Dynamics and Control of a Six-wheeled Rover with Rocker-Bogie Suspension System

Matteo Facci, Giuseppe Sensolini Arrà

Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Antonio Ruberti
Control Engineering
Sapienza Università di Roma

January 11, 2022

Contents

1 Introduction

Mobile robots are increasingly being used for applications in rough, outdoor terrain. These applications frequently necessitate robots traversing unprepared, rugged terrain in order to inspect a location or transport material. This is also true in the context of planetary exploration.
Robots are preferred over humans for planetary exploration because they require fewer resources and supplies. Rovers, in particular, have high mobility and can transport scientific equipment, making them the ideal solution to the problem.
Since the 1960s, many kinds of rovers have been developed for planetary exploration. Planetary exploration rovers can be wheeled, legged, or tracked, depending on the features of the mobile mechanism in robots. Wheeled mobile mechanisms have many advantages, including high speed on relatively flat terrain and ease of control. Planetary rovers with wheeled structures can be four-wheeled, six-wheeled, eight-wheeled, and so on.
Among these, the six-wheeled mobile rover with a rocker-bogie suspension system impresses in adaptability and obstacle climbing. These types of rovers are now used for Mars exploration.
The rocker-bogie system is the suspension arrangement developed in 1988 for use in NASA's Mars rover Sojourner, and which has since become NASA's favored design for rovers. It has been used in the 2003 Mars Exploration Rover mission robots Spirit and Opportunity, on the 2012 Mars Science Laboratory (MSL) mission's rover Curiosity, and the Mars 2020 rover Perseverance.

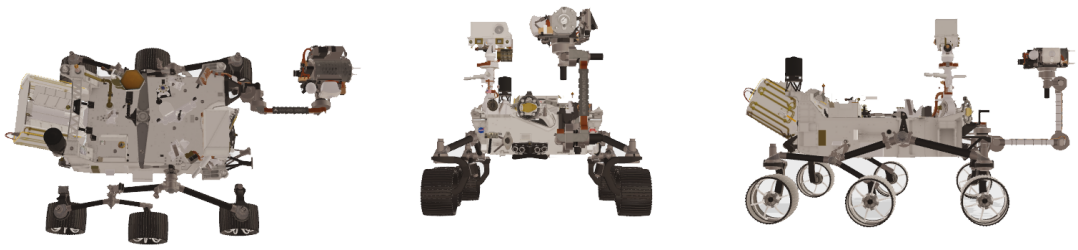

Figure 1: Top (Left), Front (Center), and Lateral (Right) views of the Perseverance rover by NASA

The rocker part of the suspension comes from the rocking aspect of the larger, body-mounted linkage on each side of the rover. These rockers are connected to each other and the vehicle chassis through a differential. Relative to the chassis, the rockers will rotate in opposite directions to maintain approximately equal wheel contact. The chassis maintains the average pitch angle of both rockers. One end of a rocker is fitted with a drive wheel, and the other end is pivoted to the bogie.
The bogie part of the suspension refers to the smaller linkage that pivots to the rocker in

the middle and which has a drive wheel at each end. Bogies were commonly used as load wheels in the tracks of army tanks as idlers distributing the load over the terrain, and were also quite commonly used in trailers of semi-trailer trucks. Both tanks and semi-trailers now prefer trailing arm suspensions.

The rocker-bogie design has no springs or stub axles for each wheel, allowing the rover to climb over obstacles (such as rocks) that are up to twice the wheel's diameter in size while keeping all six wheels on the ground. As with any suspension system, the tilt stability is limited by the height of the center of gravity. Systems using springs tend to tip more easily as the loaded side yields.

A six-wheeled rover with a rocker-bogie structure was analyzed for a planar case in this work. The rover's detailed kinematic model has been constructed, and the dynamic model has been derived based on an analysis of the entire mechanics and the formulation of the dynamics found in the literature. The simulation studies focused on the rover's climbing capability as well as its overall behavior on uneven terrains.

The study found that with the proper control action, the rover can pass through a plane surface, an inclined surface, and an inclined ditch while maintaining a constant velocity. As a result, a proper controller was created. It consists of a set of proportional integral derivative (PID) controllers applied to the rover's suspension model, and it is demonstrated that with the latter, the rover can perform well and successfully complete the required tasks.

2 Rover kinematic analysis

The rover structure is briefly described in this section. The rover is made up of various elements that are assumed to be rigid. It has six independently driven wheels that are mounted on a frame. The latter is comprised of two rocker arms that are linked to the main body. Each rocker has a rear-wheel attached to one end and a secondary rocker, known as a bogie, attached to the other. A drive wheel is located at each end of the bogie, and the bogie is connected to the rocker via a free pivoting joint. The rockers are differentially connected to the main body so that the pitch angle of the body is the average of the pitch angles of the rockers. With this design, each wheel tends to stay in contact with the ground while traversing uneven terrain. The rover's weight is evenly distributed across the wheels, resulting in good traction developed by each wheel.

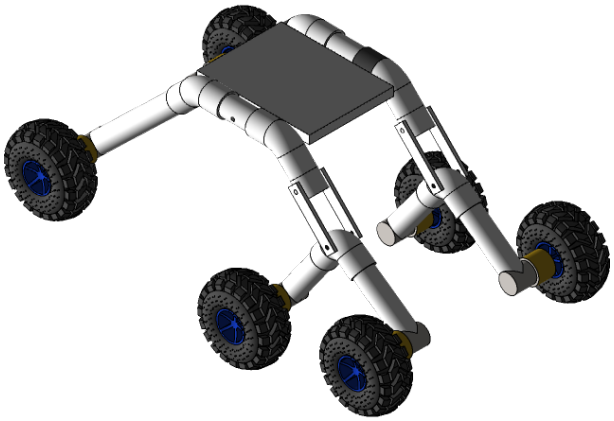

Figure 2: Rocker-bogie suspension system

Several assumptions can be made to simplify the rover analysis. Due to space system constraints such as weight and power, rovers must travel at low speeds (approximately 3 cm/s), resulting in very small dynamic effects.
Rover includes sensors such as stereo cameras and Lidar that can generate a detailed description of the environment around the robot; thus, ground characteristics are considered as input to the simulation. Another assumption is that the rover wheels will stay in contact with the ground; additionally, because the vehicle speed is so slow, the wheels will not bounce. As a result, the model assumes that all six wheels are always in contact with the ground.
The configuration and the position of the rover can be defined by the following parameters (Figure (3)):

- The position of the center of the body (coincident with the center of mass) in the

world reference frame O^w is defined by $x_r\ z_r$.

- The rover's orientation is defined only by the pitch angle ϕ, using the Euler notation.
- The configuration of the rocker-bogie mechanism is defined by the angles θ_i with $i = 1, 2, 3$ measured in the plane $x_r - z_r$, which are given by the physical structure of the suspension system itself as well as the links l_1, l_2, l_3, l_b and the γ and β angles.

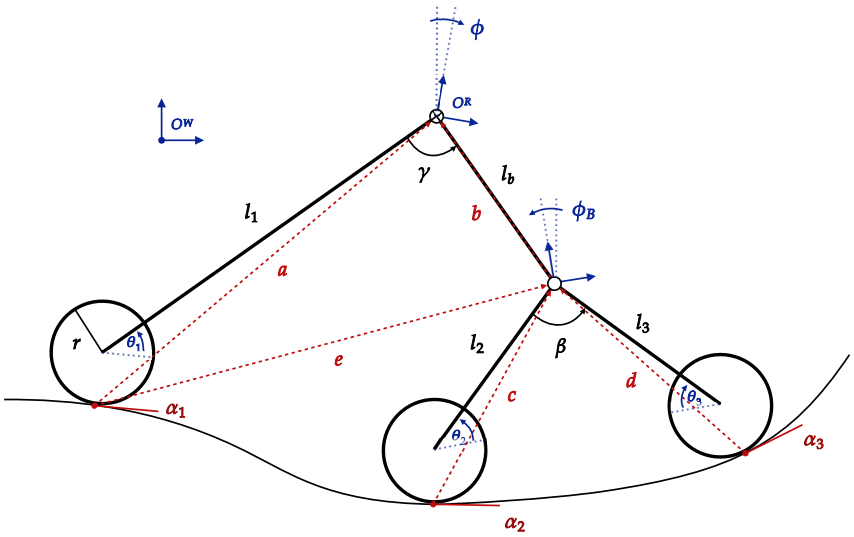

Figure 3: Rover kinematics

As for the ground-to-pivots vectors, they are:

$$a = R(\phi) \begin{pmatrix} l_1 \cos\left(\frac{\pi-\gamma}{2}\right) - r\sin(\alpha_1) \\ l_1 \sin\left(\frac{\pi-\gamma}{2}\right) + r\cos(\alpha_1) \end{pmatrix} \tag{1}$$

$$b = R(\phi) \begin{pmatrix} l_b \cos\left(\frac{\pi+\gamma}{2}\right) \\ l_b \sin\left(\frac{\pi+\gamma}{2}\right) \end{pmatrix} \tag{2}$$

$$c = R(\phi)\, R(\phi_B) \begin{pmatrix} l_2 \cos\left(\frac{\pi-\beta}{2}\right) - r\sin(\alpha_2) \\ l_2 \sin\left(\frac{\pi-\beta}{2}\right) + r\cos(\alpha_2) \end{pmatrix} \tag{3}$$

$$d = R(\phi)\, R(\phi_B) \begin{pmatrix} l_2 \cos\left(\frac{\pi+\beta}{2}\right) - r\sin(\alpha_2) \\ l_2 \sin\left(\frac{\pi+\beta}{2}\right) + r\cos(\alpha_2) \end{pmatrix} \tag{4}$$

$$e = a - b = R(\phi_B) \begin{pmatrix} l_1 \cos\left(\frac{\pi-\gamma}{2}\right) - r\sin(\alpha_1) - l_b \cos\left(\frac{\pi+\gamma}{2}\right) \\ l_1 \sin\left(\frac{\pi-\gamma}{2}\right) + r\cos(\alpha_1) - l_b \sin\left(\frac{\pi+\gamma}{2}\right) \end{pmatrix} \tag{5}$$

with

$$R(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \tag{6}$$

$$R(\phi_B) = \begin{pmatrix} \cos(\phi_{\mathrm{B}}) & -\sin(\phi_{\mathrm{B}}) \\ \sin(\phi_{\mathrm{B}}) & \cos(\phi_{\mathrm{B}}) \end{pmatrix} \tag{7}$$

as rotation matrices respectively for the rover's pitch angle and for the bogie's pitch angle. At this point, it is necessary to outline the inverse kinematics procedure. All the wheels are assumed to be in contact with the ground at only one point for each wheel. Therefore, starting from this assumption is possible to compute, given the terrain profile (and, consequently, the α_i angles), the position of each wheel center and of the other critical points.

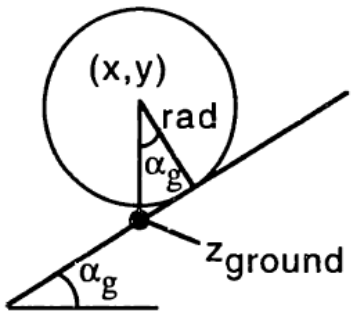

Figure 4: Contact angle and calculation of the wheel center

Given an angle α_i and the contact point of the wheel, since we assumed that each wheel is on a plane surface, from trigonometric properties, α_i is equal to the contact angle.
The contact angle between a wheel and the ground is defined as the angle between the z axis and the wheel's contact point with the ground, as shown in Figure (4). The contact angle for each wheel is zero while the rover is on a flat ground. If the front wheel climbs over a rock while the other wheels stay on flat ground, the contact angle changes from positive to negative as the wheel climbs and descends (as in the case of α_i).
It is possible to model the wheel's motion even if there were just one contact point, whose position on the wheel is defined so that the vector from the contact point to the wheel's center is orthogonal to the velocity vector.
In order to complete the kinematic analysis by obtaining the coordinates of all the critical points, such that all the wheel centers, the center of mass and the bogie position, it is possible to observe from Figure (3) that the links provide rigid distances connecting some of these points.
Equations may be constructed to determine each of the distances or rigid lengths between various coordinate pairs using the x,z coordinates of each point. In this way, distances can be simply calculated using the Euclidean distance between two points or the Pythagorean theorem, with each rigid length acting as the hypotenuse. When this procedure is applied to the present situation, one obtains the following set of equations:

$$(x_2 - x_3)^2 + (z_2 - z_3)^2 = l_{23} \tag{8}$$
$$(x_2 - x_b)^2 + (z_2 - z_b)^2 = l_2 \tag{9}$$
$$(x_3 - x_b)^2 + (z_3 - z_b)^2 = l_3 \tag{10}$$
$$(x_b - x_r)^2 + (z_b - z_r)^2 = l_b \tag{11}$$
$$(x_1 - x_r)^2 + (z_1 - z_r)^2 = l_1 \tag{12}$$
$$(x_1 - x_b)^2 + (z_1 - z_b)^2 = l_{1b} \tag{13}$$

where l_{23} and l_{1b} are two virtual links connecting the respective fixed points. Definitely, with some additional constraints, these equations can be used to successfully derive the positions of all critical points.

3 Rover force analysis

The rover's force analysis is presented in this section. The force analysis assesses whether or not the rover is physically capable of accomplishing a task. A simple test for this is to see if the rover can maintain static equilibrium in its current position. Because the rover will be moving at a slow speed, this technique of evaluation is appropriate therefore a quasi-static approximation is the most suitable choice.
Furthermore, if the rover's different positions are closely spaced and timed, the torque distribution on the wheels that verifies the rover's balance can be estimated as the actual required torques under this assumption.
The load distribution on the wheels and the vehicle's stability are affected by the position of the center of mass (CoM). As a result, determining its position is crucial. The input force to the system can be calculated from this. The CoM's position is entirely determined by the rover's kinematics.

3.1 Static equilibrium equations

The input forces can be described as a wrench that will be located at the center of the body because the rover's force analysis is performed under the quasi-static assumption.
The planar analysis for rocker-bogie mobility evaluation is a general technique that can accommodate any rocker-bogie configuration input. It can then be used to investigate the influence of kinematic characteristics on load distribution.
The model is shown in Figure (5). There are six unknowns (T_i, N_i, $i = 1, 2, 3$) in this model, and four independent static equilibrium equations. On the rocker-bogie pivot joint, there are two equilibrium equations for the forces in the x and z axes, as well as two moment equations. Both the rocker and the bogie's moment equations must be zero at that point, resulting in two equations to be solved.
The static equilibrium equations are described by:

$$T_1 \cos\alpha_1 + T_2 \cos\alpha_2 + T_3 \cos\alpha_3 - N_1 \sin\alpha_1 - N_2 \sin\alpha_2 - N_3 \sin\alpha_3 + F_x = 0 \tag{14}$$

$$T_1 \sin\alpha_1 + T_2 \sin\alpha_2 + T_3 \sin\alpha_3 + N_1 \cos\alpha_1 + N_2 \cos\alpha_2 + N_3 \cos\alpha_3 - F_z = 0 \tag{15}$$

$$T_1 \cos\alpha_1 e_y - T_1 \sin\alpha_1 e_x - N_1 \sin\alpha_1 e_y - N_1 \cos\alpha_1 e_x + M_y - F_x b_y - F_z b_x = 0 \tag{16}$$

$$\begin{aligned} &T_2 \cos\alpha_2 c_y - T_2 \sin\alpha_2 c_x - N_2 \sin\alpha_2 c_y - N_2 \cos\alpha_2 c_x + \\ &+ T_3 \cos\alpha_3 d_y - T_3 \sin\alpha_3 d_x - N_3 \sin\alpha_3 d_y - N_3 \cos\alpha_3 d_x = 0 \end{aligned} \tag{17}$$

Where the traction forces T_1, T_2, T_3 are considered to be the inputs to the system, while F_x and F_z (in the O^R reference frame) are variables to be controlled.

