Satellite Dynamics Toolbox

Principle, User Guide and tutorials

Daniel Alazard, Christelle Cumer

June 2014
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nomenclature</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>1 Rigid body dynamics based on Newton-Euler equations</strong></td>
<td>11</td>
</tr>
<tr>
<td>1.1 <strong>Newton-Euler</strong> equations at the center of mass</td>
<td>11</td>
</tr>
<tr>
<td>1.2 <strong>Newton-Euler</strong> equations at any reference point</td>
<td>12</td>
</tr>
<tr>
<td>1.3 Linearized <strong>Newton-Euler</strong> equations</td>
<td>13</td>
</tr>
<tr>
<td><strong>2 Satellite Dynamics Toolbox: principles</strong></td>
<td>17</td>
</tr>
<tr>
<td>2.1 Assumptions</td>
<td>18</td>
</tr>
<tr>
<td>2.2 Direct dynamics model</td>
<td>18</td>
</tr>
<tr>
<td>2.3 Appendage dynamics model $M_A^P(s)$</td>
<td>19</td>
</tr>
<tr>
<td>2.3.1 Case of a rigid appendage</td>
<td>20</td>
</tr>
<tr>
<td>2.3.2 Case of a flexible appendage</td>
<td>20</td>
</tr>
<tr>
<td>2.3.3 Case of an embedded angular momentum</td>
<td>21</td>
</tr>
<tr>
<td>2.4 Inverse dynamics model</td>
<td>23</td>
</tr>
<tr>
<td>2.5 Revolute joint between the appendage and the hub</td>
<td>24</td>
</tr>
<tr>
<td>2.6 Generalisation</td>
<td>26</td>
</tr>
<tr>
<td><strong>3 Implementation with Matlab®</strong></td>
<td>29</td>
</tr>
<tr>
<td>3.1 Satellite Dynamics Toolbox® functions</td>
<td>29</td>
</tr>
<tr>
<td>3.2 User defined data file <strong>nom_fich.m</strong></td>
<td>31</td>
</tr>
<tr>
<td>3.2.1 Remarks</td>
<td>31</td>
</tr>
<tr>
<td>3.2.2 A first tutorial (on inertia tensor specified at any reference point)</td>
<td>32</td>
</tr>
<tr>
<td>3.3 FAQ</td>
<td>33</td>
</tr>
<tr>
<td>3.4 Example 1: <strong>Spacecraft1.m</strong></td>
<td>35</td>
</tr>
<tr>
<td>3.5 Example 2: <strong>Spacecraft2.m</strong></td>
<td>37</td>
</tr>
<tr>
<td>3.6 Example 3: <strong>Spacecraft5.m</strong></td>
<td>38</td>
</tr>
<tr>
<td>3.7 Example 4: <strong>FOUR_CMGs.m</strong></td>
<td>40</td>
</tr>
</tbody>
</table>
## Contents

3.8 Example 5: SpaceRoboticArm.m ........................................... 45  
3.9 Example 6: Spacecraft1u.m .................................................. 48  
3.10 Exercise ........................................................................... 50  

**References** ........................................................................... 53  

### A Inline help

| A.1  | help of function main.m .................................................. 55  |
| A.2  | help of function translate_dynamic_model.m .......................... 56  |
| A.3  | help of function rotate_dynamic_model.m .............................. 56  |
| A.4  | help of function kinematic_model.m ..................................... 56  |
| A.5  | help of function antisym.m .................................................. 57  |
Nomenclature

\( \mathcal{R}_b = (O, \vec{x}_b, \vec{y}_b, \vec{z}_b) \) : Main body (hub \( \mathcal{B} \)) reference frame. \( O \) is a reference point on the main body. \( \vec{x}_b, \vec{y}_b, \vec{z}_b \) are unit vectors.

\( \mathcal{R}_a = (P, \vec{x}_a, \vec{y}_a, \vec{z}_a) \) : Appendage \( \mathcal{A} \) reference frame. \( P \) is the appendage’s anchoring point on the hub. \( \vec{x}_a, \vec{y}_a, \vec{z}_a \) are unit vectors.

\( B \) : Hub’s (\( \mathcal{B} \)) centre of mass.

\( A \) : Appendage’s (\( \mathcal{A} \)) centre of mass.

\( G \) : Overall assembly’s centre of mass \( \mathcal{B} + \mathcal{A} \).

\( T_{ba} \) : Direction cosine matrix of the rotation from frame \( \mathcal{R}_b \) to frame \( \mathcal{R}_a \), that contains the coordinates of vectors \( \vec{x}_a, \vec{y}_a, \vec{z}_a \) expressed in frame \( \mathcal{R}_b \).

\( \overrightarrow{\dot{a}}_B \) : Inertial acceleration (vector) of body \( \mathcal{B} \) at point \( B \).

\( \overrightarrow{\dot{a}}_P \) : Inertial acceleration (vector) of body \( \mathcal{B} \) at point \( P \).

\( \overrightarrow{\omega} \) : Angular speed (vector) of \( \mathcal{R}_b \) with respect to the inertial frame.

\( \overrightarrow{F}_{\text{ext}} \) : External forces (vector) applied to \( \mathcal{B} \).

\( \overrightarrow{T}_{\text{ext},B} \) : External torques (vector) applied to \( \mathcal{B} \) at point \( B \).

