計測自動制御学会東北支部 第177回研究集会 (1998.10.30-31)
資料番号 177-6

MATLABを用いた移動ベースロボットの汎用ダイナミクスシミュレーション・ツール

The SpaceDyn: a MATLAB Toolbox for Space and Mobile robots

○吉田 和哉

○ Kazuya Yoshida

東北大学 大学院 工学研究科 航空宇宙工学専攻

Dept. of Aeronautics and Space Engineering, Tohoku University

キーワード： 多節リンク系 (Multibody System), 移動ベースロボット (Moving-Base Robot) 方向余弦 (Direction Cosines), ニュートン・オイラー方程式 (Newton-Eular Equations), ダイナミクスシミュレーション (Dynamics Simulation)

連絡先： 〒980-8579 仙台市青葉区荒巻字青葉01 東北大学 大学院 工学研究科 航空宇宙工学専 吉田和哉
Tel: (022)217-6992, Fax: (022)217-6992, E-mail: yoshida@astro.mech.tohoku.ac.jp

1. Introduction

The SpaceDyn is a MATLAB Toolbox for the kinematic and dynamic analysis and simulation of articulated multi-body systems with a moving base. It can be downloaded from

`http://www.astro.mech.tohoku.ac.jp/spacedyn`

and used freely for academic purpose. Any of commercial use is not permitted.

We developed this toolbox motivated and inspired by **Robotics toolbox** developed by Peter I. Coke, which is available from

`http://www.mathworks.com/ftp/miscv4.shtml`

We took one m-file and use it as the original is, but our toolbox as a whole, does not have compatibility with the Peter Coke's toolbox unfortunately.

We hope you can find our toolbox useful for your research, and we'd appreciate your question, comments or feedback if any.

The following is a memorandum regarding our toolbox development.

- This toolbox is for the use with MATLAB 5.0 or higher. We use three dimensional array which is not supported in version 4 or lower.
- We assume the system composed of $n+1$ bodies and connected by n joints. Let the body 0 be a *reference body*. Multiple branches can attach on any single *body*, as far as the system keeps a topological tree configuration. There must be a single *joint* between two bodies. We call a terminal point or the point of interest such as manipulator hand as *endpoint*. Each body, except body 0, can have one endpoint at maximum. In this document, the terms *body* and *link* are the same.
- The toolbox allows force/torque input on (1) the centroid of the reference body, (2) each endpoint, and (3) each joint. The toolbox computes the position, velocity and acceleration of (1) the centroid of the reference body, (2) the centroid of each body, (3) each endpoint, and (4) each joint.
- Computation of input force/torque are open to user programming. You can arbitrary decide each joint as either active or passive one. If you give always zero torque, such as $\tau_i = 0$, the corresponding joint behaves as a free

joint. Or if you give such a torque as:

$$\tau_i = -Kq_i - D\dot{q}_i$$

the joint behaves as a passive visco-elastic joint. You can treat even a flexible link, by modeling it as a discrete successive chain of rigid links connected by elastic joints.

Of course, you can give any arbitrary control torque determined by your own control law, on all or arbitrary selected joints.

- We know that the Denavit-Hartenberg notation is commonly used in the field of manipulator kinematics with the advantage of unique allocation of coordinate systems with minimum parameters. But we know that the DH sometimes locates the coordinate origin away from the location of an actual joint. From the dynamics point of view, the angular velocity and the inertia tensor should be defined around the corresponding joint axis or body centroid. We then do not use the DH notation but introduce a rule to define the coordinate systems with more flexibility. Our rule locates the origin of the coordinate system on each joint and orients the primary axes so that the inertia tensor should be simpler, but admits 3 position and 3 orientation parameters among two successive coordinate systems.
- For the representation of attitude or orientation, we use 3 by 3 direction cosine matrices, coded with a symbol $\boldsymbol{C}$. For example, $\boldsymbol{C}_0$ is the direction cosines to represent the attitude of the body 0. The advantage of direction cosine is (1) singularity free, (2) we can easily derive Roll-Pitch-Yaw angles, Euler angles, or quartanions, and (3) it is easy to find the mathematical relationship with angular velocity.
- On the other hand, we frequently need Roll-Pitch-Yaw representation also. For RPY angles, we use the symbol $\boldsymbol{Q}$. For example, in order to express the twisting angles between two coordinate systems, we consider α (roll) around x axis, β (pitch) around y axis, then γ (yaw) around z axis. The set of these angles are coded by $\boldsymbol{Q}_i$.

