Robotica (2005) volume 23, pp. 771–780. © 2005 Cambridge University Press
doi:10.1017/S0263574705001712 Printed in the United Kingdom

An effective approach for dynamic analysis of rovers

A. Meghdari*, R. Karimi, H. N. Pishkenari, A. L. Gaskarimahalle and S. H. Mahboobi

Center of Excellence in Design, Robotics and Automation (CEDRA), School of Mechanical Engineering, Sharif University of Technology (Iran)

(Received in Final Form: February 8, 2005)

SUMMARY

In this paper a novel approach to dynamic formulation of rovers has been presented. The complexity of these multi-body systems especially on rough terrain, challenged us to use the Kane's method which has been preferred to others in these cases. As an example, symbolic equations of a six-wheeled rover, named CEDRA Rescue Robot which uses a shrimp like mechanism, have been derived and a simulation of forward and inverse dynamics has been presented. Due to the clear form of equations, each term defines a physical meaning which represents the effect of each parameter, resulting in a frame-work for performance comparison of rovers. Although the method has been described for a 2-D non-slipping case, it is also very useful for dimensional and dynamical optimization, high speed motion analysis, and checking various control algorithms. Furthermore, it can be extended to 3-D cases and other complicated mechanisms and rovers while conserving its inherent benefits and adding to the ease of handling nonholonomic constraints.

KEYWORDS: Rovers; Dynamic analysis; Kane's method.

I. INTRODUCTION

Various cases, such as space explorations and rescue operations, will require high mobility robots to perform intricate tasks in challenging uneven terrain. Design and control of these robots are based on their equations of motion; consequently the simpler and the more meaningful the equations of motion are, the better we can use them for purposes like optimization and checking control algorithms. Generally, the equations of motion of rovers are very complicated. This complication arises from three factors: complicated mechanism, uneven terrain and nonholonomic constraints. The movement of rovers is based on various techniques like wheeled locomotion and legged locomotion. Wheeled locomotion is the most commonly used locomotion system and probably the most studied and advanced.

One of the problems encountered in wheeled locomotion is following ground contour. Using multiple wheels improves traction and stability; suspension system and linkages keep ground contact and improve climbing ability over obstacles larger than the wheel radius. To design and control these systems, analytical modeling of the rover behavior interacting with its environment is essential.[1]

Although lots of research has dealt with the case of flat surfaces they rarely consider the dynamic analysis for rough terrain, and most of the recent research in outdoor operations have discussed only simple mechanisms. Modeling of articulated mobile robots has been studied by numerous researchers. General kinematic analysis has been studied by Sreevinasan et al.[1] Chottiner[2] and Linderman et al.[3] have studied Kinematic of six-wheeled rocker-bogie rovers such as the JPL Sojourner rover. Mechanical models of this configuration including methods for solving the inverse kinematics of the system and quasi-static force analysis have been proposed by Hacot.[4] Tarokh et al.[5] have conducted research on the direct and inverse kinematics of Rocky-7 using Denavit-Hartenberg algorithm. Force analysis of mobile robots has also been performed, which is similar to the force distribution problem in closed kinematic chains and walking machines.[6] Also dynamic modeling of a wheeled mobile robot with a suspension has been considered by Tai.[7] This approach is novel and no exact dynamical equation has been derived in the latest research, thus it can not be implemented on the control unit. Generally, no one has presented a closed form for a multi-wheeled rover without simplifying the problem.

To deal with these problems all together, we first assumed that our surface is parameterized appropriately, and then we used Kane's method to extract the minimum number of acceleration equations. Particularly the great difference between the number of generalized velocity and the number of generalized coordinates demands the use of this method. Furthermore, it is of great interest that we can utilize these meaningful equations for a reasonable definition of the optimization goal. In our previous research,[8,9] the optimization objective had been to find the rover parameters so that the path traversed by the center of gravity tends to the straight line. Now we can compare the results of this simpler optimization goal, which did not consider the dynamical aspects of the motion, with the real one. Also, the derived equations are used to check various control strategies like velocity or force control via inverse or forward dynamics.

In this paper we have concentrated our analysis on the CEDRA rescue robot (Fig. 1). The main structure is based on the shrimp mechanism. The idea of using a shrimp like mechanism in climbing the obstacles was first demonstrated in EPFL.[10] This Robot is similar to Rocky-7,[11] Sojourner[12]

* Corresponding author. E-mail: meghdari@sharif.edu

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

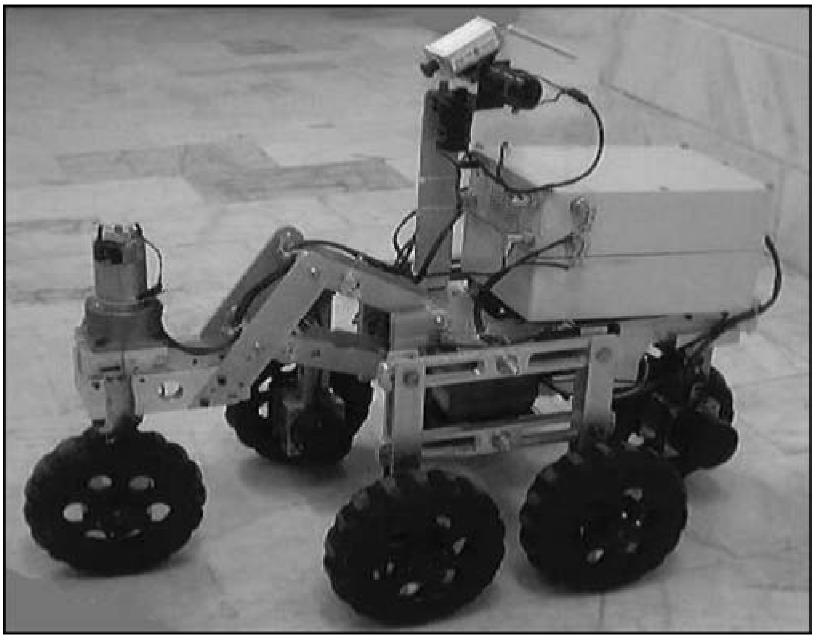


Fig. 1. CEDRA rescue robot.

and Marsokhod[13] in some parts, but adding a four-link mechanism in front of the robot has made it more efficient encountering obstacles. On the other hand, setting wheels with controllable speed and a steering system that adjusts the angle of front and back wheels makes it possible for the robot to turn in place and maneuver with high accuracy in confined areas.