\( \overrightarrow{F}_{B/A} \) : Force (vector) applied by \( \mathcal{B} \) on \( \mathcal{A} \).

\( \overrightarrow{T}_{B/A,P} \) : Torque (vector) applied by \( \mathcal{B} \) on \( \mathcal{A} \) at point \( P \).

\( D_B^B \) : Hub’s static dynamics model expressed at point \( B \).

\( m_B \) : Hub’s \( \mathcal{B} \) mass.

\( \mathbb{I}_B \) : Inertia tensor \( 3 \times 3 \) of \( \mathcal{B} \) at point \( B \).

\( M_B^A(s) \) : Appendage’s dynamics model at point \( P \).

\( m_A \) : Appendage’s mass.

\( \mathbb{I}_A \) : Inertia tensor \( 3 \times 3 \) of \( \mathcal{A} \) at point \( A \).

\( \tau_{PB} \) : Kinematic model between points \( P \) and \( B \):

\[
\tau_{PB} = \begin{bmatrix}
I_3 & (^P\vec{B}) \\
0_{3x3} & I_3
\end{bmatrix}.
\]
\((\ast \overrightarrow{PB})\): Skew symmetric matrix associated with vector \(\overrightarrow{PB}\): if \(\overrightarrow{PB} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathcal{R}_i}\)

\[\begin{bmatrix} (\ast \overrightarrow{PB}) \end{bmatrix}_{\mathcal{R}_i} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}_{\mathcal{R}_i}.\]

\(N_a\) : Number of appendages.

If there is a flexible appendage, we add the following definitions:

\(N\) : Number of flexible modes.
\(\eta\) : Modal coordinates’ vector.
\(\omega_i\) : \(i^{th}\) flexible mode’s angular frequency.
\(\xi_i\) : \(i^{th}\) flexible mode’s damping ratio.
\(l_{i,P}\) : \(i^{th}\) flexible mode’s vector of modal participations \((1 \times 6)\), expressed at point \(P\).

\(L_P\) : Matrix \(N \times 6\) of the modal participation factors expressed at point \(P\):

\[(L_P = [l_1^{T,P}, l_2^{T,P}, \ldots , l_N^{T,P}]^T).\]

If there is an embedded angular momentum, we add the following definitions:

\(\Omega\) : Angular speed \((\text{rad/s})\) of the spinning top.
\(\overrightarrow{z}_w\) : Unit vector along the embedded angular momentum.
\(I_w\) : Spinning top’s principal inertia.
\(I_r\) : Spinning top’s radial inertia.

If there is a revolute joint between the appendage and the hub, we add the following definitions:

\(\overrightarrow{r}_a\) : Unit vector along the revolute joint axis: \(\overrightarrow{r}_a = [x_{ra} \ y_{ra} \ z_{ra}]^T_{\mathcal{R}_a}\).
\(\dot{\theta}\) : Revolute joint’s angular acceleration.
\(C_m\) : Revolute joint’s torque.

General definitions:

\([X]_{\mathcal{R}_i}\) : \(X\) (model, vector or tensor) expressed in frame \(\mathcal{R}_i\).
\(\frac{dX}{dt}|_{\mathcal{R}_i} = 0\) : Derivative of vector \(\vec{X}\) with respect to frame \(\mathcal{R}_i\).
\(\vec{u} \wedge \vec{v}\) : Cross product of vector \(\vec{u}\) with vector \(\vec{v}\)
\(\vec{u} . \vec{v}\) : Scalar product of vector \(\vec{u}\) with vector \(\vec{v}\)
\(s\) : LAPLACE’s variable.
\(I_n\) : Identity matrix \(n \times n\).
\(0_{n \times m}\) : Zero matrix \(n \times m\).
\(A^T\) : Transpose of \(A\).
\(\text{diag}(\omega_i)\) : Diagonal matrix \(N \times N\): \(\text{diag}(\omega_i)(i,i) = \omega_i, i = 1, \ldots , N\).
\(P(s)(i:j,l:m)\) : Sub-system of \(P(s)\) from inputs \(l\) to \(m\) to outputs \(i\) to \(j\).
Introduction

This document presents the principles and the MATLAB® implementation of the Satellite Dynamics Toolbox (SDT version 1.3) available for download following this link:


This toolbox allows the user to compute the linear dynamics model of a satellite fitted with one or more flexible appendage (solar generators, antennas, ...) possibly taking into account the joints and actuators (solar generator driving mechanism, Control Moment Gyro (CMG), external manipulator, ...). More generally, it can model an open kinematic chain of bodies (sub-structures) in a space environment (gravity efforts are not taken into account). Such a chain or structure can be described by a tree (see Figure 1) where:

- each node $A_i$ corresponds to a body (or link or substructure),
- each edge corresponds to a joint between two bodies. 2 types of joint are considered:
  - clamped joint,
  - revolute joint (around a given axis).

Classical tree terminology will be used. In the example of Figure 1: $A_0$ is the root, $A_3$ to $A_7$ are the leaves, $A_2$ is the parent of $A_5$ and the child of $A_0$.

Each body can be either flexible or not (with the restriction that the flexible bodies must be the last elements or the leaves of the chain). The linear dynamics model is the 6 by 6 transfer between the external forces and torques applied on the main body structure at a reference point and the translational and rotational accelerations (at the same reference point). When the structure includes joints (in this version of the toolbox, only revolute joints between the sub-structures are considered), the model is augmented with the transfer between internal torque and acceleration for each joint.

