3D KINEMATIC PASSIVE-STEERING CONTROL WITH ROLL COMPENSATION FOR ROVER TRAJECTORY FOLLOWING OVER ROUGH TERRAIN

Virtual Conference 19-23 October 2020

Emanuele Aucone[1], Patrick McGarey[2]

[1]University of Pisa, via G. Caruso 16, 56122 Pisa, Italy, now at ETH Zürich, Universitatstr. 2, 8092 Zürich, Switzerland, E-mail: emanuele.aucone@usys.ethz.ch
[2]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena, CA 91109, USA, E-mail: patrick.mcgarey@jpl.nasa.gov

ABSTRACT

Some of the most appealing areas for future planetary surface exploration lie in rough, uneven terrains, such as craters and cold traps, which are currently inaccessible by state-of-the-art robotic systems [1]. To provide modularity for access and in-situ sampling within such extreme environments, Jet Propulsion Laboratory (JPL) and California Institute of Technology have collaborated to develop the DuAxel rover system, a modular robot composed of two Axel rappelling vehicles docked to a central module into a four-wheeled configuration, suited for driving long distances due to articulated passive-steering capabilities [2]. Inspired by Carnegie Mellon's Zoë rover [3], a 3D kinematic control strategy, leveraging a novel, DuAxel-centric model, has been developed to enable precise trajectory following over rough, flat terrain with presence of obstacles.

1 INTRODUCTION

Recent scientific findings suggest potential high-value science targets including lava tubes on the Earth's Moon [4] and Recurring Slope Lineae (RSL) on Mars [5] (Fig. 1). With respect to RSL, the potential for water ice at the surface has been proposed, which would make for a compelling mission in the near future. However, extreme terrains still remain difficult to access due to steep ground and the increased presence of obstacles.

From a mission perspective, the DuAxel rover system (Fig. 2) is suited to explore extreme terrains due to its unique kinematic configuration; its four-wheeled design allows for reaching a distance point from a safe landing location. Upon arrival, one of the system's two Axel rovers (which make up the wheels) undocks and rappels downslope, while the remaining Axel and the central body remain topside to act as an anchor and to provide long-range communication and energy. As the detached Axel descends into the area of interest, receiving power and support from the tether, it relays data using on-board sensors to take measurements and actuators to collect samples, all included in a bay tucked inside the wheels.

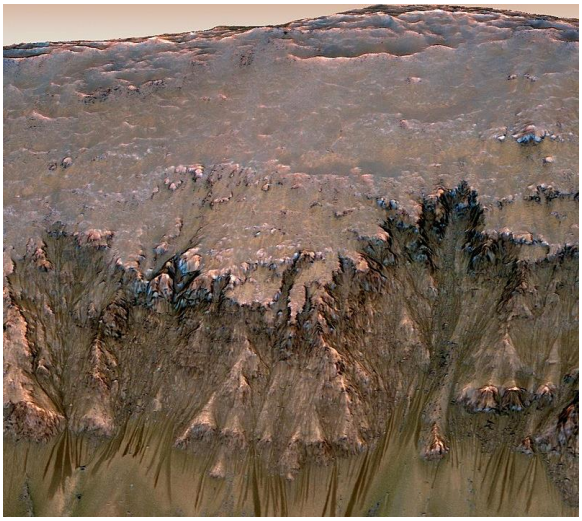

Figure 1: RSL on Mars, borrowed from [5].

Figure 2: The DuAxel four-wheeled rover system.

This work focuses on the development of a 3D kinematic, passive-steering control strategy with roll motion compensation to enhance exact trajectory following over flat terrains in presence of obstacles.

First, a novel kinematic model is formalized for DuAxel, which provides a complete and valid mathematical formulation for the kinematics of the robot. Then, to address the three-dimensional nature of the task, the presented control method extends the capabilities of a planar kinematic control integrating two additional terms in the control equations, in order to compensate roll motion caused by the interaction with the obstacles, which directly affects the orientation of the rover making it diverge from the desired path. Results in both simulation and experimental environments show effective enhancements in the mobility capabilities over obstacles, establishing the viability of DuAxel passive-steering locomotion for precise navigation from landing sites to the extreme region of interest.

2 NOVEL 3D KINEMATIC MODEL

The kinematic model is formalized to provide a complete and general mathematical model of the kinematics of the DuAxel rover in a 3D space.