Given the dynamic model of the whole rocker-bogie suspension system, since the aim is to control whether the rover (the rocker pivot) in the rocker frame coordinates follows the

position and velocity references or not, the first two equations of the model are sufficient to control the motion along the x and z axes. The other two equations as just mentioned refer to the moments of the pivots, in particular, they are useful in case of a controlled rotation of the bogie, and this is not the case, in fact, usually, the bogie pivot is considered only as a free hinge. The rover moves very slowly, therefore M_y can be replaced by the gravity torque shared by the rocker pivot. Knowing the values of the normal forces N_i we can, through a control action, determine the necessary traction forces in order to maintain the requested velocity and follow the position setpoints, satisfying the quasi-static equilibrium equations.
Equations (14), (15), (16), (17) can be further developed by introducing the motor torques τ_i as inputs, and relating motor torques and traction forces as

$$T_i = \frac{\tau_i}{r}\,, \quad i = 1, 2, 3 \tag{18}$$

where r is the wheel radius (assumed to be equal for each motor).

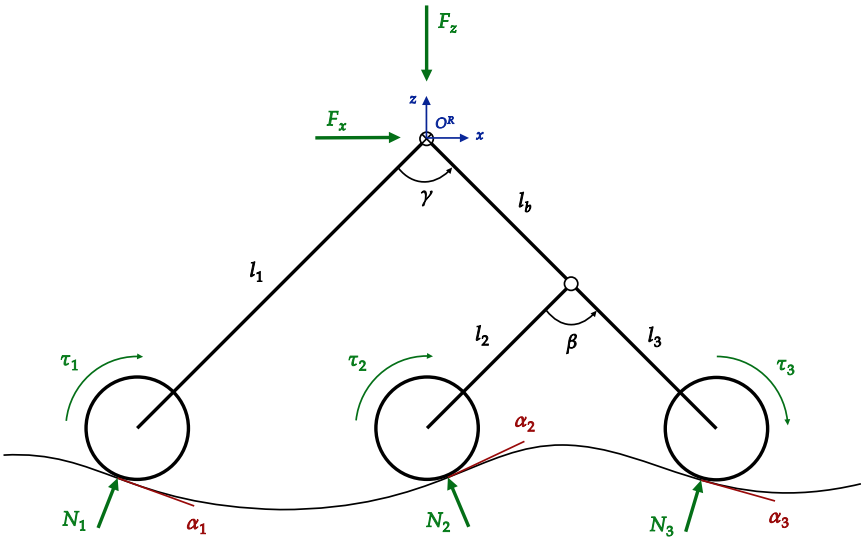

Figure 5: Rover force analysis

Previous equations can be expressed as:

$$F_x = -\frac{\tau_1}{r}\cos\alpha_1 - \frac{\tau_2}{r}\cos\alpha_2 - \frac{\tau_3}{r}\cos\alpha_3 + N_1\sin\alpha_1 + N_2\sin\alpha_2 + N_3\sin\alpha_3 \tag{19}$$

$$F_z = \frac{\tau_1}{r}\sin\alpha_1 + \frac{\tau_2}{r}\sin\alpha_2 + \frac{\tau_3}{r}\sin\alpha_3 + N_1\cos\alpha_1 + N_2\cos\alpha_2 + N_3\cos\alpha_3 \tag{20}$$

The ground's geometry is known, as are the directions of the traction and normal forces. The inverse kinematics approach, which considers the ground as a plane under each

wheel locally, determines the direction of these forces. The positions of the wheel contact points with respect to the rocker-bogie joint are necessary to perform the force analysis. Controlling or constraining some of the unknowns, on the other hand, can help with the analysis. Assuming a wheel is not slipping, $\tau_i = rT_i$, as previously stated. Because the traction forces and wheel torques are connected by the constant r, they are comparable. Additional significant constraints can be introduced. To begin, the normal force sign constraint must be checked:

$$N_i \geq 0 \ , \ \ \forall i \tag{21}$$

The ground would have to pull down on the wheel to maintain equilibrium if the normal forces were negative.
Second, the motors are not capable of producing limitless torque. It is necessary to consider the system's physical limitations:

$$|rT_i| \leq \tau_{sat} \ , \ \ \forall i \tag{22}$$

3.2 External forces

3.2.1 Normal forces

In this section we are going to compute the normal forces N_1, N_2, N_3 acting on the system. The normal forces on the three wheels and on the bogie pivot are computed as

$$N_1 = mg \ \frac{|b_x|}{|a_x| + |b_x|} \qquad N_B = mg \ \frac{|a_x|}{|a_x| + |b_x|} \tag{23}$$

$$N_2 = N_B \ \frac{|d_x|}{|c_x| + |d_x|} \qquad N_3 = N_B \ \frac{|c_x|}{|c_x| + |d_x|} \tag{24}$$

3.2.2 Aerodynamic effects

For the sake of completeness, we briefly deal with the case of the aerodynamic effects given by the longitudinal motion along an uneven terrain profile. Drag is just a force that opposes advancement and is calculated as:

$$D = \frac{1}{2} C_x \rho A v(t)^2 \tag{25}$$

where $C_x = 2.1$ (rover considered as a rectangular box) is the drag coefficient, $\rho \simeq 0.020 \ \frac{kg}{m^3}$ represents the surface air density of Mars, A is the front section area of the rover and v the speed of the robot with respect to time.
In the case of a rover on the surface of Mars, with very low speeds, the aerodynamic effects with respect to the other forces acting on the system can be neglected.

4 State-space representation

We would like to express the systems in classical nonlinear form like:

$$\begin{aligned} \dot{\xi} &= f(\xi) + \sum_i g_i(\xi) u_i + \Psi(\cdot) \\ y &= h(\xi) \end{aligned} \tag{26}$$

To do so, we define the momenta p_x and p_z of the center of mass along the x and z axes. It is now possible to introduce the state variables

$$\xi = (\xi_1,\ \xi_2,\ \xi_3,\ \xi_4)^T = (x,\ p_x,\ z,\ p_z)^T \in \mathbb{R}^4 \tag{27}$$

and the output variables

$$y = (y_1,\ y_2,\ y_3,\ y_4)^T = \xi \in \mathbb{R}^4 \tag{28}$$

while $\Psi(\cdot) \in \mathbb{R}^4$ in (26) are nonlinear functions linking subsystems through normal forces and taking into account the ground angles $\alpha_{1,2,3}$ (acting, in this context, as nonlinear disturbances). In particular

$$\Psi = \begin{bmatrix} 0 \\ \psi_x \\ 0 \\ \psi_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ \sin\alpha_1 & \sin\alpha_2 & \sin\alpha_3 \\ 0 & 0 & 0 \\ \cos\alpha_1 & \cos\alpha_2 & \cos\alpha_3 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix}$$

where normal forces N_1, N_2, N_3 are computed as in (23)-(24).

4.1 Torque-driven model

With the previous framework in mind, it is possible to define the torque inputs

$$u = (u_1,\ u_2,\ u_3)^T = [\tau_1,\ \tau_2,\ \tau_3]^T \tag{29}$$

Therefore the whole system dynamics can be rewritten as

$$\begin{aligned} \dot{\xi}_1 &= \frac{1}{m}\xi_2 \\ \dot{\xi}_2 &= g_{11}\, u_1 + g_{12}\, u_2 + g_{13}\, u_3 + \psi_x(\cdot) \\ \dot{\xi}_3 &= \frac{1}{m}\xi_4 \\ \dot{\xi}_4 &= g_{21}\, u_1 + g_{22}\, u_2 + g_{23}\, u_3 + \psi_z(\cdot) \\ y &= \xi \end{aligned} \tag{30}$$

where $g_{1i} = -\frac{\cos\alpha_i}{r}$ and $g_{2i} = \frac{\sin\alpha_i}{r}$ for $i = 1, 2, 3$.
The rover dynamics (30) can be expressed in matrix form as

$$\begin{aligned} \dot{\xi} &= A\,\xi + G(\xi)\,u + \Psi(\cdot) \\ y &= C\,\xi \end{aligned} \tag{31}$$

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1/m & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/m \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ g_{11} & g_{12} & g_{13} \\ 0 & 0 & 0 \\ g_{21} & g_{22} & g_{23} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \psi_x(\cdot) \\ 0 \\ \psi_z(\cdot) \end{bmatrix} \tag{32}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} \tag{33}$$