Chapter 1 is a reminder on Newton-Euler equations applied to a single rigid body.
Chapter 2 explains the modelling principles used in the Satellite Dynamics Toolbox. Chapter 3 presents the MATLAB® implementation of the toolbox, and more particularly:

- the setup of the data files describing the structure to be modeled,
- structure examples:
  - Spacecraft1.m is a data file describing a spacecraft composed of a main body and two cantilevered flexible appendages (see section 3.4),
  - Spacecraft2.m is the same as Spacecraft1.m but the first appendage is connected to the main body through a revolute joint and tilted with a 10 degrees angle (see section 3.5),
  - Spacecraft5.m is the same as Spacecraft1.m but includes 3 additional appendages corresponding to angular momentums along to X, Y and Z axes taken into account to represent spinned RWAs (see section 3.6),
  - FOUR_CMGs.m is a data file describing a platform fitted with four identical CMGs (see section 3.7). The CMG is described by the data file dataCMG.m. This data correspond roughly to the experimental CMG platform TETRAGYRE developed by ONERA/DCSD and presented at: http://www.onera.fr/dcsd/gyrodynes/,
  - SpaceRoboticArm.m is a data file describing a platform fitted with a 6 d.o.f. rigid space robotic manipulator (see section 3.8): the 6 links of the manipulator are described in the files Segment1.m, Segment2.m, ..., Segment6.m.

All these files and a demo script file (demoSTD.m) are included in the SDT package.
**Remark:** this new version of the toolbox can also manage **uncertain parameters** to perform sensitivity analyses. In this case, the outputs of the toolbox main functions are uncertain elements, matrices or models, compatible with the MATLAB® Robust Control Toolbox (see file `Spacecraft1u.m` as an example, section 3.9)). The development of this version of the toolbox was done with MATLAB® R2013a.
Chapter 1

Rigid body dynamics based on Newton-Euler equations

This chapter is a reminder on rigid body dynamics based on Newton-Euler equations (see also [6]). The interest of Newton-Euler equations is to consider in the same model the 6 degrees-of-fredom (3 translations and 3 rotations) of the rigid body. This chapter will also present the linearity assumptions which will be adopted in the following chapters.

1.1 Newton-Euler equations at the center of mass

Let us consider a body $B$ with a center of mass $B$ (see Figure 1.1).
The Newton-Euler equations reads:

\[
\begin{bmatrix}
\vec{F}_{ext} \\
\vec{T}_{ext,B}
\end{bmatrix}
= 
\begin{bmatrix}
\begin{bmatrix}
m^B I_3 \\
0_{3 \times 3}
\end{bmatrix} & 0_{3 \times 3} \\
0_{3 \times 3} & \mathbb{I}_B
\end{bmatrix}
\begin{bmatrix}
\vec{a}_B \\
\vec{\omega}
\end{bmatrix}
+
\begin{bmatrix}
0_{3 \times 1} \\
\vec{\omega} \wedge \mathbb{I}_B \vec{\omega}
\end{bmatrix}
\]
\tag{1.1}
\]

where:

- \(\vec{F}_{ext}\): total force acting on the body,
- \(\vec{T}_{ext,B}\): total torque acting about the centre of mass,
- \(m^B\): mass of the body,
- \(\mathbb{I}_B\): Inertia tensor of the body about the centre of mass,
- \(\vec{a}_B\): acceleration of the centre of mass,
- \(\vec{\omega}\): angular velocity of the body.

Eq. (1.1) is intrinsic and can be projected in any frame. Using the skew matrix \((\vec{\omega})\) associated with vector \(\vec{\omega}\) (see nomenclature), Eq. (1.1) can be written:

\[
\begin{bmatrix}
\vec{F}_{ext} \\
\vec{T}_{ext,B}
\end{bmatrix}
= 
\begin{bmatrix}
\begin{bmatrix}
m^B I_3 \\
0_{3 \times 3}
\end{bmatrix} & 0_{3 \times 3} \\
0_{3 \times 3} & \mathbb{I}_B
\end{bmatrix}
\begin{bmatrix}
\vec{a}_B \\
\vec{\omega}
\end{bmatrix}
+
\begin{bmatrix}
0_{3 \times 1} \\
(\vec{\omega}) \mathbb{I}_B \vec{\omega}
\end{bmatrix}
\]
\tag{1.2}
\]

1.2 Newton-Euler equations at any reference point

Let us consider another point \(P\) on the body \(B\). Since the body is rigid (assumption \([H1]\)), the velocity of \(P\) can be expressed as a function of the velocity of \(B\) and the angular velocity \(\vec{\omega}\):

\[
\vec{\nu}_P = \vec{\nu}_B + \vec{\omega} \wedge B \vec{P} = \vec{\nu}_B + \vec{P} \vec{B} \wedge \vec{\omega}.
\]
\tag{1.3}

Using the skew matrix \((\vec{\omega})\) associated with vector \(\vec{B} \vec{P}\) (see nomenclature), one can write:

\[
\vec{\nu}_P = \vec{\nu}_B + (\vec{\omega})(\vec{B} \vec{P} \vec{\omega}).
\]
\tag{1.4}

Considering the dual velocity vectors (6 components) at point \(B\) and \(P\), one can write:

\[
\begin{bmatrix}
\vec{\nu}_P \\
\vec{\omega}
\end{bmatrix}
= 
\begin{bmatrix}
I_3 & (\vec{B} \vec{P})
\end{bmatrix}
\begin{bmatrix}
\vec{\nu}_B \\
\vec{\omega}
\end{bmatrix}
\tag{1.5}
\]
\( \tau_{PB} \) is called the kinematic model (or jacobian) between points \( P \) and \( B \).

