



HAL
open science

Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation

Christophe Grand, Faïz Ben Amar, Frédéric Plumet

► **To cite this version:**

Christophe Grand, Faïz Ben Amar, Frédéric Plumet. Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation. *Mechanism and Machine Theory*, 2010, 45 (3), pp.477-495. 10.1016/j.mechmachtheory.2009.10.007 . hal-00732803

HAL Id: hal-00732803

<https://hal.science/hal-00732803>

Submitted on 17 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation

Christophe Grand^{*}, Faiz Ben Amar, Frédéric Plumet

^a *Institut des Systèmes Intelligents et de Robotique,
Université Pierre et Marie Curie - Paris6, CNRS
4 place Jussieu, 75252 Paris Cedex 05 - France*

Abstract

This paper proposes a general formulation of the kinetostatic model of articulated wheeled rovers that move on rough terrains. Differential kinematic model is used to control the generalized trajectory of the robot, composed of position and posture parameters. These posture parameters have been optimized in order to provide high stability and traction performance, during motion on irregular ground surface. Numerical simulation and experimental results, carried out on a hybrid wheeled-legged robot, show the validity of the approach presented in this paper.

Key words: kinematics; modeling; control; mobile robot; hybrid; wheeled-legged

1 Introduction

Autonomous robotic rovers have many potential applications, including space exploration, agriculture, defense, demining. Rovers such as Sojourner [1], Shrimp [2], Nomad [3] are articulated multibody structure permitting a passive adaptation to ground surface. Whereas, new kinematics of rovers like SRR [4], Gofor [5] or Lama [6] use an active suspension allowing control of some attitude parameters of the robot. These articulated wheeled systems differs from walking machines in the sense that wheeled systems maintain the contact continuously with the ground surface and velocity transmission is mainly ensured by rolling on the ground surface. High mobility systems such as Azimut [7], Hylos [8], Workpartner [9], Athlete [10] combining both rolling and crawling,

^{*} Corresponding author.

Email address: christophe.grand@upmc.fr (Christophe Grand).

inherits both advantages of wheeled and walking systems, i.e. the velocity for the first one and the ground clearance for the second.

Numerous works are related with the motion analysis of articulated wheeled systems. Kinematic analysis of motion on flat surface are developed by [11] and [12]. A classification of those systems, based on steering systems including omnidirectional wheels, are proposed in [13]. Those works are based on ideal rolling and no-side slip assumptions. Kinematics of a system composed of 2 wheels linked by an axle, evolving on 3D surface, is studied by [14] and [15]. This study proposes to use a variable length axle to prevent side slip. The rolling kinematic of a torus wheel on uneven continuous surface is investigated in [16] and the authors propose to use a passive joint allowing a lateral degree of freedom in order to overcome slippage. A methodology for developing a motion kinematics over rough ground and including various slip is proposed in [17].

This paper proposes a general kinetostatic formulation of quasi-static motion of articulated wheeled rovers. This formulation could include non-ideal contact conditions (rolling slip, side-slip, discontinuous contact, contact deformation). The method used here is based on the principle of velocities composition that allows to determine the velocity equations which link operational and joint parameters. The principle of virtual work is used to derive equilibrium equation and force transmission equation which connect contact forces, joint torques and the gravitational force. These models are applied to the motion control of a wheeled-legged rover based on the decoupling of the posture and the trajectory parameters.

Section 2 will present first a general formulation of velocity and forces transmission of wheeled-legged rover (WLR). Then in section 3, those models are applied to our experimental platform composed of four wheel-legs, and are solved in their inverse form which is convenient for the decoupled control of both the posture adjustment and the path following tasks. The section 4 discusses about the definition of the posture as a function of soil surface parameters. Finally, the section 5 presents the posture control applied to our experimental platform.

2 General kineto-static formulation of WLR

2.1 Kinematic parameters of the system

This section deals with kineto-static modelisation of systems having legs ended with wheels. The kinematic parameters are depicted in Fig. 1. The different

frames are:

- $\mathcal{R}_0 = (O, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$ the reference ground frame,
- $\mathcal{R}_p = (G, \mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$ the main platform frame,
- $\mathcal{R}_c^i = (P_i, \mathbf{t}_i, \mathbf{l}_i, \mathbf{n}_i)$ the contact frame of the i th wheel.

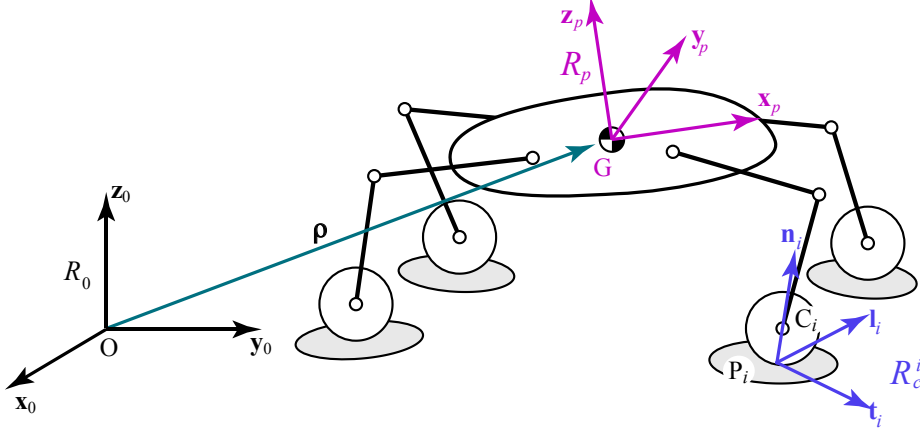


Fig. 1. Kinematic model

We define the parameters vector $\mathbf{q}^t = (\mathbf{x}^t, \boldsymbol{\theta}_i^t, \boldsymbol{\chi}_i^t)$ such as:

- $\mathbf{x}^t = (\boldsymbol{\rho}^t, \boldsymbol{\phi}^t)$ is the vector of platform parameters with respect to ground frame, where $\boldsymbol{\rho}^t = (x, y, z)$ denotes the vector of position parameters and $\boldsymbol{\phi}^t = (\varphi, \psi, \theta)$ denotes the vector of the usual consecutive yaw(θ), pitch(ψ) and roll(φ) angles;
- $\boldsymbol{\theta}_i$ is the vector of the i th leg joint parameters;
- $\boldsymbol{\chi}_i^t = (\gamma_i, \vartheta_i)$ is the i th vector of wheel's parameters, where γ_i is the steering angle and ϑ_i the rolling angle (the wheel spinning rotation rate will be denoted $\omega_i = \dot{\vartheta}_i$ that is a classical notation).

We denote $\boldsymbol{\Theta}_i$ the vector of joint parameters of the i th wheel-leg:

$$\boldsymbol{\Theta}_i^t = (\boldsymbol{\theta}_i^t, \boldsymbol{\chi}_i^t)$$

Then, if we consider a system composed of n wheel-leg kinematic chains, the vector of the robot parameters can be written as:

$$\mathbf{q}^t = (\mathbf{x}^t, \boldsymbol{\Theta}_1^t, \dots, \boldsymbol{\Theta}_n^t)$$

2.2 Wheel ground contact geometry

The contact area between the ground and wheel could have a complex geometry depending, firstly, on the flexibility of the ground and, secondly, on their

geometry. Fig. 2 gives a generic approached form of the contact area between a cylindrical flexible wheel on a soft ground. For this study, we consider a simplified geometry.

For each i th wheel, we denote:

- C_i is the center of the wheel,
- \mathbf{y}_i the unit axis of the wheel spinning motion,
- P_i the center point of the contact area,
- and $\mathcal{R}_c^i = (P_i, \mathbf{t}_i, \mathbf{l}_i, \mathbf{n}_i)$ the associated contact frame.

Where the contact frame is defined as follow: \mathbf{n}_i is the contact normal vector, \mathbf{t}_i is the longitudinal vector and $\mathbf{l}_i = \mathbf{n}_i \times \mathbf{t}_i$ is the lateral vector (see Fig.2). The vector \mathbf{t}_i is the projection of vector \mathbf{y}_i on the contact plane which depends on the system kinematic configuration:

$$\mathbf{t}_i = \frac{\mathbf{y}_i \times \mathbf{n}_i}{|\mathbf{y}_i \times \mathbf{n}_i|}$$

The center point of the contact area P_i is defined as a projection of the center of the wheel on the mean contact plan. It depends only on the wheel radius r_i and the contact normal:

$$\overrightarrow{C_i P_i} = r_i \overrightarrow{n_i}$$

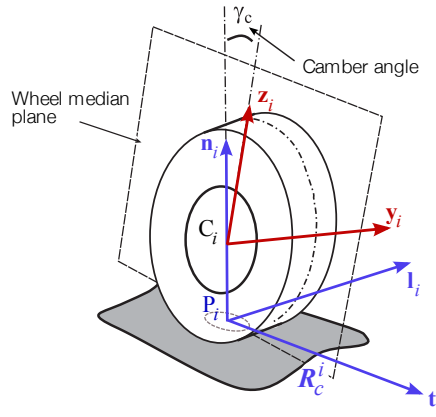


Fig. 2. Definition of the wheel-ground contact frame

2.3 Differential kinematic model of the system

Wheeled vehicles are non-holonomic system, differential kinematic equations are not explicitly integrable, they will then be established using the velocity

relations between the articulated rigid bodies. These equations assume permanent contacts but not necessary ideal rolling with the ground, i.e. the wheels could slip in both longitudinal (\mathbf{t}_i) and lateral (\mathbf{l}_i) directions of the contact plane.