II. KANE'S METHOD

Kane's method is the most recent approach to the dynamic analysis which was developed in the 1980's.[14] This method has demonstrated conspicuous predominance over others like Newton and Lagrange in complex problems. In fact, the efficacy of this approach is related to the lower number of equations, closed forms of equations, ease of deriving the constraint forces and better implementation of the numerical solutions especially in dealing with the multiplicity of masses in the system. It is also widely used in nonholonomic problems, since the Lagrange method can not handle nonholonomic constraints easily. Dynamic modeling and control of mobile manipulators using this method has been studied by Nukulwuthiopas et al.[15] They have utilized Kane's approach in developing the internal forces and compared it with the Newtonian method and Kane's approach was shown to be computationally more efficient. Other research in this area has been conducted for a one-wheeled robot and presented the preference of this method in nonholonomic systems.[16]

II.1. Dynamical equations

For a system with n-degrees of freedom (DOF) and m-generalized coordinates: q_i, the generalized speeds: u_r are defined as linear combination of $\dot{q}_i$s. The number of generalized speeds is usually equal to the number of DOFs. Definition of generalized speeds and utilizing constraint equations helps us represent all $\dot{q}_i$ versus generalized speeds. Then according to a partial derivation definition, we can obtain terms called partial velocities. Summation of internal products of actual forces and partial velocities lead to definition of generalized forces: F_r. In Kane's approach, inertial forces are considered due to the D'Alembert view of dynamic equations and inertial effects are inserted to the equation by means of generalized inertial forces: F_r^*. Further detail can be found in various references, but the final equation will be like this which mimics D'Alembert equations:

$$F_r^* + F_r = 0 \tag{1}$$

If some constraints are not included in the definition of generalized speeds, the number of generalized speeds will be more than DOF. If we assume that we have p-generalized velocities $(p \geq n)$ they will be constrained by $p - n$ linear equations and then just like the Lagrange method some terms must be added to the equation (1) as follows:

$$F_r^* + F_r = \sum_{i=1}^{p-n} \lambda_i a_{i\,r} \qquad 1 \leq r \leq p \tag{2}$$

In order to obtain the constraint forces, in the Lagrange method, a virtual displacement at the constraint point is defined, but Kane's method proposes a simpler approach by introducing a new virtual velocity. Here the partial velocities are utilized and distributed over all the system. In Kane's method there is no need to recalculate values of inertial and active forces and the constraint forces will be masked under F_r. First, we have to consider a virtual velocity in the direction of constraint force, causing a relative movement between its action and reaction point. Then this virtual separation velocity will cause some changes in the velocities of other points. Secondly, we must calculate partial velocities of force exertion points regarding these changes. Thirdly, it will be enough that we use equation (1) with newly calculated partial velocities.

III. KINEMATIC MODELING

The model presented here is a 2-D model of our 6-wheeled rover in which the number of wheels has been reduced

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

to 4. There are several reasons for this 2-D analysis. The mechanisms (like bogies and front fork) move in parallel planes and there is no linkage and flexibility in the 3rd dimension. Therefore the rover dimensions are more effective and more meaningful in the planner analysis. Moreover, the first step in design of rovers is checking the capability of climbing over obstacles and we have used this model to develop the abilities of climbing through dynamical equations of motion. As a result, we neglect its coupled dynamics in the horizontal plane with the vertical plane, and concentrate on its motion in a vertical plane when we have no steering angle, and both of the bogies have the same behavior.

III.1. Wheel path modeling

In our model, wheels are considered to be rigid and they keep their contact with the ground. The state of separation from the ground can be checked by determining the normal force on the wheel. Before simulation, the path profile should be modified and a pre-processor is required in order to generate a traversable path. First, we must have a proper mathematical definition of the terrain, thus we introduce $\vec{R}^c$ as the set of position vectors defining the path points. The profile can be estimated with a continuous function (like $\vec{R}^c(s)$) which must be a 2nd order piecewise continuous function from which the tangential vectors and the curvature can be obtained. Then, the curvature is compared with the radius of the wheel and it is determined whether the wheel can touch the ground at that point or not; and if so where will the wheel center be located?

We define the path of wheel center as the offset path (OP) and assume that the variable s is the path parameter. The position vector of the point O (wheel center) is:

$$\vec{R}^O(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix} \tag{3}$$

where the vector is twice differentiable; and the velocity and acceleration of this point are:

$$\vec{V}^O(s) = \vec{R}^{O'}(s)\dot{s} \tag{4}$$

$$\vec{a}^O(s) = \vec{R}^{O'}(s)\ddot{s} + \vec{R}^{O''}(s)\dot{s}^2 \tag{5}$$

$$\vec{R}'^O(s) = \begin{bmatrix} x'(s) \\ y'(s) \end{bmatrix} = \sqrt{x'(s)^2 + y'(s)^2} \begin{bmatrix} \cos(\theta(s)) \\ \sin(\theta(s)) \end{bmatrix} \tag{6}$$

$$\theta(s) = a\tan 2(y'(s), x'(s)) \tag{7}$$

The normal (towards robot) and tangential vectors of motion are defined as:

$$\hat{e}_l = \begin{bmatrix} \cos(\theta(s)) \\ \sin(\theta(s)) \end{bmatrix} \tag{8}$$

$$\hat{e}_n = \begin{bmatrix} -\sin(\theta(s)) \\ \cos(\theta(s)) \end{bmatrix} \tag{9}$$

Therefore the velocity of CP (contact point) is:

$$\vec{V}^O = \dot{s}\sqrt{x'(s)^2 + y'(s)^2}\hat{e}_t \tag{10}$$

According to the definition of parameter s and applying it for the location of wheel centers, diminishing normal velocity is guaranteed. Considering the wheel rotational slip, we have:

$$\vec{V}^O = \vec{V}^{(O/C)wheel} + \vec{V}^{C(wheel)/C(ground)} + \vec{V}^{C(ground)} \tag{11}$$

where $C(wheel)$ is the contact point (CP), attached to the wheel and $C(ground)$ is the CP attached the ground. Relative velocity of CP on the wheel to the wheel center equals:

$$\vec{V}^{(O/C)_{Wheel}} = \vec{\omega}^{Wheel} \times r^{(O/C)} = -r\dot{\phi}\hat{e}_t \tag{12}$$