2. Mathematical Graph Representation

In order to mathematically describe the interconnection of the bodies, we adopt a method from mathematical graph theory [1]. We simplify it with additional rules on the assignment of link and joint indices, so that we can easily and uniquely construct two types of matrices (vectors); a connection index $\boldsymbol{B}$ and incidence matrices $\boldsymbol{S}$, $\boldsymbol{S}_0$, and $\boldsymbol{S}_e$.

The procedure from indices assignment to matrix construction is summarized as follows:

1) Assign the indices of links and joints in the following manner.
   - Reference body is denoted by *link* 0.
   - The index of a *link* i in the physical connection between *link* 0 and *link* j must be $0 < i < j$.
   - One link can have multiple connections with other links. As a result of the above statement, a link i ($i > 0$) has one *lower connection* (which index is smaller than i) and zero or one or multiple *upper connection(s)* (which index is greater than i).
   - There is one single *joint* interconnecting two links.
   - Indices of joints begin from 1, and *joint* i is physically attached on *link* i.
   - Then the joint interconnecting link 0 and link 1 is *joint* 1, and the joint interconnecting link 1 and link 5 is *joint* 5, for example.
2) Connection index vector $\boldsymbol{B}$ is used to find the lower connection of a link, which exists uniquely for a link i ($i > 0$).
   - The element of $\boldsymbol{B}_i$ is the index of the lower connection of link i.
3) Incidence matrix $\boldsymbol{S}$ is used to find the upper connection of a link, which may not exist or exist one or more.
   Each element of $\boldsymbol{S}_{ij}$ $(i, j = 1, \ldots, n)$ is defined by:

$$\boldsymbol{S}_{ij} = \begin{cases} +1 & (if \quad i = \boldsymbol{B}_j) \\ -1 & (if \quad i = j) \\ 0 & (otherwise) \end{cases}$$

4) Define a matrix $\boldsymbol{S}_{0j}$ $(j = 1, \ldots, n)$ as:

$$\boldsymbol{S}_{0j} = \begin{cases} +1 & (if \quad 0 = \boldsymbol{B}_j) \\ 0 & (otherwise) \end{cases}$$

   This represents a flag to indicate if link i has a connection with the reference link 0.
5) Also define a matrix $\boldsymbol{S}_{ej}$ $(j = 1, \ldots, n)$ as:

$$\boldsymbol{S}_{ej} = \begin{cases} +1 & (if \quad \text{link } j \text{ is a terminal link}) \\ 0 & (otherwise) \end{cases}$$

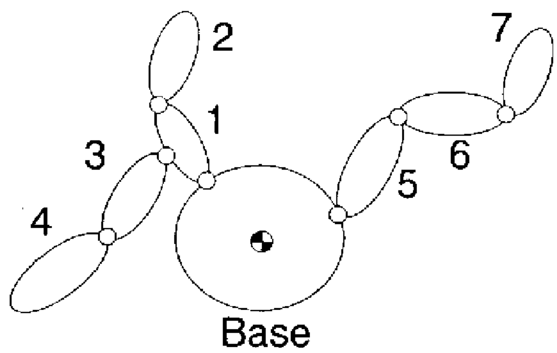


Fig. 1 Sample System

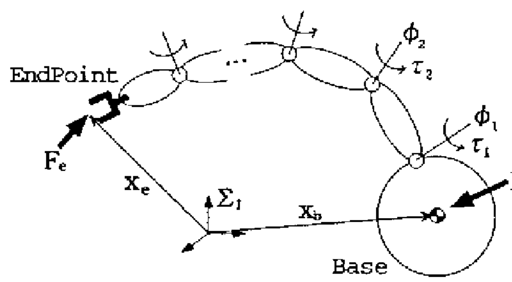


Fig. 2 Multibody System

This represents a flag to indicate if link i is a terminal endlink.

Figure 1 depicts an example of a system with multiple branches numbered by the rule presented here.

3. Coordinate System

Let the inertial reference coordinate frame be denoted by $\{\Sigma_I\}$ [1], which is stationary or lineary moving with constant velocity in the inertial space. It is not physically precise but we sometime consider the orbital fixed frame as the inertial frame in the sense of practice.

