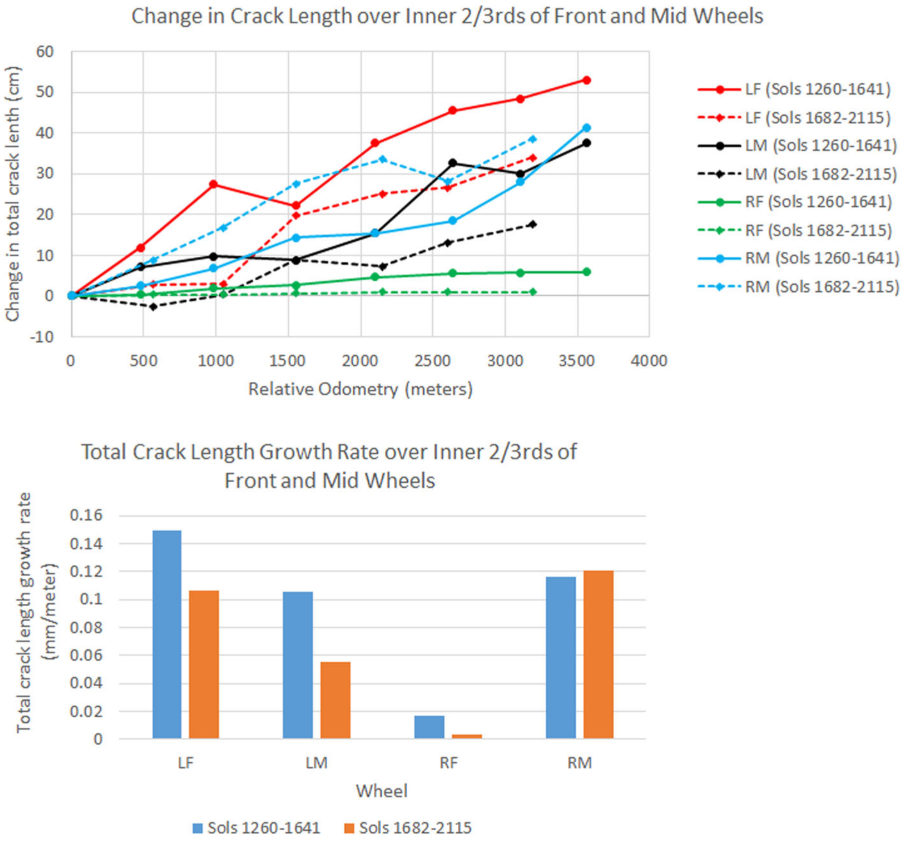


FIGURE 28 Change in total crack length (top) and total crack length growth rate (bottom) over the inner two-thirds of each front and middle wheel for the 3.6 km TRCTL disabled and enabled periods. During the TRCTL enabled period, the total crack growth rate was smaller for all front and middle wheels except for the RM. TRCTL, our algorithm inspired by traction control [Color figure can be viewed at wileyonlinelibrary.com]

enabled period of Sols 1682–2115. The change in total crack length for the LF, LM, and RF wheels are higher for the TRCTL disabled period, but the change in total crack length for the RM wheel was slightly higher for the TRCTL enabled period. Figure 28 also contains a plot of total crack length growth over the inner two-thirds of the front and middle wheels. For the TRCTL enabled period, the reduction in the total crack length growth rate for the LF, LM, and RF wheels was 28.6%, 47.6%, and 81.6%, respectively. There was a slight increase (3.8%) in the total crack length growth rate for the RM wheel. It is unclear why the RM total crack length growth rate was slightly higher for the TRCTL enabled period. Since FMWI is performed every 500 m, there is no way to pinpoint the precise locations (i.e., terrain) where the majority of RM damage occurred. During each 500 m interval between FMWI, the average difference in RM total crack length growth between the TRCTL enabled and disabled period translates to only 2.2 mm, an amount that is likely less than the measurement error.

There is anecdotal evidence that TRCTL is reducing the grouser break rate on Curiosity. The MSL Wheel Wear Tiger team had predicted, based on Earth-based experiments, three broken grousers would occur on a Curiosity wheel by the 16 km mark. The Sol 1641 FMWI was performed when the total odometry was 16.0 km. Coincidentally, two broken grousers were identified on the left middle wheel in the Sol 1641 wheel images. (This was the first time a grouser break was identified on Curiosity.) With some urgency, nominal use of TRCTL began 37 Sols later. Since Sol 1641, no additional grouser breaks have been identified in the seven subsequent FMWI, the latest performed on Sol 2115 at the 19.602 km odometry mark. The Tiger Team had predicted there would be ten grouser breaks on a wheel by this odometry mark, but the grouser break count remains two.