If we add a normal component to this velocity it can also include the model of a simple deformable wheel. Also we can consider the velocity of CP of the wheel relative to the CP of the ground as:

$$\vec{V}^{C_{Wheel}/C_{Ground}} = u_t\hat{e}_t + u_n\hat{e}_n \tag{13}$$

The tangential component of this velocity can be considered in the presence of slippage. The second term is also used for defining a virtual normal velocity, in order to find the normal force. For the nondeformable nonslipping wheel we have:

$$\sqrt{x'(s)^2 + y'(s)^2}\dot{s} + r\dot{\phi} = 0 \tag{14}$$

We prefer the parameterization of the OP curve to be so that:

$$\sqrt{x'(s)^2 + y'(s)^2} = 1 \tag{15}$$

It means that the OP curve is parameterized by the curve length, thus:

$$\vec{R}'^O(s) = \begin{bmatrix} x'(s) \\ y'(s) \end{bmatrix} = \begin{bmatrix} \cos(\theta(s)) \\ \sin(\theta(s)) \end{bmatrix} = \hat{e}_t \tag{16}$$

$$\vec{R}''^O(s) = \begin{bmatrix} x''(s) \\ y''(s) \end{bmatrix} = \begin{bmatrix} -\sin(\theta(s)) \\ \cos(\theta(s)) \end{bmatrix} \theta'(s) = \hat{e}_n\theta'(s) \tag{17}$$

$$\theta'(s) = \frac{y''x' - x''y'}{y'^2 + x'^2} \tag{18}$$

And the curvature of the path can be written as:

$$|\theta'(s)| = \kappa(s) \tag{19}$$

III.2. Coordinate definition

We have 12 parts, 14 revolute joints and 4 wheel contacts. Ignoring slip on contact points we have 0 DOF, but the parallelogram used with six revolute joints is a special case, with one degree of freedom. Finally, including slippage, we have 5 DOF. Extending this view to the 3D case we will have, 11 DOF due to, 2 steering, 2 additional wheel slip, rolling freedom, and the other parallel bogie motion.

As mentioned earlier, we only focus on the 2D mode of motion and will solve the dynamic equations for an ideal case when the controllers are designed in a manner preventing

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

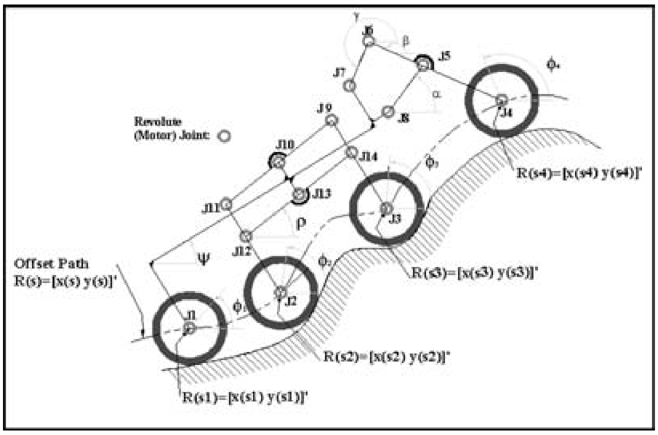


Fig. 2. Robot parameters definition.

wheels from slipping. In accordance to (Fig. 2), we will use 13 generalized coordinates and 12 constraints.

All of the coordinate frames used here are two dimensional and are defined in the Cartesian global coordinate (xy). The transformation matrix between two coordinates after rotation is defined as:

$$ {}^{(xy)_I}_{(xy)_\Theta}\Lambda(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \Lambda(\theta) \qquad (20)$$

$$\Lambda'(\theta) = \begin{bmatrix} -\sin(\theta) & -\cos(\theta) \\ \cos(\theta) & -\sin(\theta) \end{bmatrix} \qquad (21)$$

III.3. Forward kinematics

Robot kinematics can be simplified greatly by assuming curve length as its parameterization variable. This assumption simplifies both the kinematics and kinetics equations. Here, we only present forward kinematics for OPs parameterized by curve length. Also, another method has been developed for the case of other parameterization instead of curve length which is similar to this method, but computationally more complicated. Suppose that s_2 (position of second wheel on the path) is known (Fig. 3) and we need to determine the other wheel positions and links angles. Since the distance between the 2nd and 3rd wheel is constant, objective function for the 3rd wheel position determination can be defined as the form of equation (22).

$$d(s_3) = \text{distance}(\textit{Wheel}\#2, \textit{Wheel}\#3) - 2l_1 \qquad (22)$$

The goal is finding the root of this equation while we know that:

$$d(s_2 + 2l_1) \leq 0, \quad \exists \eta_1 \geq 1 : d(s_2 + \eta_1 2l_1) \geq 0 \qquad (23)$$

In the previous equations η_1 represents a parameter that can be adjusted manually based on the used OP.

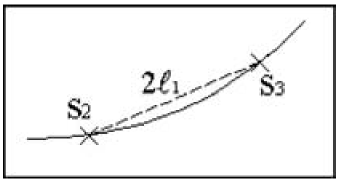


Fig. 3. 3rd wheel location.

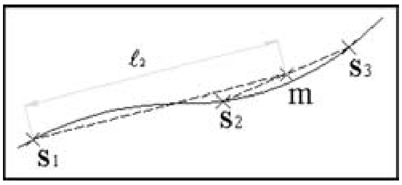


Fig. 4. 1st wheel location.

In order to find the first wheel position, we use the fact that the distance between 1st wheel and the middle point of the 2nd and 3rd one is invariant (Fig. 4); so the objective function is defined as:

$$d(s_1) = \text{distance}(\textit{Wheel}\#1, m) - l_2 \qquad (24)$$

with the following constraints:

$$d(s_2) \leq 0, \quad \exists \eta_2 \geq 1 : \; d(s_2 - \eta_2(l_2 - l_1)) \geq 0 \qquad (25)$$

Again η_2 represents a parameter that can be adjusted manually based on the used OP or found by iteration in each step.

Kinematical analysis of the 4th wheel is more complicated. Our goal is finding its position with arbitrary desired precision in the absence of any initial guess. Locus of a point attached to the middle bar of the front four-link mechanism (like point g in Fig. 5) can be formulated by the following two equations:

$$\begin{cases} X_g \cos\theta + Y_g \sin\theta = \dfrac{X_g^2 + Y_g^2 + (l_7 + l_8)^2 - l_1^2}{2(l_7 + l_8)} \\[2ex] (X_g - l_{15})\cos\theta + Y_g \sin\theta = \dfrac{(X_g - l_{15})^2 + Y_g^2 + l_8^2 - l_9^2}{2l_8} \end{cases} \qquad (26)$$

By solving these two equations for $\sin(\theta)$ & $\cos(\theta)$ and defining a function like f as:[17]

$$f = \cos^2\theta + \sin^2\theta - 1 \qquad (27)$$

We should find an acceptable value for S_4 that sets the value of f to zero.

