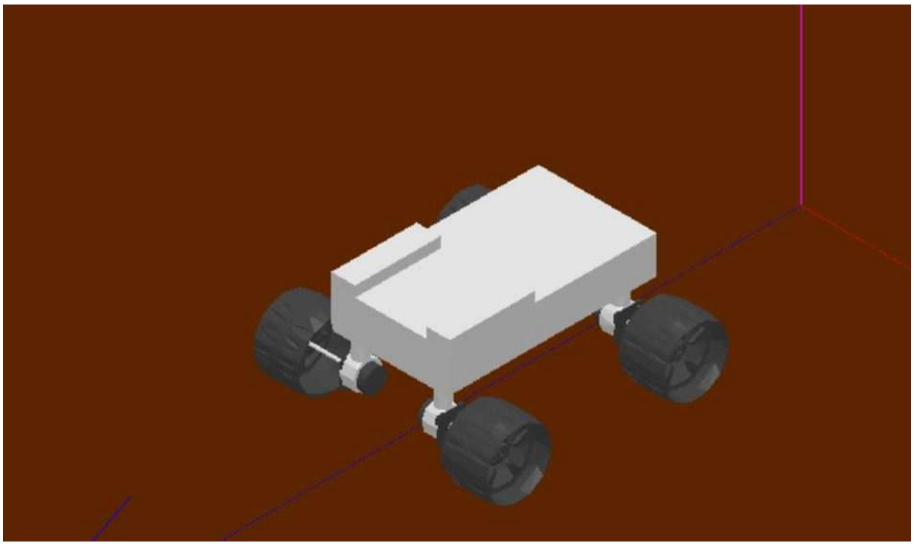
Mars Rover Simulation and Control

Jonas Chianu

April 29, 2021



1 General data of the studied mechanism

The system comprises 14 bodies (defined by the global variable `nbrbody`). Each body is called S_j (j from 0 to 13). The number of degrees of freedom of the system is 10 (`nbrdof`). The configuration parameters are denoted by q_i (i from 0 to 9).

The inertial data, given by the user, consist of the mass m_{Si} and the inertia tensor $\Phi_{G,Si}$ of each body i expressed with respect to the center of gravity.

$$m_{S0} = 1.0 \cdot 10^{-10}\, kg$$

$$m_{S1} = 42.6\, kg$$

$$m_{S2} = 2.5\, kg$$

$$m_{S3} = 0.5\, kg$$

$$m_{S4} = 2.0\, kg$$

$$m_{S5} = 2.5\, kg$$

$$m_{S6} = 0.5\, kg$$

$$m_{S7} = 2.0\, kg$$

$$m_{S8} = 2.5\, kg$$

$$m_{S9} = 0.5\, kg$$

$$m_{S10} = 2.0\,kg$$
$$m_{S11} = 2.5\,kg$$
$$m_{S12} = 0.5\,kg$$
$$m_{S13} = 2.0\,kg$$

$$\Phi_{G,S0} = \begin{bmatrix} 1.0\cdot 10^{-10} & 0 & 0 \\ 0 & 1.0\cdot 10^{-10} & 0 \\ 0 & 0 & 1.0\cdot 10^{-10} \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S1} = \begin{bmatrix} 0.84 & 0 & 0 \\ 0 & 1.81 & 0 \\ 0 & 0 & 2.44 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S2} = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.003 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S3} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 1.0\cdot 10^{-5} & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S4} = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.04 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S5} = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.003 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S6} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 1.0\cdot 10^{-5} & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S7} = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.04 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S8} = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.003 \end{bmatrix} \quad \text{, en } kg.m^2$$

$$\Phi_{G,S9} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \quad , \text{ en } kg.m^2$$

$$\Phi_{G,S10} = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.04 \end{bmatrix} \quad , \text{ en } kg.m^2$$

$$\Phi_{G,S11} = \begin{bmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.003 \end{bmatrix} \quad , \text{ en } kg.m^2$$

$$\Phi_{G,S12} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \quad , \text{ en } kg.m^2$$

$$\Phi_{G,S13} = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.04 \end{bmatrix} \quad , \text{ en } kg.m^2$$

2 Complete kinematics calculated by Sympy

The following parameters have been calculated from the user's file `Rover.py` with a *CPU* time of 3.233 second(s).

Relative motions

The motion of some bodies has been defined as a relative motion with respect to another body. It is the case of 12 bodies, for which the motion is defined in the following manner :

- Motion of body S_2 is given with respect to body S_1.
- Motion of body S_3 is given with respect to body S_2.
- Motion of body S_4 is given with respect to body S_3.
- Motion of body S_5 is given with respect to body S_1.
- Motion of body S_6 is given with respect to body S_5.
- Motion of body S_7 is given with respect to body S_6.
- Motion of body S_8 is given with respect to body S_1.
- Motion of body S_9 is given with respect to body S_8.
- Motion of body S_{10} is given with respect to body S_9.

- Motion of body S_{11} is given with respect to body S_1.
- Motion of body S_{12} is given with respect to body S_{11}.
- Motion of body S_{13} is given with respect to body S_{12}.

Homogeneous transformation matrix of each body

$$T_{0G,S0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0G,S1} = \begin{bmatrix} \cos(q_4)\cos(q_5) & -\sin(q_5)\cos(q_4) & \sin(q_4) \\ \sin(q_3)\sin(q_4)\cos(q_5) + \sin(q_5)\cos(q_3) & -\sin(q_3)\sin(q_4)\sin(q_5) + \cos(q_3)\cos(q_5) & -\sin(q_3)\cos(\\ \sin(q_3)\sin(q_5) - \sin(q_4)\cos(q_3)\cos(q_5) & \sin(q_3)\cos(q_5) + \sin(q_4)\sin(q_5)\cos(q_3) & \cos(q_3)\cos(q \\ 0 & 0 & 0 \end{bmatrix}$$