For each wheel in contact with the ground, the closed-loop velocity equation is obtained by means of the velocity composition principle:

$$\vec{V}_{(P_i, \mathcal{R}_\omega^i / \mathcal{R}_0)} = \vec{V}_{(P_i, \mathcal{R}_\omega^i / \mathcal{R}_l^i)} + \vec{V}_{(P_i, \mathcal{R}_l^i / \mathcal{R}_p)} + \vec{V}_{(P_i, \mathcal{R}_p / \mathcal{R}_0)} \quad (1)$$

where $\mathcal{R}_\omega^i = (C_i, \mathbf{x}_\omega^i, \mathbf{y}_\omega^i, \mathbf{z}_\omega^i)$ is the frame attached to the i th wheel and $\mathcal{R}_l^i = (C_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ is the frame attached to the end of the i th leg mechanism (see Fig. 3).

- $\vec{V}_s^i = \vec{V}_{(P_i, \mathcal{R}_\omega^i / \mathcal{R}_0)}$ is the sliding velocity of the contact point P_i ;
- $\vec{V}_x^i = \vec{V}_{(P_i, \mathcal{R}_p / \mathcal{R}_0)}$ is the velocity of P_i due to platform motion with respect to the ground;
- $\vec{V}_\theta^i = \vec{V}_{(P_i, \mathcal{R}_l^i / \mathcal{R}_p)}$ is the velocity of P_i due to leg's motion with respect to the platform;
- $\vec{V}_\chi^i = \vec{V}_{(P_i, \mathcal{R}_\omega^i / \mathcal{R}_l^i)}$ is the wheel circumferential velocity with respect to the leg.

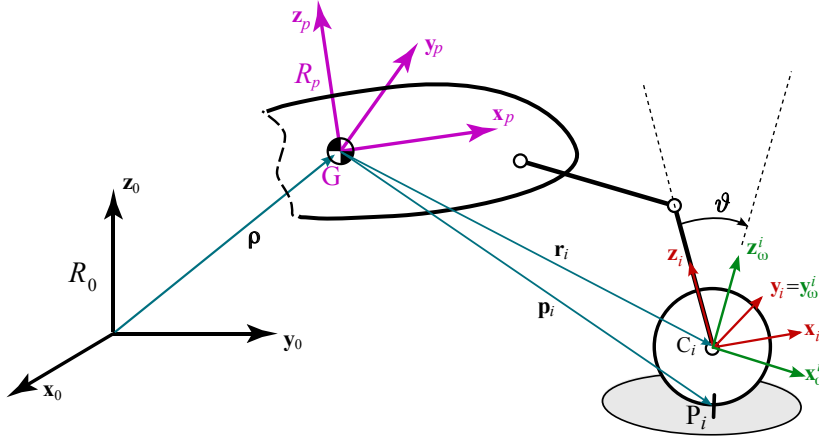


Fig. 3. Detail of the wheel-leg associated frames

2.3.1 Platform motion

First, \vec{V}_x^i (the velocity of the contact point due to platform motion with respect to the ground) is expressed in the platform frame \mathcal{R}_p :

$$\left[\mathbf{v}_x^i \right]_{\mathcal{R}_p} = \mathbf{R}^t \dot{\boldsymbol{\rho}} + \boldsymbol{\omega} \times \mathbf{p}_i \quad (2)$$

where \mathbf{p}_i is the position of the contact point in the platform frame and $(\dot{\boldsymbol{\rho}}, \boldsymbol{\omega})$ forms the platform twist. Let us remind that $\dot{\boldsymbol{\rho}}$ is the velocity of the center

of the platform expressed in the ground frame \mathcal{R}_0 and $\boldsymbol{\omega}$ is the rotation rate vector of the platform expressed in the platform frame \mathcal{R}_p .

\mathbf{R} is the rotation matrix between the platform frame and the ground, obtained by 3 successive rotations along \vec{z} , \vec{y} , \vec{x} and respectively parametrized by θ , ψ , φ angles:

$$\mathbf{R} = \mathbf{R}(\boldsymbol{\phi}) = \begin{bmatrix} C_\theta C_\psi & -S_\theta C_\varphi + C_\theta S_\psi S_\varphi & S_\theta S_\varphi + C_\theta S_\psi C_\varphi \\ S_\theta C_\psi & C_\theta C_\varphi + S_\theta S_\psi S_\varphi & -C_\theta S_\varphi + S_\theta S_\psi C_\varphi \\ -S_\psi & C_\psi S_\varphi & C_\psi C_\varphi \end{bmatrix} \quad (3)$$

with $C_\theta = \cos(\theta)$, $S_\theta = \sin(\theta)$, etc.

The rotation rate vector of the platform $\boldsymbol{\omega}$ can be computed as function of the time-derivative of orientation angles of the platform $\dot{\boldsymbol{\phi}} = (\dot{\varphi}, \dot{\psi}, \dot{\theta})^t$. This relation is given in matrix form as follow:

$$\boldsymbol{\omega} = \mathbf{T}_\phi \dot{\boldsymbol{\phi}} \quad (4)$$

with

$$\mathbf{T}_\phi(\boldsymbol{\phi}) = \begin{bmatrix} 1 & 0 & -S_\psi \\ 0 & C_\varphi & C_\psi S_\varphi \\ 0 & -S_\varphi & C_\psi C_\varphi \end{bmatrix} \quad (5)$$

Then, Equation (2) can be rewriting as function only of $\dot{\boldsymbol{\rho}}$ and $\dot{\boldsymbol{\phi}}$:

$$\left[\mathbf{v}_x^i \right]_{\mathcal{R}_p} = \mathbf{R}^t \dot{\boldsymbol{\rho}} - \tilde{\mathbf{p}}_i \mathbf{T}_\phi \dot{\boldsymbol{\phi}} \quad (6)$$

where $\tilde{\mathbf{p}}_i$ is the skew matrix corresponding to the cross-product operation, which lead to following equation in matrix form:

$$\left[\mathbf{v}_x^i \right]_{\mathcal{R}_p} = \begin{bmatrix} \mathbf{R}^t & -\tilde{\mathbf{p}}_i \mathbf{T}_\phi \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \mathbf{L}_i \dot{\boldsymbol{x}} \quad (7)$$

where $\dot{\boldsymbol{x}}$ is the platform velocity twist with respect to the ground frame \mathcal{R}_0 and \mathbf{L}_i is called locomotion matrix with a 3×6 dimensions.

In these equations, the vector \mathbf{p}_i depends on leg parameters $\boldsymbol{\theta}_i$, the wheel radius r_i and the normal contact \mathbf{n}_i :

$$\mathbf{p}_i = \mathbf{r}_i - r_i \mathbf{n}_i \quad (8)$$

where $\mathbf{r}_i = \left[\overrightarrow{GC_i} \right]_{\mathcal{R}_p}$ is the position of the center of the wheel expressed in the platform frame and is obtained from the kinematic model of the leg.

2.3.2 Leg motion

\vec{V}_θ^i , the velocity of the contact point P_i due to leg motion with respect to the platform, is expressed by classical serial chain differential kinematic model:

$$\left[\mathbf{v}_\theta^i \right]_{\mathcal{R}_p} = \left[\dot{\mathbf{r}}_i \right]_{\mathcal{R}_p} = \mathbf{J}_\theta^i \dot{\boldsymbol{\theta}}_i \quad (9)$$

In this equation, \mathbf{J}_θ^i is the classical jacobian matrix of the serial chain:

$$\mathbf{J}_\theta^i = \frac{\partial \mathbf{r}_i}{\partial \boldsymbol{\theta}_i} = \left[\mathbf{z}_1 \times \mathbf{a}_1 \dots \mathbf{z}_j \times \mathbf{a}_j \dots \mathbf{z}_m \times \mathbf{a}_m \right]_{\mathcal{R}_p}$$

where \mathbf{z}_j are the joint axes and \mathbf{a}_j the vector connecting each joint center to the platform reference point.

2.3.3 Wheel-leg loop closure equation

The vectorial equation of each loop closure condition, introduced in (1), is then expressed by its projection in the associated local contact frame \mathcal{R}_c^i :

$$\left[\mathbf{v}_s^i \right]_{\mathcal{R}_c^i} = \left[\mathbf{v}_\chi^i \right]_{\mathcal{R}_c^i} + \left[\mathbf{v}_\theta^i \right]_{\mathcal{R}_c^i} + \left[\mathbf{v}_x^i \right]_{\mathcal{R}_c^i} \quad (10)$$

So, the wheel circumferential velocity \vec{V}_χ^i which is only function of the wheel spinning velocity ω_i and the longitudinal vector \mathbf{t}_i , can be expressed in the local contact frame as:

$$\left[\mathbf{v}_\chi^i \right]_{\mathcal{R}_c^i} = -r \omega_i \left[\mathbf{t}_i \right]_{\mathcal{R}_c^i} = -r \omega_i (1, 0, 0)^t$$

Then, each loop closure condition given in the equation (10) is expressed by its projection on the local contact frame \mathcal{R}_c^i :

$$\left[\mathbf{v}_s^i \right]_{\mathcal{R}_c^i} = \mathbf{R}_i^t \mathbf{L}_i \dot{\mathbf{x}} + \mathbf{R}_i^t \mathbf{J}_\theta^i \dot{\boldsymbol{\theta}}_i - r \omega_i (1, 0, 0)^t \quad (11)$$

\mathbf{R}_i is the rotation matrix of the i th contact frame \mathcal{R}_c^i with respect to platform frame \mathcal{R}_p :

$$\mathbf{R}_i = \begin{bmatrix} | & | & | \\ \mathbf{t}_i & \mathbf{l}_i & \mathbf{n}_i \\ | & | & | \end{bmatrix}_{\mathcal{R}_p}$$

where \mathbf{n}_i is the normal contact vector expressed in the platform frame, \mathbf{t}_i is computed from the leg kinematic parameters, and $\mathbf{l}_i = \mathbf{n}_i \times \mathbf{t}_i$.

