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

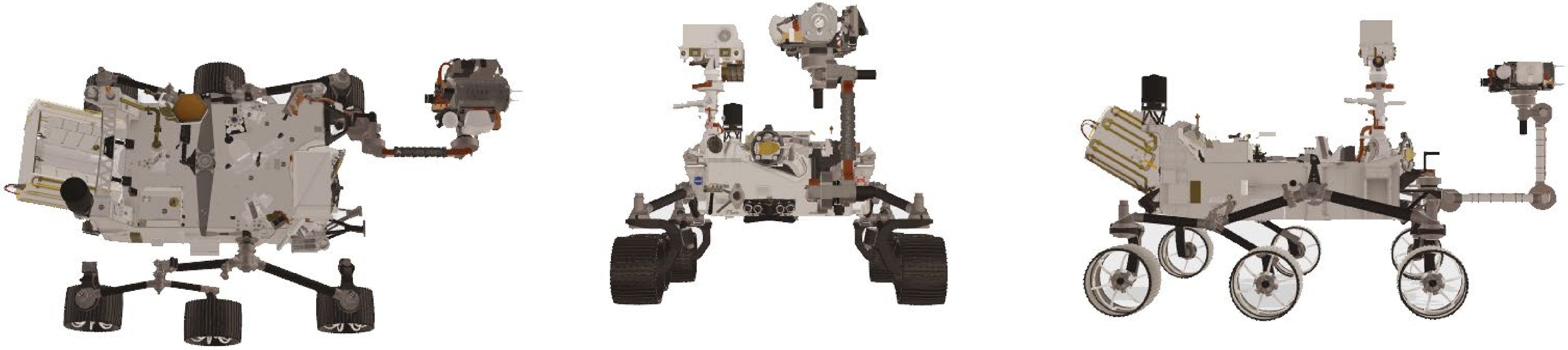
SAPIENZA
UNIVERSITÀ DI ROMA

Facci Matteo (1597454)
Sensolini Arrà Giuseppe (1661198)

VEHICLE SYSTEM DYNAMICS - A.Y. 2020-2021

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.



- 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.

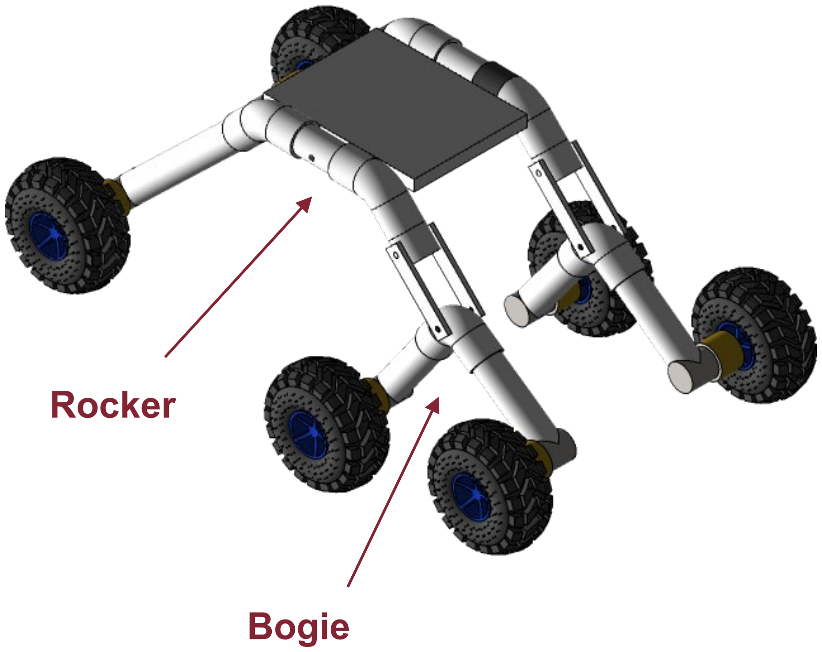
Rocker-Bogie suspension system

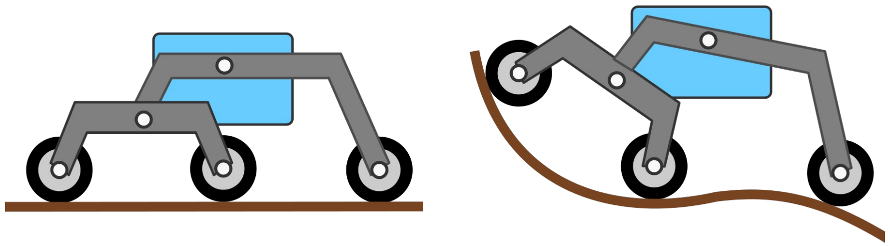
Rocker

- Rockers are larger, body-mounted linkages on each side of the rover. Rockers are connected to each other and the vehicle chassis through a differential.
- 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.

Bogie

- Bogies refers to the smaller linkages that pivot to the rockers in the middle having drive wheels 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.





Kinematics

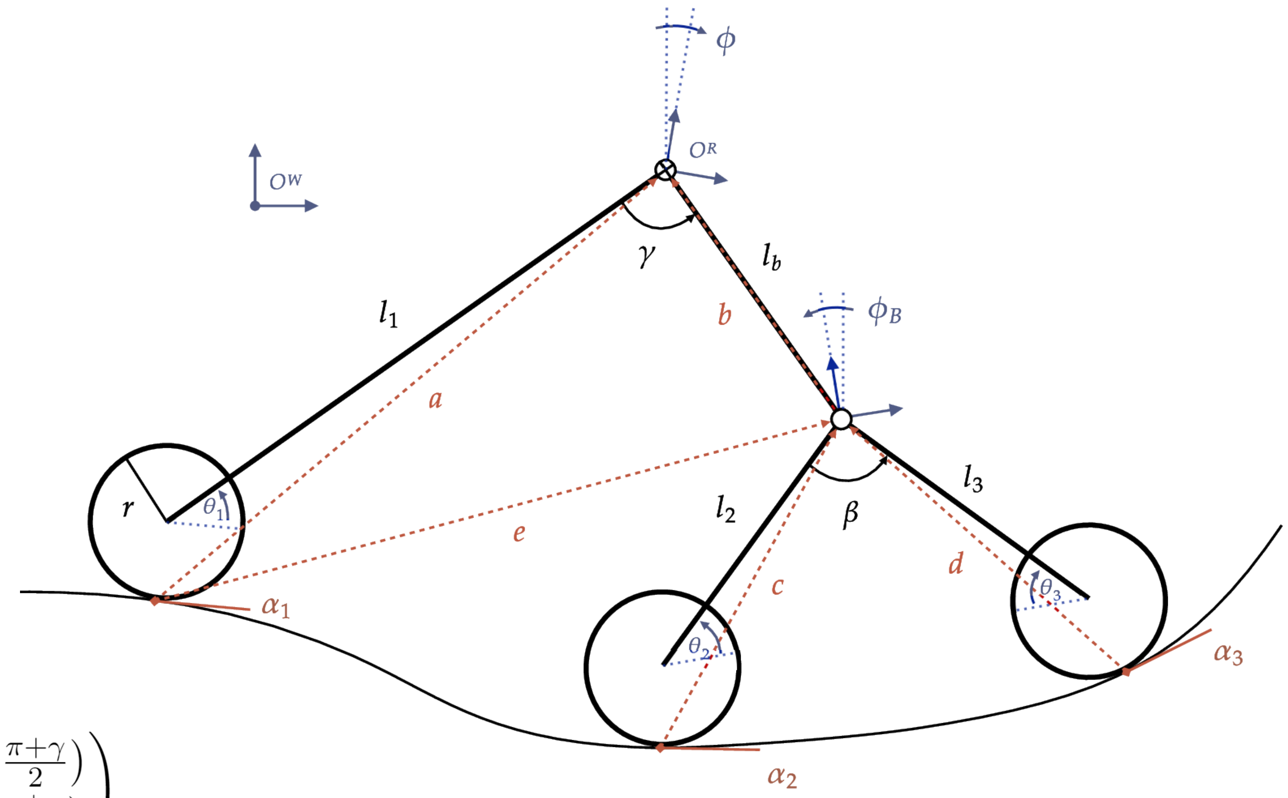
$$a = R(\phi)\begin{pmatrix} l_1\cos\left(\frac{\pi-\gamma}{2}\right) - r\sin(\alpha_1) \\ l_1\sin\left(\frac{\pi-\gamma}{2}\right) + r\cos(\alpha_1) \end{pmatrix}$$