$$T_{refG,S2/S1} = \begin{bmatrix} 1 & 0 & 0 & 0.31 \\ 0 & 1 & 0 & -0.13 \\ 0 & 0 & 1 & -0.17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S3/S2} = \begin{bmatrix} \cos(50q_6) & 0 & \sin(50q_6) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(50q_6) & 0 & \cos(50q_6) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S4/S3} = \begin{bmatrix} \cos(q_6) & 0 & \sin(q_6) & 0 \\ 0 & 1 & 0 & -0.19 \\ -\sin(q_6) & 0 & \cos(q_6) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S5/S1} = \begin{bmatrix} 1 & 0 & 0 & 0.31 \\ 0 & 1 & 0 & 0.13 \\ 0 & 0 & 1 & -0.17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S6/S5} = \begin{bmatrix} \cos(50q_7) & 0 & \sin(50q_7) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(50q_7) & 0 & \cos(50q_7) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S7/S6} = \begin{bmatrix} \cos(q_7) & 0 & \sin(q_7) & 0 \\ 0 & 1 & 0 & 0.19 \\ -\sin(q_7) & 0 & \cos(q_7) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S8/S1} = \begin{bmatrix} 1 & 0 & 0 & -0.29 \\ 0 & 1 & 0 & -0.13 \\ 0 & 0 & 1 & -0.17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S9/S8} = \begin{bmatrix} \cos(50q_8) & 0 & \sin(50q_8) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(50q_8) & 0 & \cos(50q_8) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S10/S9} = \begin{bmatrix} \cos(q_8) & 0 & \sin(q_8) & 0 \\ 0 & 1 & 0 & -0.19 \\ -\sin(q_8) & 0 & \cos(q_8) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S11/S1} = \begin{bmatrix} 1 & 0 & 0 & -0.29 \\ 0 & 1 & 0 & 0.13 \\ 0 & 0 & 1 & -0.17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S12/S11} = \begin{bmatrix} \cos(50q_9) & 0 & \sin(50q_9) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(50q_9) & 0 & \cos(50q_9) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{refG,S13/S12} = \begin{bmatrix} \cos(q_9) & 0 & \sin(q_9) & 0 \\ 0 & 1 & 0 & 0.19 \\ -\sin(q_9) & 0 & \cos(q_9) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Their time derivative

$$\dot{T}_{0G,S0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{0G,S1} = \left[\begin{matrix} -\dot{q}_4 \sin(q_4) \cos(q_5) - \dot{q}_5 \sin(q_5) \cos(q_4) \\ \dot{q}_3 \left(-\sin(q_3) \sin(q_5) + \sin(q_4) \cos(q_3) \cos(q_5)\right) + \dot{q}_4 \sin(q_3) \cos(q_4) \cos(q_5) + \dot{q}_5 \left(-\sin(q_3) \sin(q_4) \sin \right. \\ \dot{q}_3 \left(\sin(q_3) \sin(q_4) \cos(q_5) + \sin(q_5) \cos(q_3)\right) - \dot{q}_4 \cos(q_3) \cos(q_4) \cos(q_5) + \dot{q}_5 \left(\sin(q_3) \cos(q_5) + \sin \right. \\ 0 \end{matrix}\right.$$

$$\dot{T}_{refG,S2/S1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S3/S2} = \begin{bmatrix} -50\dot{q}_6 \sin(50q_6) & 0 & 50\dot{q}_6 \cos(50q_6) & 0 \\ 0 & 0 & 0 & 0 \\ -50\dot{q}_6 \cos(50q_6) & 0 & -50\dot{q}_6 \sin(50q_6) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S4/S3} = \begin{bmatrix} -\dot{q}_6 \sin(q_6) & 0 & \dot{q}_6 \cos(q_6) & 0 \\ 0 & 0 & 0 & 0 \\ -\dot{q}_6 \cos(q_6) & 0 & -\dot{q}_6 \sin(q_6) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S5/S1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S6/S5} = \begin{bmatrix} -50\dot{q}_7 \sin(50q_7) & 0 & 50\dot{q}_7 \cos(50q_7) & 0 \\ 0 & 0 & 0 & 0 \\ -50\dot{q}_7 \cos(50q_7) & 0 & -50\dot{q}_7 \sin(50q_7) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S7/S6} = \begin{bmatrix} -\dot{q}_7 \sin(q_7) & 0 & \dot{q}_7 \cos(q_7) & 0 \\ 0 & 0 & 0 & 0 \\ -\dot{q}_7 \cos(q_7) & 0 & -\dot{q}_7 \sin(q_7) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S8/S1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S9/S8} = \begin{bmatrix} -50\dot{q}_8 \sin(50q_8) & 0 & 50\dot{q}_8 \cos(50q_8) & 0 \\ 0 & 0 & 0 & 0 \\ -50\dot{q}_8 \cos(50q_8) & 0 & -50\dot{q}_8 \sin(50q_8) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S10/S9} = \begin{bmatrix} -\dot{q}_8 \sin(q_8) & 0 & \dot{q}_8 \cos(q_8) & 0 \\ 0 & 0 & 0 & 0 \\ -\dot{q}_8 \cos(q_8) & 0 & -\dot{q}_8 \sin(q_8) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S11/S1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S12/S11} = \begin{bmatrix} -50\dot{q}_9 \sin(50q_9) & 0 & 50\dot{q}_9 \cos(50q_9) & 0 \\ 0 & 0 & 0 & 0 \\ -50\dot{q}_9 \cos(50q_9) & 0 & -50\dot{q}_9 \sin(50q_9) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{T}_{refG,S13/S12} = \begin{bmatrix} -\dot{q}_9 \sin(q_9) & 0 & \dot{q}_9 \cos(q_9) & 0 \\ 0 & 0 & 0 & 0 \\ -\dot{q}_9 \cos(q_9) & 0 & -\dot{q}_9 \sin(q_9) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The velocity of the center of gravity of each body

$$\vec{v}_{G,S0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{v}_{G,S1} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S2/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S3/S2}\right\}_{S2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S4/S3}\right\}_{S3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S5/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S6/S5}\right\}_{S5} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S7/S6}\right\}_{S6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S8/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S9/S8}\right\}_{S8} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S10/S9}\right\}_{S9} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S11/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S12/S11}\right\}_{S11} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{v}_{G,S13/S12}\right\}_{S12} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The acceleration of the center of gravity of each body

$$\vec{a}_{G,S0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{a}_{G,S1} = \begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S2/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S3/S2}\right\}_{S2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S4/S3}\right\}_{S3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S5/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S6/S5}\right\}_{S5} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S7/S6}\right\}_{S6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S8/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S9/S8}\right\}_{S8} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S10/S9}\right\}_{S9} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S11/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{a}_{G,S12/S11}\right\}_{S11} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\left\{\vec{a}_{G,S13/S12}\right\}_{S12} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