2.1 Model Formulation

Starting from the theory of parallel robots, the rover is considered as a three-dimensional closed-chain mechanism (Fig. 3).

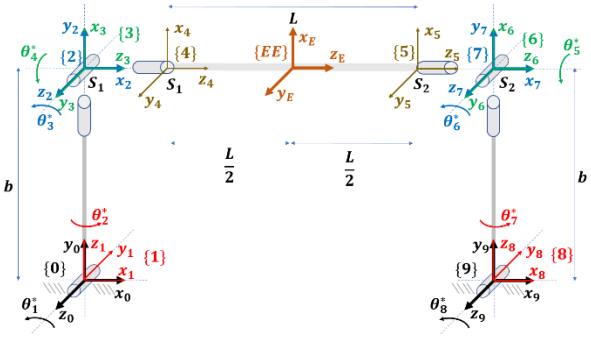

Figure 3: Coordinate reference frames of the model.

The main features and assumptions for the analysis follow the real DuAxel rover design: 1) there are only revolute joints, depicted as cylindrical forms, each one allowing a rotational motion around one single axis - the wheels are neglected in the model; 2) the three joints in the upper left (the same for the upper right) represent a unique spherical joint, so they are located in the same point even if represented in different locations - confusing and overlapped representation have been avoided; 3) the colors chosen for the frames aim to consolidate the symmetry between the two Axel vehicles and between the mechanisms behind the motion of each side of the rover; 4) each reference frame, from **{0}** to **{4}** and from **{9}** to **{5}**, is chosen according to the Denavit-Hartenberg (D-H) convention, but we apply it singularly for each Axel - this choice simplifies the analysis due to the symmetry of the model; 5) the frames **{1}**, **{2}**, **{3}** and **{8}**, **{7}**, **{6}** correspond to the yaw, pitch, roll of the rear and front Axel respectively.

2.2 Kinematic Equations

To solve the kinematics of the parallel mechanism, the following steps are performed:

1. detach the End-Effector (EE) from the system operating several 'cuts' on the frames **{4}** and **{5}**;
2. introduce a local parametrization $x \in SE(3)$ to represent the pose of the EE in the global frame **{0}** with a 4x4 homogeneous transformation matrix $T_{0,E}(x)$;
3. write the transformation matrices, expressed with the local parametrization x, related to the reference frames anchored to the 'cuts' w.r.t the global frame **{0}**, obtaining $T_{0,4}(x)$ and $T_{0,5}(x)$;
4. reduce the system to two serial chains - the two Axels – separating them from the EE;
5. write the kinematic equations of the two chains into the joint variables space q (with D-H), from the reference frames **{0}** to the frames anchored to the cuts, obtaining $T_{0,4}(q)$ and $T_{0,5}(q)$;
6. restore the integrality constraints, imposing the equalities between the calculated transformation matrices into the two task spaces.

Operating the aforementioned steps, direct and inverse kinematics of the rover can be solved. It is important for the purpose of this work to key focus on the main terms that the control method uses, which are the transformation matrices between the ground and the two passive spherical joints:

$$\begin{cases} T_{0,4}(q) = T_{0,4}(\theta_1^*, \theta_2^*, \theta_3^*, \theta_4^*, b) \\ T_{9,5}(q) = T_{0,4}(\theta_5^*, \theta_6^*, \theta_7^*, \theta_8^*, b) \end{cases} \quad (1)$$

which are similarly computed due to the symmetry of the model, and are functions of the roll, pitch and yaw joint variables of the two Axel rovers. From a mathematical point of view, the general characteristic of the kinematic model accepts constrained joints to be negated (set to zero), which allows the computation to remain consistent and valid.

3 3D KINEMATIC CONTROL

The presented control strategy aims to exploit the passive-steering capabilities of DuAxel to enable precise trajectory tracking, taking into account the roll motion when the rover is crossing over

obstacles that usually make the rover drift from the desired heading direction.

3.1 Control Assumptions and Features

The main assumptions to consider for DuAxel are reported as follows: 1) the pitch joint motions are neglected; 2) the front Axel is free to rotate in only two degrees-of-freedom (yaw and roll) whilst the rear Axel is free to rotate in only one degree-of-freedom (yaw), as the roll joint is fixed; 3) the front roll motion allows the wheels to follow the contours of uneven terrain, but this is achieved only if the wheels are always on the ground; this leads to a limitation in the size of the obstacles that can be traversed.