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

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

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

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

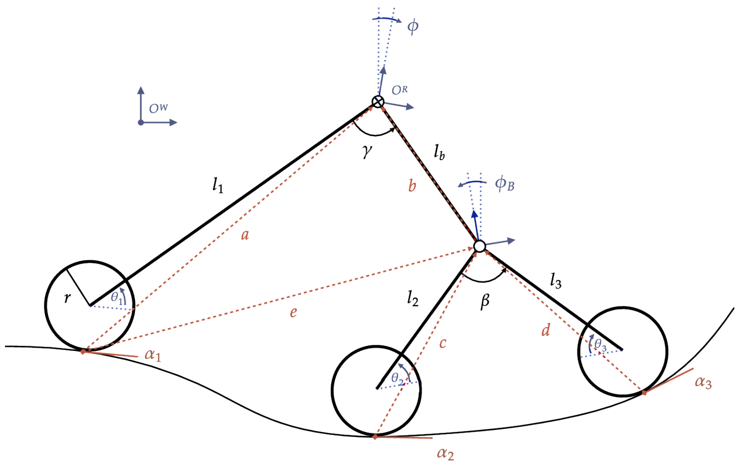


Inverse kinematics procedure

Assumption: all wheels always in contact

Given:
- A terrain profile *z = g(x)*
- An initial contact point for wheel 2

Compute:
- The position of all the relevant kinematic points



Solve this system:

$$
\begin{aligned}
(x_2 - x_3)^2 + (z_2 - z_3)^2 &= l_{23} \\
(x_2 - x_b)^2 + (z_2 - z_b)^2 &= l_2 \\
(x_3 - x_b)^2 + (z_3 - z_b)^2 &= l_3 \\
(x_b - x_r)^2 + (z_b - z_r)^2 &= l_b \\
(x_1 - x_r)^2 + (z_1 - z_r)^2 &= l_1 \\
(x_1 - x_b)^2 + (z_1 - z_b)^2 &= l_{1b}
\end{aligned}
$$

Force analysis

Full Dynamic Model

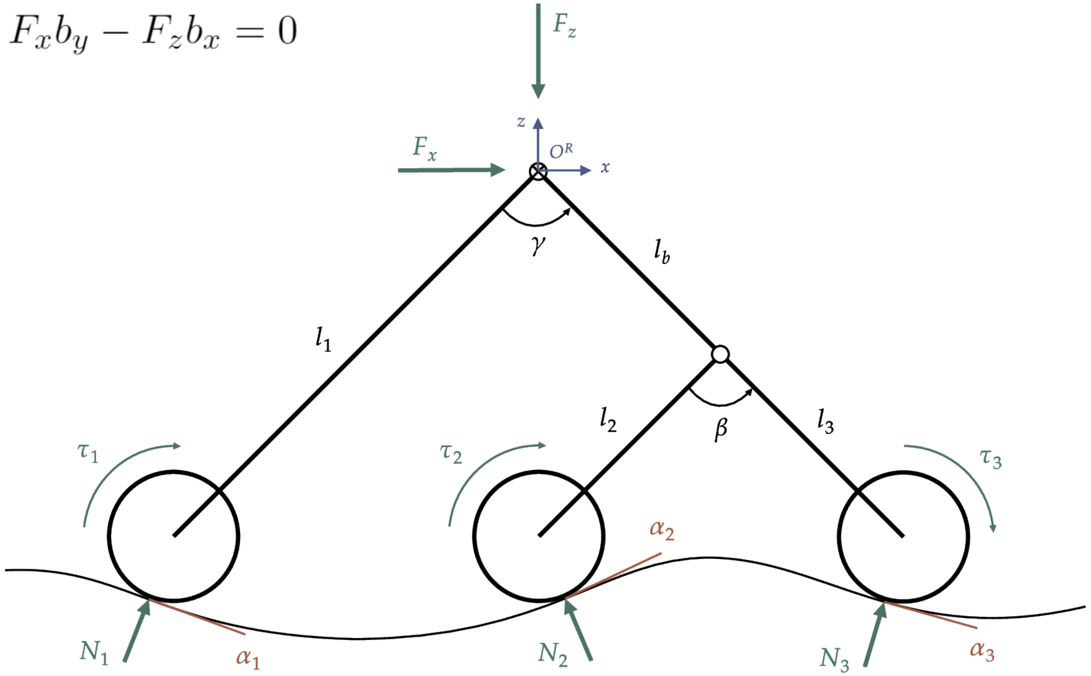
$$T_1 \cos\alpha_1 + T_2 \cos\alpha_2 + T_3 \cos\alpha_3 - N_1 \sin\alpha_1 - N_2 \sin\alpha_2 - N_3 \sin\alpha_3 + F_x = 0$$

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

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

$$T_2 \cos\alpha_2 c_y - T_2 \sin\alpha_2 c_x - N_2 \sin\alpha_2 c_y - N_2 \cos\alpha_2 c_x +$$
$$+ T_3 \cos\alpha_3 d_y - T_3 \sin\alpha_3 d_x - N_3 \sin\alpha_3 d_y - N_3 \cos\alpha_3 d_x = 0$$

- Traction forces (assuming no slip) $\quad T_i = \frac{\tau_i}{r}\ , \quad i = 1,2,3$
- Normal Forces $\quad N_i \geq 0\ , \quad \forall i$
- Torque limit $\quad |rT_i| \leq \tau_{sat}\ , \quad \forall i$



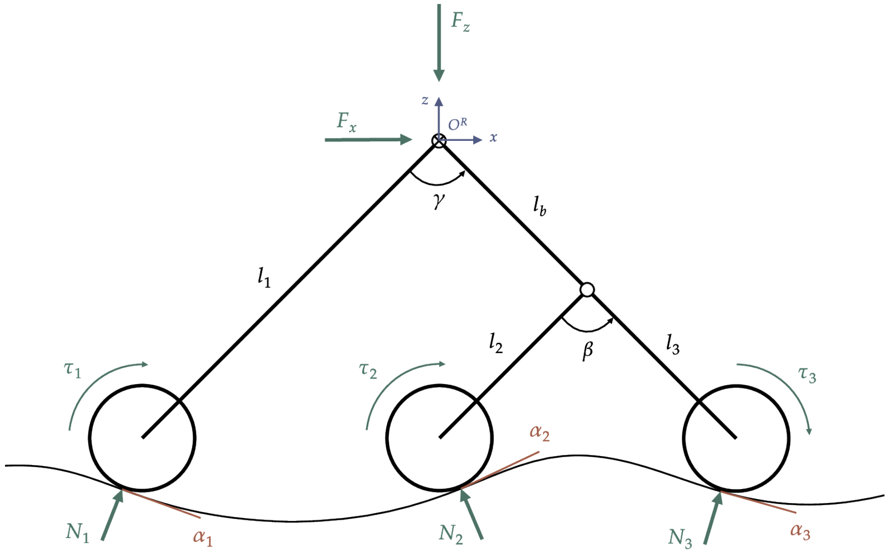
Force analysis

Partial Dynamic Model

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

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

Quasi-static approximation



Normal Forces

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

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

State-space representation

Torque-driven model

- Define the state

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

- And the motor torques as control

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

- And the disturbances:

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

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

State-space representation

Torque-driven model

$$\dot{\xi} = A\,\xi + G(\xi)\,u + \Psi(\cdot)$$
$$y = C\,\xi$$

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

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

State-space representation

Traction-driven model

$$\dot{\xi} = A\,\xi + B\,T + \Psi(\cdot)$$
$$y = C\,\xi$$

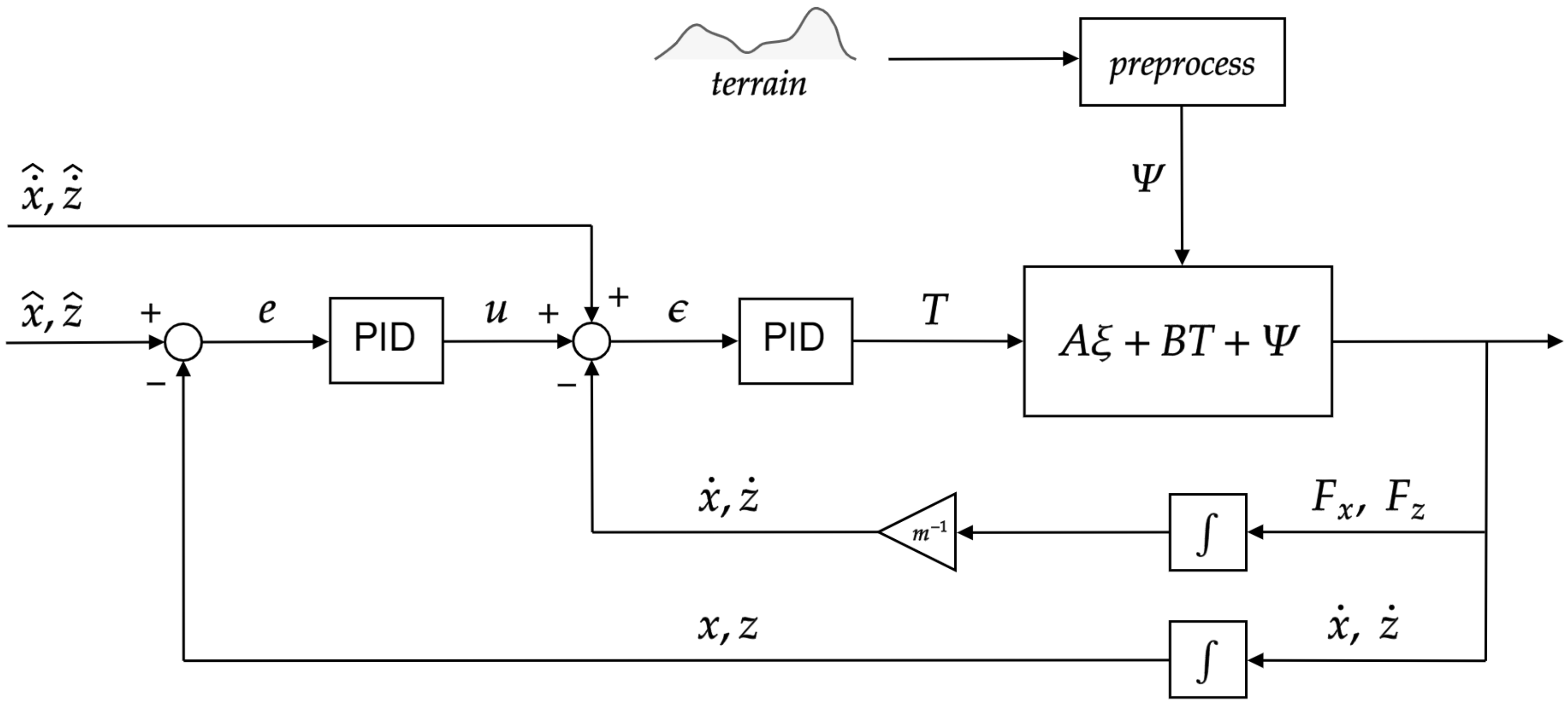
$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1/m & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/m \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 &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

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

Control System

Double-PID Control Scheme

- Aims to track position and velocity



Control System

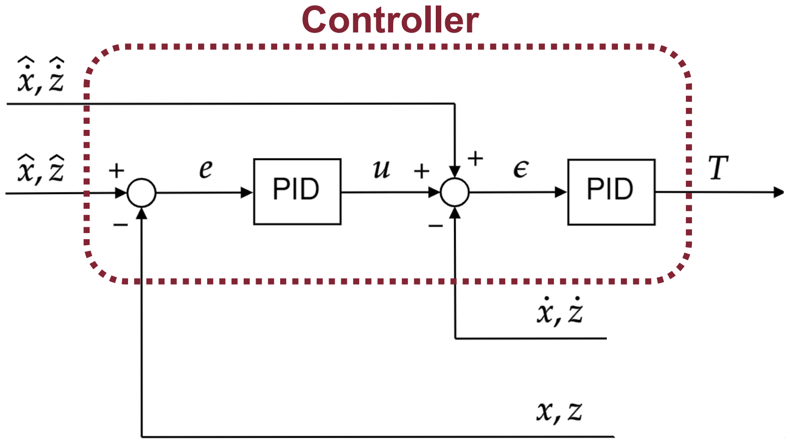
Control law

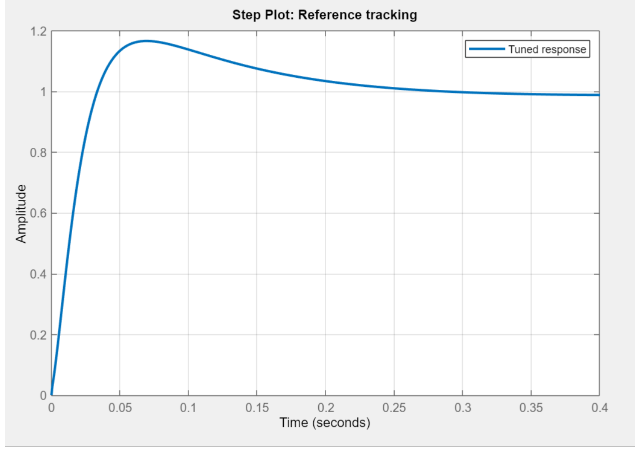
$$e(t) = \begin{bmatrix} \hat{x}(t) - x(t) \\ \hat{z}(t) - z(t) \end{bmatrix} \in \mathbb{R}^2$$

$$u(t) = \begin{bmatrix} u_x(t) \\ u_z(t) \end{bmatrix} = \begin{bmatrix} K_{p1,x}\, e_x(t) + K_{i_1,x} \int e_x(t)dt + K_{d_1,x} \frac{de_x(t)}{dt} \\ K_{p1,z}\, e_z(t) + K_{i_1,z} \int e_z(t)dt + K_{d_1,z} \frac{de_z(t)}{dt} \end{bmatrix} \in \mathbb{R}^2$$

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

$$T(t) = \begin{bmatrix} T_x(t) \\ T_z(t) \end{bmatrix} = \begin{bmatrix} K_{p2,x}\, \epsilon_x(t) + K_{i_2,x} \int \epsilon_x(t)dt + K_{d_2,x} \frac{d\epsilon_x(t)}{dt} \\ K_{p2,z}\, \epsilon_z(t) + K_{i_2,z} \int \epsilon_z(t)dt + K_{d_2,z} \frac{d\epsilon_z(t)}{dt} \end{bmatrix} \in \mathbb{R}^2$$





Operating modes and simulations

MATLAB + SIMULINK®

- Matlab
  - 1 main program to set up the environment and start the simulation
  - ~30 additional scripts and functions
    - Image processing functions
    - Inverse kinematics functions
    - Other utilities for fast computations and plots
- Simulink
  - 1 model

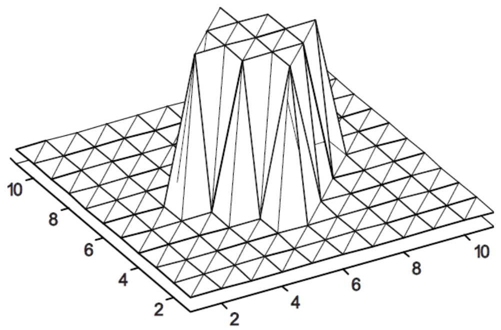
- Surface characteristics
- Action plan and commands
  - Desired position and velocity

→

- Force analysis
- Control action

Rough terrain analysis

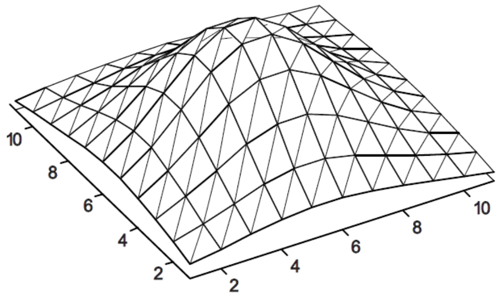
- In order to design a safe and robust motion plan, a robot must be able to identify the mobility dangers posed by steep, loose, and uneven terrain.
- Rough terrain planning algorithms must also take into account the inherent uncertainty in terrain sensing systems like rangefinder sensors.
- A two-dimensional Gaussian filter is used to pre-filter the terrain range map.