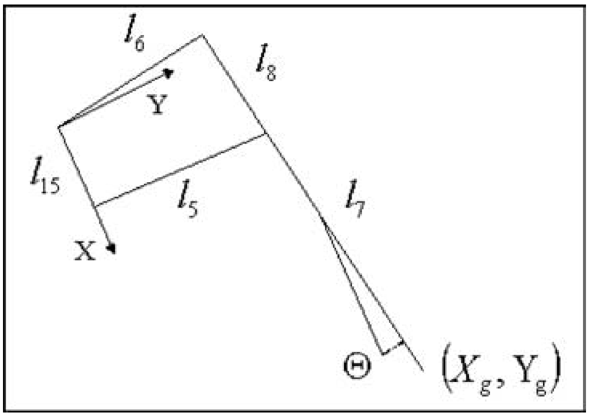


Fig. 5. Front four-link mechanism.

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

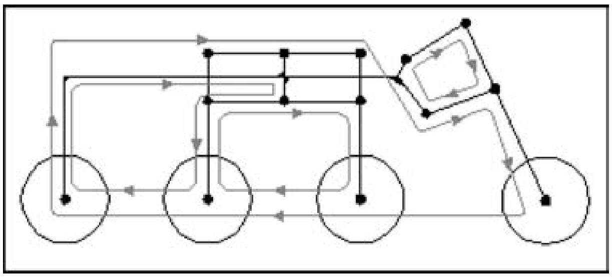


Fig. 6. Kinematic constraint loops.

III.4. Generalized speeds

The system has just one DOF based on the non-slip condition. Hence, we have to find 12 constraint equations, which 4 equations are non-slip constraints and they are:

$$\dot{s}_j\sqrt{y'(s_j)^2+x'(s_j)^2}+r\dot{\phi}_j=0 \qquad j=1,2,3,4 \tag{28}$$

The other 8 equations are closed loop chains, shown in Fig. 6. One of the preferences of Kane's method here is related to these closed loop chains; introducing the generalized speeds equal to the number of DOF makes it more practical and easier to derive the dynamical equations and to implement the numerical solution. De Jalón et. al.[18] have discussed various methods for these systems in detail. Although, it does not name Kane's method, but introduces similar methods and compares them very well in terms of number of state equations, computational cost, calculation of constraint forces, etc.... Finally, this author presents the simulation of a 4-wheeled car with a complicated suspension mechanism.

According to Fig. 6, the loop constraints can be derived like the following one:

$$\begin{bmatrix}0\\0\end{bmatrix}=\vec{R}'(s_1)\dot{s}_1+\Lambda'(\psi)\begin{bmatrix}l_3\\l_4\end{bmatrix}\dot{\psi}+\Lambda'(\alpha)\begin{bmatrix}l_5\\0\end{bmatrix}\dot{\alpha}$$
$$+\Lambda'(\beta)\begin{bmatrix}-l_8\\0\end{bmatrix}\dot{\beta}-\vec{R}'(s_4)\dot{s}_4 \tag{29}$$

In the last three equations, the relative velocity of C(wheel) to C(ground) has been considered to be zero, or we should also insert the slip velocity terms in the constraints. This method of extracting the independent closed loop chains can be extended to other wheeled mobile robots (WMR) which usually have simpler mechanisms.

We have only 1 DOF, and we define the only one generalized speed as:

$$u_1\equiv\sqrt{y'^2(s_2)+x'^2(s_2)}\dot{s}_2 \tag{30}$$

We have selected the $\dot{s}_2$ rather than the other wheels speed since this choice leads to a simpler and unique forward kinematics. Now we can write 13 equations for 12 constraints and 1 definition of u_1 in the matrix form:

$$\begin{bmatrix}A_{12\times13}\\ \mathit{Definitions\ of\ }u_1\end{bmatrix}_{13\times13}\dot{q}=\begin{bmatrix}0_{12\times1}\\u_1\end{bmatrix} \tag{31}$$

or

$$A_1(q)\dot{q}=B_1u_1 \tag{32}$$

Since the equations are independent, the matrix A_1 is not generally singular and we have:

$$\dot{q}=A_1^{-1}B_1u_1=G_1(q)_{13\times1}u_1=G_1u_1 \tag{33}$$

IV. DYNAMICAL ANALYSIS

IV.1. Inertial forces

Fig. 2 has presented the notation we have used for each body. Note that the inertia of motor gearbox (which has a high ratio and can not be neglected) has also been augmented on the wheels inertia. Although 12 substitute masses have been shown in the figure, only 6 masses have noticeable mass consisting of 4 wheels, front fork and the main body. The others have been considered as point masses added to these 6 masses in the equations. The velocities of these 6 points can be calculated like the following one:

$$\vec{V}^{c_5}=\vec{R}'(s_1)\dot{s}_1+\Lambda'(\psi)\begin{bmatrix}L_1\\L_2\end{bmatrix}\dot{\psi} \tag{34}$$

We can write the velocity of each mass as:

$$\vec{V}^{C_j}_{2\times1}=\tilde{V}^{C_j}_{2\times13}(q)\dot{q}_{13\times1}=\left\lfloor\tilde{V}^{C_j}_{2\times13}(q)G_1(q)_{13\times1}\right\rfloor u_1$$
$$=\lfloor\tilde{V}^{C_j}G_1\rfloor u_1 \tag{35}$$

$\tilde{V}^{C_j}$ is the jacobian of position vector with respect to q:

$$\tilde{V}^j=\frac{\partial\vec{R}^{Cj}}{\partial q}=Jacobian(\vec{R}^{Cj},q) \tag{36}$$

Furthermore for the rotational analysis:

$$\begin{aligned}\omega^j&=\dot{\phi}^j \qquad j=1,2,3,4\\ \omega^5&=\dot{\psi}\\ \omega^6&=\dot{\beta}\end{aligned} \tag{37}$$

$$\omega^j=\tilde{\omega}^j\dot{q}=[\tilde{\omega}^jG_1]u_1 \tag{38}$$

And similarly $\tilde{\omega}^j$ is jacobian of angular orientation vector with respect to q. Therefore the partial velocities and accelerations are derived here:

$$\vec{V}^{C_j}_1=\lfloor\tilde{V}^{C_j}G_1\rfloor \tag{39}$$

$$\omega^j_1=[\tilde{\omega}^jG_1] \tag{40}$$

$$\vec{a}^{C_j}_{2\times1}=[\tilde{V}^{C_j}G_1]\dot{u}_1+\frac{\partial[\tilde{V}^{C_j}G_1]}{\partial q}G_1u_1^2 \tag{41}$$

$$\alpha^j=[\tilde{\omega}^jG_1]\dot{u}_1+\frac{\partial[\tilde{\omega}^jG_1]}{\partial q}G_1u_1^2 \tag{42}$$

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

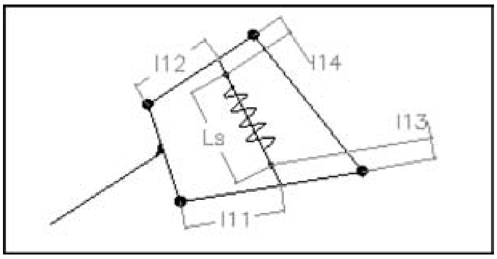


Fig. 7. Spring location.