The rotation velocity of each body

$$\vec{\omega}_{G,S0} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\vec{\omega}_{G,S1} = \begin{bmatrix}\dot{q}_3 + \dot{q}_5 \sin(q_4)\\ \dot{q}_4 \cos(q_3) - \dot{q}_5 \sin(q_3)\cos(q_4)\\ \dot{q}_4 \sin(q_3) + \dot{q}_5 \cos(q_3)\cos(q_4)\end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S2/S1}\right\}_{S1} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S3/S2}\right\}_{S2} = \begin{bmatrix}0\\50\dot{q}_6\\0\end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S4/S3}\right\}_{S3} = \begin{bmatrix}0\\\dot{q}_6\\0\end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S5/S1}\right\}_{S1} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S6/S5}\right\}_{S5} = \begin{bmatrix}0\\50\dot{q}_7\\0\end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S7/S6}\right\}_{S6} = \begin{bmatrix} 0 \\ \dot{q}_7 \\ 0 \end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S8/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S9/S8}\right\}_{S8} = \begin{bmatrix} 0 \\ 50\dot{q}_8 \\ 0 \end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S10/S9}\right\}_{S9} = \begin{bmatrix} 0 \\ \dot{q}_8 \\ 0 \end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S11/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S12/S11}\right\}_{S11} = \begin{bmatrix} 0 \\ 50\dot{q}_9 \\ 0 \end{bmatrix}$$

$$\left\{\vec{\omega}_{G,S13/S12}\right\}_{S12} = \begin{bmatrix} 0 \\ \dot{q}_9 \\ 0 \end{bmatrix}$$

The rotation acceleration of each body

$$\dot{\vec{\omega}}_{G,S0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\dot{\vec{\omega}}_{G,S1} = \begin{bmatrix} \dot{q}_4\dot{q}_5\cos(q_4) + \ddot{q}_3 + \ddot{q}_5\sin(q_4) \\ -\dot{q}_3\dot{q}_4\sin(q_3) - \dot{q}_3\dot{q}_5\cos(q_3)\cos(q_4) + \dot{q}_4\dot{q}_5\sin(q_3)\sin(q_4) + \ddot{q}_4\cos(q_3) - \ddot{q}_5\sin(q_3)\cos(q_4) \\ \dot{q}_3\dot{q}_4\cos(q_3) - \dot{q}_3\dot{q}_5\sin(q_3)\cos(q_4) - \dot{q}_4\dot{q}_5\sin(q_4)\cos(q_3) + \ddot{q}_4\sin(q_3) + \ddot{q}_5\cos(q_3)\cos(q_4) \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S2/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S3/S2}\right\}_{S2} = \begin{bmatrix} 0 \\ 50\ddot{q}_6 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S4/S3}\right\}_{S3} = \begin{bmatrix} 0 \\ \ddot{q}_6 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S5/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S6/S5}\right\}_{S5} = \begin{bmatrix} 0 \\ 50\ddot{q}_7 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S7/S6}\right\}_{S6} = \begin{bmatrix} 0 \\ \ddot{q}_7 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S8/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S9/S8}\right\}_{S8} = \begin{bmatrix} 0 \\ 50\ddot{q}_8 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S10/S9}\right\}_{S9} = \begin{bmatrix} 0 \\ \ddot{q}_8 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S11/S1}\right\}_{S1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S12/S11}\right\}_{S11} = \begin{bmatrix} 0 \\ 50\ddot{q}_9 \\ 0 \end{bmatrix}$$

$$\left\{\dot{\vec{\omega}}_{G,S13/S12}\right\}_{S12} = \begin{bmatrix} 0 \\ \ddot{q}_9 \\ 0 \end{bmatrix}$$

3 Simulation

The routine `NewmarkIntegration` performs the integration of the equations of motion up to time *FinalTime* by regular time intervals equal to *StepSave* and with the maximum allowed time step *StepMax* defined in the file `Rover`.cpp. The following values are used:

- *FinalTime* is the simulation duration (=5 s),
- *StepSave* is the time step in the numerical integration (=0.01 s),
- *StepMax* is the maximum allowed time step (=0.005 s),

the initial conditions being all zero.

4 Results

The time evolution of the different configuration parameters and their first and second derivatives can easily be plotted by `Gnuplot` as seen in figures 1 to 3 with the code listed below:

```
reset
set xlabel "Time [s]"
set grid

set term postscript eps color "Times-Roman" 20
set output "figure1.eps"
set ylabel "displacements"
plot 'Rover.res' using 1:2 title 'q_0' with line , 'Rover.res' using 1:5 title 'q_1' with line , 'Rover.res' us
set term pop
```

```
replot
pause -1 'Next plot (velocity level)?'

set term postscript eps color "Times-Roman" 20
set output "figure2.eps"
set ylabel "velocities"
plot 'Rover.res' using 1:3 title 'qd_0' with line , 'Rover.res' using 1:6 title 'qd_1' with line , 'Rover.res'
set term pop
replot
pause -1 'Next plot (acceleration level)?'

set term postscript eps color "Times-Roman" 20
set output "figure3.eps"
set ylabel "accelerations"
plot 'Rover.res' using 1:4 title 'qdd_0' with line , 'Rover.res' using 1:7 title 'qdd_1' with line , 'Rover.res
set term pop
replot
pause -1 'Next plot (Setpoint displacement level)?'

set term postscript eps color "Times-Roman" 20
set output "Setpoint_x.eps"
set ylabel "displacements"
plot 'Rover.res' using 1:33 title 'x' with line
set term pop
replot
pause -1 'Next plot (Setpoint velocity level)?'

set term postscript eps color "Times-Roman" 20
set output "Setpoint_xd.eps"
set ylabel "velocities"
plot 'Rover.res' using 1:34 title 'xd' with line
set term pop
replot
pause -1 'Next plot (Setpoint acceleration level)?'

set term postscript eps color "Times-Roman" 20
set output "Setpoint_xdd.eps"
set ylabel "accelerations"
plot 'Rover.res' using 1:35 title 'xdd' with line
set term pop
replot
pause -1 'Next plot (Tracking)?'

set term postscript eps color "Times-Roman" 20
set output "Tracking_x.eps"
set ylabel "displacements"
plot 'Rover.res' using 1:33 title 'x' with line , 'Rover.res' using 1:2 title 'q0' with line
set term pop
replot
pause -1 'Next plot (Error)?'

set term postscript eps color "Times-Roman" 20
set output "Error.eps"
set ylabel "Error [m]"
plot 'Rover.res' using 1:36 title 'xd' with line
set term pop
replot
pause -1 'Next plot (Voltage)?'

set term postscript eps color "Times-Roman" 20
set output "Voltage.eps"
set ylabel "Voltage [V]"
```

```
plot 'Rover.res' using 1:32 title 'Voltage' with line
set term pop
replot
pause -1
```