We also define moving coordinate frames fixed on each link. We do NOT take the Denavit-Hartenberg convention but introduce a simpler and flexible rule to define the link coordinates. Our rule is as follows:

1) If the joint i is revolution, then
   - locate the origin of the coordinate system $\{\Sigma_i\}$ on joint i and fixed it to the link i,
   - set its z-axis to coincide with the joint rotation axis,
   - orient its x-axis toward joint $i+1$ or the direction in which the inertia tensor is expressed easier.
2) If the joint i is prismatic, then
   - locate the origin of the coordinate system $\{\Sigma_i\}$ on the place when joint i has zero displacement and fixed it to the link $i-1$,
   - set its z-axis to coincide with the joint displacement axis, with the positive direction,
   - orient its x-axis toward the direction in which the inertia tensor is expressed easier.

We may also need a coordinate system located on the link centroid. In such a case, we define the link centroid coordinate i parallel to the coordinate located on joint i.

[1]The expression $\{\Sigma_I\}$ is use to represent the basis of a coordinate frame, a set of unit vectors: *vectrix*, see [2]

4. Direction Cosine and Coordinate Transformation Matrix

The direction cosine matrices $\boldsymbol{C}_i$ are commonly used to represent the attitude or orientation of body i in the inertial frame in the field of aseospace engineering [2]. On the other hand, the coordinate transformation matrices with the notation of ${}^I\boldsymbol{A}_i$ are commonly used in the field of robotics. These two are eventually the same:

$$\boldsymbol{C}_i = {}^I\boldsymbol{A}_i$$

Since we define the link coordinate system as above, we generally need three axis rotations to coincide from $\{\Sigma_{i-1}\}$ to $\{\Sigma_i\}$. Let $A_1(\alpha_i)$, $A_2(\beta_i)$, $A_3(\gamma_i)$ be coordinate transformation around each principle axis and $A_3(q_i)$ be the rotation angle of joint i, then we get the following relationship (see Figure 3):

$$\begin{aligned}\{\Sigma_i\} &= {}^i\boldsymbol{A}_{i-1}\{\Sigma_{i-1}\} \\ &= A_3(q_i)A_3(\gamma_i)A_2(\beta_i)A_1(\alpha_i)\{\Sigma_{i-1}\} \quad (1)\end{aligned}$$

where $A_3(\gamma_i)$ and $A_3(q_i)$ seem duplicated, but γ_i corresponds to an offset angle and should be separated from a net rotation angle q_i.

Note that the RPY representation of the attitude of link 0 is:

$$\begin{aligned}\{\Sigma_0\} &= \boldsymbol{C}_0{}^T\{\Sigma_I\} \\ &= {}^0\boldsymbol{A}_I\{\Sigma_I\} \\ &= A_3(\gamma_0)A_2(\beta_0)A_1(\alpha_0)\{\Sigma_I\} \quad (2)\end{aligned}$$

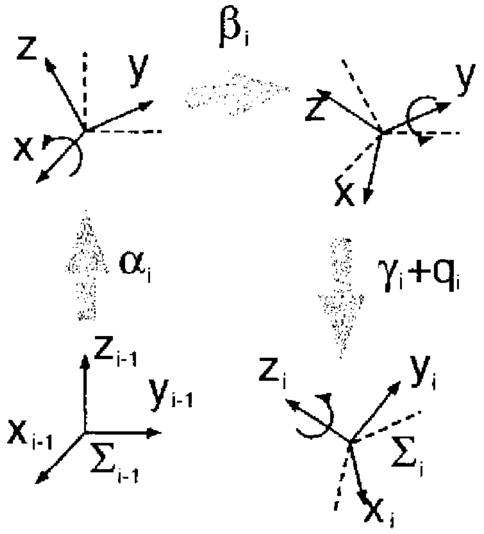


Fig. 3 Coordinate Transformation

where $\alpha_0, \beta_0, \gamma_0$ are Roll, Pitch, Yaw angles respectively.

The direction cosines are redundant way to represent attitude, but its advantage is that the relationship between attitude and angular velocity can be expressed by a simple equation, such that:

$$\dot{\boldsymbol{C}}_0 = -\tilde{\omega}_0 \boldsymbol{C}_0 \quad (3)$$

where $\dot{\boldsymbol{C}}_0$ is a time derivative of $\boldsymbol{C}_0$ and $\tilde{\boldsymbol{\omega}}$ is a skew-symmetric operation of the angular velocity ω_0. This relationship is used for the routine of singularity-free integration from angular velocity to attitude.

5. Kinematics

5.1 Link Vectors

Link vectors for a link i are defined as follows (see Figure 4).