Thus, all the variables used by the control strategy are introduced in Tab. 1: note that the angle variables, previously introduced with D-H convention (θ_i^*), now aim to express the current value of that joint angles; moreover, subscripts for referring to rear/front Axel are used only when necessary to specify it, otherwise the symbol will refer to the generic Axel.

Parameter	Symbol
Front Axel roll angle	Φ_f
Single Axel steer angle	$\Psi_{(r/f)}$
Single Axel compensated steer angle	$\Psi_{(r/f)}^{comp}$
Single wheel compensated feedforward velocity scaling (for a generic wheel)	s^{comp}
DuAxel roll angle	φ
DuAxel pitch angle	ϑ
DuAxel yaw angle	ψ
DuAxel heading angle	α
Rover length (distance between steering joints)	L
Single Axel drop (height of steering joints above wheel centers axis)	b
Single Axel width (distance between wheel centers)	d
Wheel radius	r

Table 1: 3D kinematic control common variables

3.2 Control Equations

The common form of kinematic control for passively steered rover is composed of a feedforward term, which maps the commanded twist into the wheel velocities, and a feedback term, based on the error between the desired steering and the current steer angle. Therefore, steering motion can be achieved with closed-loop control of each Axel steer/yaw angle so that the global yaw angle of the rover could consequently follow the desired heading angle.

The presented 3D control extends the capabilities of a 2D kinematic control, to address driving over obstacles. Such 2D form, described for a single Axel (rear/front) in Eq. 2, assumes that all wheels drive in a plane.

$$\begin{bmatrix}\hat{v}_l\\ \hat{v}_r\end{bmatrix} = \begin{bmatrix}\frac{1}{\cos(\hat{\Psi})} & -\frac{d}{2}\\ \frac{1}{\cos(\hat{\Psi})} & \frac{d}{2}\end{bmatrix}\begin{bmatrix}\hat{V}\\ \hat{\omega}\end{bmatrix} + K_p\begin{bmatrix}-(\hat{\Psi}-\Psi)\\ \hat{\Psi}-\Psi\end{bmatrix} \qquad (2)$$

$$\text{with: } \hat{\Psi} = atan2\left(\frac{L}{2\hat{R}}\right),\ \ \hat{R} = \frac{\hat{V}}{\hat{\omega}}$$

The symbol ^ is used to denote commanded variables: in detail, $\hat{v}_l$ and $\hat{v}_r$ denote the commanded left and right wheel velocities for a single Axel, $\hat{V}$ and $\hat{\omega}$ are the commanded longitudinal and angular velocities respectively, K_p is the proportional gain used for the feedback term, $\hat{\Psi}$ is the commanded steer angle for a single generic Axel, and $\hat{R}$ is the commanded turn radius (positive for a counter-clockwise arc, negative for clockwise).

3.3 Three-dimensional Expansion

When the DuAxel rover drives over obstacles, the interaction with the external object causes non-zero front Axel roll angle, breaking the planar assumptions. Because of this motion, the steering/yaw of the rover is affected in two ways:

1) the robot yaw angle directly changes, so the rover will not follow the desired heading angle ($|\psi - \alpha| > 0$). Being the steering angles measured in the rover frame, the commanded steer angles must be changed in order to align the two Axels with the desired heading at the same time: a new term $\hat{\Psi}^{comp}$, already introduced in Tab. 1 as commanded compensated steer angle, must therefore be computed for each Axel;

2) the 3D wheel velocity vector is no longer parallel to the terrain plane: when projected onto the ground, wheels that are traversing obstacles appear to move slower, which if uncorrected causes deviations in the rover yaw angle. The roll compensation, in this case, is performed by computing a feedforward velocity scaling factor for each wheel (and of course for each Axel), denoted as s^{comp}, which correctly projects the velocity vector.