4.2 Traction-driven model

The nonlinear input matrix $G(\xi)$ in (32) can be substituted by a linear matrix B (not depending on trigonometric functions of α_i's) using as inputs horizontal and vertical components of the traction forces of the whole rover system, namely

$$T = (T_x,\ T_z)^T \tag{34}$$

with these inputs the system can be rewritten as

$$\begin{aligned} \dot{\xi} &= A\,\xi + B\,T + \Psi(\cdot) \\ y &= C\,\xi \end{aligned} \tag{35}$$

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1/m & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/m \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_x \\ T_z \end{bmatrix} + \begin{bmatrix} 0 \\ \psi_x(\cdot) \\ 0 \\ \psi_z(\cdot) \end{bmatrix} \tag{36}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} \tag{37}$$

It is important to notice that this approach relies on the assumptions that a mapping between system traction forces T and motor torques τ holds as in (18) at each instant of time. Also, in both models, we have to assume the absence of slip for each wheel (pure rotational motion) and we must assume that each wheel is always in contact with the ground at one point. The latter is in fact a plausible hypothesis as given the reduced speeds of the rover in relation to the size, mass and capacity of the rocker-bogie suspensions of traverse obstacles higher than the diameter of the rover's wheels, the robot is able to keep all wheels firmly on the ground.
In fact the rocker-bogie design eliminates the need for springs or stub axles in each wheel, allowing the rover to climb over obstacles twice the diameter of the wheel while retaining all six wheels on the ground. The tilt stability of any suspension system is limited by the height of the center of gravity. As the weighted side yields, systems with springs tend to tip more easily.
In presence of an obstacle the front wheels are driven against the obstacle by the middle and rear wheels in order to go over a vertical obstacle face. The front of the vehicle is subsequently lifted up and over the obstruction by the rotation of the front wheel. The middle wheel is then pressed against the impediment by the back wheels and pulled against it by the front wheels until it is raised up and over. Finally, the front two wheels drag the rear wheel over the obstruction. The vehicle's forward progress is slowed or entirely stopped during each wheel's traverse of the impediment. For the operational speeds at which these vehicles have been operated to far, this is not an issue.
This allows the system to operate also on steep sloping surfaces providing good stability. For instance, Perseverance rover by NASA is designed to withstand a tilt of 45 degrees in any direction without tipping over. For added protection and safe driving, the rover drivers avoid terrains that would cause a tilt of more than 30 degrees.
Each of the six wheels of the rover has its own motor. Individual steering motors are installed in the two front and two rear wheels, allowing the vehicle to turn in position. Each tire is also equipped with cleats, which provide traction when climbing in soft sand or scrambling over rocks. The maximum speed of these robots is limited to remove as many dynamic effects as possible, allowing the motors to be geared down and each wheel to lift a considerable amount of the vehicle's mass separately.

5 Control system for position and velocity tracking

The control strategy aims to track a reference trajectory mantaining a fixed velocity; in this section the control system is illustrated under the following assumptions:

- ground angles $\alpha_i(t)$ are known $\forall t$,
- normal forces $N_i(t)$ are known $\forall t$;

which are reasonable assumptions if an internal odometry is present.

The proposed control scheme (figure 6) uses the positions $\bar{x}(t)$, $\bar{z}(t)$ as a reference and the velocities $\hat{\dot{x}}(t)$, $\hat{\dot{z}}(t)$ serves as a feedforward.
The first controller is a classical PID controller, it acts as a position regulator. The velocity inputs $u(t)$ are generated as follows:

$$e(t) = \begin{bmatrix} \hat{x}(t) - x(t) \\ \hat{z}(t) - z(t) \end{bmatrix} \in \mathbb{R}^2 \tag{38}$$

$$u(t) = \begin{bmatrix} u_x(t) \\ u_z(t) \end{bmatrix} = \begin{bmatrix} K_{p_1,x}\, e_x(t) + K_{i_1,x} \int e_x(t)dt + K_{d_1,x} \frac{de_x(t)}{dt} \\ K_{p_1,z}\, e_z(t) + K_{i_1,z} \int e_z(t)dt + K_{d_1,z} \frac{de_z(t)}{dt} \end{bmatrix} \in \mathbb{R}^2 \tag{39}$$

The second controller (a PID controller again) takes as input a velocity error computed as

$$\epsilon(t) = \begin{bmatrix} \hat{\dot{x}}(t) - \dot{x}(t) + u_x(t) \\ \hat{\dot{z}}(t) - \dot{z}(t) + u_z(t) \end{bmatrix} \in \mathbb{R}^2 \tag{40}$$

and generates a traction control

$$T(t) = \begin{bmatrix} T_x(t) \\ T_z(t) \end{bmatrix} = \begin{bmatrix} K_{p_2,x}\, \epsilon_x(t) + K_{i_2,x} \int \epsilon_x(t)dt + K_{d_2,x} \frac{d\epsilon_x(t)}{dt} \\ K_{p_2,z}\, \epsilon_z(t) + K_{i_2,z} \int \epsilon_z(t)dt + K_{d_2,z} \frac{d\epsilon_z(t)}{dt} \end{bmatrix} \in \mathbb{R}^2 \tag{41}$$

Where $K_{p_j} = [K_{p_j,x},\, K_{p_j,z}] \in \mathbb{R}^2$, $K_{i_j} = [K_{i_j,x},\, K_{i_j,z}] \in \mathbb{R}^2$ and $K_{d_j} = [K_{d_j,x},\, K_{d_j,z}] \in \mathbb{R}^2$ for $j = 1, 2$ are the control gains for both PID controllers. As will be shown in the simulation section, this double-PID controller has proven to achieve the desired control objective.

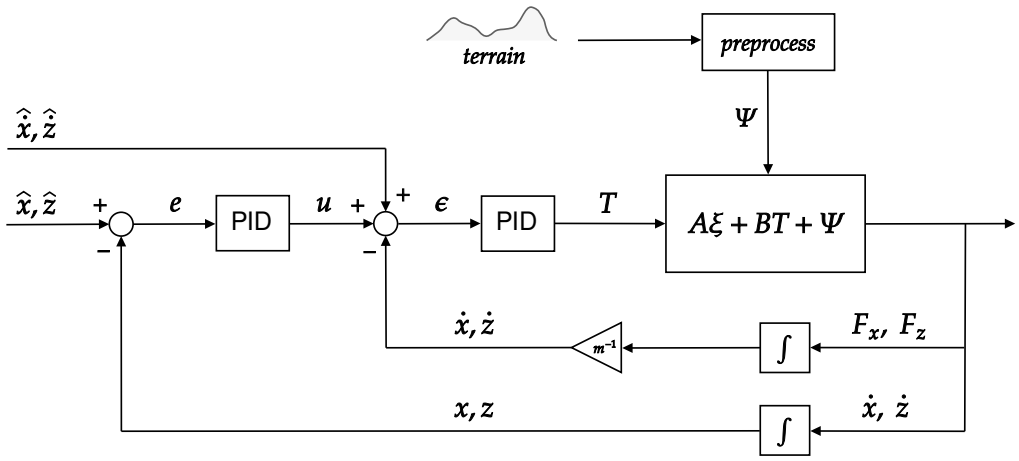

Figure 6: double-PID control scheme for position and velocity tracking

PID controllers is one of the most suitable choices to face this kind of problem. For sake of completeness a PID is a controller which perform a proportional, integral and derivative action on the controlled system.
For systems with a long lag time between the control action and the control response, proportional control alone can be sufficient. Proportional control produces a consistent offset in most systems. The control action gets less and smaller as the process variable approaches the setpoint until it is never quite enough to attain the setpoint. Another possibility is that the system has a too high proportional gain, causing it to oscillate as it overshoots, undershoots, and so on the setpoint.
Integral control removes the offset by integrating (summing) the error, multiplying it by the integral gain, and finally adding it to the control output. The instantaneous error is zero when the system reaches the setpoint, and the integrated value stops changing, but there may be a residual integrated error, resulting in a non-zero output from the integral control. Unlike proportional control, where zero error equals zero output, this residual action allows the controller to eliminate the offset.
Derivative action is especially beneficial for speeding up the responsiveness of slow, low-noise systems. Derivative control works with the error's derivative. The derivative control will act to reduce the output if the error is rapidly reducing (big negative derivative) (by adding in a large negative number). There will be little derivative action if the error is changing slowly or not at all. This allows for bigger proportional and integral gain levels to be used, with the derivative control counteracting those actions if the system responds rapidly enough. In a noisy system, however, noise causes the derivative of the error to change frequently, resulting in poor control.
Since the rover is a system which moves at a very low speed, each control action is carried out a few moments in time from the previous one, thus allowing to keep the error small

and make this kind of double-PID control possible, in fact for this reason, it would not have been possible with other types of faster vehicles.