Before Filtering

Gaussian filter

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



After Filtering

Elevation boundaries

$$z_n^+ = z_n + \sigma_Z$$
$$z_n^- = z_n - \sigma_Z$$

Image processing for the recognition of the terrain profile

- Goal: derive the profile of the terrain on which the robot will have to advance directly from a satellite image of Mars.

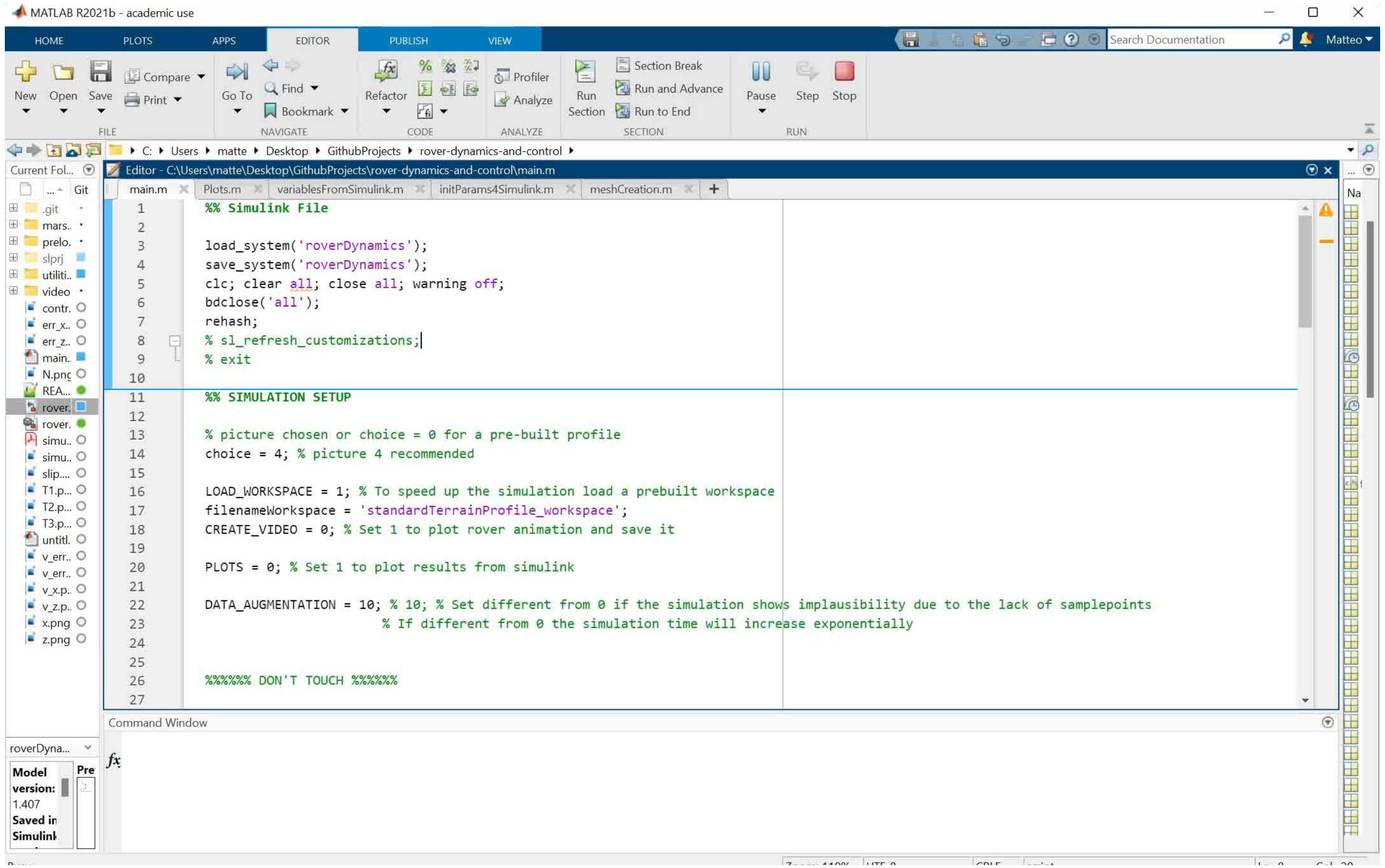
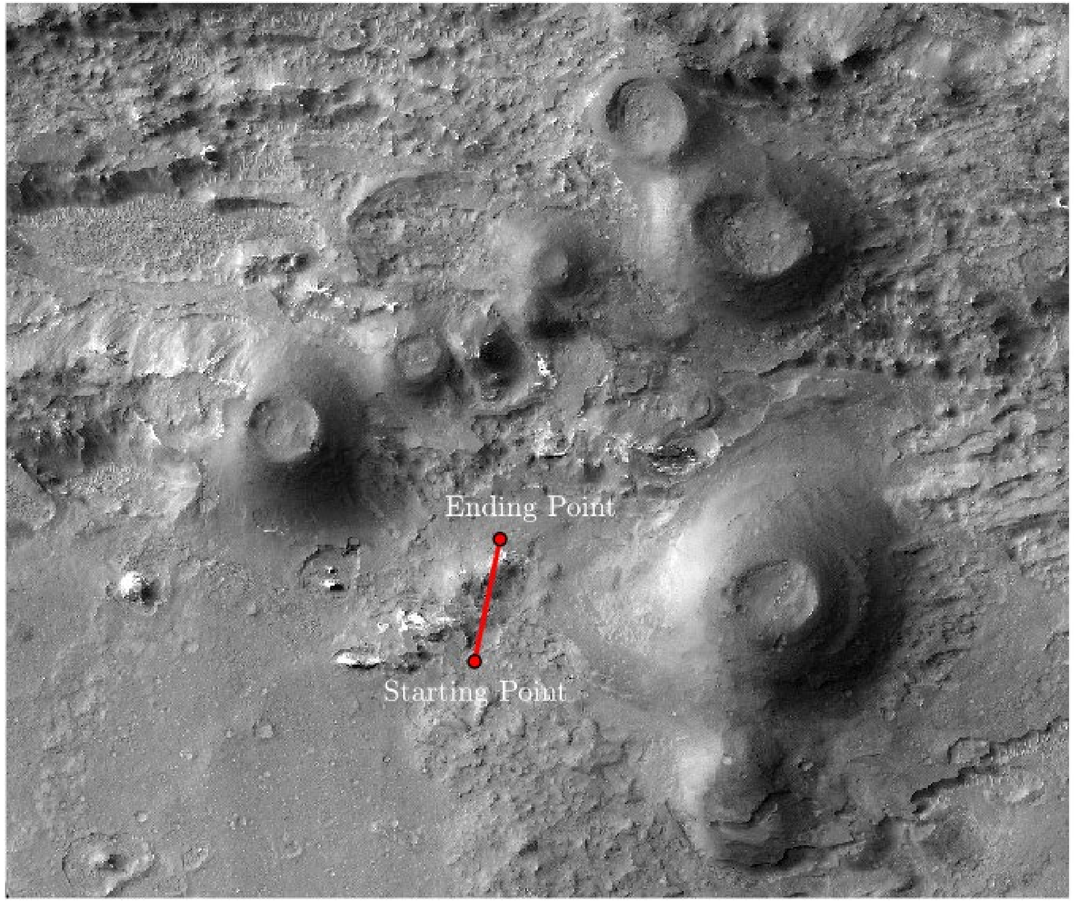
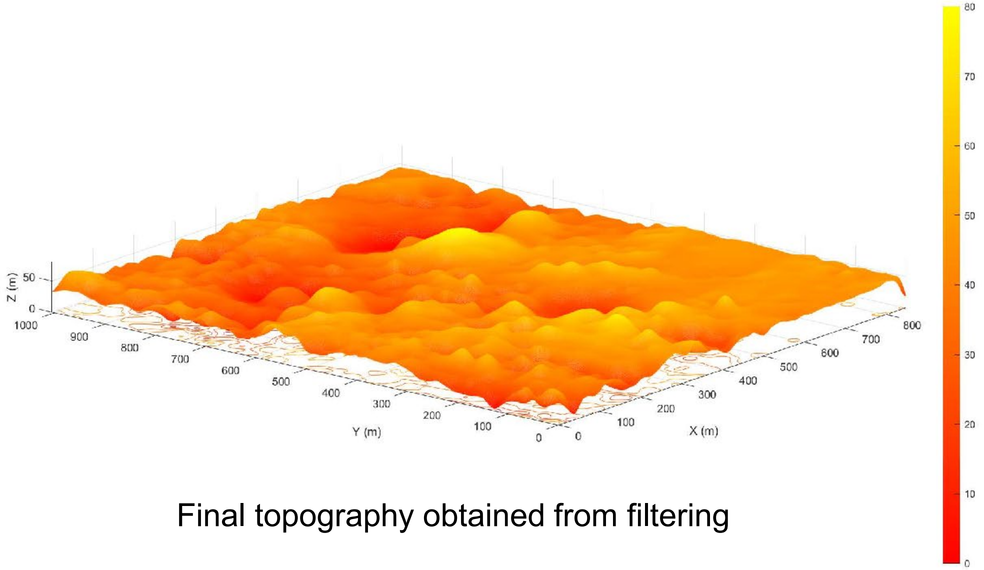