After having introduced these new terms, the 3D

passive-steering, kinematic control law, for a single Axel, is formalized as follows:

$$\begin{bmatrix}\hat{v}_l\\ \hat{v}_r\end{bmatrix} = \begin{bmatrix}s_l^{comp} & 0\\ 0 & s_r^{comp}\end{bmatrix}\begin{bmatrix}\frac{1}{\cos(\hat{\Psi}^{comp})} & -\frac{d}{2}\\ \frac{1}{\cos(\hat{\Psi}^{comp})} & \frac{d}{2}\end{bmatrix}\begin{bmatrix}\hat{V}\\ \hat{\omega}\end{bmatrix} + K_p\begin{bmatrix}-(\hat{\Psi}^{comp}-\Psi)\\ \hat{\Psi}^{comp}-\Psi\end{bmatrix} \quad (3)$$

From a different point of view, the effect of such 3D control approach is to make the projection of the 3D wheel velocity onto the ground plane match the wheel velocity commanded by the 2D controller in both direction and magnitude:

$$proj(\hat{v}^{3D}) = \hat{v}^{2D} \quad (4)$$

3.4 Wheel-Terrain Contact Points

The 3D kinematic model formalized in the previous section is exploited to compute wheel-terrain contact points in the global frame, which are required to calculate the two roll compensation terms.

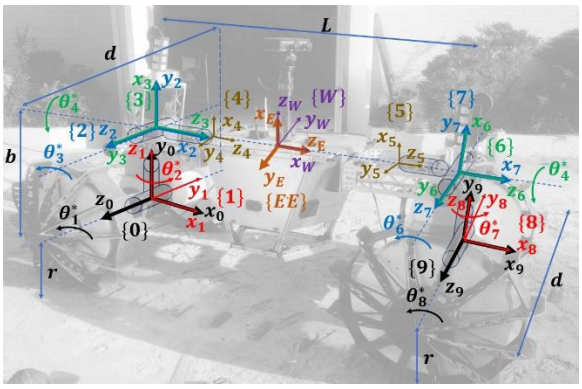

Figure 4: 3D kinematic model coordinate frames overlapped on the real DuAxel system.

The wheel-terrain contact points for rear and front, left and right, wheels can be generally expressed in the global frame through the following equations:

rear (l/r): $$P_W = T_{W,EE}\ T_{EE,4}\ T_{4,0}\ P_0 \quad (5)$$

front (l/r): $$P_W = T_{W,EE}\ T_{EE,5}\ T_{5,9}\ P_9 \quad (6)$$

where P_0 denotes the contact point for the left/right wheel in the rear Axel base frame $\{0\}$ and P_9 the contact point for the left/right wheel in the front Axel base frame $\{9\}$. Such points are equal, due to the symmetry of the two kinematic chains, and also constant: $P_0 = P_9 = [0, -r, \pm d/2, 1]^T$. It is easy to notice that the third element has positive sign for the right wheels and negative for left ones.

Introducing the variables expressed in Tab. 1, the Eqs. 5-6 can be expanded as:

$$P_W = T_{ZYX}(\varphi,\vartheta,\psi)\ T_{R^y}(-\pi/2)\ T_{R^z}(\pi)\ T_{T^z}(-L/2)\ T_{0,4}^{-1}(\Psi_r, b)P_0 \quad (7)$$

$$P_W = T_{ZYX}(\varphi,\vartheta,\psi)\ T_{R^y}(-\pi/2)\ T_{R^z}(\pi)\ T_{T^z}(L/2)\ T_{9,5}^{-1}(\Phi_f, \Psi_f, b)\ P_9 \quad (8)$$

where $T_{0,4}^{-1}$ and $T_{9,5}^{-1}$ are exactly the inverses of the homogeneous transformation matrices $T_{0,4}$ and $T_{9,5}$ computed in the kinematic analysis of Sec. 2.2; furthermore, T_{R^y}, T_{R^z} and T_{T^z}express two rotations around the y- and z-axis. Finally, $T_{ZYX}(\varphi,\vartheta,\psi)$ denotes the homogeneous transformation matrix between the global and the End-Effector frames, which is obviously function of the orientation of the rover (roll, pitch and yaw motions of DuAxel, introduced in Tab. 1) – note that in such specific case, the global frame {W} has been considered as co-located with the {EE} frame. Expressing in such complete form is very useful because if a parameter has to be neglected it can be directly be set to zero, but the formulation still remains general for every case, thus perfectly suited for future usage of the same model analysis.

4 ROLL COMPENSATION TERMS

In this section the calculation of both roll compensation terms is explained. These terms modify the commanded steer angles and velocity of the wheels, so that the rover can maintain the desired trajectory when is traversing complex terrain.