The generalized forces can be obtained by substituting the above parameters can be calculated as following equation.

$$F_1^* = -\sum_{j=1}^{6}\left(m_j\left(\vec{V}_1^{C_j}\cdot\vec{a}^{C_j}\right) + I_j\left(\omega_1^j\alpha^j\right)\right) \tag{43}$$

IV.2. Gravity force

The partial velocities associated with each weight force exertion point are equal to the partial velocities of center of gravity of bodies. Also the weight forces are simply described by:

$$\vec{F}^{W_j} = -m_j g\begin{bmatrix}0\\1\end{bmatrix} \tag{44}$$

IV.3. Spring effect

The spring is mounted on the front fork between two confronting links and warranties the continuous contact with the surface. It helps the rover climb the obstacles reserving some energy in case of moving upward and releasing it while the other wheels face the obstacle. The internal damping of the spring also absorbs the severe vibrations and impulses on the mechanism. (Fig. 7) The vector connecting two end points of the spring equals:

$$\vec{R}^{Spring} = -\Lambda(\alpha)\begin{bmatrix}l_{11}\\l_{13}\end{bmatrix} + \Lambda(\varphi)\begin{bmatrix}-l_9\\l_{10}\end{bmatrix} - \Lambda(\gamma)\begin{bmatrix}l_{12}\\-l_{14}\end{bmatrix} \tag{45}$$

And the force generated by the spring is:

$$\begin{aligned}\vec{F}^{Spring} &= -k(|\vec{R}^{Spring}| - l_0)\frac{\vec{R}^{Spring}}{|\vec{R}^{Spring}|}\\ &= -k\left(1 - \frac{l_0}{\sqrt{\vec{R}^{Spring}\,.\,\vec{R}^{Spring}}}\right)\vec{R}^{Spring}\end{aligned} \tag{46}$$

which moves with the velocity of:

$$\vec{V}^{Spring} = [\tilde{V}^{Spring}G_1]\,u_1 \tag{47}$$

and the partial velocity of spring equals:

$$\vec{V}_1^{Spring} = [\tilde{V}^{Spring}G_1] \tag{48}$$

IV.4. Motor torques

The only controlling effort inserted into the system is due to the motor torques. For the 4 torques exerting on 4 wheels, we define a vector as:

$$\tau = [\tau_1\;\tau_2\;\tau_3\;\tau_4]^T \tag{49}$$

and the angular velocities of these torques are:

$$\begin{aligned}\omega^{\tau_j} &= \dot{\phi}_j - \dot{\psi} \quad j = 1,2,3\\ \omega^{\tau_4} &= \dot{\phi}_4 - \dot{\beta}\end{aligned} \tag{50}$$

These velocities can also be obtained from wheel rotation Jacobian:

$$\omega^{\tau_j} = \tilde{\omega}^{\tau_j}\dot{q} = [\tilde{\omega}^{\tau_j}G_1]u_1 \quad j = 1,2,3,4 \tag{51}$$

$$\omega_1^{\tau_j} = [\tilde{\omega}^{\tau_j}G_1] \tag{52}$$

Now by defining E in the form of:

$$E = [\tilde{\omega}^{\tau_1}\tilde{\omega}^{\tau_2}\tilde{\omega}^{\tau_3}\tilde{\omega}^{\tau_4}]^T\,G_1 \tag{53}$$

We can derive the closed form equation for the dynamics of the rover:

$$M(q)\dot{u}_1 + N(q)u_1^2 + gW(q) + kK(q) = \tau\cdot E(q) \tag{54}$$

Where the coefficients are:

$$\begin{aligned}M(q) &= \sum_{j=1}^{6}(m_j[\tilde{V}^{C_j}G_1]\cdot[\tilde{V}^{C_j}G_1] + I_j[\tilde{\omega}^jG_1]\cdot[\tilde{\omega}^jG_1])\\ N(q) &= \sum_{j=1}^{6}\left(m_j[\tilde{V}^{C_j}G_1]\cdot\frac{\partial[\tilde{V}^{C_j}G_1]}{\partial q}\right.\\ &\qquad\left. + I_j[\tilde{\omega}^jG_1]\cdot\frac{\partial[\tilde{\omega}^jG_1]}{\partial q}\right)G_1\\ K(q) &= \left(1 - \frac{l_0}{\sqrt{\vec{R}^{Spring}\,.\,\vec{R}^{Spring}}}\right)[\tilde{V}^{Spring}G_1]\cdot\vec{R}^{Spring}\\ W(q) &= \sum_{j=1}^{6}m_j[\tilde{V}^{C_j}G_1]\cdot\begin{bmatrix}0\\1\end{bmatrix}\end{aligned} \tag{55}$$

IV.5. Comments on the equations

The equation obtained above is a closed form symbolic equation that can be applied for simulation and optimization (especially Gradient Methods). Many other planetary rovers like Rocky7 with the rocker-bogie mechanism may also be analyzed using this general method. Except for the yaw rotation, efficiency of all of these mechanisms is highly dependent on the links dimension and can be optimized by proper definition of optimization goals (e.g.: energy efficiency, path of CM, maximum torques or powers, etc. . .). Moreover, the equations of motions can be derived for

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

all of these mechanisms in the ideal form (non-slip and planner motion) to compare their advantages in the face of hindrances.