7 | SUMMARY

The TRCTL algorithm presented in this paper was developed in response to the increased Curiosity rover wheel damage rate observed in October 2013. Based upon the MSL project investigation into the causes of wheel damage, reducing the forces imparted on individual wheels while climbing sharp, embedded rocks may result in extending wheel life. This TRCTL algorithm departs from the previously used Ackermann steering model, which assumes level terrain. The algorithm merges real-time data from the rocker-bogie suspension system and IMU data to estimate wheel contact points with the terrain, and commands speeds based on the climbing behavior of each individual wheel.

The algorithm was implemented as a hot patch to the current testbed version of flight software, rather than being released as a new version of flight software. This decision allowed for a faster implementation, because the patch size was less than 0.5% the size of a new version of flight software, and required only regression testing of mobility and fault protection capabilities. During V&V testing of the TRCTL algorithm, all mobility commands executed nominally while running the software patch. These comprehensive ground tests on the flight-like rover testbed included testing on mixed terrain types and various rock heights; the results presented here demonstrate modest reductions in the wheel loading. The average reduction in resultant loads on relatively benign terrain was 19% for leading front wheels, while leading middle wheels experienced an 11% reduction. Smaller, but consistent, reductions in wheel loads and drive torques were seen on more complex terrain. Development and V&V testing also demonstrated the ancillary benefits of reduced unintended yaw and rover slip.

The first drive using TRCTL successfully occurred on Sol 1646 during a 5 m checkout test. A later 25 m checkout test was successfully performed on Sol 1662. The MSL project approved the TRCTL flight software patch for nominal use in flight in April 2017, and nominal use of TRCTL commenced on Sol 1678. Between Sols 1678 and 2160, TRCTL has been used in 146 of the 149 Curiosity drives. Over this period, 99.38% of Curiosity's 3587.672 m odometry has been performed with TRCTL enabled. Performance results from flight data indicates that drives using the TRCTL software result in lower peak and average drive actuator current.

The 3.6 km period before and after TRCTL nominal use started was evaluated. The increase in rover elevation was the same for both periods, however the average rover pitch was 2.1° higher for the TRCTL enabled period. The average reduction in wheel currents for the front, middle, and rear wheels was 11.1%, 2.5%, and 7.2%, respectively. Over all wheels, the reduction in the mean of mean current and mean of the max current for each drive step was 18.7% and 16.8%, respectively. While we cannot directly measure wheel loads or torques in flight, these findings suggest that the wheels are experiencing lower forces. The ancillary benefit of reduction on average unintended yaw was 52.3% for all successful drive steps. For each 1° interval of average rover tilt from 0° to 20°, average percent wheel slip was lower during the TRCTL enabled period. During the 1.0 km period before and after TRCTL nominal use started, the TRCTL enabled period had a 27 m higher increase in elevation and a 3.6° larger average rover pitch. Over all wheels, the reduction in the mean of mean current and mean of the max current for each drive step was 8.3% and 6.3%, respectively.

At Curiosity's current odometry (19,947.2 m), based on wheel longevity experiments in the JPL Mars Yard, the MSL Wheel Wear Tiger Team had predicted there would be 10 grouser breaks on a Curiosity wheel. But during the TRCTL enabled period, the wheel grouser break rate has been significantly slower than predicted by the Tiger Team; only two grousers have broken thus far. Based on these results, the TRCTL algorithm is performing as expected. Overall, the primary costs of the TRCTL software (15.4% longer traverse time and 1.81× larger mobility data products) are outweighed by the benefit to MSL of reduced forces on the wheels.

7.1 | Future work

The performance of the TRCTL software will continue to be assessed on Mars. Additional tools to aid in the trending of flight data and to determine the efficacy of TRCTL are currently in development. In addition, improvements to the TRCTL software are currently being discussed. A version which incorporates torque feedback into wheel speed commanding to better achieve desired torque at each wheel as a function of contact angle has been proposed. This modification is hypothesized to further reduce the resultant contact force on each wheel, although this potential improvement in performance has not yet been quantified. This algorithm is currently in development, and the discussion of its implementation is ongoing.

Future rover missions are also evaluating TRCTL for possible use, largely in part for the potential reduced yaw error and reduced slip benefits demonstrated here.

ACKNOWLEDGMENTS

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors would like to thank the Mars Science Laboratory Program for supporting this study.

ORCID

Olivier Toupet http://orcid.org/0000-0003-4993-7390
Arturo Rankin http://orcid.org/0000-0001-6592-6566

REFERENCES

Arvidson, R., DeGrosse, P., Grotzinger, J., Heverly, M., Shechet, J., Moreland, S., & Stilly, E. (2017). Relating geologic units and mobility system kinematics contributing to curiosity wheel damage at gale crater, mars. *Journal of Terramechanics*, *73*, 73–93.