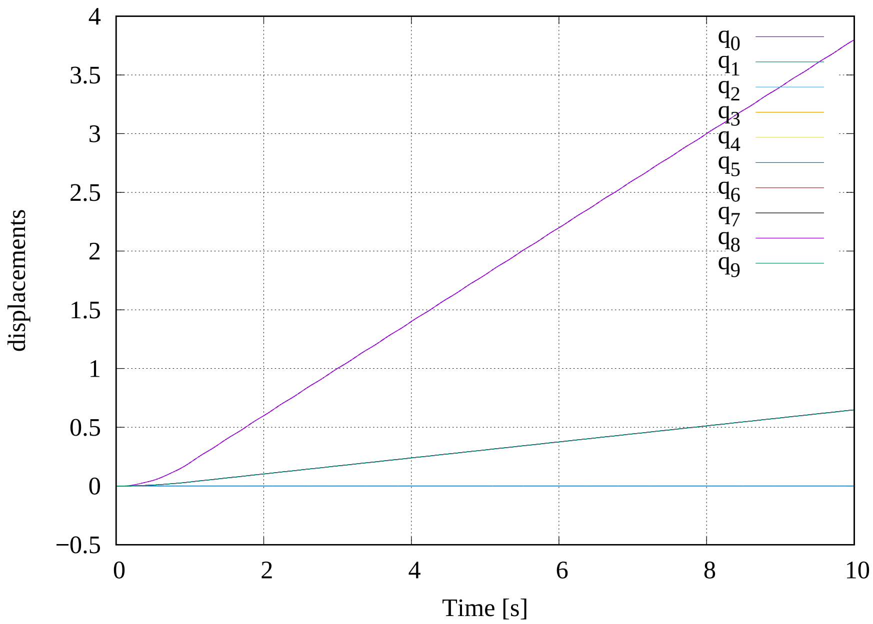


Figure 1: Time evolution of configuration parameters

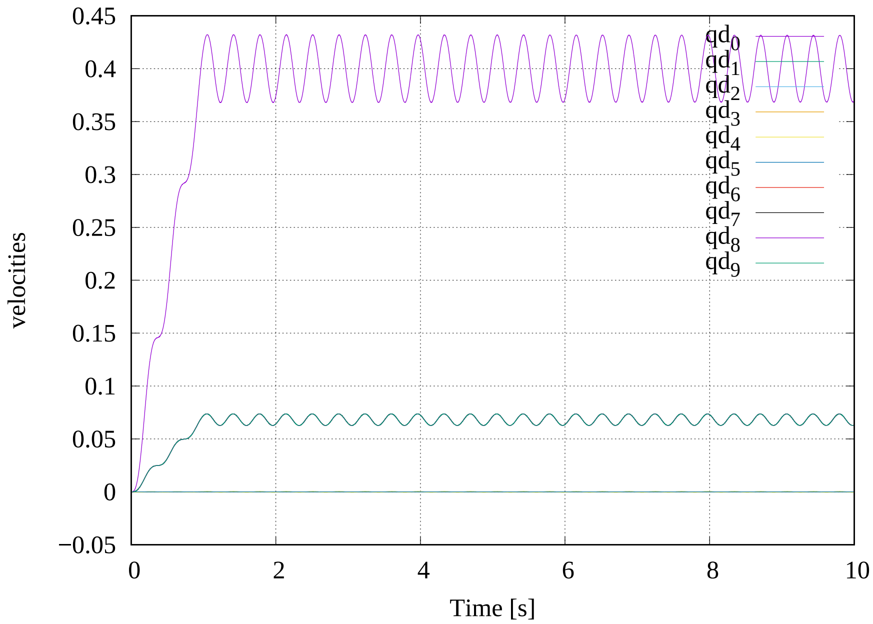


Figure 2: Time evolution of time derivatives of configuration parameters

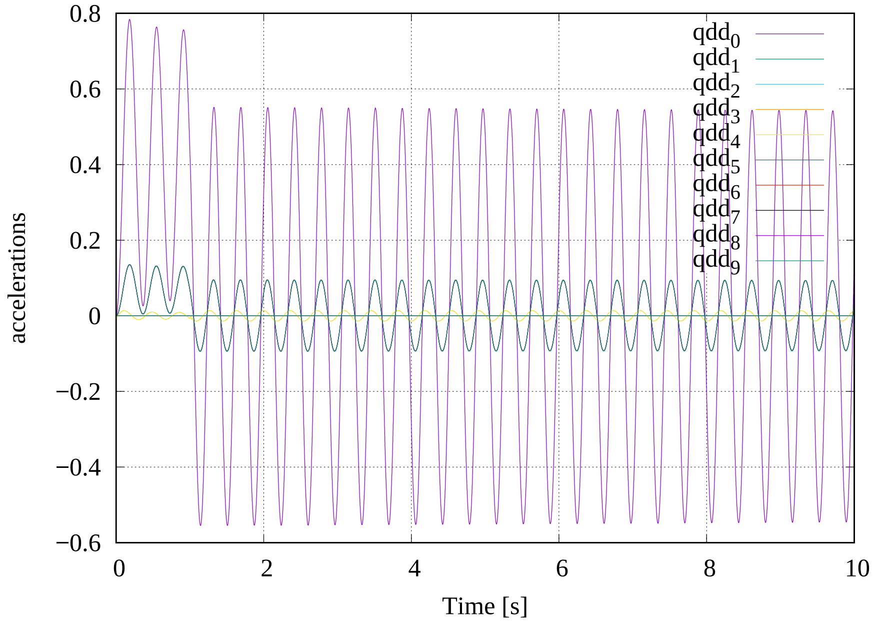


Figure 3: Time evolution of second time derivatives of configuration parameters

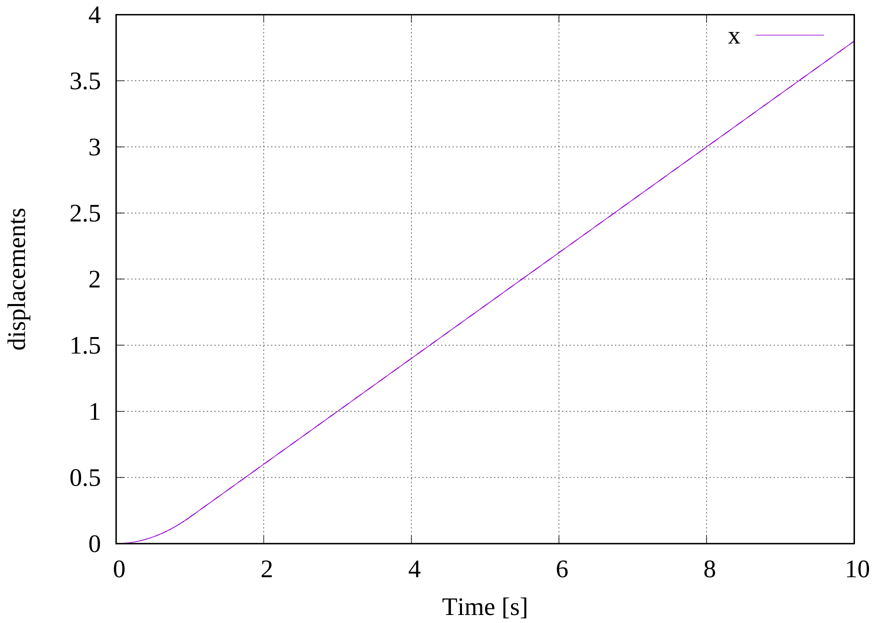


Figure 4: Time evolution of Setpoint

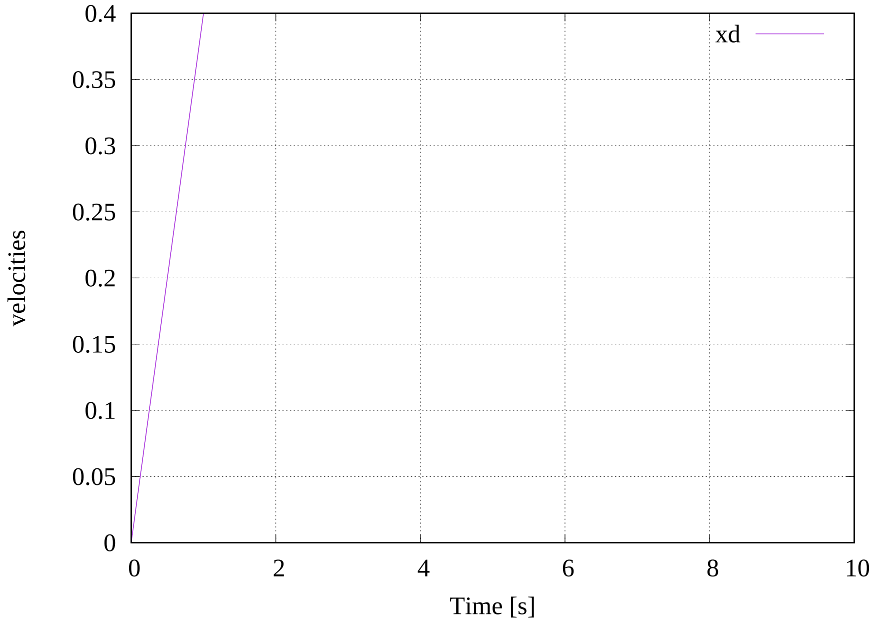


Figure 5: Time evolution of first time derivatives of Setpoint

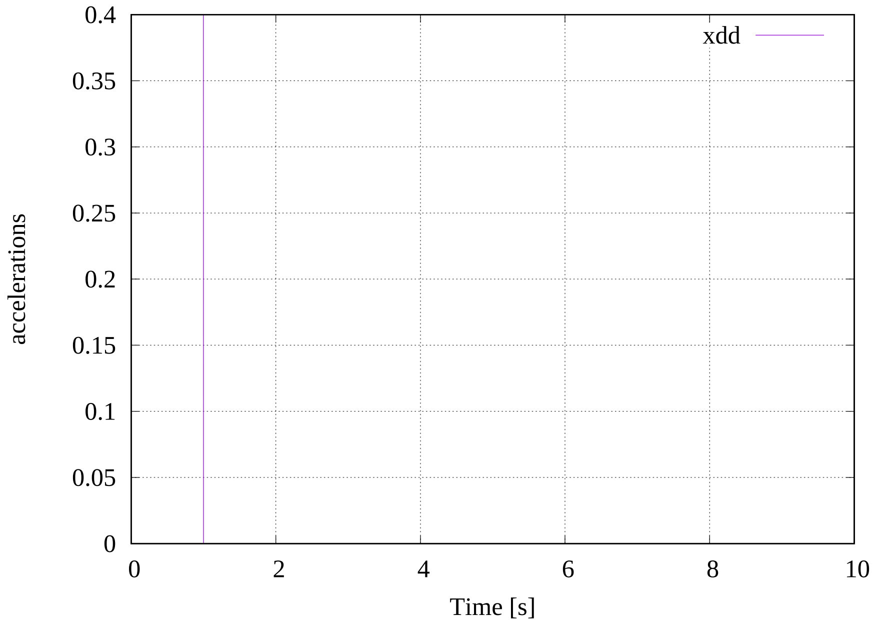


Figure 6: Time evolution of second time derivatives of Setpoint

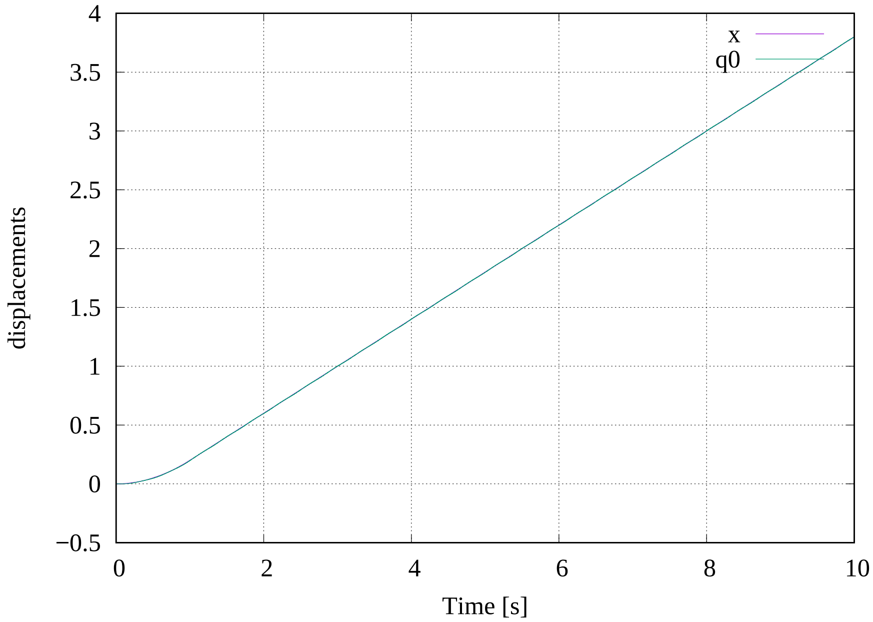


Figure 7: Time evolution of Tracking

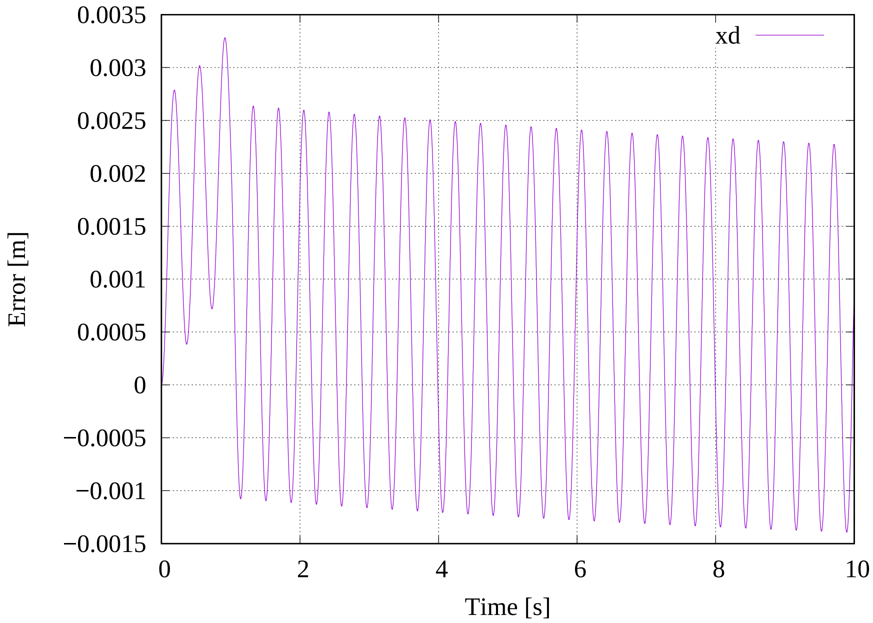


Figure 8: Time evolution of Error

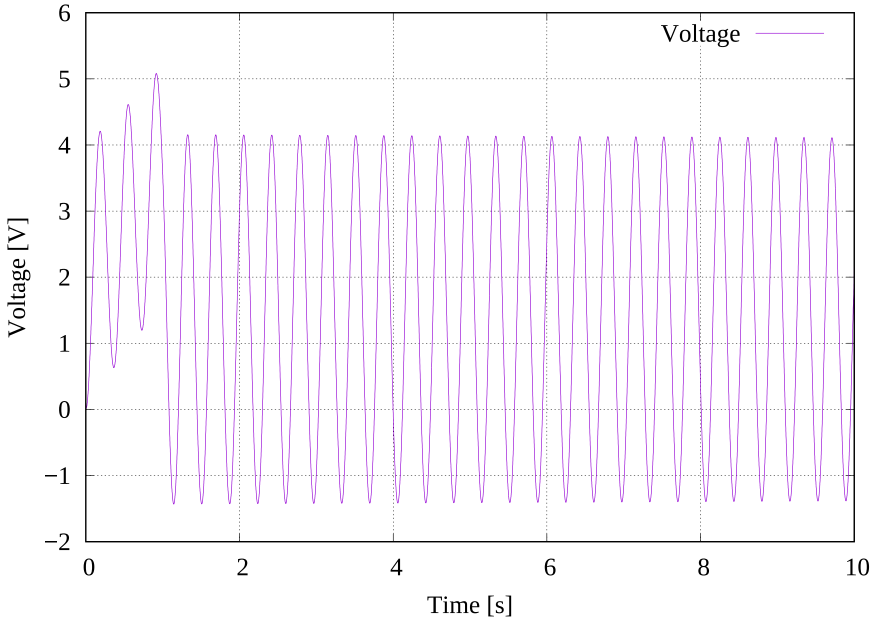


Figure 9: Time evolution of Voltage

A User's Python code

```
#!/usr/bin/python3

import sys
sys.path.insert(0, '../../../..')
sys.dont_write_bytecode = True # Disable pycache folder

from cagem import * # Import functions to derive system kinematics

#~~~~~~~~#
# Mars Rover Simulation and Control #
#~~~~~~~~#

#-------------------#
# System properties #
#-------------------#
Rvr=MBSClass(nbrbody=14,\
nbrdof=10,\
nbrdep=0,\
    ApplicationTitle="Mars Rover Simulation and Control",\
    ApplicationFileName="Rover")
# nbrbody: Number of bodies
# nbrdof:  Number of degrees of freedom
# nbrdep:  Number of dependent parameters
q,p,pexpr,t=Rvr.Getqpt() # Declare system variables

# Gravity vector

# Mars Gravity = -3.72 m/s^2
Rvr.SetGravity(0,0,-3.72)

# Write in cpp format
# Set global variables
Rvr.SetGlobalVar("""
/* Motor constants */
double n=50.0; // Reduction ratio
double Kt[nActuator] = {0.4,0.4,0.4,0.4};
double Ra[nActuator] = {0.85,0.85,0.85,0.85};