Unlike other researches done in this area, this equation is not confined to the quasi-static condition;[1,19] high speed motion can also be analyzed here (of course we can check this exact method with the others by eliminating some terms). The non quasi-static terms appear in the form of $N(q)u^2$ and include the Coriolis, centrifugal and rotational terms. Although this term is derived for 1 DOF case, in the case of slippage and studying systems with more DOFs, it doesn't change its general form; u_1^2 and $\dot{u}_1$ are substituted by the vectors uu^T and $\dot{u}$. Assume that all of the q & $\dot{q}$ s and local path slopes are sensed with sensors in an arbitrary moment, then we have:

$$\begin{aligned} A(q)_{m\times m}\dot{q}_{m\times 1} &= B_{m\times n}(q)u_{n\times 1} \Rightarrow \dot{q} \\ &= A(q)\backslash B(q)u = G(q)_{m\times n}u \end{aligned} \qquad (56)$$

By solving the above equation with a numerical method, all of the coefficients of equation (54) are calculated except $N(q)$. For the $N(q)$ term we have to know the value of $\frac{\partial[\tilde{V}^{C_j}G]}{\partial q}$, $\frac{\partial[\tilde{\omega}^j G]}{\partial q}$. If we have used numerical values of G based on sensed q, knowing N(q) demands a huge amount of numerical differentiating techniques which isn't reasonable even for this 1 DOF mechanism. The other method, we have used, is evaluating the symbolic relation for G and deriving a formula for $\frac{\partial[\tilde{V}^{C_j}G]}{\partial q}$, $\frac{\partial[\tilde{\omega}^j G]}{\partial q}$. Although we have used the symbolic form of G for differentiation, but for the simulation, the Gaussian method and numerical solution of equation $A\dot{q} = Bu$ have been implemented.

IV.6. Ground forces

The ground forces on the wheels are substantial in all rovers. They can determine the state of slippage and missing ground contact. The contact forces are assumed in this form for every wheel:

$$\vec{F}^{C_{Ground}\rightarrow C_j} = f_j\hat{e}_{t-j} + N_j\hat{e}_{n-j} \quad j = 1, 2, 3, 4 \qquad (57)$$

To make these forces into evidence, a virtual relative velocity is considered in their contact point regarding equation (13). These relative velocities are defined:

$$\vec{V}^{C_j/C_{Ground}} = u_{t_j}\hat{e}_{t-j} + u_{nj}\hat{e}_{n-j} \quad j = 1, 2, 3, 4 \qquad (58)$$

Eight virtual generalized velocities can be defined, and when they are distributed over the closed chains:

$$A_1(q)\dot{q} = B_1(q)u_1 + \sum_{i=2}^{9} B_i(q)u_i \qquad (59)$$

It is very important that A$_1$ has not changed and consequently in every "system function evaluation", G$_r$ (link matrices between $\dot{q}$ and u_r) can be computed numerically by the simple Gaussian method:

$$\dot{q} = [G_1\ G_2\ldots]u = A_1\backslash[B_1\ B_2\ldots]u \qquad (60)$$

With known G_r, partial velocities associated with each of velocities can be computed by equation like (39, 40). Care must be taken into account because in this case velocities are not just based on $\dot{q}$ and an additional term must be added to them regarding broken constraints:

$$\vec{V}_r = [\tilde{V}G_r] + \vec{V}^{Broken\_Constraint\_r^{th}} \qquad (61)$$

$$\omega_r = [\tilde{\omega}G_r] + \omega^{Broken\_Constraint\_r^{th}} \qquad (62)$$

From this point of view, Kane's equation of motion is the dual of unit load theorem in mechanics of materials. So the unit load displacement is the dual of partial velocity. In numerical computation we do not need to use partial differentiation and instead we solve equation (60) for a "u" equaling unity.

By substituting of these partial velocities in equation (1), and using the numerical values of forces we can reach the desired equations. By multiplying each contact force by the partial velocity of its exertion point, it can be added to the F_r in equation (1). Hence by our proper definition of extra virtual generalized velocities in each equation (for rth generalized velocity $2 \leq r \leq 9$) just one of the contact forces appears by unity coefficient and they can be rephrased as:

$$N = \mathbf{N}_x(q, u) + \mathbf{N}_\tau(q)\tau + \mathbf{N}_u(q)\dot{u} \qquad (63)$$

The above equation means that we can calculate the vector of wheel normal forces (and also other desired forces) as a summation from $\mathbf{N}_x$: an only state dependent value (including effect of spring, weight and some rotational terms), $\mathbf{N}_{\dot{u}}\dot{\mathbf{u}}$: acceleration (or rate of change of generalized velocity) linearly dependent term and $\mathbf{N}_\tau\tau$: linearly dependent term to active controlling effort. But if we use the acceleration equation (54) we can rephrase it as:

$$\dot{u} = \dot{u}_x + \dot{u}_\tau\tau \qquad (64)$$

Combination of the two above equations leads to:

$$N = N_x + N_\tau\tau \qquad (65)$$

$$f = f_x + f_\tau\tau \qquad (66)$$

By eliminating acceleration terms we are ready to apply constraint equations as:

$$N_i \geq 0, \quad |f_i| \leq \mu_i N_i \quad 1 \leq \mathrm{i} \leq 4 \qquad (67)$$

The above equations (three equations per wheel) represent 12 linear inequality constraints on available efforts. Furthermore motor torques are restricted by characteristic curves of motors and their drivers; taking into account that motor inertias are augmented into wheel inertias and neglecting Coulomb frictional torques in bearings we have:

$$|\tau_i| \leq \tau_{\max\_i}(\omega_{motor\_i}) \quad \mathbf{1} \leq \mathbf{i} \leq 4 \qquad (68)$$

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

The above equation limits the 4-dimensional effort space to a state dependent (velocity of motor) hyper-cube. However for low velocities, boundaries of these cubes can be estimated by stall torques. Within this hyper-cube and the space formed between hyper-planes of equations (67) is located the possible solution space. It is seen that generally by increasing velocities in uneven-terrain this space shrinks to a null space which means some wheels are slipping or losing their ground contact that can lead to tip-over. Also in some instances (for example in inverse dynamics) we focus only on a hyper-plane in the solution space formed by equation (64), which means that we set our torques such that it results in an especial value of acceleration value.

Not only must we know the limits on our efforts, but also we must know an appropriate way to handle redundancies in our actuators. Based on the almost equal division of weight to various wheels (in almost even-terrain) it seems that driving the robot by equal torques is a good initial guess, and in addition a simple method for implementing a control unit. However as mentioned by Iagnemma[19] and also other publications, it is much better if we use our freedom in arranging actuator values in a more reasonable way that leads to the optimization of some variables. For our simulations we minimize the lost power in our motors which is a widely used technique.

For our DC motors we simply assume that torques are proportional to motor currents. In addition, we assume that the lost power is only a resistive one which is proportional to currents squared. Finally, lost power will be a summation of powered values of torques. As far as we have used the same motor and control board for all wheels lost power can be expressed as:

$$\textit{Lost Power} \propto \sum_{i=1}^{6} \tau_{Wheel\_i}^2 = \left(\tau_1^2 + \frac{\tau_2^2 + \tau_3^2}{2} + \tau_4^2\right) \quad (69)$$

Therefore it is seen that optimization function is a quadratic function with some linear equality and inequality constraints in a hyper-cube limit. Thus both the exact and numerical methods in the quadratic optimization can be used to find its solution fast enough to be really implemented in a control board.