Property: \( \tau_{PB}^{-1} = \tau_{BP} \).

Considering the dual force vectors (6 components) at point \( B \) and \( P \), one can also write:

\[
\begin{bmatrix}
    \vec{F}_{ext} \\
    \vec{T}_{ext,B}
\end{bmatrix}
= \begin{bmatrix}
    I_3 & 0_{3\times3} \\
    -(\vec{PB})^* & I_3
\end{bmatrix}
\begin{bmatrix}
    \vec{F}_{ext} \\
    \vec{T}_{ext,P}
\end{bmatrix},
\]

(1.6)

Indeed: \( \vec{T}_{ext,B} = \vec{T}_{ext,P} + \vec{BP} \wedge \vec{F}_{ext} \).

The time-derivation of (1.3) in the inertial frame gives:

\[
\vec{a}_P = \vec{a}_B + \vec{PB} \wedge \vec{\omega} + \left. \frac{d\vec{PB}}{dt} \right|_B \wedge \vec{\omega} + \vec{\omega} \wedge (\vec{PB} \wedge \vec{\omega}),
\]

(1.7)

or equivalently:

\[
\begin{bmatrix}
    \frac{d\vec{a}_P}{dt} \\
    \vec{\omega}
\end{bmatrix}
= \begin{bmatrix}
    I_3 & (\vec{PB})^* \\
    0_{3\times3} & I_3
\end{bmatrix}
\begin{bmatrix}
    \vec{a}_B \\
    \vec{\omega}
\end{bmatrix}
+ \begin{bmatrix}
    (\vec{\omega})^*(\vec{PB}) \vec{\omega} \\
    0_{3\times1}
\end{bmatrix}.
\]

(1.8)

Using (1.2), (1.6) and (1.8), one can derive the NEWTON-EULER equations at point \( P \):

\[
\begin{bmatrix}
    \vec{F}_{ext} \\
    \vec{T}_{ext,P}
\end{bmatrix}
= \begin{bmatrix}
    m^B I_3 & m^B (\vec{PB})^* \\
    -m^B (\vec{PB})^* & \vec{n}_B^b - m^B (\vec{PB})^* 2
\end{bmatrix}
\begin{bmatrix}
    \frac{d\vec{a}_P}{dt} \\
    \vec{\omega}
\end{bmatrix}
+ \begin{bmatrix}
    m^B (\vec{\omega})^*(\vec{PB}) \vec{\omega} \\
    (\vec{\omega})^*(\vec{n}_B^b - m^B (\vec{PB})^* 2 - \vec{\omega})
\end{bmatrix}
\]

(1.9)

### 1.3 Linearized Newton-Euler equations

Assuming that \( \vec{\omega} \) is small (linearity assumption [H2]), quadratic terms in \( (\vec{\omega})^* X_{3\times3} \vec{\omega} \) can be neglected. Then equations (1.2) and (1.9) becomes:

\[
\begin{bmatrix}
    \vec{F}_{ext} \\
    \vec{T}_{ext,B}
\end{bmatrix}
= \begin{bmatrix}
    m^B I_3 & 0_{3\times3} \\
    0_{3\times3} & \vec{n}_B^b
\end{bmatrix}
\begin{bmatrix}
    \frac{d\vec{a}_B}{dt} \\
    \vec{\omega}
\end{bmatrix}
\]

\( \vec{n}_B^b \) is the direct dynamics model of the body \( B \) about the centre of mass \( B \).
• \[
\begin{bmatrix}
\frac{\mathbf{F}_{\text{ext}}}{\mathbf{T}_{\text{ext,P}}}
\end{bmatrix} = \begin{bmatrix}
\frac{m_B \mathbf{I}_3 - m_B (\mathbf{B}/\mathbf{P})}{D_B^{\mathbf{r}_B\rightarrow\mathbf{r}_B}} & \frac{m_B (\mathbf{B}/\mathbf{P})}{\mathbf{I}_B - m_B (\mathbf{B}/\mathbf{P})^2} & \frac{\mathbf{d}_P}{\omega}
\end{bmatrix}
\]

\[D_B^{\mathbf{r}_P} = \tau_{BP} D_B^{\mathbf{r}_B \tau_{BP}}\] (1.10)

is the direct dynamics model of the body \(B\) about the point \(P\).

Equations (1.10) is very convenient to transport the direct dynamics model from one point to another: considering a third point \(R\), one can also write:

\[D_B^{\mathbf{r}_R} = \tau_{BR} D_B^{\mathbf{r}_B \tau_{BR}} = \tau_{BR}^T \tau_{PB} D_B^{\mathbf{r}_P \tau_{PB} \tau_{BR}}\] (1.11)

\[D_B^{\mathbf{r}_R} = \tau_{PR}^T D_B^{\mathbf{r}_P \tau_{PR}}.\] (1.12)

Without loss of generality, one can write the linear dynamics model of a mechanical system at its center of mass \(B\) and then transport it to any other point \(R\) (where the external force is applied for instance).