The projection in the contact frame allows to express directly the sliding condition in the contact. We denote \mathbf{v}_s^i the relative velocity in the i th contact:

$$\mathbf{v}_s^i = \left[\mathbf{v}_s^i \right]_{\mathcal{R}_c^i} = (s_t^i, s_l^i, s_n^i)^t \quad (12)$$

where:

- s_t^i expresses the longitudinal slippage,
- s_l^i expresses the lateral slippage,
- s_n^i could represent contact deformation velocity in the normal direction or the contact detachment velocity.

2.3.4 Differential kinematic model of the full system

From this point, it is supposed that all velocities equations are expressed by their projection in the contact frame \mathcal{R}_c^i of each wheel-leg chain. And finally, we obtain the velocity equation for the all system composed of n wheel-leg chains:

$$\mathbf{L}(\mathbf{x}, \Theta, \mathbf{n}) \dot{\mathbf{x}} + \mathbf{J}(\Theta, \mathbf{n}) \dot{\Theta} = \mathbf{v}_s \quad (13)$$

with

$$\mathbf{L} = \begin{bmatrix} \mathbf{R}_1^t \mathbf{L}_1 \\ \mathbf{R}_2^t \mathbf{L}_2 \\ \vdots \\ \mathbf{R}_n^t \mathbf{L}_n \end{bmatrix}_{3n \times 6} \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2 & & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \mathbf{J}_n \end{bmatrix}_{3n \times nm} \quad \Theta = \begin{bmatrix} \theta_1 \\ \chi_1 \\ \theta_2 \\ \chi_2 \\ \vdots \\ \theta_n \\ \chi_n \end{bmatrix}_{nm \times 1}$$

$$\mathbf{v}_s = \begin{bmatrix} \mathbf{v}_s^1 \\ \mathbf{v}_s^2 \\ \mathbf{v}_s^3 \\ \vdots \\ \mathbf{v}_s^n \end{bmatrix}_{nm \times 1} \quad \mathbf{J}_i = \begin{bmatrix} \mathbf{R}_i^t \mathbf{J}_\theta^i \\ \begin{bmatrix} -r \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}_{3 \times m}$$

and assuming that all chains have the same degree of freedom m .

One can notice that the Jacobian matrix \mathbf{J}_i of each wheel-leg chain depends on the normal vectors of wheel-ground contacts \mathbf{n}_i .

2.4 Quasi-static model of the system

We denote $\mathbf{f}^t = (\mathbf{f}_1^t, \mathbf{f}_2^t, \dots, \mathbf{f}_n^t)$ with $\mathbf{f}_i = (f_t^i, f_l^i, f_n^i)^t$ the vector of contact force components. We will use the principle of virtual power to determine the static equation assuming a virtual velocity field $(\dot{\mathbf{x}}^*, \dot{\Theta}^*, \mathbf{V}_s^*)$ which must satisfy kinematic equation (13).

Let us denote \mathbf{w} the 6×1 wrench vector that balances all the external forces applied to the system (including gravitational and inertial forces), expressed at the platform center of gravity. We denote $\boldsymbol{\tau}$ the vector of actuator torques applied on joints. The total power developed by external forces, contact forces and joint torques is:

$$\begin{aligned} P^* &= -\mathbf{w}^t \dot{\mathbf{x}}^* + \boldsymbol{\tau}^t \dot{\Theta}^* - \mathbf{f}^t \mathbf{V}_s^* \\ &= -\mathbf{w}^t \dot{\mathbf{x}}^* + \boldsymbol{\tau}^t \dot{\Theta}^* - \mathbf{f}^t (\mathbf{L} \dot{\mathbf{x}}^* + \mathbf{J} \dot{\Theta}^*) \\ &= (-\mathbf{w}^t - \mathbf{f}^t \mathbf{L}) \dot{\mathbf{x}}^* + (\boldsymbol{\tau}^t - \mathbf{f}^t \mathbf{J}) \dot{\Theta}^* \end{aligned} \quad (14)$$

The principles of virtual power states that:

$$P^* = 0 \quad \forall (\dot{\mathbf{x}}^*, \dot{\Theta}^*) \Leftrightarrow \begin{cases} -\mathbf{L}^t \mathbf{f} = \mathbf{w} \\ \mathbf{J}^t \mathbf{f} = \boldsymbol{\tau} \end{cases} \quad (15)$$

These equations assume that the total mass of the system is concentrated on the platform. The second equation should be corrected by adding \mathbf{w}_s which is the generalized force due to the weight of wheel-leg parts and associated to $\dot{\Theta}$ parameters:

$$\mathbf{J}^t \mathbf{f} = \boldsymbol{\tau} + \mathbf{w}_s \quad (16)$$

The system has a high degree of static indeterminacy i.e. scalar equilibrium equations are less than unknown contact forces. This indeterminacy is due to in one part to external contact with the environment (frictional contacts with 3 unknown force components at each contact) and in the another part to internal redundant actuation (for example all wheels are in general driven in off-road application). The resolution of this model gives the contact load distribution which are important for determining the traction torque applied to the wheel. In order to solve this model, we have to add relationships or assumptions generally on contact forces. Waldron [18] proposes to use the zero interaction principle to raise the static indeterminacy, based on a equiprojectivity of tangential contact forces. This principle establishes that all tangential contact forces work in the same direction to propel the vehicle. The second way to overcome the indeterminacy is to use tyre-terrain models as the magic formula for on-road vehicle [19] or terramechanics relations for off-road ones

[20]. Those models express relationships between contact force components, slippage parameters and mechanical properties of the ground.

2.5 Grasping analogy

We denote $\mathbf{G} = -\mathbf{L}^t$, then in the case of ideal rolling, equation (13) becomes:

$$\mathbf{G}^t \dot{\mathbf{x}} = \mathbf{J} \dot{\Theta} = \mathbf{v} \quad (17)$$

and equations (15) become :

$$\boldsymbol{\tau} = \mathbf{J}^t \mathbf{f} \quad (18)$$

$$\mathbf{w} = \mathbf{G} \mathbf{f}$$

This kineto-static duality can be summarized as for grasping system as shown in Fig. 4.

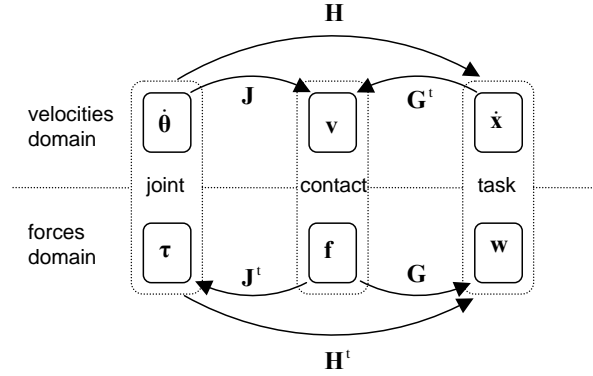


Fig. 4. Velocities and forces transmission in locomoting-grasping systems

3 Inverse velocity model of the robot Hylos

In this section, we will focus on the particular kinematics of Hylos robot. Then, we will use pure rolling assumption in order to compute the inverse velocity model which will be used in the posture control loop. As the jacobian \mathbf{J} is not square, the inverse kinematic model can be given by using the Moore-Penrose inverse matrix \mathbf{J}^+ :

$$\dot{\Theta} = -\mathbf{J}^+ \mathbf{L} \dot{\mathbf{x}} \quad (19)$$

This solution is generally used for redundant system. However, it is not adapted for our locomotion system. For example, in case of motion on flat horizontal plane, this solution will give greater importance to the legged locomotion.

Here, we will propose a method which separate horizontal motion parameters (x, y, θ) from attitude ones (z, φ, ψ) .

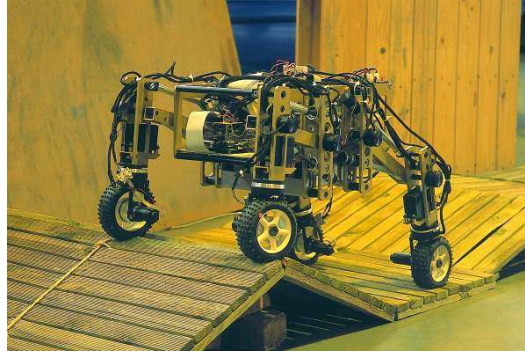


Fig. 5. Hylos robot

3.1 Hylos kinematics

Hylos robot, illustrated in Fig. 5, is approximately 70 cm long and its mass is about 12 kg. It is composed by four legs ($n = 4$), each one has two revolute joints with parallel axes and is ended by a driven and steered cylindrical wheel. Leg joints are actuated by means of ball screws and pantographic mechanisms. Thus, each wheel-leg has 4 degrees of freedom ($m = 4$) and the system has at all $m \times n = 16$ degrees of freedom actuated by DC motors.