6 Operating modes and simulations

This section explains how the rover analysis is organized in the simulation environment. The entire simulator was developed on Matlab 2021b as regards the setup of the workspace with the preprocessing of the images, the kinematic analysis, and the return of the results in appropriate plots and simulation videos. For the analysis of the control system and the forces acting on the system, an ad hoc model was built on Simulink.
Even if it is an ideal setup, the implemented logic is easily applicable to real cases, thanks to its versatility and its modularity, this makes the simulator a useful tool for more complex planning methodologies, which require the repeated execution of the simulation more times.
As a result, it's critical to keep the model as computationally simple as possible while still ensuring a high level of accuracy for future applications.
The ability of a rover to traverse a given terrain is determined by the ground profile and surface characteristics. Sensors on rovers, such as stereo cameras and Lidar-based systems, can create maps of the environment around them. Estimates of soil properties are far more difficult to obtain. Because online identification methods are possible, the ground features are taken into account as input to the simulation.
Another input is the action plan, which is part of the control strategy. It consists of a list of tasks that the rover must complete, such as traveling forward along the entire profile at a constant speed and determining whether or not the task is possible and how difficult it can be.
They are rover commands that define the vehicle's next coordinates and desired velocity. These actions consider distances that are tiny enough to be physically significant.
The actions are analyzed, and the new rover position is determined. The inverse kinematics algorithm uses this position to determine the geometry of the wheel/ground contact. Following that, a quasi-static analysis is carried out as described in the previous sections. It determines the behavior of the entire system in response to the control action. Inverse kinematics and force analysis for these kinds of systems may be extremely difficult and time-consuming to compute, thus the goal in this case was to make all of the simulations computationally efficient.

6.1 Rough terrain analysis

The impacts of robot-terrain interaction must be explicitly considered in motion planning algorithms for mobile robots on tough terrain. In order to design a safe and robust motion plan, a robot must be able to identify the mobility dangers posed by steep, loose, and uneven terrain.
Real-world implementation issues must be taken into account while developing rough terrain motion planning algorithms. Algorithms must be efficient enough to run on a system with limited processing resources on board, such as a planetary exploration rover.

Rough terrain planning algorithms must also take into account the inherent uncertainty in terrain sensing systems like rangefinder sensors. Finally, due to path errors, planning algorithms must account for the fact that the mobile robot cannot precisely track a planned path.

6.1.1 Terrain measurement uncertainty

Uncertainty in terrain measurement leads to a misrepresentation of terrain features' true position. This poses a risk to robot safety because inaccurate range sensor data could result in motion plans that are physically impossible to traverse. Misrepresentation of the position (as opposed to elevation) of uneven terrain regions is of particular practical importance, as it may result in motion plans intersecting with barriers (such as huge stones) or untraversable sections (such as steep slopes). As a result, the consequences of terrain measurement uncertainty must be considered in the algorithm. To do so, a two-dimensional Gaussian filter is used to pre-filter the terrain range map.
The filter is of the form:

$$z'(x,y) = \frac{e^{-\frac{x_s^2+y_s^2}{2\sigma_Z^2}}}{\sigma_Z\sqrt{2\pi}} \tag{42}$$

where the standard deviation σ_Z of the filter is defined by the terrain measurement uncertainty given by miscalibration and sensor noise, and x_s and y_s refer to the distance from the sensor to the terrain point. As shown in the following Figure (7), the filter has the effect of "blurring" terrain data.

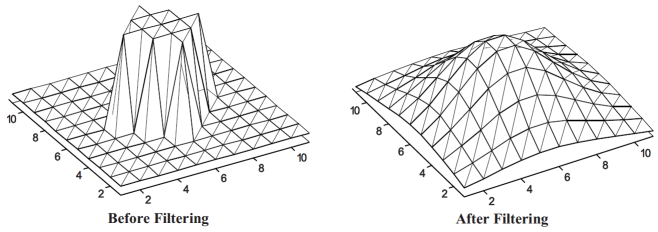

Figure 7: Example of terrain smoothing before and after Gaussian filter

Uncertainty in terrain measurement has an impact on the evaluation since it renders the true robot configuration at a given terrain point ambiguous. Static stability, kinematic validity, and terrain traversability are all affected by this. Because of the significant cost of robot failure in real-world tasks, a conservative approach to terrain measurement uncertainty should be used. Potential worst-case configurations generally exist at possible terrain point location boundaries. Here we consider variation in elevation only.
The elevation boundaries of a given terrain point z_n can be computed as

$$z_n^+ = z_n + \sigma_Z \tag{43}$$
$$z_n^- = z_n - \sigma_Z \tag{44}$$

Even though advanced tools for dealing with uncertainty exist, since the present work is based on a planar case of longitudinal motion and the soil profile is only simulated and generated by filtering a satellite image, the result obtained from the "smoothing" of the Gaussian filter will be considered as optimal and reliable, as better specified in the following section in the image processing algorithm.

6.1.2 Image processing for the recognition of the terrain profile

The main purpose of the image processing algorithm is to derive the profile of the terrain on which the robot will have to advance directly from a satellite image of Mars. The images analyzed with the related topographical parameters are taken directly from the HiRISE database of the University of Arizona.
The HiRISE (High-Resolution Imaging Science Experiment) is a camera on board the Mars Reconnaissance Orbiter that has been orbiting and analyzing Mars since 2006. It has a 0.5 m aperture reflecting telescope, the biggest of any deep space mission, and can take images of Mars with levels of 0.3 m/pixel.
HiRISE has captured images of the Opportunity rover, the Curiosity rover, and the current Perseverance expedition on the surface of Mars.
The entire simulation process starts from the choice of the path (straight for simplicity) that the robot will have to analyze and take. One of the satellite images of the aforementioned dataset is shown to the user and he can freely trace a segment from any pixel coordinates to an endpoint (Figure (8)).

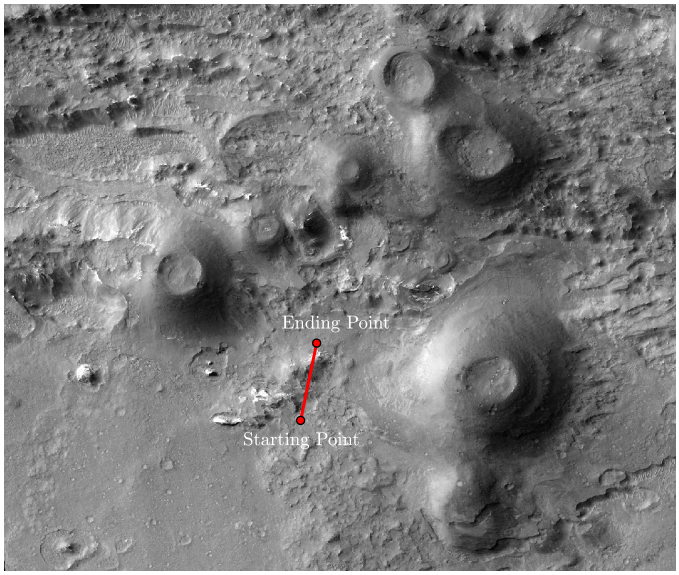

Figure 8: Straight path to follow traced on the image taken by HiRISE

From the choice of the path to be carried out, the actual image processing begins. A Gaussian filter is applied to the entire image for the first smoothing step.
The result of the smoothing is thus processed in order to convert the RGB image into grayscale and scale it according to the real values of the photographed surface. In particular, for scaling it is necessary to measure the height of the highest point photographed, the total extension of the photographed surface, and therefore the number of meters per pixel shown. In this way, the value of each pixel in height is obtained, assuming that the brighter the pixel, the higher the surface height will be.
No preprocessing of the image to face the problem of the shadows created by the position of the Sun with respect to the surface photographed by the satellite has been done, in any case there are methods in the literature capable of eliminating the shadows effectively. It was therefore assumed that the data obtained from the filtering of the original image were reliable despite the shadows.
The results obtained are thus used to create a mesh plot with a contour plot underneath (level surfaces). A mesh plot is a three-dimensional surface that uses the values in the grayscale matrix as heights above a grid in the x-y plane defined by X and Y coordinates. The edge colors vary according to the heights specified by the grayscale Z coordinates (Figure (9)).
At the end the algorithm returns the sampled pixel values along the line segment traced by the user on the image. In this way the terrain profile is obtained and ready to be used

as input for the simulation of the system (Figure (10)).