4.1 Compensated Steer Angles

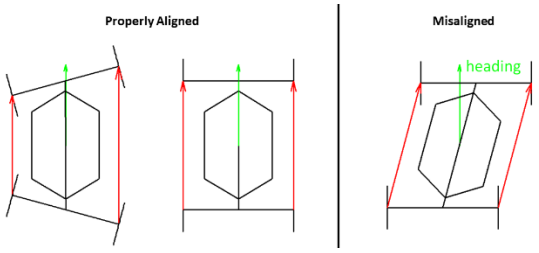

Figure 5: Alignment of the left and right side vectors from rear to front wheels with the heading direction, borrowed from [3].

To compute a unique solution for the commanded compensated steer angles $\hat{\Psi}^{comp}$ (for rear and Front Axel) one additional constraint is required: in the desired steering configuration the vectors

from rear-left to front-left and from rear-right to front-right wheel-terrain contact points has to be aligned with the DuAxel heading direction, as shown in Fig. 5. The compensation terms for both Axels are calculated iteratively using Newton's method - it has been developed and processed analytically, without using any solver.

***Algorithm 1*: calculation for both Axels**

1: $\underline{\Phi^{comp}} = \underline{\Phi}$
2: $\underline{error} = \underline{threshold}$
3: **while** $\underline{error} \geq \underline{threshold}$ **do**
4: compute P_W for rear wheels
5: compute P_W for front wheels
6: $P_W = T_{R_z}(\psi - \alpha)P_W$ (for all wheels)
7: $\underline{error} = \underline{\Phi} - \underline{\alpha^{Axel}}$ (for each Axel)
8: $\underline{\Phi^{comp}} = \underline{\Phi^{comp}} - \mathbb{J}^{-1}\ \underline{error}$
9: **end while**

Alg. 1 summarize the computation of $\underline{\Phi^{comp}}$, which denotes a 2x1 vector containing the compensated terms for rear and front Axel respectively - all the other underlined terms express 2x1 vectors themselves:

Line 1-to-3: the commanded compensated steer angles are initialized with the commanded steer angles without compensation, whilst Axels heading errors are initialized as the threshold, in order to perform the while loop at least once; the while loop is used to compute the Newton's method;

Line 4: the locations of the wheel-terrain contact points, for the rear Axel are, calculated in the global frame using Eq. 7, introducing the rear $\underline{\Phi^{comp}}$ instead of Ψ_r;

Line 5: the same procedure is computed for the front Axel using Eq. 8, introducing the front $\underline{\Phi^{comp}}$ instead of Ψ_f;

Line 6: the vectors from rear to front contact points, for both left and right sides, are constrained to be aligned with the heading direction. To perform this step, it is necessary to rotate about the z-axis of the global frame {W} by the difference between the rover yaw and the heading angle $(\psi - \alpha)$,which is geometrically estimated as follows:

$$\psi - \alpha = \frac{\Delta h_{left}\ (\psi - \alpha)_{right} + \Delta h_{right}\ (\psi - \alpha)_{left}}{\Delta h_{left} + \Delta h_{right}} \qquad (9)$$

with: $\Delta h_{left/right} = \left|P_{front}(z) - P_{rear}(z)\right|$,

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

where the elements $P_{rear/front}\ (x/y/z)$ stand for the x, y or z components of the rear/front wheel-terrain contact points P_W. With this computation it is easy to derive that $(\psi - \alpha)_{right}$ and $(\psi - \alpha)_{left}$ are the angles of the rear-to-front vectors into the ground plane, illustrated in Fig. 5, for left and right sides respectively, while $(\psi - \alpha)$ is a weighted average of these two angles: more weight is given to the side with a smaller difference in height between the wheels, i.e. the side driving on flatter terrain;

Line 7: the Axels heading errors are calculated: the single error is defined as the difference between the nominal commanded steer angle on flat terrain Φ and the single Axel heading α^{Axel} , measured by projecting the left and right contact points onto the ground plane:

$$\alpha^{Axel} = atan2(P_{right}(x) - P_{left}(x),\ -P_{right}(y) + P_{left}(y)) \qquad (10)$$

Where, by analogy, the elements $P_{right/left}(x/y)$ stand for the x or y components of the left/right contact points P_W for the single Axel – the two computed angles are englobed in the 2x1 vector $\underline{\alpha^{Axel}}$;

Line 8: according to Newton's method, the update to the commanded compensated steer angle vector $\underline{\Phi^{comp}}$ is calculated using the 2x2 Jacobian matrix of the error vector with respect to $\underline{\Phi^{comp}}$ itself - the Jacobian has been computed analytically.