For each wheel-leg, we denote α_i and β_i the first two joint parameters, γ_i and ϑ_i respectively the steering angle and the wheel rolling angle:

$$\boldsymbol{\theta}_i = (\alpha_i, \beta_i)^t \quad \text{and} \quad \boldsymbol{\chi}_i = (\gamma_i, \vartheta_i)^t \quad \longrightarrow \quad \boldsymbol{\Theta}_i = (\alpha_i, \beta_i, \gamma_i, \vartheta_i)^t$$

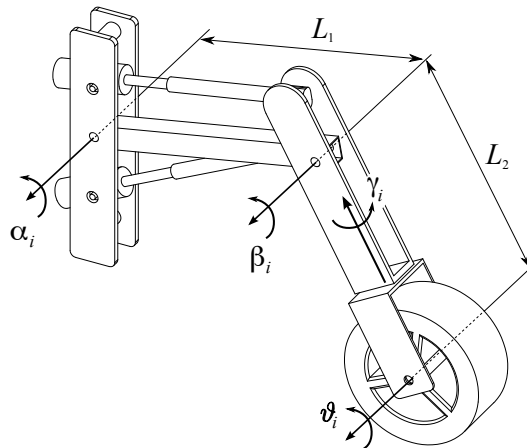


Fig. 6. CAD view of one wheel-leg

In the particular case of the Hylos robot design, the kinematic model of each leg corresponds to the classical 2 dof serial manipulator model:

$$\mathbf{r}_i = \begin{cases} x_i = \pm L_x/2 + L_1 \cos \alpha_i + L_2 \cos(\alpha_i + \beta_i) \\ y_i = \pm L_y/2 \\ z_i = L_1 \sin \alpha_i + L_2 \sin(\alpha_i + \beta_i) \end{cases} \quad (20)$$

where L_x and L_y are geometric parameters of the platform (Fig. 9), L_1 and L_2 are the lengths of the two leg segments (Fig. 6).

The jacobian matrix of each leg can be easily obtained by expressing the time-derivative of the previous equation:

$$\mathbf{J}_\theta^i = \begin{bmatrix} L_1 \sin \alpha_i + L_2 \sin(\alpha_i + \beta_i) & L_2 \sin(\alpha_i + \beta_i) \\ 0 & 0 \\ L_1 \cos \alpha_i + L_2 \cos(\alpha_i + \beta_i) & L_2 \cos(\alpha_i + \beta_i) \end{bmatrix} \quad (21)$$

3.2 Hylos mobility analysis

Ideal rolling assumption deals with non-slippage condition in wheel-ground contact. The inverse kinematic problem consists of determining the joint velocity for a given desired operational trajectory. The operational parameters should be defined as function of the general mobility of the system, which will be first investigated in this section.

The non slippage condition leads to:

$$\mathbf{L}\dot{\mathbf{x}} + \mathbf{J}\dot{\Theta} = \mathbf{0} \quad \text{or} \quad \mathbf{A}\dot{\mathbf{q}} = \mathbf{0} \quad (22)$$

with $\mathbf{A} = [\mathbf{L} \quad \mathbf{J}]$ and $\dot{\mathbf{q}}^t = (\dot{\mathbf{x}}^t, \dot{\Theta}^t)$.

An ideal rolling contact is equivalent to an instantaneous spherical joint located in the contact point, so it can be approached as a 3 dof joint. Then we obtain a structure with 18 bodies (including ground) and 20 joints with 28 dof at all. The general Gruebler mobility index is:

$$m_g = 28 - 6(20 - 18 + 1) = 10 \quad (23)$$

This mobility index can also be computed from equation (22) as it is the difference between the 12 equations and the 22 (=6+16) velocity parameters.

However, this general index does not consider the rank of the kinematic equations system and the geometry of joint axes. The real mobility index i.e. the

number of independent time-derivative parameters in the equation (22) can be defined as:

$$m_r = \dim(\mathbf{q}) - \text{rank}(\mathbf{A}) \quad (24)$$

The Fig. 7 depicts real kinematic mobility index as function of contact normals and rover configurations. For a general configuration of the robot and the ground, this mobility is equal to 10, i.e. all equations in (22) are independents. However, some particular configurations exhibit higher mobility (11 or 12), where the rank of matrix \mathbf{A} is equal to (11 or 10). In these cases, mobility increases and represents a partial internal mobility of the steering axes where the joint velocities become independents of all other vehicle velocity parameters. In these configurations, the additional mobilities seem to be located in the steering axes which are in these cases collinear to the contact normals.

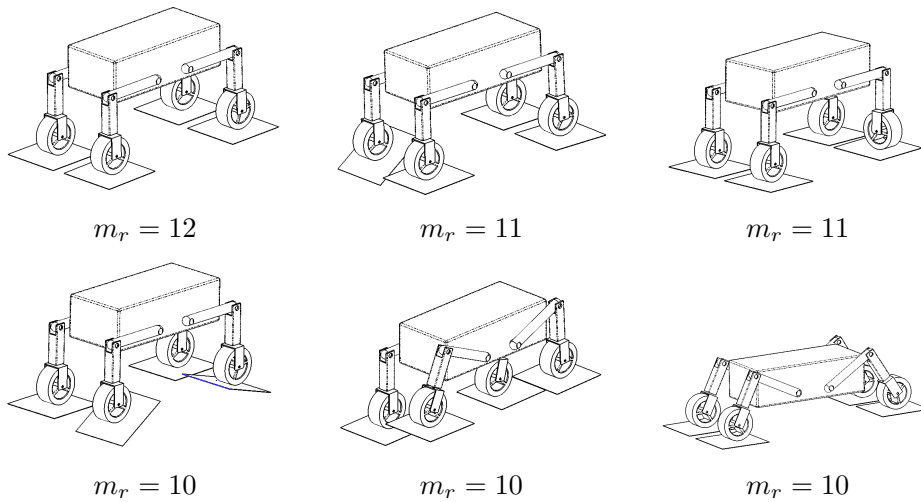


Fig. 7. Mobility for some cases as function of contact planes and of configurations

3.3 Velocity space reduction

As said in the previous section, the velocity parameter of the steering axis is independent of other velocity parameters when the steering axis is collinear to the contact normal. This configuration introduces a singularity in the Jacobian matrix as the steering axis passes through the contact point. For other configurations (mainly depending of the caster angle between the steering axis and the contact normal), the column of \mathbf{J}_i , of equation (13), associated to the steering rate $\dot{\gamma}_i$ is almost null which leads to an ill-conditioned matrix. Furthermore, the called caster angle (Fig. 8) must be as small as possible in order to keep the contact area on the rolling tread of the cylindrical wheel during steering. Then, this column of the jacobian matrix and the associated time-derivative rate will be removed in the following development. In parallel to this, we will split velocity equations in two groups:

- The first one corresponds to other kinematic constraint i.e. longitudinal non-slippage condition $\mathbf{t}_i^t \mathbf{v}_s^i = 0$ and permanent contact condition $\mathbf{n}_i^t \mathbf{v}_s^i = 0$,
- The second one corresponds to lateral slippage constraints $\mathbf{I}_i^t \mathbf{v}_s^i = 0$.

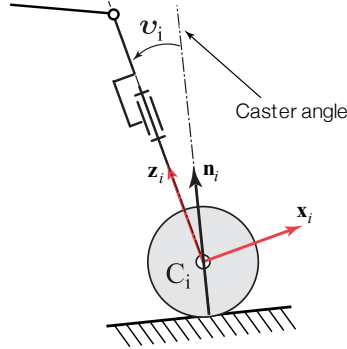


Fig. 8. Wheel caster angle

The first group can be written as:

$$\mathbf{S}_x \mathbf{L} \dot{\mathbf{x}} + (\mathbf{S}_x \mathbf{J} \mathbf{S}_u^t) (\mathbf{S}_u \dot{\boldsymbol{\Theta}}) = 0 \quad (25)$$

where \mathbf{S}_x is a reduction matrix selecting equations along the \mathbf{t}_i and \mathbf{n}_i axis:

$$\mathbf{S}_x = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & & & \mathbf{0} \\ & \ddots & & \\ & & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \\ \mathbf{0} & & & \end{bmatrix}_{8 \times 12}$$

and \mathbf{S}_u a selection matrix eliminating γ_i parameters and the associated column in the jacobian matrix \mathbf{J} :

$$\mathbf{S}_u = \begin{bmatrix} \begin{bmatrix} \mathbf{I}_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & & & \mathbf{0} \\ & \ddots & & \\ & & \begin{bmatrix} \mathbf{I}_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \\ \mathbf{0} & & & \end{bmatrix}_{12 \times 16}$$

where \mathbf{I}_k is the $k \times k$ identity matrix.

The second group can be written as:

$$\mathbf{S}_\gamma \mathbf{L} \dot{\mathbf{x}} + (\mathbf{S}_\gamma \mathbf{J} \mathbf{S}_u^t) (\mathbf{S}_u \dot{\boldsymbol{\Theta}}) = 0 \quad (26)$$

where \mathbf{S}_γ is a reduction matrix selecting equations along the \mathbf{l}_i axis:

$$\mathbf{S}_\gamma = \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} & \mathbf{0}_{3 \times 1} \\ \vdots & \vdots \\ \mathbf{0}_{3 \times 1} & \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix}_{4 \times 12}$$

This separation allows the resolution of the inverse differential kinematic problem. First, we will solve the first group by computing the reduced command vector $\mathbf{u} = \mathbf{S}_u \dot{\boldsymbol{\Theta}} = (\dot{\alpha}_1, \dot{\beta}_1, \omega_1, \dots, \dot{\alpha}_4, \dot{\beta}_4, \omega_4)$ for a given desired twist components of the platform. Then, in a second phase, we will compute the steering angle γ_i for each wheel that provides the desired motion direction of the platform.