Benowitz, E., & Maimone, M. (2015). Patching flight software on Mars. *Workshop on Spacecraft Flight Software*, Laurel, Maryland.

Biesiadecki, J. J., Liebersbach, R., & Maimone, M. W. (2008). Mars exploration rover mobility and IDD downlink analysis tools. *International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS)*, Los Angeles, CA.

Condus, T. C., & Arvidson, R. E. (2018). Personal communication.

Goldberg, S. B., Maimone, M. W., & Matthies, L. (2002). Stereo vision and rover navigation software for planetary exploration. *IEEE Aerospace Conference*, Big Sky, MT, vol. 5, 2025-2036.

Iagnemma, K., & Dubowsky, S. (2000a). Vehicle wheel-ground contact angle estimation: With application to mobile robot traction control, *Advances in robot kinematics* (pp. 137–146). Dordrecht: Springer.

Iagnemma, K., & Dubowsky, S. (2000b). Mobile robot rough-terrain control (rtc) for planetary exploration. *ASME Design Engineering Technical Conferences*, Baltimore, MD.

Iagnemma, K., & Dubowsky, S. (2004). Traction control of wheeled robotic vehicles in rough terrain with application to planetary rovers. *The International Journal of Robotics Research*, *23*(10-11), 1029–1040.

Ishigami, G., Nagatani, K., & Yoshida, K. (2009). Slope traversal controls for planetary exploration rover on sandy terrain. *Journal of Field Robotics*, *26*, 264–286.

Johnson, A., Goldberg, S., Cheng, Y., & Matthies, L. (2008). Robust and efficient stereo feature tracking for visual odometry. *International Conference on Robotics and Automation*, Pasadena, CA.

Maimone, M., Cheng, Y., & Matthies, L. (2007). Two years of visual odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, *24*, 169–186.

Maimone, M., Johnson, A., Cheng, Y., Willson, R., & Matthies, L. (2006). Autonomous navigation results from the Mars Exploration Rover (MER) mission, *Springer tracts in advanced robotics* (21, pp. 3–13). Berlin, Heidelberg: Springer.

Maimone, M., Maxwell, S., Biesiadecki, J., & Algermissen, S. (2018). RP-check: An architecture for spaceflight command sequence validation. *IEEE Aerospace Conference*, Big Sky, MT.

Sreenivasan, S., & Wilcox, B. (1994). Stability and traction control of an actively actuated micro-rover. *Journal of Robotic Systems*, *11*, 487–502.

Tarokh, M., McDermott, G., Hayati, S., & Hung, J. (1999). Kinematic modeling of a high mobility mars rover. *International Conference on Robotics and Automation*, Detroit, MI, 992–998.

Toupet, O., Biesiadecki, J., Rankin, A., Steffy, A., Meirion-Griffith, G., Levine, D., Schadegg, M., & Maimone, M. (2018). Traction control design and integration onboard the mars science laboratory curiosity rover. IEEE Aerospace Conference, Big Sky, MT.

Wright, J., Hartman, F., Cooper, B., Maxwell, S., Yen, J., & Morrison, J. (2006). Driving on mars with RSVP. *IEEE Robotics and Automation*, *13*(2), 37–45.

Yoshida, K., & Hamano, H. (2002). Motion dynamics of a rover with slip-based traction model. *IEEE International Conference on Robotics and Automation*, Washington, DC, 3155–3160.

Yoshida, K., Hamano, H., & Watanabe, T. (2003). Slip-based traction control of a planetary rover, *Experimental robotics VIII*, *Springer tracts in advanced robotics* (5, pp. 644–653). Berlin, Heidelberg: Springer.

How to cite this article: Toupet O, Biesiadecki J, Rankin A, et al. Terrain-adaptive wheel speed control on the curiosity mars rover: Algorithm and flight results. *J Field Robotics*. 2020;37:699–728. https://doi.org/10.1002/rob.21903

APPENDIX: NOMENCLATURE

Table A1 below defines the key variables and parameters referenced in this paper.

APPENDIX: WHEEL ANGULAR RATES

The equations to compute the idealized (no-slip) wheel angular rates as a function of the rover's origin linear velocity, suspension angles and angular rates, attitude rates, steering angles, wheel contact angles, and rover geometry are provided as follows:

TABLE A1 Description of key symbols

Symbol	Description
${}^bR_a$	Rotation from frame *a* to frame *b*
${}^a\overrightarrow{AB}$	Vector $\overrightarrow{AB}$ expressed in frame *a*
${}^av_A$	Velocity of point *A* expressed in frame *a* (relative to inertial frame)
O	Rover origin point
D	Rocker pivot point
B_1	Left bogie pivot point
B_2	Right bogie pivot point
A_i	Center of wheel *i*
l_{fd}	Length between front wheel and rocker in body *x-z* plane
l_{db}	Length between rocker and bogie in body *x-z* plane
l_{bm}	Length between bogie and middle wheel in body *x-z* plane
l_{br}	Length between bogie and rear wheel in body *x-z* plane
$\varkappa_1$	Angle between $\overrightarrow{DA_1}$ and body *x* axis on flat ground
$\varkappa_2$	Angle between body *x* axis and $\overrightarrow{B_1D}$ on flat ground
$\varkappa_3$	Angle between $\overrightarrow{B_1A_3}$ and body *x* axis on flat ground
$\varkappa_4$	Angle between body *x* axis and $\overrightarrow{A_5B_1}$ on flat ground
R_w	Wheel radius
x_{fm}	Longitudinal distance between front and middle wheels on flat ground
x_{mr}	Longitudinal distance between middle and rear wheels on flat ground
y_{of}	Lateral distance between origin and front wheels on flat ground

(Continues)

TABLE A1 (Continued)

Symbol	Description
y_{om}	Lateral distance between origin and middle wheels on flat ground
y_{or}	Lateral distance between origin and rear wheels on flat ground
x_{od}	Longitudinal distance between rover origin and rocker on flat ground
z_{od}	Signed vertical distance between rover origin and rocker on flat ground
r	Turn radius (distance between rover origin and center of rotation)
$\dot{x}$, $\dot{y}$, $\dot{z}$	Linear velocity of rover origin along x, y, z body axes
ϕ	Rover roll angle
θ	Rover pitch angle
ψ	Rover yaw angle
$\omega_{x,y,z}$	Rover angular rates along body x, y, z axes (relative to inertial frame)
β	Left rocker angle
ρ_1	Left bogie angle
ρ_2	Right bogie angle
η_i	Contact angle of wheel i
ψ_i	Steering angle of wheel i
$\dot{\theta}_i$	Angular rate of wheel i
${}^{y}\zeta_i$	Angular rate of drive actuator for wheel i along its y axis

$$\begin{aligned}\dot{\theta}_1 = {} & ((\cos(\beta)\sin(\eta_1) + \cos(\eta_1)\cos(\psi_1)\sin(\beta))(\omega_y x_{od} - \dot{z} + \omega_x y_{of} \\ & + l_{fd}\cos(\beta - \kappa_1)(\dot{\beta} + \omega_y)) \\ & - (\sin(\beta)\sin(\eta_1) - \cos(\beta)\cos(\eta_1)\cos(\psi_1))(\dot{x} + \omega_y z_{od} + y_{of}\omega_z \\ & - l_{fd}\sin(\beta - \kappa_1)(\dot{\beta} + \omega_y)) \\ & + R_w(\cos(\psi_1)(\dot{\beta} + \omega_y) - \omega_x\cos(\beta)\sin(\psi_1) + \omega_z\sin(\beta)\sin(\psi_1)) \\ & + \cos(\eta_1)\sin(\psi_1)(-\omega_x z_{od} + x_{od}\omega_z \\ & + l_{fd}\omega_z\cos(\beta - \kappa_1) + l_{fd}\omega_x\sin(\beta - \kappa_1)))/R_w.\end{aligned} \tag{B1}$$

$$\begin{aligned}\dot{\theta}_2 = {} & ((\sin(\beta)\sin(\eta_2) + \cos(\beta)\cos(\eta_2)\cos(\psi_2))(\dot{x} + \omega_y z_{od} - y_{of}\omega_z \\ & - l_{fd}\sin(\beta + \kappa_1)(\dot{\beta} - \omega_y)) \\ & - R_w(\cos(\psi_2)(\dot{\beta} - \omega_y) + \omega_x\cos(\beta)\sin(\psi_2) \\ & + \omega_z\sin(\beta)\sin(\psi_2)) - (\cos(\beta)\sin(\eta_2) - \cos(\eta_2) \\ & \cos(\psi_2)\sin(\beta))(\dot{z} - \omega_y x_{od} + \omega_x y_{of} \\ & + l_{fd}\cos(\beta + \kappa_1)(\dot{\beta} - \omega_y)) + \cos(\eta_2)\sin(\psi_2)(x_{od}\omega_z - \omega_x z_{od} \\ & + l_{fd}\omega_z\cos(\beta + \kappa_1) - l_{fd}\omega_x\sin(\beta + \kappa_1)))/R_w. \\ \dot{\theta}_3 = {} & (R_w(\dot{\beta} + \omega_y + \dot{\rho}_1) + \dot{x}\cos(\beta + \eta_3 + \rho_1) - \dot{z}\sin(\beta + \eta_3 + \rho_1) \\ & + \dot{\beta}l_{db}\sin(\kappa_2 - \eta_3 - \rho_1) \\ & + l_{db}\omega_y\sin(\kappa_2 - \eta_3 - \rho_1) + \omega_y z_{od}\cos(\beta + \eta_3 + \rho_1) \\ & + y_{om}\omega_z\cos(\beta + \eta_3 + \rho_1) + \omega_y x_{od}\sin(\beta + \eta_3 + \rho_1) \\ & + \omega_x y_{om}\sin(\beta + \eta_3 + \rho_1) + \dot{\beta}l_{bm}\sin(\kappa_3 + \eta_3) \\ & + l_{bm}\omega_y\sin(\kappa_3 + \eta_3) + l_{bm}\dot{\rho}_1\sin(\kappa_3 + \eta_3))/R_w.\end{aligned} \tag{B2}$$