double i_current[nActuator] = {0.0,0.0,0.0,0.0};

// Trajectory profile
double Acceleration_Time = 1.0; // [s]
double DesiredVelocity = 0.4; // [m/s]
double SetPoint = 0.0;

double Acceleration_Cst = 0.0;
double Current_Position = 0.0;
double Current_Velocity = 0.0;
double Current_Acceleration = 0.0;

double t_final = 10.0; // Simulation time
double dt = 1e-3; // Time step

// Discrete PI controller
double Ki_controller = 100.0;
double Kp_controller = 1500.0;
double Error_Current = 0.0;
double Error_Old = 0.0;
double ControllerCommand = 0.0;
double ControllerCommand_old = 0.0;
```

```
// Animation speed up
int SaveFrameCounter = 0;
int SpacedFrame = 100; // Every X passages in SaveData, save the frame for EasyAnim
""")

# Some constants
n=50 # Reduction ratio

#--------------------#
# Inertia properties #
#--------------------#
Rvr.body[0].Set( mass=1e-10,Ixx=1e-10, Iyy=1e-10, Izz=1e-10) # Ground
Rvr.body[1].Set( mass=42.6 ,Ixx=0.84 , Iyy=1.81 , Izz=2.44)  # Main Body
Rvr.body[2].Set( mass=2.5  ,Ixx=0.003, Iyy=0.003, Izz=0.003) # Left  Motor 1 Front
Rvr.body[3].Set( mass=0.5  ,Ixx=0.001, Iyy=1e-5 , Izz=0.001) # Left  Rotor 1 Front
Rvr.body[4].Set( mass=2.0  ,Ixx=0.04 , Iyy=0.06 , Izz=0.04)  # Left  Wheel 1 Front
Rvr.body[5].Set( mass=2.5  ,Ixx=0.003, Iyy=0.003, Izz=0.003) # Right Motor 1 Front
Rvr.body[6].Set( mass=0.5  ,Ixx=0.001, Iyy=1e-5 , Izz=0.001) # Right Rotor 1 Front
Rvr.body[7].Set( mass=2.0  ,Ixx=0.04 , Iyy=0.06 , Izz=0.04)  # Right Wheel 1 Front
Rvr.body[8].Set( mass=2.5  ,Ixx=0.003, Iyy=0.003, Izz=0.003) # Left  Motor 2 Rear
Rvr.body[9].Set( mass=0.5  ,Ixx=0.001, Iyy=1e-5 , Izz=0.001) # Left  Rotor 2 Rear
Rvr.body[10].Set(mass=2.0  ,Ixx=0.04 , Iyy=0.06 , Izz=0.04)  # Left  Wheel 2 Rear
Rvr.body[11].Set(mass=2.5  ,Ixx=0.003, Iyy=0.003, Izz=0.003) # Right Motor 2 Rear