3.4 Computing joint rates

The 12 joint velocities $\mathbf{u} = (\dot{\alpha}_i, \dot{\beta}_i, \omega_i, \dots)$ are computed by using the reduced velocity model on equation (25), for a given desired platform velocity $\dot{\mathbf{x}}$. A basic kinematic analysis shows that the system is redundant as there is 12 joints velocities and 8 independent velocity equations. So, there is an infinite set of solutions for \mathbf{u} that produces a desired motion $\dot{\mathbf{x}}$. Thus, we propose to define a new operational vector $\boldsymbol{\xi}^t = (\mathbf{x}^t, \mathbf{e}^t)$ of dimension 10 based on the 6 platform parameters and 4 new internal parameters $\mathbf{e} = (e_1, e_2, e_3, e_4)^t$ which are the half wheelbases. They are defined for each wheel-leg as the position of the wheel center with respect to the platform frame and projected along the longitudinal axis of the platform \mathbf{x}_p : $e_i = \mathbf{x}_p^t \mathbf{r}_i$ (Fig. 9).

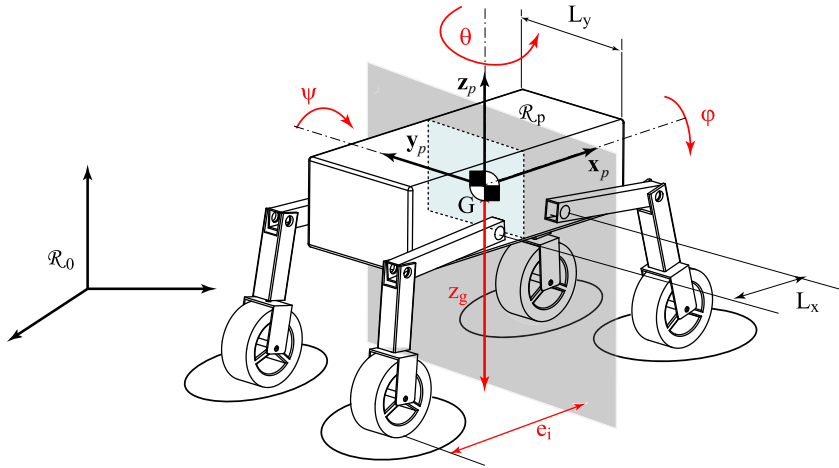


Fig. 9. The 7 posture parameters: 3 for the platform and 4 additional half-wheelbases parameters

By writing the differential equations for each kinematic wheel-leg chain, $\dot{\mathbf{e}}$ could be written as function of the command vector \mathbf{u} . We denote $\mathbf{u}_i = (\dot{\alpha}_i, \dot{\beta}_i, \omega_i)^t$ the part of \mathbf{u} corresponding to the i th wheel-leg, and we obtain:

$$\dot{\mathbf{e}}_i = \mathbf{J}_e^i \begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \\ \omega_i \end{bmatrix} = \mathbf{J}_e^i \mathbf{u}_i \quad (27)$$

This equation expresses the motion of the wheel contact points along \mathbf{x}_p axis. It is function only of legs joint rates (without the wheel rate). It could be simply deduced from the wheel-leg Jacobian matrix \mathbf{J}_θ^i defined in equation (21) by considering its projection along \mathbf{x}_p :

$$\mathbf{J}_e^i = [L_1 \sin \alpha_i + L_2 \sin(\alpha_i + \beta_i) \quad L_2 \sin(\alpha_i + \beta_i) \quad 0]_{1 \times 3}$$

Then $\dot{\mathbf{e}}$ is written in matrix form as function of \mathbf{u} :

$$\dot{\mathbf{e}} = \mathbf{J}_e(\Theta)\mathbf{u} \quad \text{with} \quad \mathbf{J}_e(\Theta) = \begin{bmatrix} \mathbf{J}_e^1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_e^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_e^3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_e^4 \end{bmatrix}_{4 \times 12} \quad (28)$$

Thus, equations (25) and (28) can be concatenated in matrix form to obtain the following state velocity equation:

$$\begin{bmatrix} \mathbf{S}_x \mathbf{L} & \mathbf{0}_{8 \times 4} \\ \mathbf{0}_{4 \times 6} & \mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} + \begin{bmatrix} \mathbf{S}_x \mathbf{J} \mathbf{S}_u^t \\ \mathbf{J}_e \end{bmatrix} \mathbf{u} = \mathbf{0} \quad (29)$$

And finally this equation is rewritten as:

$$\tilde{\mathbf{L}} \dot{\boldsymbol{\xi}} + \tilde{\mathbf{J}} \mathbf{u} = \mathbf{0} \quad (30)$$

with

$$\tilde{\mathbf{L}} = \begin{bmatrix} \mathbf{S}_x \mathbf{L} & \mathbf{0}_{8 \times 4} \\ \mathbf{0}_{4 \times 6} & \mathbf{I}_4 \end{bmatrix}_{12 \times 10} \quad \text{and} \quad \tilde{\mathbf{J}} = \begin{bmatrix} \mathbf{S}_x \mathbf{J} \mathbf{S}_u^t \\ \mathbf{J}_e \end{bmatrix}_{12 \times 12}$$

$\tilde{\mathbf{J}}$ is a regular square 12x12 matrix (except some singular cases not studied in this paper), then:

$$\mathbf{u} = - \left(\tilde{\mathbf{J}}^{-1} \tilde{\mathbf{L}} \right) \dot{\boldsymbol{\xi}} \quad (31)$$

3.5 Computing the steering angles

Each equation of the second group (26) could be solved separately in order to give the steering angles which are compatibles with the desired platform velocity $\dot{\mathbf{x}}$ and the other internal velocities $\dot{\boldsymbol{\theta}}$ computed in the last section. Assuming that the lateral contact vector \mathbf{l}_i is collinear to the wheel axis (i.e. no camber angle), we show easily from equation (26) that:

$$\tan \gamma_i = \frac{\nu_i}{\mu_i} \quad (32)$$

with

$$\begin{aligned} \nu_i &= v_y + \omega_z x_i - \omega_x z_i \\ \mu_i &= (v_x + \omega_y z_i - \omega_z y_i + \dot{x}_i) \sin(\alpha_i + \beta_i) + \dots \\ &\quad (v_z + \omega_x y_i - \omega_y x_i + \dot{z}_i) \cos(\alpha_i + \beta_i) \end{aligned} \quad (33)$$

and $\mathbf{v} = (v_x, v_y, v_z)^\top$ et $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^\top$ are platform twist parameters in the local frame, then $\mathbf{v} = \mathbf{R}^\top \dot{\boldsymbol{\rho}}$, $\boldsymbol{\omega} = \mathbf{T}_\phi \dot{\boldsymbol{\phi}}$.

For a classical wheeled system moving on a plane, the steering angle is related directly to the lateral velocity of the platform v_y and its yaw rate ω_z . This is observed in the numerator of the later equation. However, the term $-\omega_x z_i$ is not usual. In fact, roll platform reconfiguration $-\omega_x$ needs a roll motion with a non-null steering angle. This motion coupling is illustrated in Fig. 10.

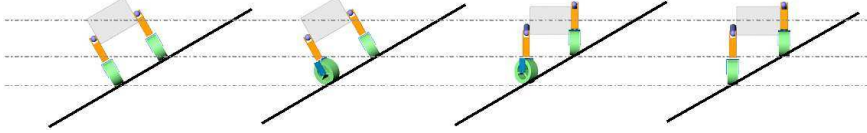


Fig. 10. Rolling motion needs steering the wheels to prevent side-slippage

4 Posture control of Hylos based on velocity model

4.1 Posture definition

As it was previously demonstrated, when assuming that lateral non-slippage conditions are satisfied by controlling the suitable steering angles at each contact, the system has 10 independent velocity parameters. There are 8 velocity equations (25) for 18 velocity parameters (6 for the platform and 12 for joints including legs and wheel's roll). In the operational space, the 10 parameters correspond to 3 parameters (x, y, θ) of the platform horizontal trajectory and

7 posture parameters. These posture parameters correspond to 3 platform attitude parameters (z, ϕ, ψ) and 4 internal parameters (e_1, e_2, e_3, e_4) defined previously by the half wheelbases of each contact.

The posture parameters vector $\boldsymbol{\pi}$ is defined as:

$$\boldsymbol{\pi} = (\varphi, \psi, z_g, e_1, e_2, e_3, e_4)^t$$

where $z_g = z - z_s$ is the rover elevation relative to the local terrain altitude (z_s), thus z_g can also be defined as the rover ground clearance. Furthermore, z_s is considered to be locally constant which means that $\dot{z}_s \approx 0$.

4.2 Posture optimisation

The problem of posture optimization could be treated by considering various performance criteria as stability, traction, energy consumption... These criterion could be expressed as function of contact force components (normal component and tangential ones). However, it is difficult to estimate contact normal vectors and thus to carry out a real-time efficient optimization of contact force distribution. Moreover, the adaptation of the posture as function of the local contact normals could develop excessive energy consumption without a real increase of the global performance. Furthermore, this posture adaptation has some response delay and could be efficient only for very slow robot motion.