It is also useful to note that we can use rotational acceleration equation of wheels to derive equations (66) in a simpler way and indeed we have used this equation for evaluation of wheel friction forces:

$$I_j\ddot{\phi} = \tau_j + rf_j \quad j = 1, 2, 3, 4 \quad (70)$$

V. SIMULATION

The obtained equations can be used for various purposes such as: dynamics optimization, checking available control strategies, inverse and forward dynamic simulation, and comparison between various rovers. But as far as we have focused on obtaining equations themselves, we only describe two simple simulations. In both simulations we force the robot to pass a bump generated by a function like "$he^{-(\frac{x}{w})^2}$". The "h" is selected equal to "w" and is about 2.25 times the wheel radius.

V.1. Inverse dynamics

In this part of the simulation we assume that all six wheels torques are arranged based on the stated optimization. Also we assume that the robot is driven with a constant velocity (20 cm/s which equals 40% robot maximum speed) in the second wheel (point s_2). This velocity is very close to the velocity of CM. In addition, from a control point of view it is a measurable velocity that can be set to a constant value with a velocity controller.

Results of the simulation are plotted in Fig. 8 part (a). It can be inferred that this obstacle makes little change to the angles of robots because of its special passive mechanism. Also it is seen that torques (based on optimization scheme) are almost equal (unless near slippage or losing ground contact). However, in critical cases like when the front wheel passes the top of the bump and starts to move downward it is very critical to perform traction control.

Also various terms in the acceleration equation are plotted. It is beneficial that we can use the terms obtained by u^2. In most literature, this term is neglected due to the quasi-static simulations or linearized formulas; but the effect of inertia forces on our motion appears only in this term. Also as can be seen from the figure, most of the effort is expended to balance energy required for weight forces. Torques are slightly decreased by the spring effect in both the ascending and descending phases of motion, indeed the main effect of the spring is increasing ground contact of front and back wheels. It is valuable to mention that if we discard our constraints on torques (traction control) and set them to equal values we must pass the obstacle with almost 10% of maximum speed and near stall torques. This result emphasizes our method of driving the robot at a high speed and low torques case over challenging obstacles.

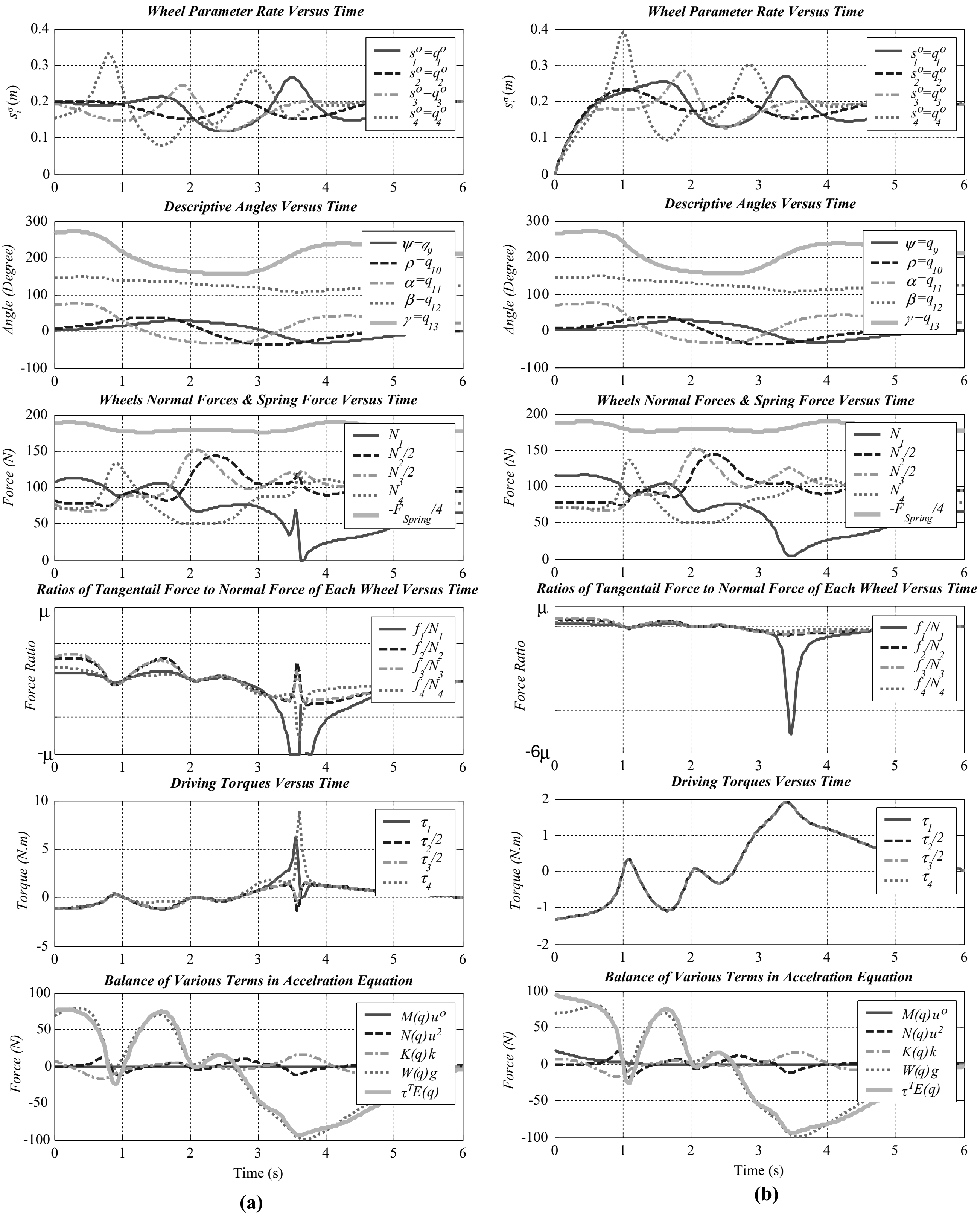
It is estimated that each step takes about 25 ms calculation in a 2.66 MHz computer for a low-level language, like *FORTRAN*. Consequently, we hope that we can implement this method in designing control algorithms based on current or future computers.