Rvr.body[12].Set(mass=0.5  ,Ixx=0.001, Iyy=1e-5 , Izz=0.001) # Right Rotor 2 Rear
Rvr.body[13].Set(mass=2.0  ,Ixx=0.04 , Iyy=0.06 , Izz=0.04)  # Right Wheel 2 Rear


#-------------------#
# Position matrices #
#-------------------#

# Ground
Rvr.body[0].T0F=Tdisp(0,0,0)
# Main body
# Frame of body 1 is free to move at the center of mass of the chassis
Rvr.body[1].T0F=Tdisp(q[0],q[1],q[2]+0.285)*Trotx(q[3])*Troty(q[4])*Trotz(q[5])

# Left Motor 1 Front
Rvr.body[2].TrefF=Tdisp(0.31,-0.13,-0.17)
Rvr.body[2].ReferenceFrame(1)
# Left Rotor 1 Front
Rvr.body[3].TrefF=Troty(n*q[6])
Rvr.body[3].ReferenceFrame(2)
# Left Wheel 1 Front
Rvr.body[4].TrefF=Tdisp(0,-0.19,0)*Troty(q[6])
Rvr.body[4].ReferenceFrame(3)

# Right Motor 1 Front
Rvr.body[5].TrefF=Tdisp(0.31,0.13,-0.17)
Rvr.body[5].ReferenceFrame(1)
# Right Rotor 1 Front
Rvr.body[6].TrefF=Troty(n*q[7])
Rvr.body[6].ReferenceFrame(5)
# Right Wheel 1 Front
Rvr.body[7].TrefF=Tdisp(0,0.19,0)*Troty(q[7])
Rvr.body[7].ReferenceFrame(6)

# Left Motor 2 Rear
Rvr.body[8].TrefF=Tdisp(-0.29,-0.13,-0.17)
Rvr.body[8].ReferenceFrame(1)
# Left Rotor 2 Rear
Rvr.body[9].TrefF=Troty(n*q[8])
```

```
Rvr.body[9].ReferenceFrame(8)
# Left Wheel 2 Rear
Rvr.body[10].TrefF=Tdisp(0,-0.19,0)*Troty(q[8])
Rvr.body[10].ReferenceFrame(9)

# Right Motor 2 Rear
Rvr.body[11].TrefF=Tdisp(-0.29,0.13,-0.17)
Rvr.body[11].ReferenceFrame(1)
# Right Rotor 2 Rear
Rvr.body[12].TrefF=Troty(n*q[9])
Rvr.body[12].ReferenceFrame(11)
# Right Wheel 2 Rear
Rvr.body[13].TrefF=Tdisp(0,0.19,0)*Troty(q[9])
Rvr.body[13].ReferenceFrame(12)

#----------------------#
# Initial configuration #
#----------------------#

Rvr.qini[0]=0 # [m/rad]
Rvr.qini[1]=0 # [m/rad]