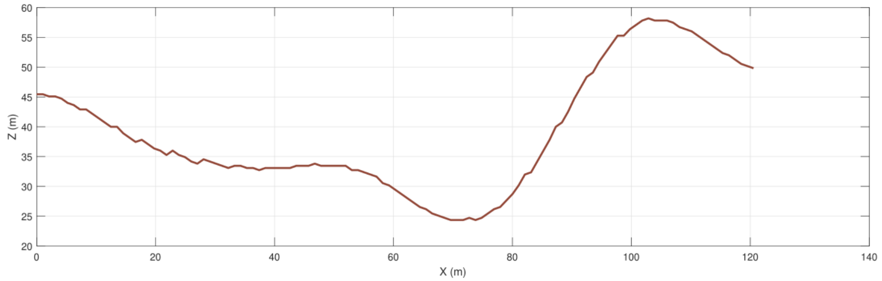
Image processing for the recognition of the terrain profile



Satellite image from HiRISE database of the University of Arizona

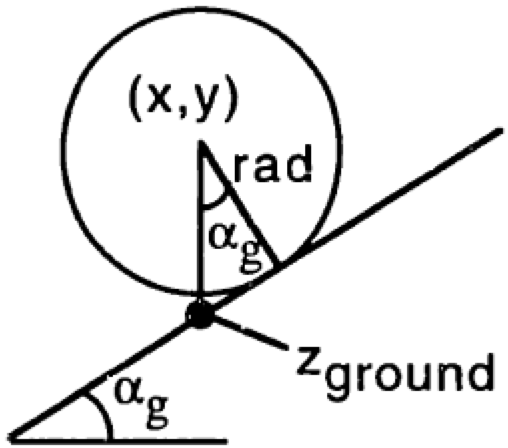


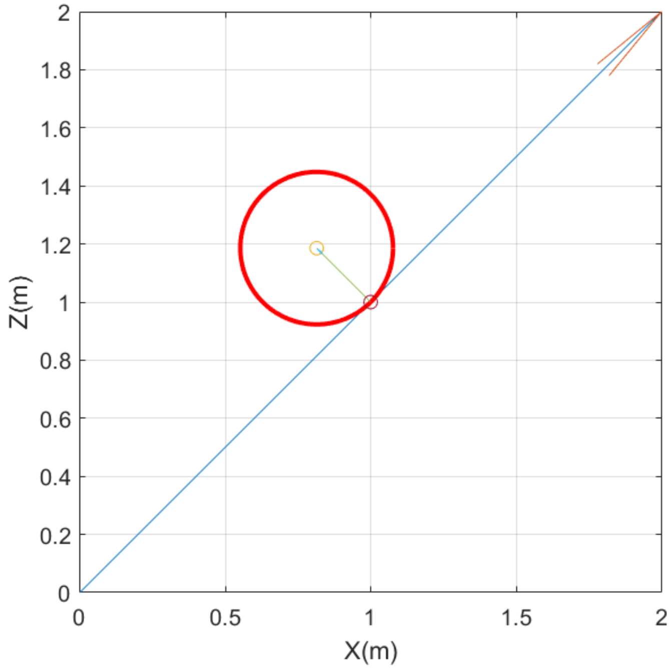
Final topography obtained from filtering



Resulting terrain profile

Wheel-Ground contact





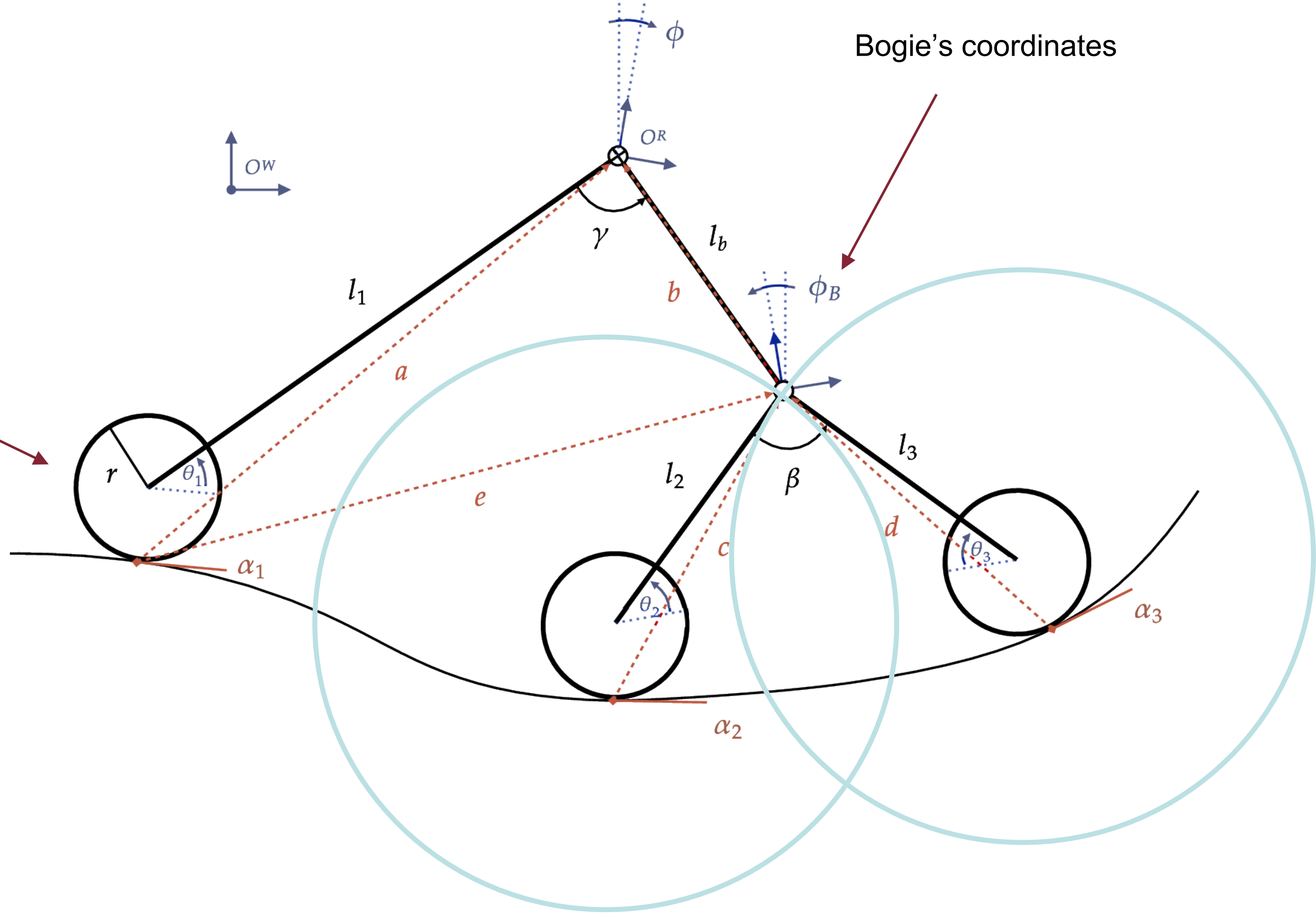
- Given an angle α and the contact point of the wheel, from trigonometric properties, α is equal to the contact angle.
- The contact angle for each wheel is zero while the rover is on a flat ground.
- If the front wheel climbs over a rock while the other wheels stay on flat ground, the contact angle changes from positive to negative as the wheel climbs and descends.