4.2 Compensated Feedforward Velocities

***Algorithm 2*: calculation for a single wheel**

1: **loop**
2: **if** (rear wheel) **then**
3: compute P_W^t using Eq. 7
4: **else**
5: compute P_W^t using Eq. 8
6: $P_W^t(z) = P_W^t(z) + \Delta h_{rover}$
7: $\Delta h_{wheel} = P_W^t(z) - P_W^{t-1}$
8: $slope_{wheel} = sin^{-1}(\Delta h_{wheel}/(r\omega_{wheel}\Delta t))$
9: $s^{comp} = 1/cos(slope_{wheel})$
10: $P_W^{t-1} = P_W^t(z)$
11: **end loop**

The compensated feedforward velocity terms s^{comp} are necessary to scale the commanded wheel velocities when driving over the obstacles: indeed, after scaling by such terms s^{comp}, the projection of the 3D wheel velocity vector onto the ground plane is equivalent in both magnitude and direction to the 2D velocity vector on flat terrain. The calculation of the compensated term for a single wheel is explained by Alg. 2:

Line 2-to-5: at each timestep, the location of the single wheel-terrain contact point is computed in the global frame - differently from the previous algorithm, the current values of Ψ_r and Ψ_f are used;

Line 6: the offset Δh_{rover} denotes the difference in height of the {EE} frame origin between the current and previous timestep, expressed in the global frame. This value is summed to the z component of the contact point, depending on where the obstacle is located, i.e., the difference in height of the rover is summed only on the side (left/right) where the obstacle is traversed;

Line 7: by analogy, the difference in height of the contact points between the current and previous timestep, called Δh_{wheel}, is calculated;

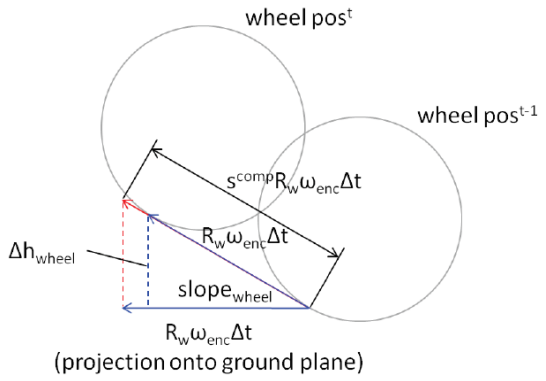

Figure 6: Wheel position at two adjacent timesteps as the wheel climbs a slope/obstacle, borrowed from [3].

Line 8: starting from Fig. 6, the slope angle is calculated geometrically: this value refers to a measure of the inclination of the obstacle and is useful to have a qualitative idea of "how much the projection will affect the wheel velocity";

Line 9: the compensated feedforward velocity scaling term s^{comp} is calculated, based on a simple trigonometric relation. It is important to note that these scaling terms increase with the steepness of the terrain/obstacle, as it should be;

Line 10: the previous wheel-terrain contact point is updated with the current value for the next iteration.

From a technical point of view, the algorithm has been developed to compute the compensated feedforward velocity scaling terms for all the wheels simultaneously.

5 RESULTS

In this section the results of the control architecture are exposed, proving enhanced mobility of the DuAxel rover system in presence of obstacles. The strategy is entirely implemented as a *ROS* node, using *C++* language: this high-level node performs exactly the algorithms described in the previous section.

5.1 Dynamic Simulation Results

Evaluation of the control method is performed exploiting *Gazebo* to visualize the effect of the control action into a 3D simulator - a desert compact terrain and two small ramps are added to the environment to simulate a ramp obstacle (dimensions are set in order to respect the physical limits imposed by the DuAxel wheels).