The development of (1.10) is:

\[D_B^{\mathbf{r}_P} = \tau_{BP} D_B^{\mathbf{r}_B \tau_{BP}} = \begin{bmatrix}
m_B \mathbf{I}_3 & m_B (\mathbf{B}/\mathbf{P}) & \mathbf{I}_B - m_B (\mathbf{B}/\mathbf{P})^2
\end{bmatrix}
\]

One can recognize in the bottom-right-hand term, the HUYGENS theorem between the inertia tensor \(\mathbb{I}_B\) of \(B\) at point \(B\) and the inertia tensor \(\mathbb{I}_P\) of \(B\) at point \(P\):

\[\mathbb{I}_P = \mathbb{I}_B - m_B (\mathbf{B}/\mathbf{P})^2\] (1.13)

or in projection in the frame \(R_b = (0, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)\) attached to the body \(B\), with \(\mathbf{B}/\mathbf{P}\) :

\[\mathbb{I}_P^{\mathbb{R}_b} = \mathbb{I}_B^{\mathbb{R}_b} + m_B \begin{bmatrix}
y^2 + z^2 & -xy & -xz
-yx & x^2 + z^2 & -yz
-xz & -yz & x^2 + y^2
\end{bmatrix}^{\mathbb{R}_b}\] (1.14)

Note that the equation (1.12) works on the \(6 \times 6\) direct dynamics model and is valid for any points \((P\) and \(R\)) on the body \(B\) while HUYGENS theorem (equation (1.13)) works on the \(3 \times 3\) tensor of inertia and is valid only if \(B\) coincides with the centre of mass of the body \(B\). In the general case:

\[\mathbb{I}_P^{\mathbb{R}_b} \neq \mathbb{I}_P - m_B (\mathbf{B}/\mathbf{P})^2.\]
Nevertheless Huygens theorem (equation (1.13)) can be used to find $I^B_B$ from $I^B_O$, in the case where the data of the tensor of inertia are provided w.r.t. the body reference frame $R_B$:

$$I^B_B = I^B_O + m^B (\overrightarrow{OB})^2. \quad (1.15)$$

Indeed, in the MATLAB® implementation of the SDT (detailed in chapter 3), the tensor of inertia of each body must be defined w.r.t. to its centre of mass by the user. In some applications, data are provided at a reference point $O$ different from the centre of mass $B$, then equation (1.15) can then be used and can be easily implemented in the SDT (see section 3.2.2 as a tutorial).
Chapter 2

Satellite Dynamics Toolbox: principles

We consider a satellite as a main body $B$ (the hub) and one appendage $A$ cantilevered to the hub at point $P$, as shown on Figure 2.1. The appendage is considered as a dynamic sub-structure either because of its flexibility or because of an embedded angular momentum (reaction wheels or CMG). The objective is two-fold: first, computing the direct dynamics model $M_{B}^{A+B}(s)$, i.e. the 6 by 6 transfer between the dual vector of the translational and rotational accelerations $\left[ \frac{\ddot{d}_{B}}{\dot{\omega}} \right]$ of the hub seen from the hub centre of mass $B$ and the dual vector of the external forces and torques $\left[ \frac{\overrightarrow{F}_{\text{ext}}}{\overrightarrow{T}_{\text{ext},B}} \right]$ applied to the hub at point $B$:

$$\left[ \frac{\overrightarrow{F}_{\text{ext}}}{\overrightarrow{T}_{\text{ext},B}} \right] = M_{B}^{A+B}(s) \left[ \frac{\ddot{d}_{B}}{\dot{\omega}} \right].$$

Second, computing the inverse dynamics model $[M_{B}^{A+B}(s)]^{-1}$.

Considering the result of the previous chapter (equation (1.10)), one can then compute the direct and inverse dynamics models about any arbitrary reference point $R$, that is: the models between dual vectors of accelerations $\left[ \frac{\ddot{d}_{R}}{\dot{\omega}} \right]$ and forces $\left[ \frac{\overrightarrow{F}_{\text{ext}}}{\overrightarrow{T}_{\text{ext},R}} \right]$ seen from the reference point $R$:

$$M_{R}^{A+B}(s) = \tau_{BR}^{T} M_{B}^{A+B}(s) \tau_{BR}$$

$$[M_{R}^{A+B}(s)]^{-1} = \tau_{RB} [M_{B}^{A+B}(s)]^{-1} \tau_{RB}^{T}. \quad (2.2)$$
2.1 Assumptions

[H1] The hub is rigid: \( \frac{d \vec{r}_{B/P}}{dt} \big|_{R_b} = 0. \)

[H2] Non-linear terms (in \( \vec{\omega} \wedge X_{3 \times 3} \vec{\omega} \)) of second or higher order are disregarded.

[H3] The only external force (resp. torque) applied to the appendage \( A \) is the force \( \vec{F}_{B/A} \) (resp. torque \( \vec{T}_{B/A,P} \)) applied by the hub \( B \) at the appendage’s anchorage point \( P \).