One obvious posture vector could be defined by a constant attitude of the platform (zero pitch and roll angles and a nominal ground clearance z_g) and a constant nominal wheelbases e_n :

$$\boldsymbol{\pi}_n = (0, 0, z_g, \frac{e_n}{2}, \frac{e_n}{2}, -\frac{e_n}{2}, -\frac{e_n}{2})^t$$

This posture is a good compromise: it preserves stability, ground clearance and force transmission from actuators to contacts. Furthermore, zero attitude angles of the platform provide stabilized video with horizon captured by the embedded vision system.

4.3 Posture configuration on a sloping terrains

Another way to define a global performance criteria for the desired posture is to consider what we call the equal-distribution of contact vertical loads. We define an average contact plane computed from the four contact points which

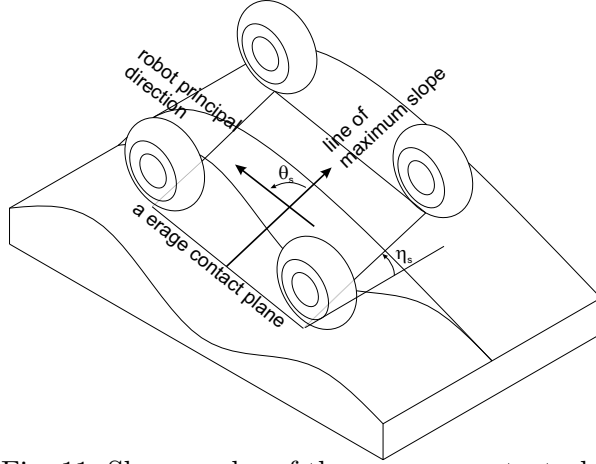


Fig. 11. Slope angles of the average contact plane

are estimated quite easily from joint encoders and a 2-axes inclinometer sensor. The problem is then reduced to finding an optimal posture on a sloping ground defined by two parameters: η_s the maximum slope angle and θ_s the robot yaw angle relative to the slope direction (Fig. 11).

As the gauge of the robot (lateral distance between wheels) remains constant, an equal-distribution of loads between left and right contacts requires a zero roll angle of the platform (Fig. 11). However, the system is redundant in its sagittal plane causing the equal load distribution between front and rear contacts to have an infinite set of solutions. We notice that the solution of zero pitch angle set up the joint legs near their singularities or their joint limits, whereas assessing the platform longitudinal axis parallel to the slope gives better solutions of joint leg configuration close the nominal one.

Assuming this constraint, the horizontal wheelbases must be equal in order to give the same load between front and left. Another constraint raised by the steering system is that the steering axis of the steered wheels should be normal to the plane, in order to keep the contact on the rolling tread of the wheel. We denotes ψ' the slope angle along the robot longitudinal direction, which could be expressed by:

$$\psi' = \arcsin(\cos(\theta_s) \sin(\eta_s)) \quad (34)$$

Angles η_s, θ_s are estimated from the direct kinematic model and from measured roll and pitch angle of the platform. By choosing a desired ground clearance z_g and a nominal wheelbase e_n , the desired posture which gives equal vertical load distribution is:

$$\begin{aligned} \boldsymbol{\pi}_s = & (0, \psi', z_g, \frac{e_n}{2} - |z_g| \tan \psi', \frac{e_n}{2} - |z_g| \tan \psi', \\ & -\frac{e_n}{2} + |z_g| \tan \psi', -\frac{e_n}{2} + |z_g| \tan \psi')^t \end{aligned} \quad (35)$$

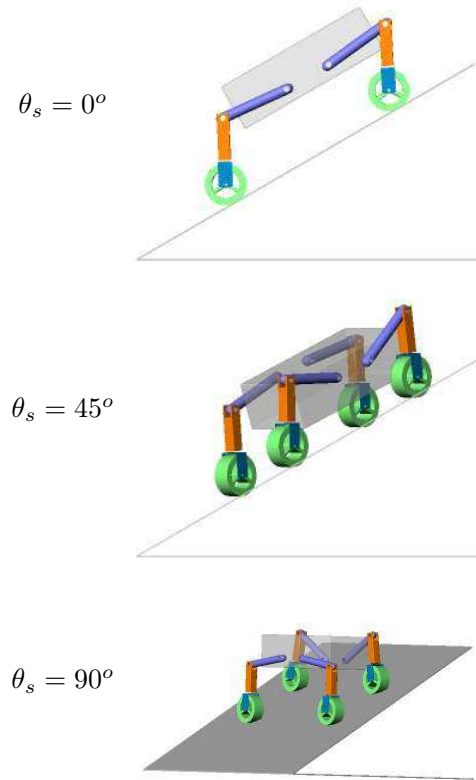


Fig. 12. Posture of equal distribution of contact loads for different configurations on a slope

Those postures are represented in Fig. 12 for a slope with a maximal angle of $\eta_s = 30^\circ$ and for different values of the yaw angle $\theta_s = \{0^\circ, 45^\circ, 90^\circ\}$.

Those postures could be corrected in order to decrease the caster angle of the steering axis (relatively the slope normal), as represented in Fig. 13 where only the front steering wheels would be used to steer the robot. This is to ensure that the contact remains on the rolling tread of the tyre for high steering angle value.

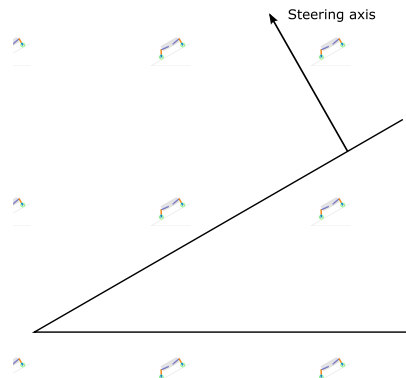


Fig. 13. Posture correction as function of the manoeuvrability constraint

4.4 Posture control

When the robot crosses an irregular surface, it must maintain its posture around a desired posture $\boldsymbol{\pi}_d$. We use a classical feedback linearization control law for the posture control:

$$\dot{\boldsymbol{\pi}} = \mathbf{K}_\pi (\boldsymbol{\pi}_d - \boldsymbol{\pi}) \quad (36)$$

where \mathbf{K}_π is a diagonal positive matrix.

Let us remind that as the ground elevation is considered to be locally constant $\dot{z}_s \approx 0$. So, we have the following relation $\dot{z}_g = \dot{z}$ and the posture variation is:

$$\dot{\boldsymbol{\pi}} = (\dot{\varphi}, \dot{\psi}, \dot{z}, \dot{e}_1, \dot{e}_2, \dot{e}_3, \dot{e}_4)^t$$

Then, we can compute the platform posture velocity by using the following equations:

$$\begin{cases} v_z = -\dot{z} + \omega_y \frac{\sum_i x_i}{4} - \omega_x \frac{\sum_i y_i}{4} \simeq -\dot{z} \\ \omega_x = \dot{\varphi} - \dot{\theta} \sin \psi \simeq \dot{\varphi} \\ \omega_y = \dot{\psi} \cos \varphi + \dot{\theta} \cos \psi \sin \varphi \simeq \dot{\psi} \cos \varphi \end{cases}$$

The first equation assumes that the projection of the contact center on the horizontal plane is closer to the one of the platform center. In the two other equations, we neglect the effect of yaw velocity $\dot{\theta}$. Those posture parameters and the other velocities parameters $(v_x, v_y, \dot{\theta})^t$ (given by path tracking control) are used in the inverse velocity model which has to compute from equations (31, 32) the actuator velocity inputs (except for steering actuators, which have position low-level control). One must notice that those equations require the knowledge of normals \mathbf{n}_i at each contact. The equation (11) shows that tangential vector \mathbf{t}_i can be determined from the measure of the system velocity parameters, including platform parameters and joint ones $(\dot{\alpha}_i, \dot{\beta}_i)$. However, measuring the 3 components of the instantaneous linear velocity is not simple. In our experiment, we make an estimation of contact normals from the average contact plane.

The Fig. 14 describes the general schema of the control system. This controller, based on the inverse velocity model, is divided in two parts. The main idea, is to be able to decouple the posture control and the trajectory control.

First, the posture controller, as presented previously, computes the reconfiguration velocities $\dot{\boldsymbol{\pi}}$ in order to reach a desired posture $\boldsymbol{\pi}_d$, using measurements from inclinometers and joint sensors. The current posture $\boldsymbol{\pi}$ is obtained from this measures by solving the direct kinematic model.