$$\begin{aligned}\dot{\theta}_4 = {} & (\dot{x}\cos(\beta - \eta_4 - \rho_2) + \dot{z}\sin(\beta - \eta_4 - \rho_2) + R_w(\omega_y - \dot{\beta} + \dot{\rho}_2) \\ & - \dot{\beta}l_{db}\sin(\kappa_2 - \eta_4 - \rho_2) \\ & + \omega_y z_{od}\cos(\beta - \eta_4 - \rho_2) - y_{om}\omega_z\cos(\beta - \eta_4 - \rho_2) \\ & + l_{db}\omega_y\sin(\kappa_2 - \eta_4 - \rho_2) - \omega_y x_{od}\sin(\beta - \eta_4 - \rho_2) \\ & + \omega_x y_{om}\sin(\beta - \eta_4 - \rho_2) - \dot{\beta}l_{bm}\sin(\kappa_3 + \eta_4) \\ & + l_{bm}\omega_y\sin(\kappa_3 + \eta_4) + l_{bm}\dot{\rho}_2\sin(\kappa_3 + \eta_4))/R_w.\end{aligned} \tag{B3}$$

$$\begin{aligned}\dot{\theta}_5 = {} & -((\cos(\beta)(\cos(\rho_1)\sin(\eta_5) + \cos(\eta_5)\cos(\psi_5)\sin(\rho_1)) \\ & - \sin(\beta)(\sin(\eta_5)\sin(\rho_1) - \cos(\eta_5)\cos(\psi_5)\cos(\rho_1))) \\ & \quad (\dot{z} - \omega_y x_{od} - \omega_x y_o r + \dot{\beta}l_{br}\cos(\beta + \kappa_4 + \rho_1) \\ & + l_{br}\omega_y\cos(\beta + \kappa_4 + \rho_1) + l_{br}\dot{\rho}_1\cos(\beta + \kappa_4 + \rho_1) \\ & + \dot{\beta}l_{db}\cos(\beta + \kappa_2) + l_{db}\omega_y\cos(\beta + \kappa_2)) \\ & - R_w(\cos(\psi_5)(\dot{\beta} + \omega_y + \dot{\rho}_1) - \omega_x\cos(\beta + \rho_1)\sin(\psi_5) \\ & + \omega_z\sin(\beta + \rho_1)\sin(\psi_5)) \\ & + (\cos(\beta)(\sin(\eta_5)\sin(\rho_1) - \cos(\eta_5)\cos(\psi_5)\cos(\rho_1)) \\ & + \sin(\beta)(\cos(\rho_1)\sin(\eta_5) \\ & + \cos(\eta_5)\cos(\psi_5)\sin(\rho_1)))(\dot{x} + \omega_y z_{od} + y_{or}\omega_z \\ & + \dot{\beta}l_{br}\sin(\beta + \kappa_4 + \rho_1) + l_{br}\omega_y\sin(\beta + \kappa_4 + \rho_1) \\ & + l_{br}\dot{\rho}_1\sin(\beta + \kappa_4 + \rho_1) + \dot{\beta}l_{db}\sin(\beta + \kappa_2) \\ & + l_{db}\omega_y\sin(\beta + \kappa_2)) + \cos(\eta_5)\sin(\psi_5)(\omega_x z_{od} - x_{od}\omega_z \\ & + l_{br}\omega_z\cos(\beta + \kappa_4 + \rho_1) + l_{br}\omega_x\sin(\beta + \kappa_4 + \rho_1) \\ & + l_{db}\omega_z\cos(\beta + \kappa_2) + l_{db}\omega_x\sin(\beta + \kappa_2)))/R_w.\end{aligned} \tag{B4}$$