# Need to be determined for the static equilibrium
# Method: Let the system stabilize without calling
# StaticEquilibrium function and plot the graph with
# the degrees of freedom and see how much q[2] (elevation
# of the rover) was decreased, then put the value
# as initial condition for q[2]
# Call StaticEquilibrium() function
# q[2] = -0.01 also works, as long as the
# the tire is a little bit inside the ground
# as indicated in the exam hint
Rvr.qini[2]=0 # [m/rad]

Rvr.qini[3]=0 # [m/rad]
Rvr.qini[4]=0 # [m/rad]
Rvr.qini[5]=0 # [m/rad]
Rvr.qini[6]=0 # [m/rad]
Rvr.qini[7]=0 # [m/rad]
Rvr.qini[8]=0 # [m/rad]
Rvr.qini[9]=0 # [m/rad]

#--------------------------#
# Forces #
#--------------------------#
Rvr.Force("""

///----------///
/// Actuators ///
///----------///

/// Actuator current
for(int i_motor=0; i_motor<nActuator; i_motor++)
    i_current[i_motor] = (u[0] - Kt[i_motor] * n*qd[6+i_motor])/Ra[i_motor];

/// Torques applied on the rotors

body[3].MG+= (Kt[0]*i_current[0])* body[3].TOG.R.uy();
body[2].MG-= (Kt[0]*i_current[0])* body[3].TOG.R.uy();

body[6].MG+= (Kt[1]*i_current[1])* body[6].TOG.R.uy();
body[5].MG-= (Kt[1]*i_current[1])* body[6].TOG.R.uy();
```

```
body[9].MG+= (Kt[2]*i_current[2])* body[9].TOG.R.uy();
body[8].MG-= (Kt[2]*i_current[2])* body[9].TOG.R.uy();

body[12].MG+= (Kt[3]*i_current[3])* body[12].TOG.R.uy();
body[11].MG-= (Kt[3]*i_current[3])* body[12].TOG.R.uy();

///-------///
/// Tires ///
///-------///

/* Tire data */
structtyre tyre_rover_Front;
tyre_rover_Front.r1=0.115;
tyre_rover_Front.r2=0.03;
tyre_rover_Front.Kz=1000000.0;
tyre_rover_Front.Cz=600.0;
// Total mass of rover = 62.6
tyre_rover_Front.Fznom=  (62.6*3.72)*0.29/(0.29+0.31)/2.0; // For one front wheel

tyre_rover_Front.Clongnom=1000.0;
tyre_rover_Front.nlong=0.1;
tyre_rover_Front.Clatnom=1000.0;
tyre_rover_Front.nlat=0.1;