2.2 Direct dynamics model

The direct dynamics model of the assembly \( M_{B}^{A+B}(s) \) at the hub \( B \) centre of mass is the sum ([2], [4]):

- of the hub’s direct dynamics model at point \( B \): \( D_{B}^{B} \):
  \[
  D_{B}^{B} = \begin{bmatrix}
  m_{B} I_{3} & 0_{3 \times 3} \\
  0_{3 \times 3} & \vec{I}_{B}^{B}
  \end{bmatrix}
  \]
  which can be projected in frame \( R_{b} \) as follows:
  \[
  \left[ D_{B}^{B} \right]_{R_b} = \begin{bmatrix}
  m_{B} I_{3} & 0_{3 \times 3} \\
  0_{3 \times 3} & \left[ \vec{I}_{B}^{B} \right]_{R_b}
  \end{bmatrix}_{R_b},
  \]

- and of the appendage’s direct dynamics model at point \( P \): \( M_{P}^{A}(s) \) moved to point \( B \) thanks to the kinematic model \( \tau_{PB} \) (see nomenclature) and the transport equation (1.10), that leads to:

\[
M_{B}^{A}(s) = \tau_{PB}^{T} M_{P}^{A}(s) \tau_{PB}.
\]  (2.3)
When summing, we can write that:

\[ M_B^{A+B}(s) = D_B^B + \tau_{PB}^T M_F^A(s) \tau_{PB} . \]  

In projection in the hub’s frame \( \mathcal{R}_b \), the assembly’s direct dynamics model is the following:

\[ [M_B^{A+B}]_{\mathcal{R}_b}(s) = [D_B^B]_{\mathcal{R}_b} + [\tau_{PB}]_{\mathcal{R}_b} \begin{bmatrix} T_{ba} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{ba} \end{bmatrix} [M_F^A(s)]_{\mathcal{R}_a} \begin{bmatrix} T_{ba} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{ba} \end{bmatrix}^T [\tau_{PB}]_{\mathcal{R}_b} \]

where \( T_{ba} \) is the direction cosine matrix of the rotation from frame \( \mathcal{R}_b \) to frame \( \mathcal{R}_a \) (see nomenclature). Figure 2.2 represents this model’s block diagram and matching physical signals.

The following operation allows for the transport of the direct dynamics model to point \( G \), the global assembly’s centre of mass:

\[ M_G^{A+B}(s) = \tau_{BG}^T M_B^{A+B}(s) \tau_{BG} . \]

### 2.3 Appendage dynamics model \( M_P^A(s) \)

There are 3 different cases to consider in order to write the appendage’s dynamics model \( M_P^A(s) \), at anchoring point \( P \) between \( A \) and \( B \) (see [2]).
2.3.1 Case of a rigid appendage

\[ M^A_P(s) = D^A_P = \tau^T_{AP} \begin{bmatrix} m^A I_3 & 0_{3\times3} \\ 0_{3\times3} & I^A_A \end{bmatrix} \tau_{AP}. \] (2.5)

This is a static model.

2.3.2 Case of a flexible appendage

\[ M^A_P(s) = D^A_P - \sum_{i=1}^{N} l^T_{i,P} l_{i,P} \frac{s^2}{s^2 + 2\xi_i \omega_i s + \omega_i^2}, \] (2.6)

in which \( D^A_P \) comes from (2.5) and the various parameters \( \omega_i, \xi_i \) et \( l_{i,P} \) (see nomenclature) can be provided by finite element software used to model the appendage (see [2] or [3] for more explanations). Other representations can be used for \( M^A_P(s) \):

- the hybrid-cantilevered model:

\[
\begin{bmatrix}
\overrightarrow{\dot{F}_{B/A}} \\
\overrightarrow{\dot{T}_{B/A,P}}
\end{bmatrix} = D^A_P \begin{bmatrix}
\overrightarrow{\ddot{a}}_P \\
\overrightarrow{\ddot{\omega}}
\end{bmatrix} + L^T_P \dot{\eta} \\
\dot{\eta} + \text{diag}(2\xi_i \omega_i) \dot{\eta} + \text{diag}(\omega_i^2) \eta = -L_P \begin{bmatrix}
\overrightarrow{\ddot{a}}_P \\
\overrightarrow{\ddot{\omega}}
\end{bmatrix}
\]

- the second order generic model:

\[
\begin{bmatrix}
D^A_P & L^T_P \\
L_P & I_N
\end{bmatrix} \begin{bmatrix}
\dot{\eta} \\
\dot{\eta}
\end{bmatrix} + \begin{bmatrix}
0_{6\times6} & 0_{6\times N} \\
0_{N\times6} & \text{diag}(2\xi_i \omega_i)
\end{bmatrix} \begin{bmatrix}
\dot{\eta} \\
\dot{\eta}
\end{bmatrix} + \begin{bmatrix}
0_{6\times6} & 0_{6\times N} \\
0_{N\times6} & \text{diag}(\omega_i^2)
\end{bmatrix} \begin{bmatrix}
\eta \\
\eta
\end{bmatrix} = \begin{bmatrix}
F_{ext} \\
0_{N\times1}
\end{bmatrix}
\]

where \( \ddot{\eta} = \begin{bmatrix}
\overrightarrow{\ddot{a}}_P \\
\overrightarrow{\ddot{\omega}}
\end{bmatrix} \) and \( F_{ext} = \begin{bmatrix}
\overrightarrow{\dot{F}_{B/A}} \\
\overrightarrow{T_{B/A,P}}
\end{bmatrix} \).