Rover kinematics

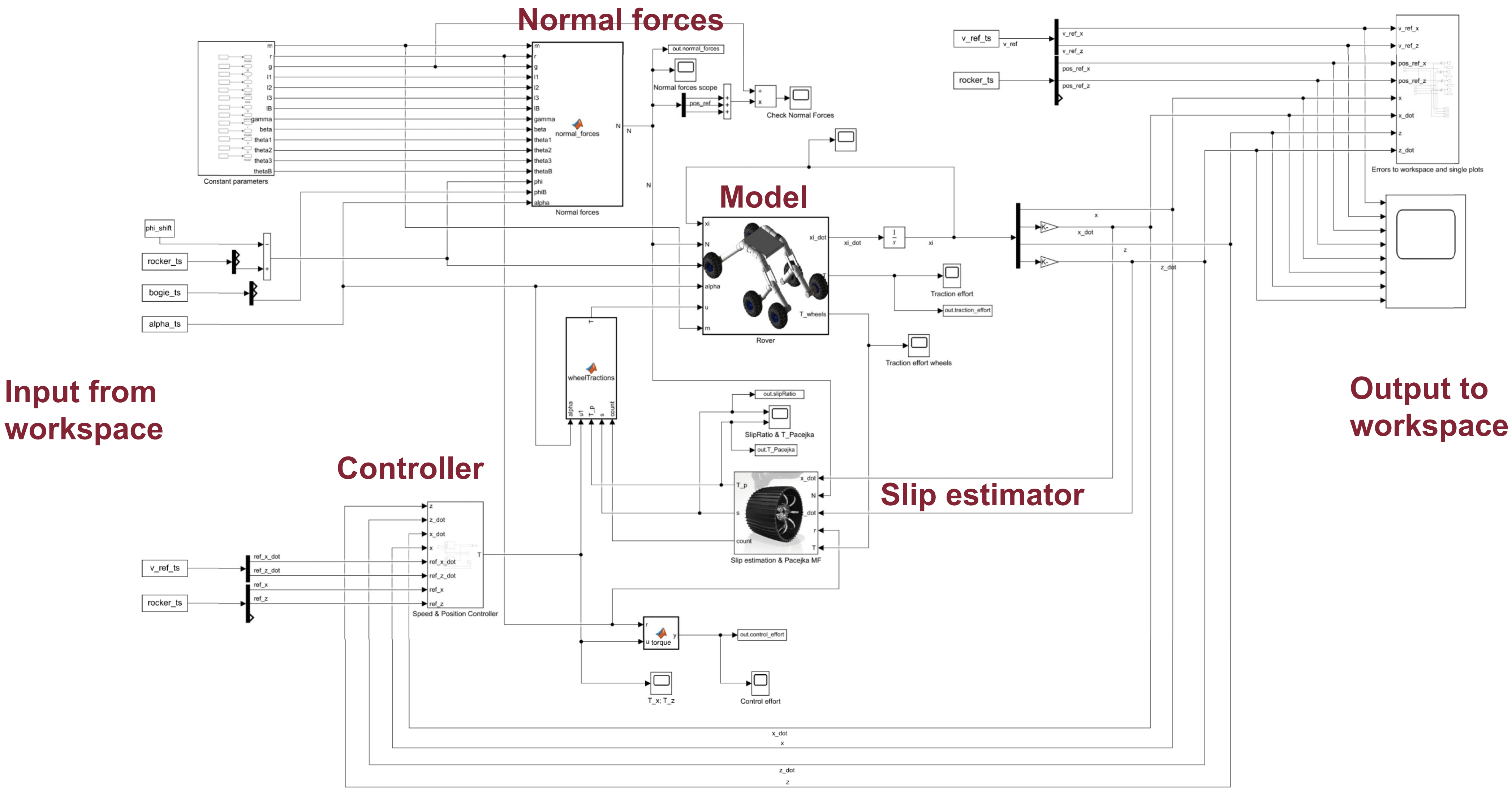
$$(x_2 - x_3)^2 + (z_2 - z_3)^2 = l_{23}$$
$$(x_2 - x_b)^2 + (z_2 - z_b)^2 = l_2$$
$$(x_3 - x_b)^2 + (z_3 - z_b)^2 = l_3$$
$$(x_b - x_r)^2 + (z_b - z_r)^2 = l_b$$
$$(x_1 - x_r)^2 + (z_1 - z_r)^2 = l_1$$
$$(x_1 - x_b)^2 + (z_1 - z_b)^2 = l_{1b}$$