$\boldsymbol{c}_{ij}$: vector from the centroid of link i to joint j.

$\boldsymbol{\ell}_{ij}$: vector from joint i to joint j.

$$\boldsymbol{\ell}_{ij} = \boldsymbol{c}_{ij} - \boldsymbol{c}_{ii} \quad (4)$$

$\boldsymbol{c}_{ie}$: vector from the centroid of link i to the end-point if link i is an end-link.

$\boldsymbol{\ell}_{ie}$: vector from joint i to the end-point.

$$\boldsymbol{\ell}_{ie} = \boldsymbol{c}_{ie} - \boldsymbol{c}_{ii} \quad (5)$$

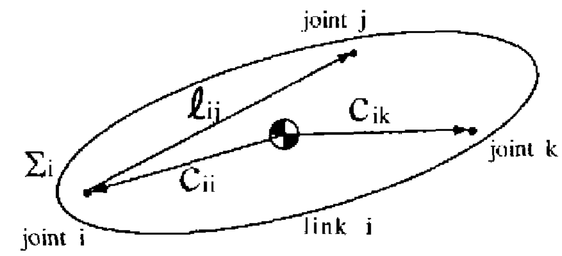


Fig. 4 Position vectors

5.2 Revolution Joint

For a successive set of links connected by a revolution joint, velocity $\boldsymbol{v}_i$ and angular velocity $\boldsymbol{\omega}_i$ are calculated recursively (see Figure 5). When $\boldsymbol{v}_o$ and $\boldsymbol{\omega}_0$ are given,

$$\begin{aligned} {}^I\boldsymbol{\omega}_i &= {}^I\boldsymbol{\omega}_{B_i} + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\dot{\phi}_i & (6) \\ {}^I\boldsymbol{v}_i &= {}^I\boldsymbol{v}_{B_i} + {}^I\boldsymbol{\omega}_{B_i} \times {}^I\boldsymbol{c}_{B_i i} - {}^I\boldsymbol{\omega}_i \times {}^I\boldsymbol{c}_{ii} & (7) \end{aligned}$$

And accelerations are calculated as the following.

$$\begin{aligned} {}^I\dot{\boldsymbol{\omega}}_i &= {}^I\dot{\boldsymbol{\omega}}_{B_i} + {}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\dot{\phi}_i) + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\ddot{\phi}_i & (8) \\ {}^I\dot{\boldsymbol{v}}_i &= {}^I\dot{\boldsymbol{v}}_{B_i} + {}^I\dot{\boldsymbol{\omega}}_{B_i} \times {}^I\boldsymbol{c}_{B_i i} + {}^I\boldsymbol{\omega}_{B_i} \times ({}^I\boldsymbol{\omega}_{B_i i} \times {}^I\boldsymbol{c}_{B_i i}) & \\ &\quad - {}^I\dot{\boldsymbol{\omega}}_i \times {}^I\boldsymbol{c}_{ii} - {}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{\omega}_i \times {}^I\boldsymbol{c}_{ii}) & (9) \end{aligned}$$

5.3 Prismatic Joint

If a joint is prismatic, the kinematic relationship becomes as follwos, for verocities:

$$\begin{aligned} {}^I\boldsymbol{\omega}_i &= {}^I\boldsymbol{\omega}_{B_i} & (10) \\ {}^I\boldsymbol{v}_i &= {}^I\boldsymbol{v}_{B_i} + {}^I\boldsymbol{\omega}_{B_i} \times {}^I\boldsymbol{c}_{B_i i} - {}^I\boldsymbol{\omega}_i \times {}^I\boldsymbol{c}_{ii} & \\ &\quad + {}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i) + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\dot{\phi}_i & (11) \end{aligned}$$

And for accelerations:

$$\begin{aligned} {}^I\dot{\boldsymbol{\omega}}_i &= {}^I\dot{\boldsymbol{\omega}}_{B_i} & (12) \\ {}^I\dot{\boldsymbol{v}}_i &= {}^I\dot{\boldsymbol{v}}_{B_i} + {}^I\dot{\boldsymbol{\omega}}_{B_i} \times \boldsymbol{c}_{B_i i} + {}^I\boldsymbol{\omega}_{B_i} \times ({}^I\boldsymbol{\omega}_{B_i} \times \boldsymbol{c}_{B_i i}) & \\ &\quad - {}^I\dot{\boldsymbol{\omega}}_i \times \boldsymbol{c}_{ii} - {}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{\omega}_i \times \boldsymbol{c}_{ii}) + {}^I\dot{\boldsymbol{\omega}}_i \times ({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i) & \\ &\quad + {}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i)) + 2{}^I\boldsymbol{\omega}_i \times ({}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\phi_i) & \\ &\quad + {}^I\boldsymbol{A}_i{}^i\boldsymbol{k}_i\ddot{\phi}_i & (13) \end{aligned}$$