- the state-space representation with a feedforward matrix \( D^A_{P_0} \), which is the residual mass of the appendage \( A \) rigidly cantilevered to the hub \( B \) at point \( P \):

\[
\begin{bmatrix}
\dot{\eta} \\
\dot{\eta}
\end{bmatrix} = \begin{bmatrix}
0_{N\times N} & I_N \\
-\text{diag}(\omega_i^2) & -\text{diag}(2\xi_i \omega_i)
\end{bmatrix} \begin{bmatrix}
\eta \\
\dot{\eta}
\end{bmatrix} + \begin{bmatrix}
0_{N\times6} \\
-L_P
\end{bmatrix} \begin{bmatrix}
\overrightarrow{\ddot{a}}_P \\
\overrightarrow{\ddot{\omega}}
\end{bmatrix} + \begin{bmatrix}
D^A_P - L^T_P L_P \\
D^A_{P_0}
\end{bmatrix} \begin{bmatrix}
\overrightarrow{\ddot{a}}_P \\
\overrightarrow{\ddot{\omega}}
\end{bmatrix}
\]

- or the block diagram on Figure 2.3. This diagram easily allows for the introduction of parametric uncertainties on \( \omega_i, \xi_i, L_P \) and \( D^A_{P_0} \) because they appear with a minimum number of occurences.
2.3 Appendix dynamics model $M^A_P(s)$

2.3.3 Case of an embedded angular momentum

$$M^A_P(s) = D^A_P + \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & -\frac{1}{s}I_w \Omega (\vec{z}_w) \end{bmatrix},$$

(2.7)

where $D^A_P$ comes from (2.5), where $I_w$ (Kg m$^2$) is the appendage’s inertia about its spin axis (spinning top) defined by unit vector $\vec{z}_w$ and where $\Omega$ is the spinning top’s speed (rad/s).

Complementary assumptions

When the appendage is an embedded angular momentum, we will further suppose that:

[H4] $\Omega$ and $\vec{z}_w$ (the angular speed and the spin axis of the spinning top) are constants in frame $\mathcal{R}_a$,

[H5] the spinning top is balanced,

[H6] (non-restrictive) the spin axis is along the $z$ axis of the appendage body frame $\mathcal{R}_a$ (i.e. $\vec{z}_w = \vec{z}_a$).
Under assumptions [H5] and [H6], one can write:

\[
[I_A]^R_a = \begin{bmatrix}
I_r & 0 & 0 \\
0 & I_r & 0 \\
0 & 0 & I_w
\end{bmatrix}^R_a
\]

where \(I_w\) is the spinning top’s principal inertia (about \(\vec{z}_w\)) and \(I_r\) its radial inertia.

When expressed in frame \(R_a\), equation (2.7) becomes:

\[
[M_P^A]_{R_a}(s) = [\tau_{AP}^T]_{R_a} \begin{bmatrix}
m^A I_3 & 0_{3x3} \\
0_{3x3} & [I_A]^R_a
\end{bmatrix}_{R_a} [\tau_{AP}]_{R_a} + \begin{bmatrix}
0_{3x3} \\
0_{3x3}
\end{bmatrix}_{R_a} \begin{bmatrix}
0_{3x3} \\
0_{3x3}
\end{bmatrix}_{R_a} \begin{bmatrix}
0 & 1/s & 0 \\
-1/s & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}_{R_a}.
\]

(2.8)

**Proof of the model (2.7)**

At point \(A\), the appendage’s centre of mass, one can write:

\[
\vec{F}_{B/A} = m^A \vec{d}_A
\]

(2.9)

\[
\vec{T}_{B/A,A} = \frac{d\vec{H}^A}{dt}|_{R_b} + \vec{\omega} \wedge \vec{H}^A = \frac{d\vec{H}^A}{dt}|_{R_a} + \vec{\omega} \wedge \vec{H}^A \quad \text{from [H1]}
\]

(2.10)

where \(\vec{H}^A = I_A^A \vec{\omega} + I_w \vec{\omega}_w\) is the total angular momentum of the appendage.

\[
\vec{T}_{B/A,A} = \begin{bmatrix}
I_A^A \vec{\omega} \\
I_w \vec{\omega}_w
\end{bmatrix}_{R_a} \begin{bmatrix}
d(\vec{\omega}_w)_{R_a} \\
\vec{\omega}_w \wedge \vec{\omega} + \vec{\omega} \wedge \vec{\omega}_w
\end{bmatrix}_{R_a} = 0 \quad \text{([H4])}
\]

(2.11)

From (2.9) and (2.11), the direct dynamics model of the appendage at point \(A\) is:

\[
M_A^A(s) = \begin{bmatrix}
m^A I_3 & 0_{3x3} \\
0_{3x3} & I_A^A
\end{bmatrix} + \begin{bmatrix}
0_{3x3} \\
0_{3x3}
\end{bmatrix} \begin{bmatrix}
0_{3x3} \\
0_{3x3}
\end{bmatrix} \begin{bmatrix}
0 & 1/s & 0 \\
-1/s & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}.
\]

Transporting this model to point \(P\), we get \(M_P^A(s) = \tau_{AP}^T M_A^A(s) \tau_{AP}\), that leads to equation (2.7) when taking the first of the following remarks into account.
2.4 Inverse dynamics model

Remarks (independent of the projection frame used):

- \( \tau_{AP}^T = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\frac{1}{s}I_w\Omega (^{\ast}z_w) \end{bmatrix} \tau_{AP} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\frac{1}{s}I_w\Omega (^{\ast}z_w) \end{bmatrix} \), \( \forall \tau_{AP} \).