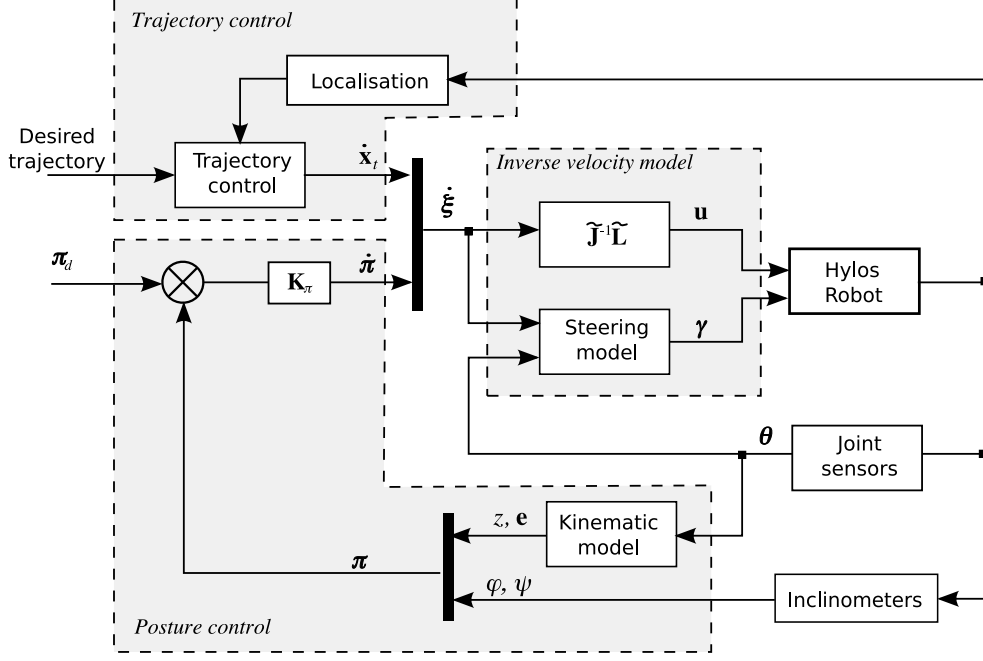


Fig. 14. General control schema

Secondly, the trajectory controller aims to compute the following platform velocities components $\dot{\mathbf{x}}_t = (\dot{x}, \dot{y}, \dot{\theta})^t$. They correspond to the horizontal velocities of the robot projected in the reference frame \mathcal{R}_0 and its yaw rate. These velocities can be computed using a trajectory controller. This part is not detailed in this paper. For the evaluation of the posture control which is presented in the next paragraph, we will use a simple open-loop control of the trajectory. The platform velocities $\dot{\mathbf{x}}_t$ are directly computed by differentiating the trajectory input $\mathbf{x}_t(t)$ with respect to time.

Then, these two control velocities vectors ($\dot{\pi}$ and $\dot{\mathbf{x}}_t$) are combined to form the state velocity vector $\dot{\xi}$ which is then used to solve, through the inverse velocity model, the joint velocities applied to the robot. The steering angles γ are then obtained by solving the non-holonomic constraints (32).

Furthermore, our model supposes that wheel-soil contacts are maintained continuously. This function is guaranteed by adding a correction term in the control law:

$$s_n^i = K_f(f_n^i - f_0^i) \quad (37)$$

where s_n^i is the contact detachment velocity introduced in equation (12), f_n^i is contact normal force measured by means of 3 axis force sensor which is integrated on each leg, f_0^i is the reference value of the i th wheel contact force along normal axis and K_f is a positive matrix. The set of force reference values of all wheels f_0^i are computed from the force balance equation (see Eq. (15)).

4.5 Evaluation of the posture control algorithm

In the experiment, the robot moves straightforward at a speed of 0.08 m/s with heading angle $\theta = 0^\circ$ on an asymmetric irregular terrain. The Fig. 15 shows the evolution of the rover on an irregular asymmetrical ground profile, with a posture reconfiguration. The rover succeeds the crossing of this ground surface with very good stability margin, whereas it was not able to cross this surface in pure rolling mode (without reconfiguring legs) since a lateral tip-over has been noticed.

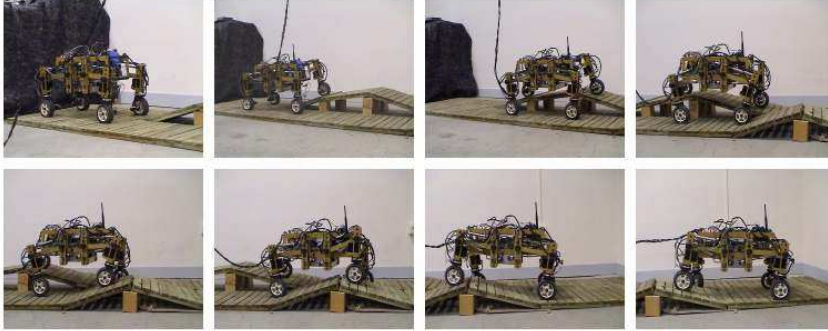


Fig. 15. Hylas crossing an asymmetrical terrain profile with a constant absolute posture

The desired posture consists in keeping the platform horizontal with a null pitch and roll angles ($\phi = \psi = 0$) and in maintaining constant the ground clearance z_g and the half wheelbase e_i . The measured pitch and roll angles are plotted in Fig. 16. The maximum deviation of pitch and roll errors are respectively 3° and 4° using the posture control law. So, these experiments show the ability of the control algorithm to maintain a certain posture with a small error on pitch and roll angles when evolving on an unknown irregular terrain.

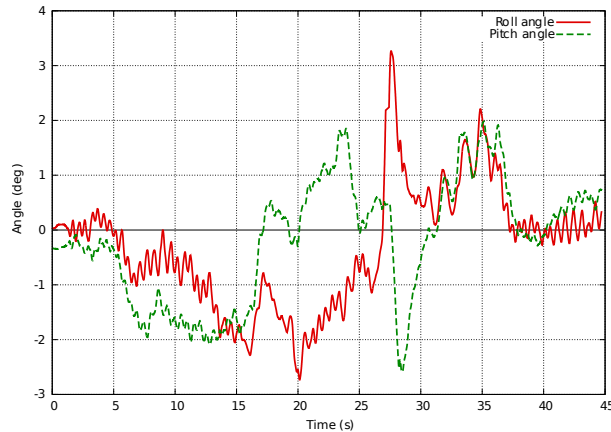


Fig. 16. Evolution of the controlled pitch and roll angles of the rover evolving on irregular terrain

The applicability of the posture control algorithm has also been evaluated for a generic path on uneven terrain (not only a straightforward path). This evaluation was done using dynamics simulation. The Fig. 17 shows the evolution of the robot when is following a complex path.

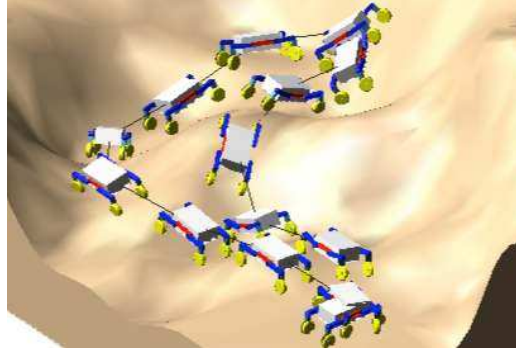


Fig. 17. Simulation of the rover evolving on a irregular terrain and following a curved path

In this simulation, the robot trajectory is controlled through a simple open loop controller. The desired vehicle velocity $(\dot{x}, \dot{y}, \dot{\theta})^t$ are computed from the given trajectory and applied to the system in coordination with the posture controller. In this case, this velocity term $(\dot{x}, \dot{y}, \dot{\theta})^t$ is combined with the posture reconfiguration velocity term $\dot{\pi} = (\dot{\varphi}, \dot{\psi}, \dot{z}, \dot{e}_1, \dot{e}_2, \dot{e}_3, \dot{e}_4)^t$ to form the full state velocity $\dot{\xi}$. Then, the equations (31) and (32) are used to compute the joint velocity \mathbf{u} and the steering angles γ_i .

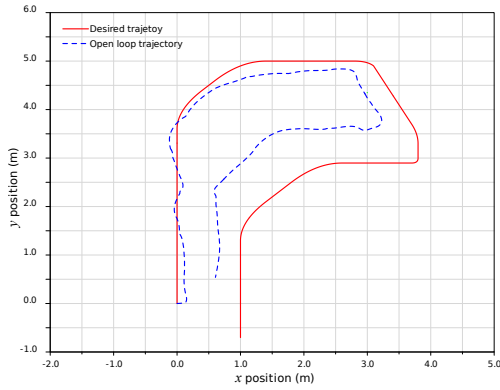


Fig. 18. Motion trajectory

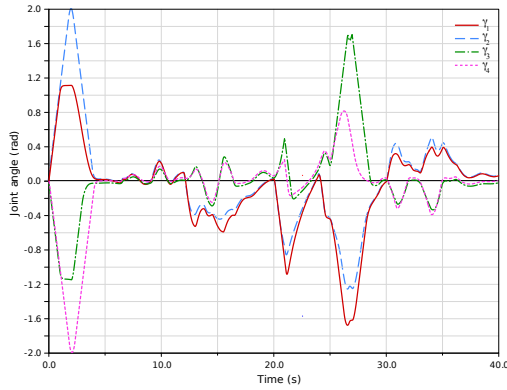


Fig. 19. Evolution of the steering angles

The desired trajectory and the real one followed by the robot are represented in the Fig. 18, while the evolution of the steering angle of each wheel-leg is depicted in the Fig. 19. These results show that the inverse differential kinematic model proposed in this paper allows to control simultaneously both the posture parameters of the robot and its horizontal position $\mathbf{x}_t = (x, y, \theta)^t$, in spite of the drift noted on the trajectory really followed by the robot.

In this simulation, the trajectory is driven with an open-loop control. The corresponding velocities are computed by using the derivative of the reference

trajectory $\mathbf{x}_t(t)$. Since we use a simple open-loop control, the real trajectory deviates from the reference trajectory. This error can be reduced by considering a more sophisticated close-loop trajectory control. In spite of this point, these results show the ability of our control schema to perform a control of the robot posture while evolving on irregular terrain at a specified velocity.