$$\begin{aligned}\dot{\theta}_6 = {} & -((\cos(\beta)(\cos(\rho_2)\sin(\eta_6) + \cos(\eta_6)\cos(\psi_6)\sin(\rho_2)) \\ & + \sin(\beta)(\sin(\eta_6)\sin(\rho_2) - \cos(\eta_6)\cos(\psi_6)\cos(\rho_2))) \\ & (\dot{z} - \omega_y x_{od} + \omega_x y_o r + l_{br}\dot{\rho}_2\cos(\beta - \kappa_4 - \rho_2) \\ & - \dot{\beta}l_{db}\cos(\beta - \kappa_2) + l_{db}\omega_y\cos(\beta - \kappa_2) \\ & - \dot{\beta}l_{br}\cos(\beta - \kappa_4 - \rho_2) + l_{br}\omega_y\cos(\beta - \kappa_4 - \rho_2)) \\ & + R_w(\omega_x\sin(\psi_6)\cos(\beta - \rho_2) - \cos(\psi_6)(\omega_y - \dot{\beta} + \dot{\rho}_2) \\ & + \omega_z\sin(\beta - \rho_2)\sin(\psi_6)) + (\cos(\beta)(\sin(\eta_6)\sin(\rho_2) \\ & - \cos(\eta_6)\cos(\psi_6)\cos(\rho_2)) - \sin(\beta)(\cos(\rho_2)\sin(\eta_6) \\ & + \cos(\eta_6)\cos(\psi_6)\sin(\rho_2)))(\dot{x} + \omega_y z_{od} - y_{or}\omega_z \\ & + \dot{\beta}l_{br}\sin(\beta - \kappa_4 - \rho_2) - l_{br}\omega_y\sin(\beta - \kappa_4 - \rho_2) \\ & - l_{br}\dot{\rho}_2\sin(\beta - \kappa_4 - \rho_2) + \dot{\beta}l_{db}\sin(\beta - \kappa_2) \\ & - l_{db}\omega_y\sin(\beta - \kappa_2)) - \cos(\eta_6)\sin(\psi_6)(x_{od}\omega_z - \omega_x z_{od} \\ & - l_{br}\omega_z\cos(\beta - \kappa_4 - \rho_2) + l_{br}\omega_x\sin(\beta - \kappa_4 - \rho_2) \\ & - l_{db}\omega_z\cos(\beta - \kappa_2) + l_{db}\omega_x\sin(\beta - \kappa_2)))/R_w.\end{aligned} \tag{B5}$$

APPENDIX: INTERMEDIARY VARIABLES

To calculate the linear velocity of the rover origin that corresponds to one of the wheels reaching its maximum speed, we define the following intermediary variables, used in Subsection 2.7 and expressed here for conciseness:

$$a_{11} = \cos(\eta_1)\sin(\beta) + \cos(\beta)\cos(\psi_1)\sin(\eta_1), \tag{C1}$$

$$a_{12} = \cos(\beta)\cos(\eta_1) - \cos(\psi_1)\sin(\beta)\sin(\eta_1), \tag{C2}$$

$$\begin{aligned} a_{13} = {} & y_{of}(\cos(\eta_1)\sin(\beta) + \cos(\beta)\cos(\psi_1)\sin(\eta_1)) \\ & + \sin(\eta_1)\sin(\psi_1)(x_{od} + l_{fd}\cos(\beta - \varkappa_1)), \end{aligned} \tag{C3}$$

$$\begin{aligned} b_1 = {} & (\cos(\beta)\cos(\eta_1) - \cos(\psi_1)\sin(\beta)\sin(\eta_1))(\omega_y x_{od} + \omega_x y_{of} \\ & + l_{fd}\cos(\beta - \varkappa_1)(\dot{\beta} + \omega_y)) \\ & - (\cos(\eta_1)\sin(\beta) + \cos(\beta)\cos(\psi_1)\sin(\eta_1))(\omega_y z_{od} \\ & - l_{fd}\sin(\beta - \varkappa_1)(\dot{\beta} + \omega_y)) \\ & + \sin(\eta_1)\sin(\psi_1)(\omega_x z_{od} - l_{fd}\omega_x\sin(\beta - \varkappa_1)), \end{aligned} \tag{C4}$$

$$a_{21} = \cos(\beta)\cos(\psi_2)\sin(\eta_2) - \cos(\eta_2)\sin(\beta), \tag{C5}$$

$$a_{22} = \cos(\beta)\cos(\eta_2) + \cos(\psi_2)\sin(\beta)\sin(\eta_2), \tag{C6}$$

$$\begin{aligned} a_{23} = {} & y_{of}(\cos(\eta_2)\sin(\beta) - \cos(\beta)\cos(\psi_2)\sin(\eta_2)) \\ & + \sin(\eta_2)\sin(\psi_2)(x_{od} + l_{fd}\cos(\beta + \varkappa_1)), \end{aligned} \tag{C7}$$