V.2. Forward dynamics

In this part of the simulation we assume that all six wheels torques are equal. Again the same task has been applied to a PID like controller and the results are shown in Fig. 8 part (b). However appropriate feedback for cancellation of nonlinear terms requires knowledge of path curvature which is itself a challenging question for implementation in an actual robot.

In this simulation after a 1.5 second transient response we will follow the same behavior as before. But to emphasize the effect of traction control we have not utilized the traction control method. Consequently, the plot of normal forces, friction forces & their ratios has changed. Now we have a better margin from surface separation and less torques are required but indeed this does not happen because of reaching limits of friction force.

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press



Wheel Parameter Rate Versus Time

Descriptive Angles Versus Time

Wheels Normal Forces & Spring Force Versus Time

Ratios of Tangentail Force to Normal Force of Each Wheel Versus Time

Driving Torques Versus Time

Balance of Various Terms in Accelration Equation

(a) **(b)**

Fig. 8. (a) Inverse dynamic simulation, (b) Forward dynamic simulation.

VI. CONCLUSIONS

Based on Kane's method a systematic method for deriving equations of motion governing a rover has been presented. The method is very efficient for numerical purposes. In addition, it has been shown that the method is capable of extracting exact symbolic equations for a rover with one of the most complicated mechanism including four closed chains. Also it is capable of calculating constraint forces easily and handling them to generate traction control algorithms for high velocities in uneven terrains. Both kinematics and kinetics are presented and this method offers an appropriate set of coordinate to fully and easily describe rover configuration.

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press

This method which can be used in selection of state variables, extraction of rate equations and also in meaningful description of various terms of equations, is very useful and novel. This work has been mainly an introductory example and has focused mostly on the detailed steps of dynamic formulation rather than dynamic analysis of CEDRA robot. Future works are required to perform more exact analysis of this robot and to extend the gamut of this method.

There are several works to be performed in the future. Kinematic and dynamic analysis can be extended to 3-D case. Derived system equations may be utilized for more complicated methods in traction control. The objective of traction control is to optimize torque distribution of the robot motors to obtain proper traction in various cases and difficult terrain conditions. Also, dynamic parameter optimization while minimizing energy consumption is another major subject in mobile robot mechanics that can be fulfilled based on system dynamic modeling. Finally, implementation of this method on other known rover mechanisms and comparison between them would lead us to a better understanding of the rover design and analysis field.

References

1. S. Sreevinasan and B. Wilcox, "Stability and Traction Control of an Actively Actuated Micro-Rover", *Journal of Robotic Systems* **11**, 487–502 (1994).
2. J. E. Chottiner, "Simulation of a Six-Wheeled Martain Rover Called the Rocker-Bogie", *M.Sc. Thesis* (The Ohio State University, Columbus, Ohio, 1992).
3. R. Linderman and H. Eisen, "Mobility Analysis, Simulation and Scale Model Testing for the Design of Wheeled Planetary Rovers", *Conference of Missions, Technologies, and Design of Planetary Mobile Vehicle*, Toulouse, France (September 28–30, 1992) pp. 1–6.
4. H. Hacot, "The Kinematic Analysis and Motion Control of a Planetary Rover", *M.Sc. Thesis* (Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, May, 1998).
5. M. Tarokh, G. McDermott and J. Hung, "Kinematics and Control of Rocky-7 Mars Rover", *Technical Report* (Department of Mathematical and Computer Science, San Diego State University, 1998).
6. V. Kumar, and J. Gardner, "Kinematics of Redundantly Actuated Closed Chains", *IEEE Transaction on Robotics and Automation* **6**, No. 2, 269–274 (1990).
7. M. Tai, "Modeling of Wheeled Mobile Robot on Rough Terrain", *CD-ROM Proc. of ASME International Mechanical Engineering Congress*, Washington, D.C. (November 15–21, 2003) pp. 1–7.
8. A. Meghdari, H. N. Pishkenari, A. L. Gaskarimahalle, S. H. Mahboobi and R. Karimi, "Optimal Design and Fabrication of "CEDRA" Rescue Robot Using Genetic Algorithm", *CD-ROM Proc. of International Design Engineering Technical Conferences, DETC2004*, Salt Lake City, Utah (2004) pp. 1–8.
9. A. Meghdari, F. Amiri, S. H. Mahboobi, A. Baghani, A. L. Gaskarimahalle, H. N. Pishkenari, R. Karimi and Y. Khalighi, "Design and Fabrication of a Mobile Robot For Rescue Purposes", (in Persian), *CD-ROM Proc. of the Iranian Society of Mechanical Engineering Conferences, ISME2004*, Tehran, Iran (2004) pp. 1–8.
10. R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet, "Innovative design for wheeled locomotion in rough terrain", *J. Robotics and Autonomous Systems* **40**, 151–162 (2002).
11. R. Volpe, J. Balaram, T. Ohm and R. Ivlev, "Rocky 7: A Next Generation Mars Rover Prototype", *Journal of Advanced Robotics* **11**, No. 4, 341–358 (Dec., 1997).
12. J. Matijevic, "Mars Pathfinder Microrover -Implementing a Low Cost Planetary Mission Experiment", *Proceeding of the Second IAA International Conference on Low-Cost Planetary Missions,* John Hopkins Applied Physics Laboratory, Maryland, USA (April 16–19, 1996) paper # IAA-L-0510.
13. A. L. Kemurdjian, V. Gromov, V. Mishkinyuk, V. Kucherenko and P. Sologub, "Small Marsokhod Configuration", *International Conference on Robotics and Automation*, Nice (1992) pp. 165–168.
14. T. R. Kane and D. A. Levinson, *Dynamics: Theory and Applications* (McGraw-Hill, 1985).
15. H. G. Tanner and K. J. Kyriakopoulos, "Mobile manipulator modeling with Kane's approach", *Robotica* 19, part 6, 675–690 (2001).
16. W. Nukulwuthiopas, T. Maneewan and S. Laowattana, "Dynamic Modeling of a One-Wheel Robot by Using Kane's Method", *IEEE International Conference on Industrial Technology*, Bangkok Thailand (2002) (Dec., 2002) pp. 524–529.
17. R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages* (McGraw-Hill Book Co., 1964).
18. J. G. De Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge*, (Springer-Verlag, New-York, 1994) pp. 156–201.
19. K. D. Iagnemma, "Rough Terrain Mobile Robot Planning and Control with Application to Planetary Exploration", *Ph.D. Thesis* (Mechanical Engineering Dept., Massachusetts Institute of Tech., 2001).

https://doi.org/10.1017/S0263574705001712 Published online by Cambridge University Press