5.4 End-Point Kinematics

The kinematic relationship around the end-points is expressed as follows:

$$\dot{\boldsymbol{x}}_h = \boldsymbol{J}_m\dot{\boldsymbol{\phi}} + \boldsymbol{J}_b\dot{\boldsymbol{x}}_b \quad (14)$$

$$\ddot{\boldsymbol{x}}_h = \boldsymbol{J}_m\ddot{\boldsymbol{\phi}} + \dot{\boldsymbol{J}}_m\dot{\boldsymbol{\phi}} + \boldsymbol{J}_b\ddot{\boldsymbol{x}}_b + \dot{\boldsymbol{J}}_b\dot{\boldsymbol{x}}_b \quad (15)$$

$\boldsymbol{x}_b$	$\in R^6$	: position/orientation of the base
$\boldsymbol{x}_h$	$\in R^6$	: position/orientation of the end-points
$\boldsymbol{\phi}$	$\in R^n$	: joint variables
$\boldsymbol{J}_b$	$\in R^{6\times 6}$	: Jacobian matrix for base variables
$\boldsymbol{J}_m$	$\in R^{6\times n}$	: Jacobian matrix for joint variables

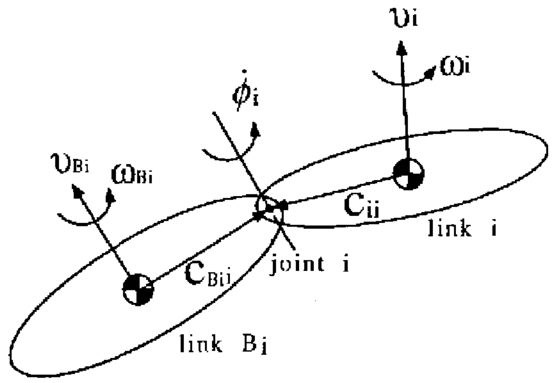


Fig. 5 Kinematics

6. Equation of Motion

The equation of motion of the system is expressed in the following form [3, 4]:

$$\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} = \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \mathcal{F}_h \quad (16)$$

where

$$\boldsymbol{H}_b \in R^{6\times 6} \equiv \begin{bmatrix} w\boldsymbol{E} & w\tilde{\boldsymbol{r}}_{0g}^T \\ w\tilde{\boldsymbol{r}}_{0g} & \boldsymbol{H}_\omega \end{bmatrix} \quad (17)$$

$$\boldsymbol{H}_{bm} \in R^{6\times n} \equiv \begin{bmatrix} \boldsymbol{J}_{Tw} \\ \boldsymbol{H}_{\omega\phi} \end{bmatrix} \quad (18)$$

$$\boldsymbol{H}_\omega \in R^{3\times 3} \equiv \sum_{i=1}^{n} (\boldsymbol{I}_i + m_i \tilde{\boldsymbol{r}}_{0i}^T \tilde{\boldsymbol{r}}_{0i}) + \boldsymbol{I}_0 \quad (19)$$

$$\boldsymbol{H}_{\omega\phi} \in R^{3\times n} \equiv \sum_{i=1}^{n} (\boldsymbol{I}_i \boldsymbol{J}_{Ri} + m_i \tilde{\boldsymbol{r}}_{0i} \boldsymbol{J}_{Ti}) \quad (20)$$

$$\boldsymbol{H}_m \in R^{n\times n} \equiv \sum_{i=1}^{n} (\boldsymbol{J}_{Ri}^T \boldsymbol{I}_i \boldsymbol{J}_{Ri} + m_i \boldsymbol{J}_{Ti}^T \boldsymbol{J}_{Ti}) \quad (21)$$

$$\boldsymbol{J}_{Tw} \in R^{3\times n} \equiv \sum_{i=1}^{n} m_i \boldsymbol{J}_{Ti} / w \quad (22)$$