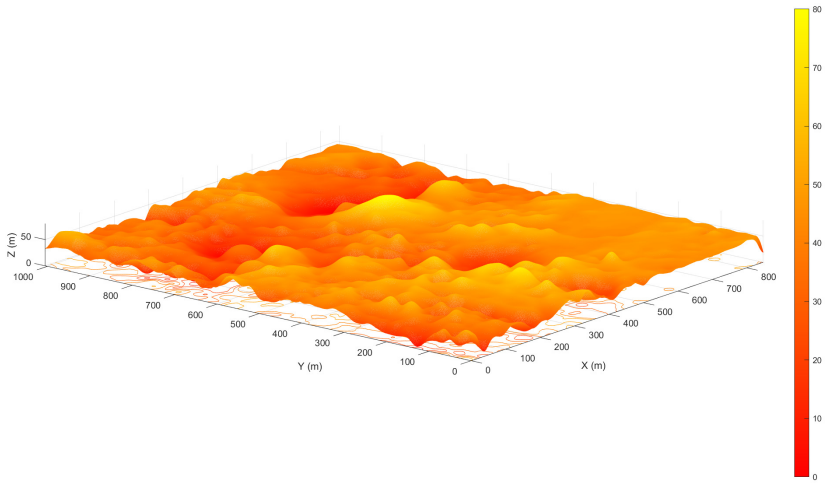

Figure 9: Final topography obtained from filtering

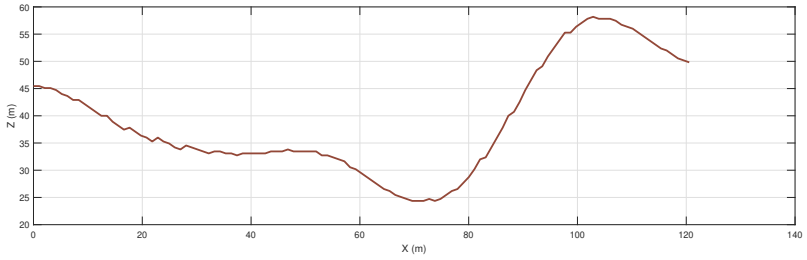

Figure 10: Resulting terrain profile

6.2 Simulation results

Once the terrain profile is obtained from the image processing algorithm, therefore also the α_i angle, everything can be computed efficiently thanks to the simulator whose logic was largely explained in the previous sections. A simple task for showing the simulator results is then performed (Figure (11)).

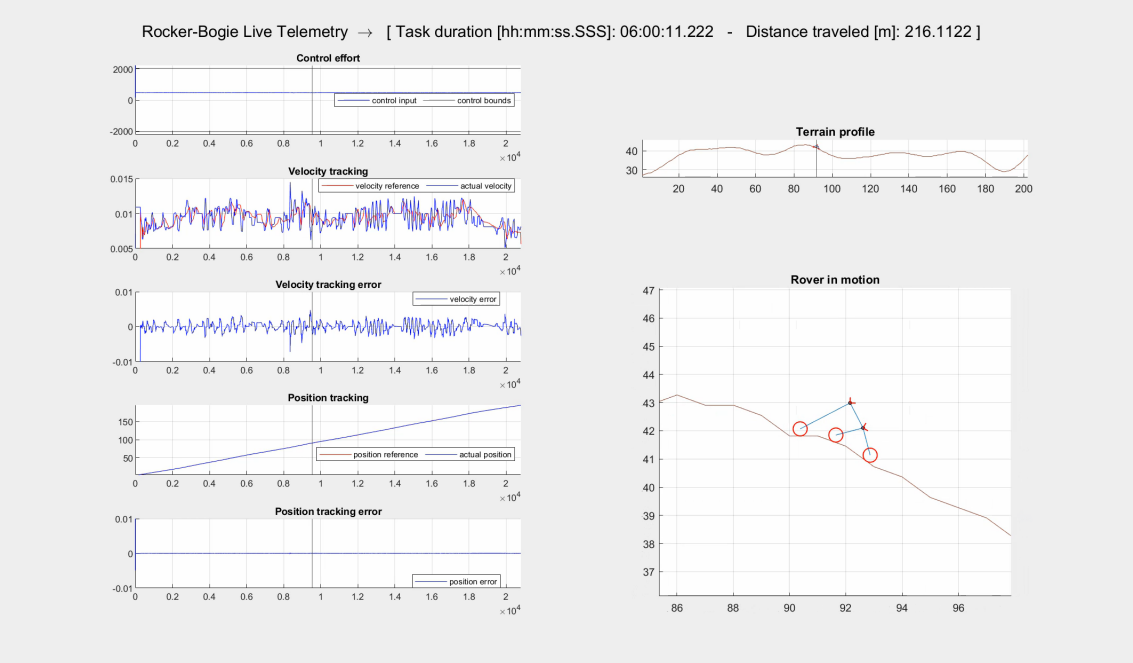

Figure 11: Rocker-bogie live telemetry

It is possible to see from the graphs on the left that the task has been completed successfully.
Analyzing more in detail the system behavior, the trajectory on the x and z axes are perfectly tracked thanks to the adopted control strategy (Figures (12)-(15)).

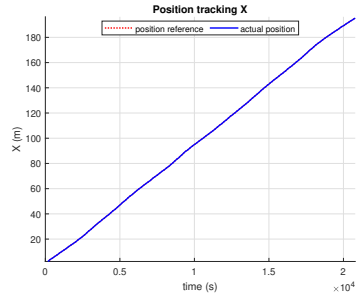

Figure 12: x-axis tracked position

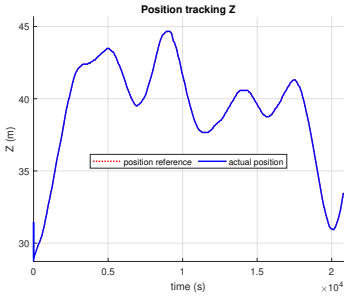

Figure 13: z-axis tracked position

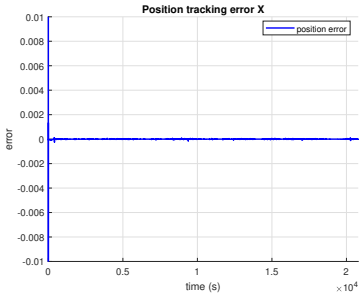

Figure 14: x-axis tracked position error

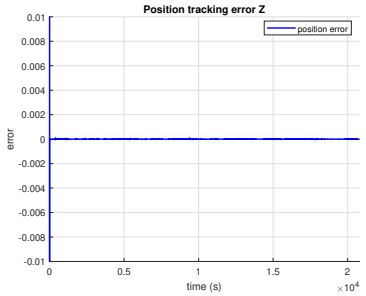

Figure 15: z-axis tracked position error

The same holds for the velocity tracking, in fact, apart from an infinitesimal error due to the small instantaneous variations of the reference, also in this case the controller is able to track the reference well, making sure to complete the requested task in the best possible way (Figures (16)-(19)).