Numerical solution of the system

Bogie's coordinates



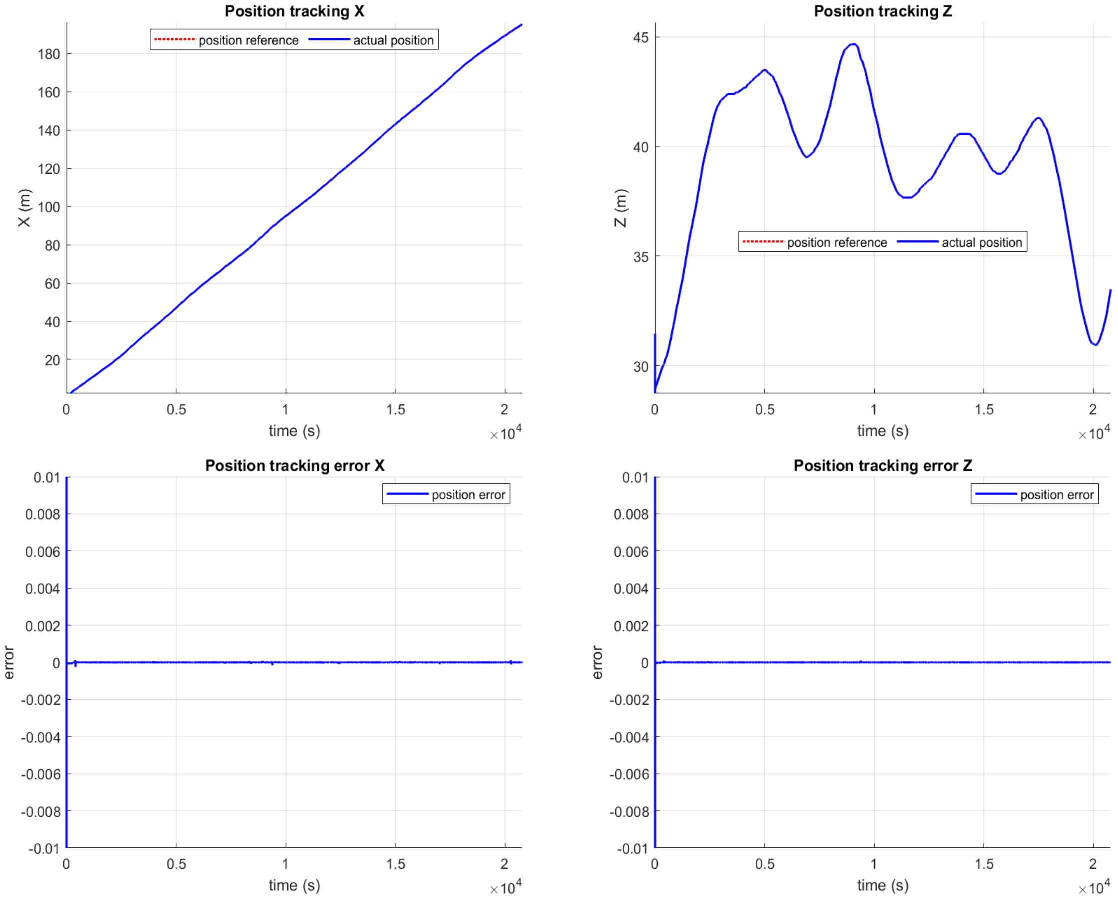
Simulink model



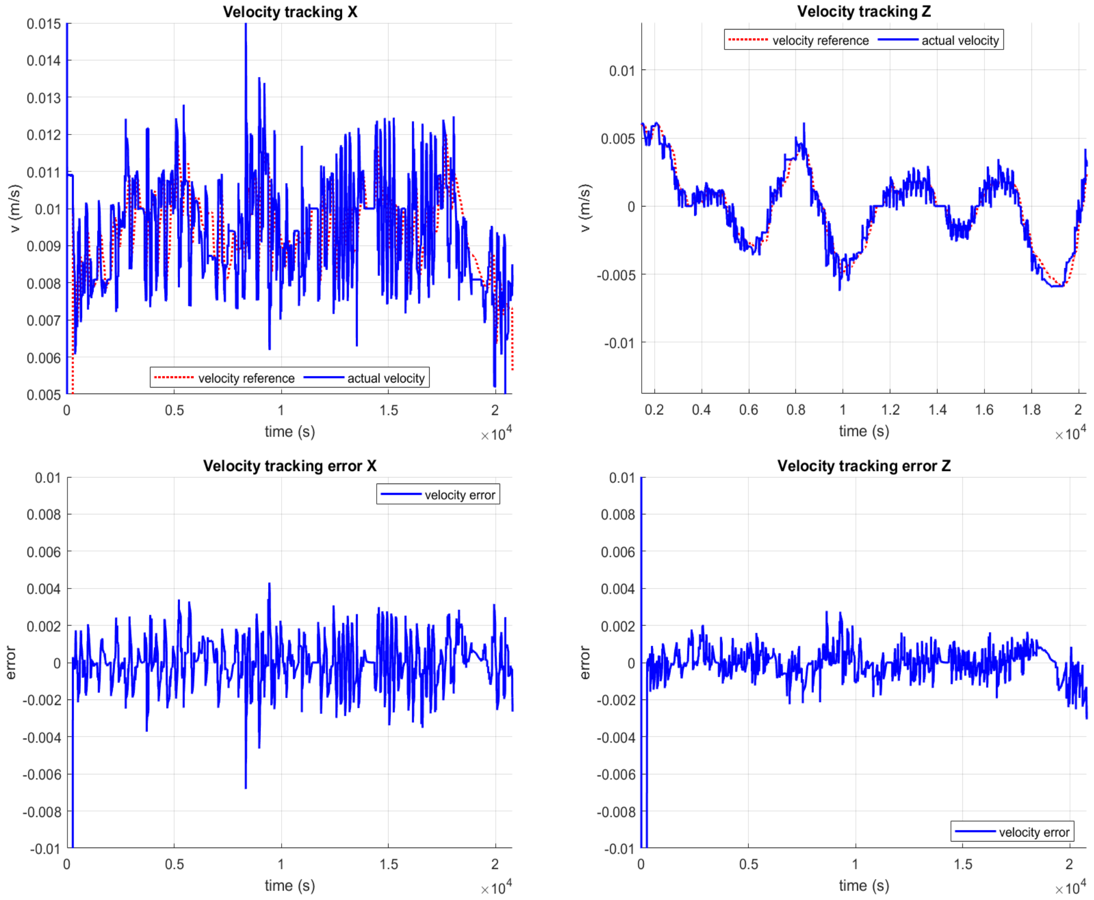
Simulation results

Simulation results

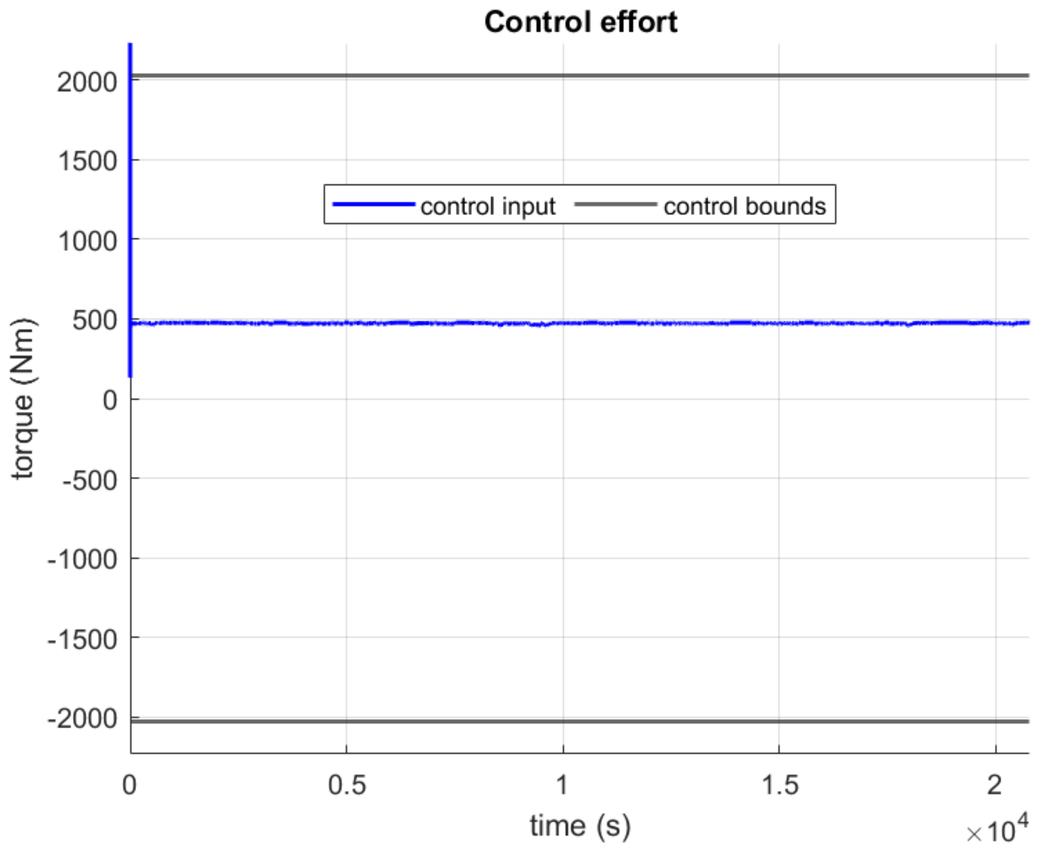
Position tracking

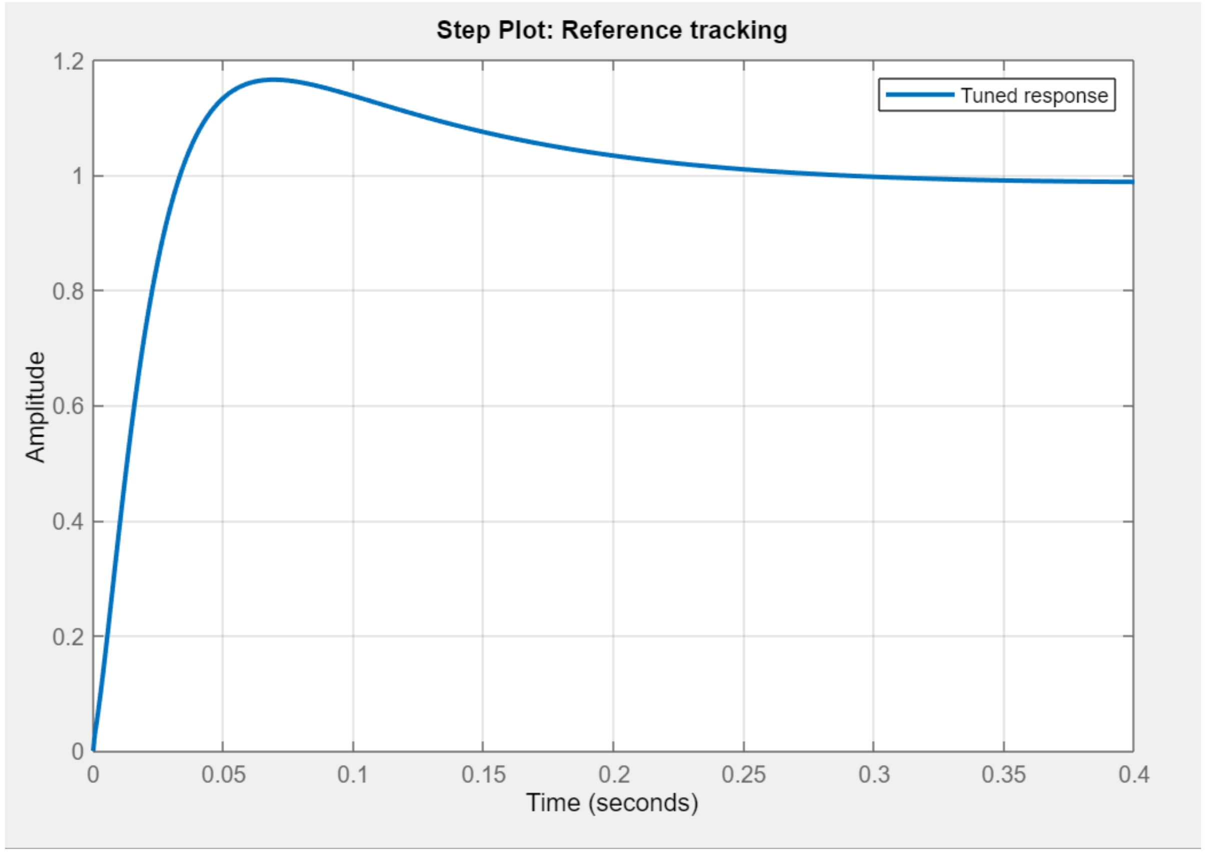


Velocity tracking



Simulation results





Slip estimator and Pacejka's magic formula

- In order to reduce wheel slip the normal force interaction must be regulated.