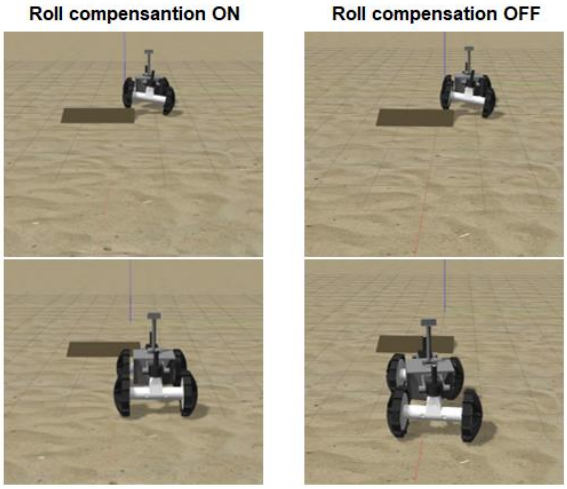

Figure 7: Gazebo simulation tests to compare the behavior of the controller with roll compensation enabled and disabled.

Several tests have been performed making the rover cross the ramp obstacle, positioned on the right, while following a straight trajectory (desired heading equal to zero), with a linear velocity of 0.1 m/s and a controller gain equal to 0.1, in order to compare the behavior of the rover when roll compensation is enabled or disabled - rover locomotion is addressed through a Differential Drive controller, which receives the wheel velocities commands from the high-level control node, whilst angle values are obtained by link and joint states or estimated from the central module

IMU. From Fig. 7 it is easy to visualize that with roll compensation disabled the kinematic controller behaves as the rover is driving on flat terrain, causing a final error in the difference between the yaw angle of the rover and the desired heading - the rover follows a curve that drift from the desired path. With roll compensation on, instead, the kinematic controller incorporates the steering compensation term, which takes into account the roll motion, and the velocity compensation terms, which correctly project the wheel velocity onto the ground: after the obstacle is traversed, the yaw and the heading are aligned and the rover is able to follow the straight direction.

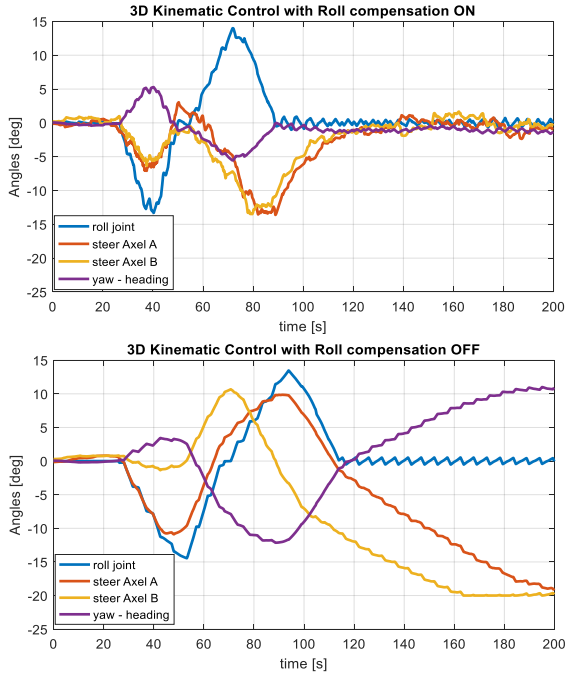

Figure 8: Results form a set of Gazebo simulations: comparison between roll compensation enabled and disabled demonstrates the enhancement in the trajectory following capabilities when the roll motion is incorporated into the control strategy formulation.

In Fig. 8, the same results are reported from a different perspective, plotting the values (on average over a set of simulations) of the front roll joint, the front and rear steer joints, and the difference between the yaw of the rover and the desired heading, over the entire trajectory. Plots effectively confirm that DuAxel faces an incorrect final configuration when the roll compensation is disabled, as the two steer angles are completely misaligned and the yaw angle diverges from the desired heading angle - this last value is easy to obtain or measure in Gazebo, but not in the real world because DuAxel does not have a state estimator or a specific sensor. On the other hand, the 3D kinematic control with roll compensation enabled definitely enhances the mobility of the rover, as it is able to traverse the obstacle and being aligned to the desired heading.

5.2 Experimental Results

Figure 9: Physical experiments with roll compensation enabled: the rover is able to follow a desired straight path.

Figure 10: Physical experiments with roll compensation disabled: the rover diverges from the desired trajectory.

The entire architecture is implemented on the DuAxel rover to check for trajectory following on real-time operations. The control method is developed as a *ROS* node, using *C++* and *Python*,

and incorporated in the locomotion framework of the rover system. Evaluation and comparison of roll compensation on and off are performed therefore through physical experiments - the Gazebo environment has been replicated in the JPL's mini Mars-Yard, where a similar ramp obstacle is built with concrete bricks.