5 Conclusion

This paper develops an original kineto-static formalism which characterizes the motion of a wheeled-legged robot on 3D ground surface. The model is used to control the posture of the robot and its static stability. This model has been evaluated with dynamic simulations and with experiments on a physical prototype. Results show the validity of the model and the feasibility of this approach. The model is a generic one and could be used for any articulated rover with active suspensions. It assumes continuous contact with the ground and not necessarily pure rolling. Slippage velocities can be introduced easily in the model if they are correctly estimated. This velocity model can be used also for internal or external parameters estimation such as contact normals, 6D platform trajectory. Future works will consider the integration of more sophisticated close-loop control for the trajectory tracking.

References

- [1] R. Volpe, Rocky 7: A next generation mars rover prototype, *Journal of Advanced Robotics* 11 (4) (1997) 341–358.
- [2] R. Siegwart, P.Lamon, T. Estier, M. Lauria, R. Piguet, Innovative design for wheeled locomotion in rough terrain, *Robotics and Autonomous Systems* 40 (2002) 151–162.
- [3] E. Rollins, J. Luntz, A. Foessel, B. Shamah, W. Whittaker, Nomad: A demonstration of the transforming chassis, in: *Proc. of the Int. Conf. on Intelligent Components for Vehicles*, 1998.
- [4] K. Iagnemma, K. Rzepniewski, S. Dubowsky, P. Schenker, Control of robotic vehicles with actively articulated suspensions in rough terrain, *Autonomous Robots* 14 (1).
- [5] S. Sreenivasan, B. Wilcox, Stability and traction control of an actively actuated microrover, *Journal of Robotics Systems* 11 (6) (1994) 487–502.
- [6] G. Andrade, F. BenAmar, Ph.. Bidaud, R. Chatila, Modeling wheel-sand interaction for optimization of a rolling-peristaltic motion of a marsokhod robot,

in: Proc. of IROS : IEEE/Int. Conf. on Intelligent Robots and Systems, 1998, pp. 576–581.

- [7] F. Michaud, D. Letourneau, M. Arsenault, Y. Bergeron, R. Cadrin, F. Gagnon, M.-A. Legault, M. Millette, J.-F. Pare, M.-C. Tremblay, P. Lepage, Y. Morin, J. Bisson, S. Caron, Multi-modal locomotion robotic platform using leg-track-wheel articulations, *Autonomous Robots* 18 (2) (2005) 137–156.
- [8] C. Grand, F. BenAmar, F. Plumet, P. Bidaud, Stability and traction optimisation of high mobility rover, *The International Journal of Robotics Research* 23 (10-11) (2004) 1041–1058.
- [9] A. Halme, S. Ylönen, Further development and testing of the hybrid locomotion of workpartner robot, in: Proc. of CLAWAR : Int. Conf. on Climbing and Walking Machine, 2002.
- [10] K. Hauser, T. Bretl, J.-C. Latombe, B. Wilcox, Motion planning for a six-legged lunar robot, in: Proc. of the The Seventh Int. Workshop on the Algorithmic Foundations of Robotics, 2006.
- [11] P. Muir, C. Neuman, Kinematic modeling of wheeled mobile robots, *Journal of robotics systems* 4 (2) (1987) 281–340.
- [12] J. Alexander, J. Maddocks, On the kinematics of wheeled mobile robot, *Int. Journal of Robotics Research* 8 (5) (1989) 15–27.
- [13] G. Campion, G. Bastin, B. d’Andréa-Novel, Structural properties and classification of kinematic and dynamic models of wheeled mobile robots, *IEEE Trans. on Robotics and Automation* 12 (1) (1996) 47–62.
- [14] S. Sreenivasan, K. Waldron, Displacement analysis of an actively articulated wheeled vehicle configuration with extensions to motion planning on uneven terrain, *Transactions of the ASME* 118 (6) (1996) 312–317.
- [15] B. Choi, S. Sreenivasan, Gross motion characteristics of articulated mobile robots with pure rolling capability on smooth uneven surfaces, *IEEE Trans. on Robotics and Automation* 15 (2) (1999) 340–343.
- [16] N. Chakraborty, A. Ghosal, Kinematics of wheeled mobile robots on uneven terrain, *Mechanism and Machine Theory* 39 (2004) 1273–1287.
- [17] M. Tarokh, G. J. McDermott, Kinematics modeling and analyses of articulated rovers, *IEEE Trans. on Robotics and Automation* 21 (4) (2005) 539–554.
- [18] K. Waldron, Force and motion management in legged locomotion, *IEEE Journal of Robotics and Automation* 2 (4) (1986) 214–220.
- [19] H. Pacejka, *Tyre and Vehicle Dynamics*, Elsevier, 2002.
- [20] M. Bekker, *Introduction to terrain-vehicle systems*, The University of Michigan Press, 1969.

A Notations

Frames definition	
$\mathcal{R}_0 = (O, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$	Reference ground frame
$\mathcal{R}_p = (G, \mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$	Main platform frame
$\mathcal{R}_c^i = (P_i, \mathbf{t}_i, \mathbf{l}_i, \mathbf{n}_i)$	Contact frame of the i th wheel
$\mathcal{R}_l^i = (C_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$	Frame attached to the end of the i th leg mechanism
$\mathcal{R}_\omega^i = (C_i, \mathbf{x}_\omega^i, \mathbf{y}_\omega^i, \mathbf{z}_\omega^i)$	Frame attached to the i th wheel

Kinematic parameters	
$\boldsymbol{\rho}^t = (x, y, z)$	position of the platform frame with respect to (w.r.t.) the ground
$\boldsymbol{\phi}^t = (\varphi, \psi, \theta)$	orientation angles of the platform frame w.r.t. the ground (roll, pitch and yaw angles, respectively)
$\boldsymbol{\theta}_i$	vector of the i th leg joint parameters
$\boldsymbol{\chi}_i^t = (\gamma_i, \vartheta_i)$	vector of the i th wheel's parameters
$\mathbf{x}^t = (\boldsymbol{\rho}^t, \boldsymbol{\phi}^t)$	vector of the platform parameters
$\boldsymbol{\Theta}_i^t = (\boldsymbol{\theta}_i^t, \boldsymbol{\chi}_i^t)$	vector of joint parameters of the i th wheel-leg
$\mathbf{q}^t = (\mathbf{x}^t, \boldsymbol{\theta}_i^t, \boldsymbol{\chi}_i^t)$	aggregated vector of the robot parameters
γ_i	steering angle of the i th wheel
ϑ_i	rolling angle of the i th wheel
ν_i	caster angle of the i th wheel
r_i	radius of the i th wheel
\mathbf{p}_i	position of the contact point of the i th wheel w.r.t. the platform frame
$\mathbf{R} = \mathbf{R}(\varphi, \psi, \theta)$	rotation matrix of the platform frame w.r.t. the ground

Velocity model	
$\dot{\rho}$	velocity of the center of the platform expressed in the ground frame
ω	rotation rate vector of the platform expressed in the platform frame
ω_i	spinning rotation rate of the i th wheel $\omega_i = \dot{\vartheta}_i$
\mathbf{T}_ϕ	coupling matrix between ω and $\dot{\phi}$
\mathbf{L}_i	locomotion matrix associated to the motion of the i th wheel-leg
\mathbf{J}_θ^i	jacobian matrix of the i th leg
\mathbf{v}_s^i	vector of the relative velocity in the i th contact
s_t^i	longitudinal slippage of the i th wheel
s_l^i	lateral slippage of the i th wheel
s_n^i	contact deformation velocity in the normal direction
\mathbf{L}	locomotion matrix of the full system
\mathbf{J}	jacobian matrix of the full system
Θ	aggregated vector of the wheel-legs parameters
\mathbf{V}_s	aggregated vector of the wheel-soil slippage parameters

Static model	
\mathbf{f}_i	vector of the i th wheel contact force expressed in \mathcal{R}_c^i
\mathbf{f}	aggregated vector of all wheel contact forces
\mathbf{w}	total wrench vector of all external forces applied to the system
$\boldsymbol{\tau}$	vector of actuator torques applied on wheel-leg joints

Hylos specific notations	
α_i, β_i	joint angles of the i th leg of the robot Hylos
L_1, L_2	length of each leg's links of the robot Hylos
L_x, L_y	geometric parameters of the main platform of Hylos

Inverse velocity model	
e_i	half wheelbase of the i th wheel-leg
\mathbf{e}	vector of the all half wheelbases
\mathbf{J}_e^i	jacobian matrix associated to the i th wheelbase rate \dot{e}_i
\mathbf{J}_e	jacobian matrix associated to the vector of wheelbase rates $\dot{\mathbf{e}}$
$\boldsymbol{\xi}$	vector of operational parameters $\boldsymbol{\xi}^t = (\mathbf{x}^t, \mathbf{e}^t)$
\mathbf{u}	control input vector of the controller
$\mathbf{S}_x, \mathbf{S}_u, \mathbf{S}_\gamma$	selection matrices
$\tilde{\mathbf{L}}$	augmented locomotion matrix of the controllable system
$\tilde{\mathbf{J}}$	augmented jacobian matrix of the controllable system
z_g	rover ground clearance
$\boldsymbol{\pi}$	vector of the posture parameters $\boldsymbol{\pi} = (\varphi, \psi, z_g, e_1, e_2, e_3, e_4)^t$
\mathbf{x}_t	vector of the platform position parameters used in trajectory control $\mathbf{x}_t = (x, y, \theta)^t$