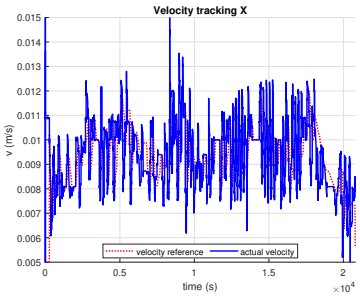

Figure 16: x-axis tracked velocity

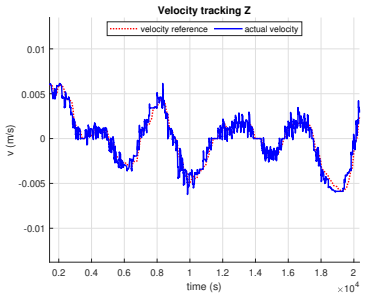

Figure 17: z-axis tracked velocity

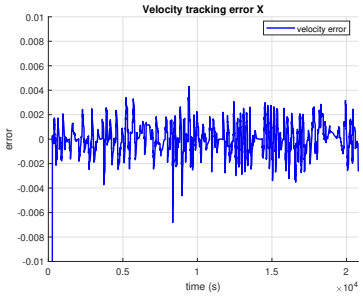

Figure 18: x-axis tracked velocity error

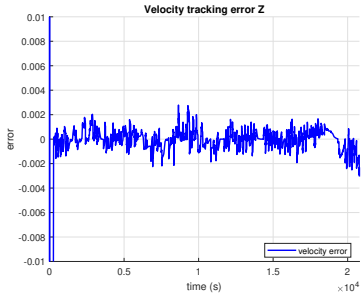

Figure 19: z-axis tracked velocity error

It is possible to verify also that, from the Figure (20) showing the total control effort in terms of torque, the torque's constraint (eq. (22)) is respected, therefore, under the assumptions made for completing the task, the goal is reached successfully. And the step response in Figure (21) shows the efficiency of the control strategy adopted.

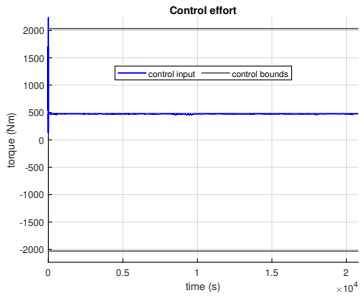

Figure 20: Total torque with constraints

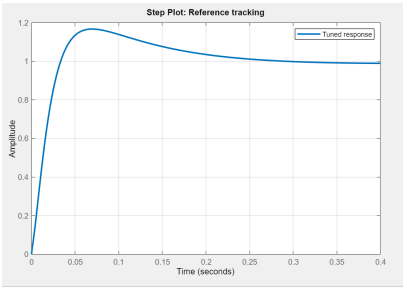

Figure 21: Step response

6.3 Slip estimator and Pacejka's magic formula

The assumption of a pure rotational motion (absence of slip) for each wheel is removed here.
In order to reduce wheel slip, the normal force interaction must be carefully regulated. A slipping wheel penetrates into soft soil quickly, which might cause rover positioning errors if they are based on wheel rotations.
To improve wheel-soil interaction, it is therefore important to build a planar and hence simple simulation of the system.

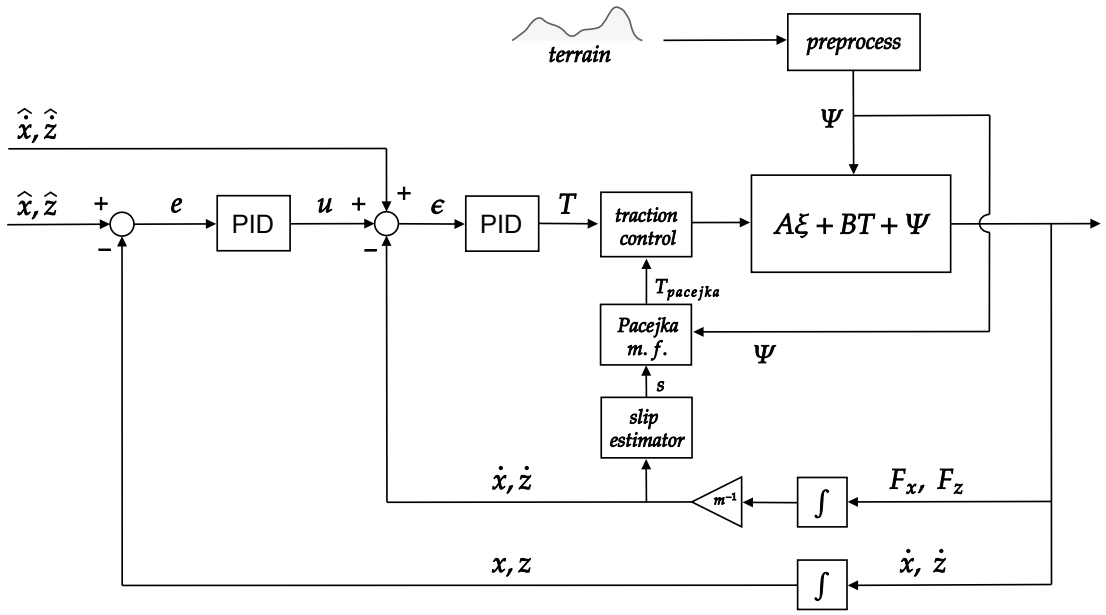

Figure 22: double-PID control scheme with slip estimator and traction control

For what concerns the slip ratio, when it is high, the tire forces are highly nonlinear. When moving the vehicle in the tire nonlinear zone, it is critical to have a realistic nonlinear tire force model for the vehicle dynamics.
So, given the longitudinal driving speed and the equivalent circumferential velocity of the wheel we can define the longitudinal slip ratio as:

$$\sigma_i = \frac{r\omega_i - v_{x,i}}{v_{x,i}}, \quad r\omega_i < v_{x,i} \quad Braking \tag{45}$$

$$\sigma_i = \frac{r\omega_i - v_{x,i}}{r\omega_i}, \quad r\omega_i > v_{x,i} \quad Accelerating \tag{46}$$

The angular velocity of the wheels is ω_i. A slip ratio of zero indicates that the forward velocity and tire rolling speed are equal, implying that neither accelerating or braking torque exists. The tire has a positive finite rolling velocity and the vehicle has a greater finite forward velocity when the slip ratio is positive. A negative slip ratio indicates that the vehicle's forward velocity is finite and the tire's equivalent positive rolling velocity is greater. The wheel is either locked at zero speed or rotating with the car at zero speed at each extreme, i.e. $\sigma_i = +1$ and $\sigma_i = -1$.
It is necessary therefore to build a tire model in order to analyze the performance on the ground. Empirical models are typically highly parameterized, making them difficult to apply in areas where no measurements are available. These models, on the other hand,

are frequently highly compact, with only a few analytical equations, and computationally fast, which can be a significant benefit for real-time simulation.
In order to develop a relationship for the slip ratio and the forces and torques generated by the soil onto the wheel, a soil-vehicle mathematical model must be utilized. A simple model that has been used for testing is the Pacejka's magic tire formula.
The Pacejka tire model is a complex nonlinear semi-empirical model that can represent the nonlinear and coupled behavior of tire forces over a wide variety of operations, it is a trigonometric function (made up of sine and arctangent) with generally four terrain specific coefficients.
The tire forces are described as functions of the tire normal force, slip ratio, slip angle, and surface friction coefficient in Pacejka's model.
It is crucial to have a realistic tire model that accounts for the slip phenomenon; otherwise, the robot would not be able to move efficiently. Calculating the longitudinal forces operating as a function of slip ratio is useful in the current planar example.
As shown below, the longitudinal forces in this model are supposed to be dependent on the normal force, surface friction, and longitudinal slip ratio:

$$T_i = T(\sigma_i, \mu, N_i) \tag{47}$$

The longitudinal force is calculated using the Pacejka tire model based on the percentage of longitudinal slip while the friction coefficient is calculated using the Pacejka's model as a function of longitudinal slip ratio:

$$\mu(\sigma_i) = Dsin(Ctan^{-1}(B\sigma_i - E(B\sigma_i - tan^{-1}(B\sigma_i)))) \tag{48}$$

The values of B, C, D, and E for various ground types are generated from constants determined through terramechanics measurements, and the overall model coefficients B, C, D, and E are then calculated from a combination of these constants. The longitudinal traction force T_i is computed using the friction coefficient $\mu(\sigma_i)$ and the normal force N_i acting on the wheel:

$$T_i = \mu(\sigma_i)N_i \tag{49}$$

Considering the same setup of the simulation of the previous section, a slip estimator has been added in order to compute the slip ratio of the aforementioned task and check whether a wheel is slipping or not. The results are shown in Figure (26):