$$\begin{aligned} b_2 = {} & (\cos(\eta_2)\sin(\beta) - \cos(\beta)\cos(\psi_2)\sin(\eta_2))(\omega_y z_{od} \\ & - l_{fd}\sin(\beta + \varkappa_1)(\dot{\beta} - \omega_y)) \\ & - (\cos(\beta)\cos(\eta_2) + \cos(\psi_2)\sin(\beta)\sin(\eta_2))(\omega_x y_{of} - \omega_y x_{od} \\ & + l_{fd}\cos(\beta + \varkappa_1)(\dot{\beta} - \omega_y)) \\ & + \sin(\eta_2)\sin(\psi_2)(\omega_x z_{od} + l_{fd}\omega_x\sin(\beta + \varkappa_1)), \end{aligned} \tag{C8}$$

$$a_{31} = \sin(\beta + \eta_3 + \rho_1), \tag{C9}$$

$$a_{32} = \cos(\beta + \eta_3 + \rho_1), \tag{C10}$$

$$a_{33} = y_{om}\sin(\beta + \eta_3 + \rho_1), \tag{C11}$$

$$\begin{aligned} b_3 = {} & \omega_y x_{od}\cos(\beta + \eta_3 + \rho_1) + \omega_x y_{om}\cos(\beta + \eta_3 + \rho_1) \\ & - \omega_y z_{od}\sin(\beta + \eta_3 + \rho_1) \\ & - \dot{\beta} l_{db}\cos(\eta_3 - \varkappa_2 + \rho_1) \\ & - l_{db}\omega_y\cos(\eta_3 - \varkappa_2 + \rho_1) + \dot{\beta} l_{bm}\cos(\eta_3 + \varkappa_3) \\ & + l_{bm}\omega_y\cos(\eta_3 + \varkappa_3) + l_{bm}\dot{\rho_1}\cos(\eta_3 + \varkappa_3), \end{aligned} \tag{C12}$$

$$a_{41} = \sin(\eta_4 - \beta + \rho_2), \tag{C13}$$

$$a_{42} = \cos(\eta_4 - \beta + \rho_2), \tag{C14}$$

$$a_{43} = -y_{om}\sin(\eta_4 - \beta + \rho_2), \tag{C15}$$

$$\begin{aligned} b_4 = {} & \dot{\beta} l_{db}\cos(\eta_4 - \varkappa_2 + \rho_2) - l_{db}\omega_y\cos(\eta_4 - \varkappa_2 + \rho_2) \\ & + \omega_y x_{od}\cos(\eta_4 - \beta + \rho_2) \\ & - \omega_x y_{om}\cos(\eta_4 - \beta + \rho_2) \\ & - \omega_y z_{od}\sin(\eta_4 - \beta + \rho_2) - \dot{\beta} l_{bm}\cos(\eta_4 + \varkappa_3) \\ & + l_{bm}\omega_y\cos(\eta_4 + \varkappa_3) + l_{bm}\dot{\rho_2}\cos(\eta_4 + \varkappa_3), \end{aligned} \tag{C16}$$

$$\begin{aligned} a_{51} = {} & \cos(\beta)(\cos(\eta_5)\sin(\rho_1) + \cos(\psi_5)\cos(\rho_1)\sin(\eta_5)) \\ & + \sin(\beta)(\cos(\eta_5)\cos(\rho_1) \\ & - \cos(\psi_5)\sin(\eta_5)\sin(\rho_1)), \end{aligned} \tag{C17}$$

$$\begin{aligned} a_{52} = {} & \cos(\beta)(\cos(\eta_5)\cos(\rho_1) - \cos(\psi_5)\sin(\eta_5)\sin(\rho_1)) \\ & - \sin(\beta)(\cos(\eta_5)\sin(\rho_1) \\ & + \cos(\psi_5)\cos(\rho_1)\sin(\eta_5)), \end{aligned} \tag{C18}$$

$$\begin{aligned} a_{53} = {} & y_{or}(\cos(\beta)(\cos(\eta_5)\sin(\rho_1) + \cos(\psi_5)\cos(\rho_1)\sin(\eta_5)) \\ & + \sin(\beta)(\cos(\eta_5)\cos(\rho_1) \\ & - \cos(\psi_5)\sin(\eta_5)\sin(\rho_1)) - \sin(\eta_5)\sin(\psi_5)(l_{db}\cos(\beta + \varkappa_2) \\ & - x_{od} + l_{br}\cos(\beta + \varkappa_4 + \rho_1)), \end{aligned} \tag{C19}$$