$$\boldsymbol{J}_{Ti} \in R^{3\times n} \equiv [\boldsymbol{k}_1 \times (\boldsymbol{r}_i - \boldsymbol{p}_1), \boldsymbol{k}_2 \times (\boldsymbol{r}_i - \boldsymbol{p}_2), \ldots, \ldots, \boldsymbol{k}_i \times (\boldsymbol{r}_i - \boldsymbol{p}_i), \boldsymbol{o}, \ldots, \boldsymbol{o}] \quad (23)$$

$$\boldsymbol{J}_{Ri} \in R^{3\times n} \equiv [\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_i, \boldsymbol{o}, \ldots, \boldsymbol{o}] \quad (24)$$

$$\boldsymbol{r}_{0g} \in R^3 \equiv \boldsymbol{r}_g - \boldsymbol{r}_0 \quad (25)$$

$$\boldsymbol{r}_{0i} \in R^3 \equiv \boldsymbol{r}_i - \boldsymbol{r}_0 \quad (26)$$

m_i	: mass of link i of arm k
w	: total mass of the system ($w = \sum_{i=1}^{n} m_i$)
$\boldsymbol{r}_i$	: position vector of centroid of link i
$\boldsymbol{p}_i$	: position vector of joint i
$\boldsymbol{k}_i$	: unit vector indicating joint axis direction of link i
$\boldsymbol{r}_0$	: position vector of centroid of satellite base body
$\boldsymbol{r}_g$	: position vector of a total centroid of the system
$\boldsymbol{c}_b, \boldsymbol{c}_m$	: velocity dependent non-linear terms
$\mathcal{F}_b$	: external force/moment on the base
$\boldsymbol{\tau}$	: joint torque of the arm
$\mathcal{F}_h$	: external force/moment on the hand
$\boldsymbol{E}$	: 3×3 identity matrix

and a tilde operator stands for a cross product such that $\tilde{\boldsymbol{r}}\boldsymbol{a} \equiv \boldsymbol{r} \times \boldsymbol{a}$. All position and velocity vectors are defined with respect to the inertial reference frame.

7. Forward Dynamics: Simulation Procedure

The procedure to compute a forward dynamics solutions are summarized as follows:

1) At time t, compute link positions and velocities, recurcively from link 0 to n.

2) Compute the inertia matrices using equations (17)-(26).

3) Set accelerations $\ddot{\boldsymbol{x}}_b$ and $\ddot{\boldsymbol{\phi}}$ zero, and external forces $\mathcal{F}_b$ and $\mathcal{F}_h$ zero, then compute the inertial forces recursively from link n to 0. The resultant forces on the coordinates $\boldsymbol{x}_b$ and $\boldsymbol{\phi}$ are equal to the non-liner forces $\boldsymbol{c}_b$ and $\boldsymbol{c}_m$, respectively.

4) Determine joint control forces $\boldsymbol{\tau}$ and thruster forces on the base $\mathcal{F}_b$ from a control law.

5) Compute the accelerations by:

$$\begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \mathcal{F}_h - \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} \right\} \quad (27)$$

6) Integrate the above accelerations to yield the velocities and positions at time $t + \Delta t$.

7) go to 1. and continue.

8. Inverse Dynamics

Inverse dynamic computation is useful for a computed torque control. It is also needed for the forward dynamics in numerical computation the velocity dependent non-linear terms as described in the last section.

For the inverse dynamic computation, an order-n, recursive Newton-Euler approach [5] is well-known.

Newton and Euler equations for a link i are expressed as:

$$\boldsymbol{F}_i = m_i \dot{\boldsymbol{v}}_i \qquad (28)$$

$$\boldsymbol{N}_i = \boldsymbol{I}_i \dot{\omega}_i + \omega_i \times (\boldsymbol{I}_i \omega_i) \qquad (29)$$

where $\boldsymbol{F}_i, \boldsymbol{N}_i$ are inertial force and moment exert on the link centroid. Together with the following force and moment exerting on the joint or end-point,

$\boldsymbol{f}_i, \boldsymbol{n}_i$: Force and moment on joint i.