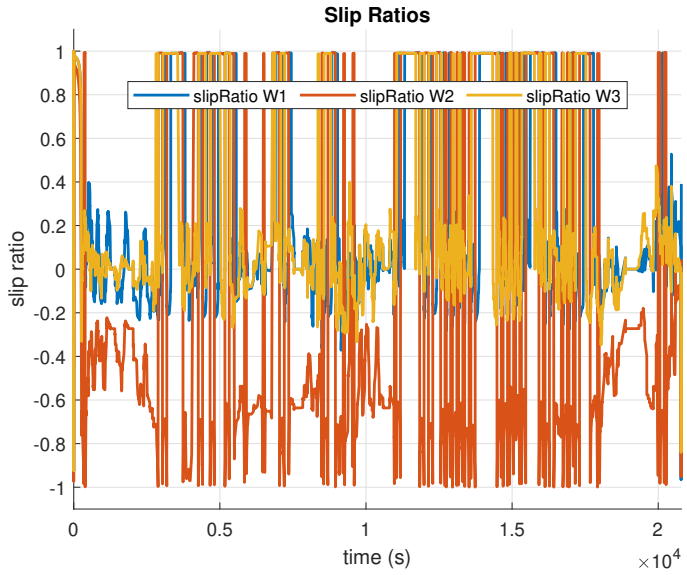

Figure 23: Slip ratio of the 3 wheels

It is evident that if all the wheels are slipping, the robot will be stuck and cannot complete the required task. Therefore the slip estimator is a sort of checking point to check whether a task is feasible or not.
The next step is to compute the traction forces necessary to avoid each wheel from slipping. Therefore the new traction forces are computed through the Pacejka's magic formula and ready to be used as new inputs to the system through an ad hoc new control law.

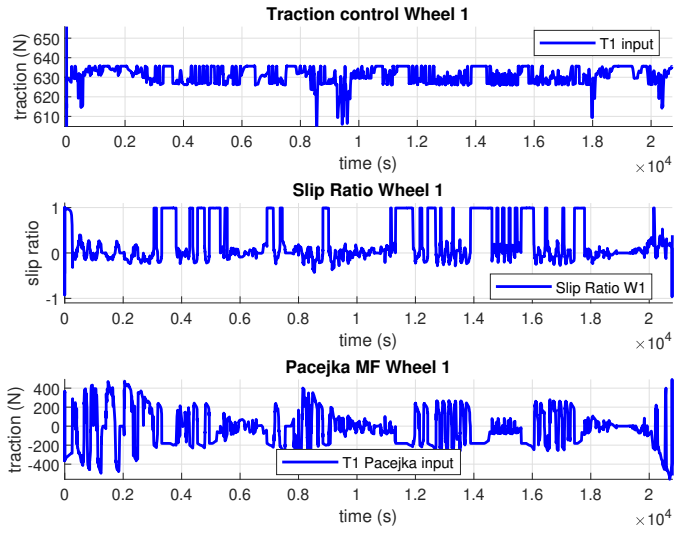

Figure 24: Original traction force, slip ratio and new traction force from Pacejka's magic formula of wheel 1

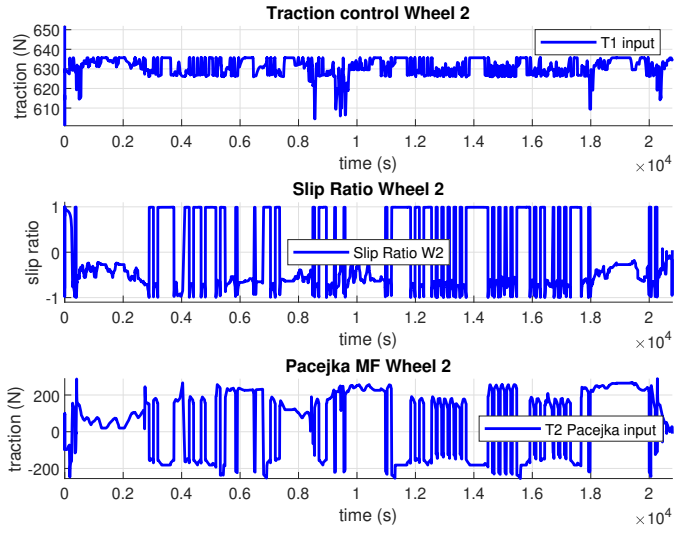

Figure 25: Original traction force, slip ratio and new traction force from Pacejka's magic formula of wheel 2

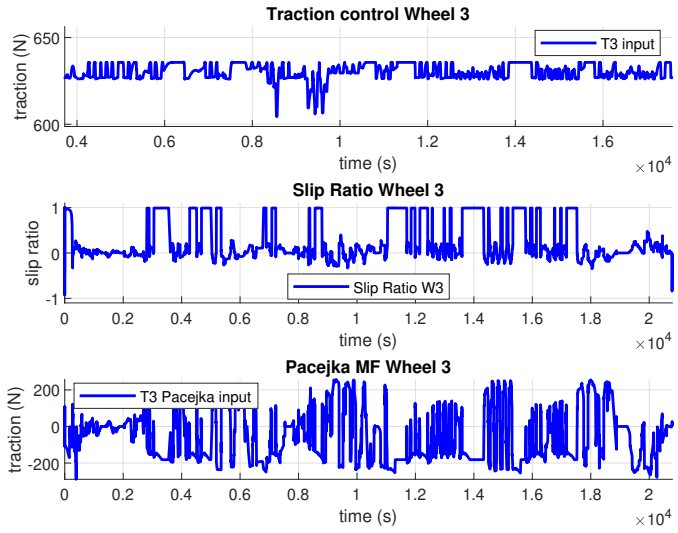

Figure 26: Original traction force, slip ratio and new traction force from Pacejka's magic formula of wheel 3

7 Conclusion

The study and simulation of a rocker-bogie rover were reported in this paper. These rovers are mostly employed for planetary exploration, therefore they must be able to traverse long distances and be partially autonomous. As a result, it's critical to create a precise model for such systems.

The bidimensional case of a rover travelling in a longitudinal motion is considered in this study. The inverse kinematics of the system is solved using a procedure that is developed and presented. This method simplifies the problem and produces precise results, making it ideal for methodology planning.

The rover's quasi-static force analysis is then shown. The ground-tire interaction is crucial to this research since it offers information on the longitudinal forces. Two techniques based on the assumption of slip presence have been successfully developed and shown using a control strategy for the rover's position and velocity tracking.

References

[1] Hervé Hacot
Analysis and Traction Control of a Rocker-Bogie Planetary Rover
June, 1998

[2] Bruno Siciliano, Oussama Khatib, Frans Groen - Springer
Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers
2004

[3] Kazuya Yoshida, Hiroshi Hamano, Toshinobu Watanabe
Slip-Based Traction Control of a Planetary Rover

[4] Kanfeng Gua, Guoyong Yang
A Judgment Criterion of Crossing Obstacle Ability for Lunar Rover with Track Angle
January, 2016

[5] Ranjan Vepa
Dynamics and Control of Autonomous Space Vehicles and Robotics
2019

[6] Erin J. E. Austen
Development of Kinematic and Dynamic Models for the Argo J5 Rover
2019

[7] Lin-hui Li, Jing Lian, Bai-chao Chen, Jing Chang, Hai-yang Huang
Trajectory tracking and traction coordinating controller design for lunar rover based on dynamics and kinematics analysis
July, 2014

[8] Viboon Sangveraphunsiri, Mongkol Thianwiboon
Traction Control for a Rocker-Bogie Robot with Wheel-Ground Contact Angle Estimation
October, 2005

[9] Pushpendra Kumar, Pushparaj Mani Pathak
Dynamic Modeling, Simulation and Velocity Control of Rocker-Bogie Rover for Space Exploration
February, 2015

[10] Moshe P. Mann, Zvi Shiller
Dynamic Stability of a Rocker Bogie Vehicle: Longitudinal Motion
April, 2005

[11] Giancarlo Genta - Microcosm Press and Springer
Introduction to the mechanics of space robots
2012