- Longitudinal slip ratio:

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

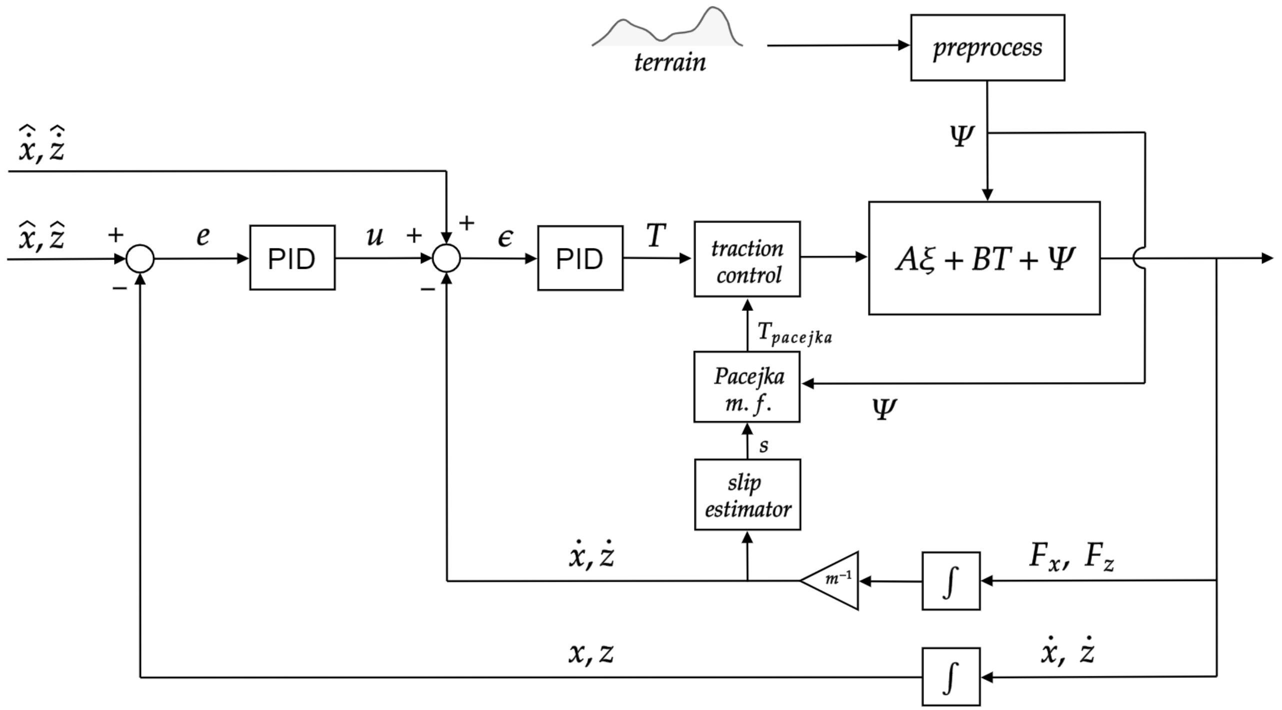
$$\sigma_i = \frac{r\omega_i - v_{x,i}}{r\omega_i}, \quad r\omega_i > v_{x,i} \quad Accelerating$$

- Longitudinal forces are supposed to be dependent on the normal force, surface friction, and longitudinal slip ratio.

$$T_i = T(\sigma_i, \mu, N_i)$$

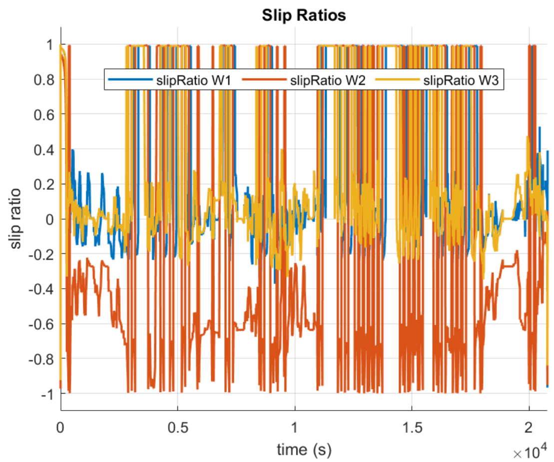
- Pacejka tire model is used. Tire forces are described as functions of the slip ratio

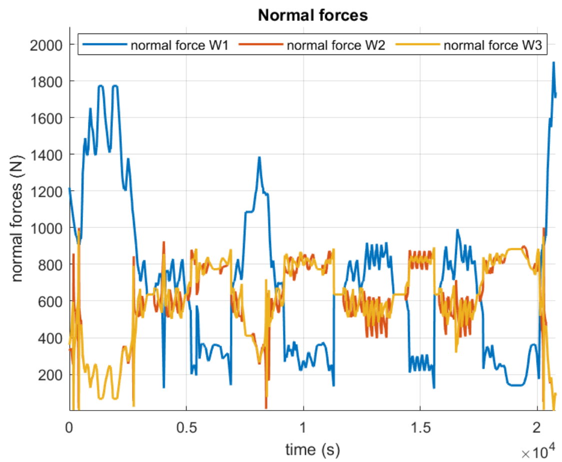
$$\mu(\sigma_i) = Dsin(Ctan^{-1}(B\sigma_i - E(B\sigma_i - tan^{-1}(B\sigma_i))))$$

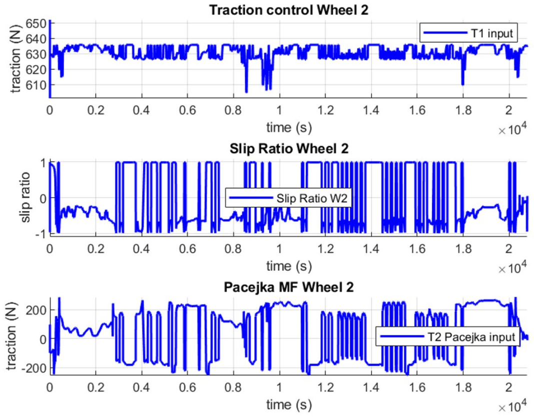


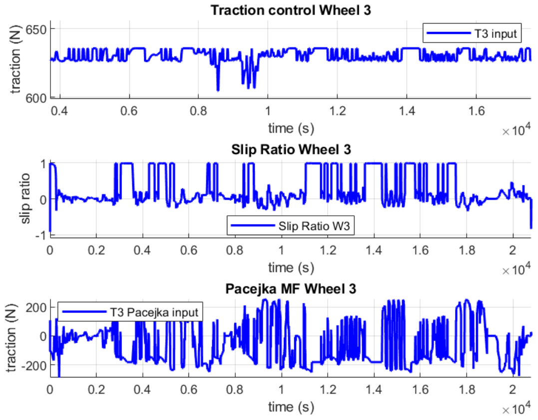
Slip estimator and Pacejka's magic formula

- Longitudinal traction forces: $T_i = \mu(\sigma_i)N_i$
- Slip estimator checks for task feasibility
- If a task is feasible, traction forces are computed through Pacejka m.f.









Conclusion

- The planar analysis of the rover dynamics has been conducted in this study.
- Strong assumptions has been made about the dynamic model. A relaxation of these would require a completely different control design.
- The system with the designed controller, under the specified assumptions, is capable of tracking the desired reference position and velocity with satisfactory results.
- 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.
- Traction control could be further developed including the complete knowledge of the motor dynamics and a better understanding of the ground-tire interaction for this particular kind of application.


NASA
THANKS
FOR YOUR
ATTENTION