tyre_rover_Front.Ccambernom=100.0;
tyre_rover_Front.ncamber=0.1;
tyre_rover_Front.fClbs=0.6;
tyre_rover_Front.fClbd=0.4;

structtyre tyre_rover_Rear;
tyre_rover_Rear.r1=0.115;
tyre_rover_Rear.r2=0.03;
tyre_rover_Rear.Kz=1000000.0;
tyre_rover_Rear.Cz=600.0;
// Total mass of rover = 62.6
tyre_rover_Rear.Fznom=  (62.6*3.72)*0.31/(0.29+0.31)/2.0; // For one rear wheel
tyre_rover_Rear.Clongnom=1000.0;
tyre_rover_Rear.nlong=0.1;
tyre_rover_Rear.Clatnom=1000.0;
tyre_rover_Rear.nlat=0.1;
tyre_rover_Rear.Ccambernom=100.0;
tyre_rover_Rear.ncamber=0.1;
tyre_rover_Rear.fClbs=0.6;
tyre_rover_Rear.fClbd=0.4;

/* Definition of the tire forces */
AddTyreEfforts(4,vcoord(0,1,0),tyre_rover_Front); /// Left Wheel 1 Front
AddTyreEfforts(7,vcoord(0,1,0),tyre_rover_Front); /// Right Wheel 1 Front

AddTyreEfforts(10,vcoord(0,1,0),tyre_rover_Rear); /// Left Wheel 2 Rear
AddTyreEfforts(13,vcoord(0,1,0),tyre_rover_Rear); /// Right Wheel 2 Rear

""")

#------------------------#
# Integration parameters #
#------------------------#
Rvr.SetIntegrationParameters(tfinal=5, hsave=0.01, hmax=0.005)
# tfinal = duration of simulation [s]
# hsave = time step for saving results [s]
# hmax = adaptive integration time step [s]

#---------#
```

```
# Options #
#---------#
STATIC=1 # Set STATIC to 1 to search the static equilibrium before integration
POLE=1 # Set POLE to 1 to perform an eigen value analysis
TEST=0 # Set TEST to 1 to perform the efficiency tests

Rvr.ComputeKinematics() # Derive the system kinematics symbolically
Rvr.EasyDynFlags(STATIC,POLE,TEST) # Define flags
Rvr.ExportEasyDynProgram() # Write .cpp file enclosing the kinematics

Rvr.ExportUK_Latex_Report() # Uncomment to generate the English LaTeX report
Rvr.ExportGnuplotScript() # Uncomment to output script for graphs of variables
```