$\boldsymbol{f}_{ei}, \boldsymbol{n}_{ei}$: Force and momnet on end-point (if link i is an end-link)

the dynamic equilibrium expressed in the following form (see Figure 6):

$$\boldsymbol{f}_i = \boldsymbol{F}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} \boldsymbol{f}_j + \boldsymbol{S}_{ei} \boldsymbol{f}_{ei} \qquad (30)$$

$$\boldsymbol{n}_i = \boldsymbol{N}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} (\boldsymbol{\ell}_{ij} \times \boldsymbol{f}_j + \boldsymbol{n}_j) + \boldsymbol{S}_{ii} \boldsymbol{c}_{ii} \times \boldsymbol{F}_i + \boldsymbol{S}_{ei} (\boldsymbol{\ell}_{ie} \times \boldsymbol{f}_{ei} + \boldsymbol{n}_{ei}) \qquad (31)$$

for around a revolution joint, and

$$\boldsymbol{f}_i = \boldsymbol{F}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} \boldsymbol{f}_j + \boldsymbol{S}_{ei} \boldsymbol{f}_{ei} \qquad (32)$$

$$\boldsymbol{n}_i = \boldsymbol{N}_i + \sum_{j=i+1}^{n} \boldsymbol{S}_{ij} (\boldsymbol{\ell}_{ij} \times \boldsymbol{f}_j + \boldsymbol{n}_j) \qquad (33)$$
$$+ \boldsymbol{S}_{ii} (\boldsymbol{c}_{ii} - \phi_i) \times \boldsymbol{F}_i + \boldsymbol{S}_{ei} (\boldsymbol{\ell}_{ie} \times \boldsymbol{f}_{ei} + \boldsymbol{n}_{ei})$$

for around a prismatic joint.

After the computation of whole $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ for $i = 1$ to n, we can obtain joint torque as:

$$\tau_i = \boldsymbol{n}_i^{TI} \boldsymbol{k}_i \quad (if \;\; \text{revolution joint}) \qquad (34)$$

$$\tau_i = \boldsymbol{f}_i^{TI} \boldsymbol{k}_i \quad (if \;\; \text{prismatic joint}) \qquad (35)$$

And the reaction force/moment on the base centroid is obtained as follows:

$$\boldsymbol{F}_0 = \sum_{i=1}^{n} \boldsymbol{S}_{0i} \boldsymbol{f}_i \qquad (36)$$

$$\boldsymbol{N}_0 = \sum_{i=1}^{n} \boldsymbol{S}_{0i} (\boldsymbol{c}_{0i} \times \boldsymbol{f}_i + \boldsymbol{n}_i) \qquad (37)$$

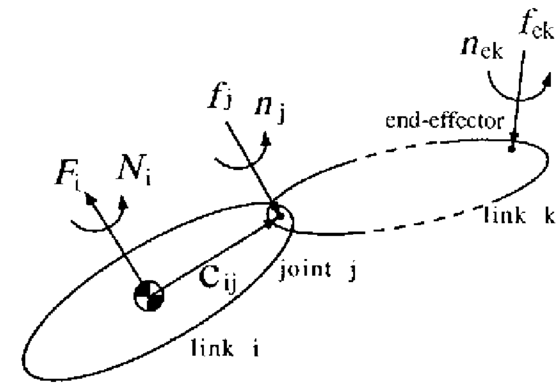


Fig. 6 Dynamic equilibrium

9. Application Examples

Here, some of applications for dynamic simulation of moving-base sytems are illustrated, which all are relevant to actual space flight missions.

Figure 7 (a) depicts a simulation model of ETS-VII, a Japanese free-flying space robot with 2 meter long 6 DOF manipulator arm. The satellite was launched November, 1997. It is currently flying in orbit, as of August 1998, and a number of significant experiments on space robotics are conducting on the satellite. Free-flying system dynamics including manipulator reaction and the vibrations of solar paddles can be analyzed with the SpaceDyn toolbox.

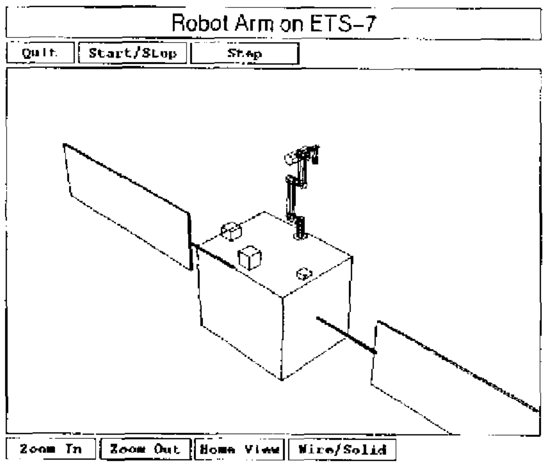
Figure 7 (b) depicts a flexible-base robot. Practical examples of such a system are SRMS-SPDM system, a Canadian made space station manipulator system and JEMRMS, a macro-mini manipulator system for the Japanese Experimental Module of the station. For these systems the internal dynamics, as presented in the following section, will be a key technology in terms of the reaction and vibration management.