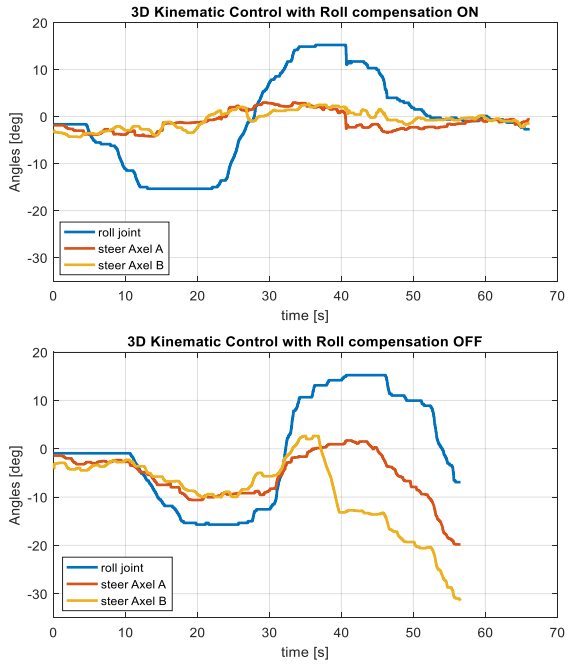

Figure 11: Analysis form a set of experiments aiming to replicate the simulation tests setup: results validate the control strategy on the physical system as well.

By analogy with the simulation testing, a set of field experiments confirms improvements in the mobility of DuAxel. From Figs. 9-to-11 it is easy to visualize that with roll compensation off the rover drifts from the desired path, as the kinematic control is not able to re-align the two steer angles (the rover is hitting the wall, as it is following an undesired curve). With roll compensation enabled, instead, the kinematic controller includes the two roll compensation terms: the result shows that after the obstacle is traversed the two steering angles are aligned with the heading angle, and the rover can continue to follow the straight direction - note that the difference between the rover yaw angle and the desired heading is not reported because the rover framework does not provide neither a state estimation nor an odometry system. Such results not only are comparable with the ones obtained from the simulation tests, but they are also physically coherent with the model, validating therefore the control strategy and demonstrating enhancements in the trajectory following capability.

6 CONCLUSIONS

In this work a 3D kinematic passive-steering control with roll compensation, which leverages a novel 3D model formalized, has been proposed to enhance trajectory following capabilities of the DuAxel rover system over flat terrain in presence of obstacles. Both simulation and experimental results have proved the advantages of integrating roll compensation terms to successfully address precise locomotion. A drawback of the presented method is the lack of information about the obstacles the rover has to climb, as well as the constraint on their size.

Future works could include the integration of a higher-level planner to manage the trajectory definition. Moreover, the estimation of some parameters needed could be handled leveraging on more robust approaches. Finally, a system for terrain mapping could surely enhance the control strategy in terms of defining the ground plane and estimating obstacles. To accomplish all these tasks, DuAxel would need additional capabilities and autonomy by leveraging visual and perception systems, as well as additional locomotion assets.

Acknowledgement

All the research has been carried out at the NASA's JPL, California Institute of Technology, Pasadena CA, with the Robotics Surface Mobility group, sponsored by JPL Visiting Student Research Program and the National Aeronautics and Space Administration.

References

[1] I. A. Nesnas, J. B. Matthews, P. Abad-Manterola, J. W. Burdick, J. A. Edlund, J. C. Morrison, R. D. Peters, M. M. Tanner, R. N. Miyake, B. S. Solish, R. C. Anderson (2012) Axel and DuAxel Rovers for the Sustainable Exploration of Extreme Terrains. *Journal of Field Robotics*, vol. 29(4), pp. 663–685.

[2] P. McGarey, W. Reid, and I. Nesnas (2019) Towards articulated mobility and efficient docking for the duaxel tethered robot system. *IEEE Aerospace Conference*, pp. 1–9.

[3] Neal Seegmiller and David Wettergreen (2011) Control of a passively steered rover using 3-d kinematics. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 607–612.

[4] Ronald Greeley (1971) Lava tubes and channels in the lunar marius hills. *The Moon*, 3:289–314.

[5] Runyon Kirby and Lujendra Ojha (2014) Recurring Slope Lineae. *Encyclopedia of Planetary Landforms*, pp.1–6.