$$\begin{aligned} b_5 = {} & \omega_x\sin(\eta_5)\sin(\psi_5)(z_{od} + l_{db}\sin(\beta + \varkappa_2) + l_{br}\sin(\beta + \varkappa_4 + \rho_1)) \\ & - (\cos(\beta)(\cos(\eta_5)\sin(\rho_1) \\ & + \cos(\psi_5)\cos(\rho_1)\sin(\eta_5)) + \sin(\beta)(\cos(\eta_5)\cos(\rho_1) \\ & - \cos(\psi_5)\sin(\eta_5)\sin(\rho_1)))(\omega_y z_{od} + \dot{\beta} l_{br}\sin(\beta + \varkappa_4 \\ & + \rho_1) + l_{br}\omega_y\sin(\beta + \varkappa_4 + \rho_1) + l_{br}\dot{\rho_1}\sin(\beta + \varkappa_4 + \rho_1) \\ & + \dot{\beta} l_{db}\sin(\beta + \varkappa_2) + l_{db}\omega_y\sin(\beta + \varkappa_2)) \\ & - (\cos(\beta)(\cos(\eta_5)\cos(\rho_1) - \cos(\psi_5)\sin(\eta_5)\sin(\rho_1)) \\ & - \sin(\beta)(\cos(\eta_5)\sin(\rho_1) + \cos(\psi_5)\cos(\rho_1)\sin(\eta_5))) \\ & (\dot{\beta} l_{br}\cos(\beta + \varkappa_4 + \rho_1) - \omega_x y_{or} - \omega_y x_{od} \\ & + l_{br}\omega_y\cos(\beta + \varkappa_4 + \rho_1) + l_{br}\dot{\rho_1}\cos(\beta + \varkappa_4 + \rho_1) \\ & + \dot{\beta} l_{db}\cos(\beta + \varkappa_2) + l_{db}\omega_y\cos(\beta + \varkappa_2)), \end{aligned} \tag{C20}$$

$$\begin{aligned} a_{61} = {} & \cos(\beta)(\cos(\eta_6)\sin(\rho_2) + \cos(\psi_6)\cos(\rho_2)\sin(\eta_6)) \\ & - \sin(\beta)(\cos(\eta_6)\cos(\rho_2) \\ & - \cos(\psi_6)\sin(\eta_6)\sin(\rho_2)), \end{aligned} \tag{C21}$$

$$\begin{aligned} a_{62} = {} & \cos(\beta)(\cos(\eta_6)\cos(\rho_2) - \cos(\psi_6)\sin(\eta_6)\sin(\rho_2)) \\ & + \sin(\beta)(\cos(\eta_6)\sin(\rho_2) \\ & + \cos(\psi_6)\cos(\rho_2)\sin(\eta_6)), \end{aligned} \tag{C22}$$

$$\begin{aligned} a_{63} = {} & - y_{or}(\cos(\beta)(\cos(\eta_6)\sin(\rho_2) + \cos(\psi_6)\cos(\rho_2)\sin(\eta_6)) \\ & - \sin(\beta)(\cos(\eta_6)\cos(\rho_2) \\ & - \cos(\psi_6)\sin(\eta_6)\sin(\rho_2))) \\ & - \sin(\eta_6)\sin(\psi_6)(l_{br}\cos(\varkappa_4 - \beta + \rho_2) - x_{od} \\ & + l_{db}\cos(\beta - \varkappa_2)), \end{aligned} \tag{C23}$$

$$\begin{aligned} b_6 = {} & \sin(\eta_6)\sin(\psi_6)(\omega_x z_{od} - l_{db}\omega_x\sin(\beta - \varkappa_2) + l_{br}\omega_x\sin(\varkappa_4 - \beta \\ & + \rho_2)) - (\cos(\beta)(\cos(\eta_6)\sin(\rho_2) \\ & + \cos(\psi_6)\cos(\rho_2)\sin(\eta_6)) - \sin(\beta)(\cos(\eta_6)\cos(\rho_2) \\ & - \cos(\psi_6)\sin(\eta_6)\sin(\rho_2)))(\omega_y z_{od} + \dot{\beta} l_{db}\sin(\beta - \varkappa_2) \\ & - l_{db}\omega_y\sin(\beta - \varkappa_2) - \dot{\beta} l_{br}\sin(\varkappa_4 - \beta + \rho_2) \\ & + l_{br}\omega_y\sin(\varkappa_4 - \beta + \rho_2) + l_{br}\dot{\rho_2}\sin(\varkappa_4 - \beta + \rho_2)) \\ & - (\cos(\beta)(\cos(\eta_6)\cos(\rho_2) - \cos(\psi_6)\sin(\eta_6)\sin(\rho_2)) \\ & + \sin(\beta)(\cos(\eta_6)\sin(\rho_2) + \cos(\psi_6)\cos(\rho_2)\sin(\eta_6))) \\ & (\omega_x y_{or} - \omega_y x_{od} - \dot{\beta} l_{db}\cos(\beta - \varkappa_2) + l_{db}\omega_y\cos(\beta - \varkappa_2) \\ & - \dot{\beta} l_{br}\cos(\varkappa_4 - \beta + \rho_2) \\ & + l_{br}\omega_y\cos(\varkappa_4 - \beta + \rho_2) + l_{br}\dot{\rho_2}\cos(\varkappa_4 - \beta + \rho_2)). \end{aligned} \tag{C24}$$