Both figures 7 (a) and (b) are illustrated using a useful animation tool named "XAnimate," which can be freely downloaded from [6].

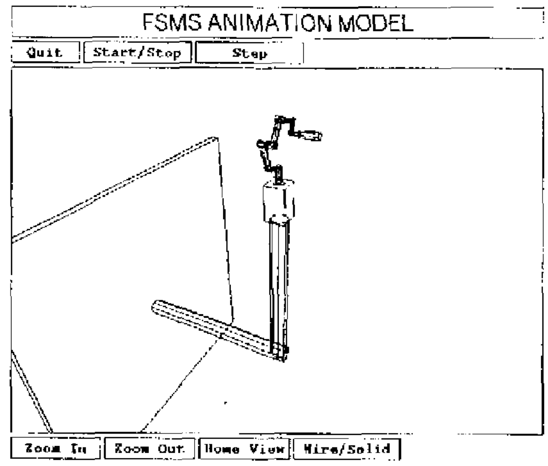
Figure 7 (c) is a touch-down simulation of MUSES-C asteroid sample-return satellite. For this simulation, impulsive ground contact is a key issue and the contact model discussed in the previous subsection is applied [7]. With the development of the contact model for tire mechanics, the dynamic motion of off-road articulated vehicle can be also simulated, as shown in(d) [8], such an application is found in a mission of a planetary exploration rover.

参考文献

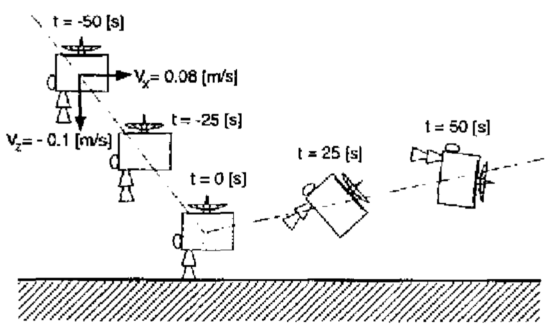
1) Jens Wittenburg: *Dynamics of Systems of Rigid Bodies,* B. G. Teubner Stuttgart, 1977.



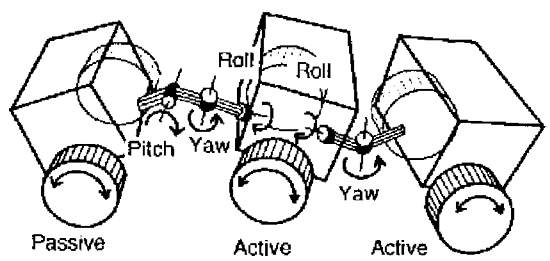
(a) A simulation model for Free-Flying Space Robot



(b) A simulation model for Flexible-Base Robot



(c) Touch-down simulation of MUSES-C Asteroid Sample-Return Satellite



(d) An example of an articulated off-road vehicle as a potential design of a planetary rover

Figure 7 Practical applications of the dynamics simulation of moving-base robots by "SpaceDyn" toolbox

2) Peter C. Hughes: *Spacecraft Attitude Dynamics*, John Wiley & Sons, Inc., 1986.

3) *Space Robotics: Dynamics and Control*, edited by Xu and Kanade, Kluwer Academic Publishers, 1993.

4) K. Yoshida: "A General Formulation for Under-Actuated Manipulators," *Proc. 1997 IEEE/RSJ Int. Conf. on Intelligent Robots and Syatems*, pp.1651-1957, Grenoble, France, 1997.

5) J. S. Y. Luh, M. W. Walker and R. P. C. Paul: "On-Line Computational Scheme for Mechanical Manipulators," *Trans. ASME J. Dynamic Systems, Measurements and Control*, vol.120, pp.69-76, 1980.

6) ftp://eeftp.eng.ohio-state.edu/pub/orin/

7) K. Yoshida and T. Hiraoka, "Touch Down Simulation of the MUSES-C Satellite for Asteroid Sampling," AIAA Modeling & Simulation Technologies Conf. Paper 98-4376, Boston, MA, August, 1998.

8) K. Yoshida, T. Shiwa, and M. Oda, "Dynamic Simulation of an Articulated Off-Road Vehicle," AIAA Modeling & Simulation Technologies Conf. Paper 98-4362, Boston, MA, August, 1998.