- It clearly appears in (2.8) that the model of an embedded angular momentum appendage is a second order 6 \( \times \) 6 model containing 2 integrators. These 2 integrators were used to model the hub’s rotation vector \( \vec{\omega} \) from the angular acceleration \( \ddot{\vec{\omega}} \) (equation (2.11)), that is the last 3 inputs of direct dynamics model.

- When computing a minimal realization of the inverse dynamics model \( [M_B^{A+B}(s)]^{-1} \) augmented with the 6 integrators needed to compute the speeds from the accelerations, these previous integrators disappear.

- When considering a hub fitted with \( N_a \) embedded angular momentum appendages \( \mathcal{A}_i, i = 1, \cdots, N_a \) (for instance, reaction wheels), the dynamics model of the assembly \( M_B^{\sum_{i=1}^{N_a} A_i+B}(s) \), which is a 2\( N \) order model, should reduce to a second-order model because the matrix (\( ^{\ast}\vec{h} \)) is rank 2 for all vectors \( \vec{h} \) and:

\[
\sum_{i=1}^{N_a} I_{w_i}\Omega_i (^{\ast}z_{w_i}) = \frac{1}{s} \sum_{i=1}^{N_a} I_{w_i}\Omega_i (^{\ast}z_{w_i}) = \frac{1}{s} (^{\ast}\vec{h})
\]

where \( \vec{h} = \sum_{i=1}^{N_a} I_{w_i}\Omega_i z_{w_i} \) is the total angular momentum of the global assembly, containing the \( N_a \) spinning tops.

2.4 Inverse dynamics model

Figure 2.4 shows the block diagram of the inverse dynamics model \( [M_B^{A+B}(s)]^{-1} \), expressed in frame \( R_b \). It also shows that the appendage’s direct dynamics model is a feedback for the hub’s inverse dynamics model:

\[
(D_B^B + M_B^A(s))^{-1} = [D_B^B]^{-1} (I_6 + M_B^A(s)[D_B^B]^{-1})^{-1}.
\]

The usefulness of this representation instead of the direct inversion turns up when conducting sensitivity or robustness analyses with respect to parametric variations of the appendage’s model. Indeed, there is no need to invert the direct dynamics model of the appendage \( M_B^A(s) \). Moreover, one can assure that each characteristic parameter of the model \( M_B^A(s) \) (\( m^A, l_i^A, \omega_i, \xi_i, l_{i,P}, \Omega, \cdots \)) appear a minimum number of occurrences.
2.5 Revolute joint between the appendage and the hub

Let us consider Figure 2.5 in which the joint between the hub $B$ and the appendage $A$ at point $P$ is a revolute joint along $\vec{r}_a$ axis. We use the following definitions:

- $\ddot{\theta}$ is the angular acceleration inside the revolute joint:
  \[
  \ddot{\omega}_{A/B} = \ddot{\theta} \vec{r}_a \quad \text{or} \quad \begin{bmatrix} \ddot{\omega}_{A/B} \end{bmatrix}_{R_a} = \ddot{\theta} \begin{bmatrix} x_{ra} \\ y_{ra} \\ z_{ra} \end{bmatrix}_{R_a},
  \]

- $C_m$ is the torque (if present) along $(P, \vec{z}_a)$ applied by an actuator located inside the revolute joint.

The objective is to compute the augmented direct model $P_B^{A+B}(s) (7 \times 7)$ of the assembly $A + B$ such that:

\[
\begin{bmatrix}
\vec{F}_{ext} \\
\vec{T}_{ext,B} \\
C_m
\end{bmatrix}_{R_b} = P_B^{A+B}(s) \begin{bmatrix}
\vec{a}_B \\
\vec{\omega} \\
\ddot{\theta}
\end{bmatrix}_{R_b}
\]

Figure 2.4: Inverse dynamics model block diagram of the appendage $[M_{A+B}^B(s)]^{-1}$ expressed in frame $R_b$. 

ISAE/DMIA/ADIS
Page 24/58
2.5 Revolute joint between the appendage and the hub

Because of the revolute joint, the projection of the torque $\vec{T}_{B/A,P}$, exerted by the base on the appendage at point $P$, along $\vec{r}_a$ axis is either: null in case of a free revolute joint or equal to $C_m$ in case of an actuated joint.

$$C_m = \vec{T}_{B/A,P} \cdot \vec{r}_a.$$  \hspace{1cm} (2.12)

Expressing the direct dynamics model $M_A^P(s)$ of the appendage at point $P$ in frame $\mathcal{R}_a$ enables us to write that:

$$\begin{bmatrix}
\vec{F}_{B/A} \\
\vec{T}_{B/A,P} \\
C_m
\end{bmatrix}_{\mathcal{R}_a} = M_A^P(s) \begin{bmatrix}
\vec{d}_P \\
\vec{\omega} + \vec{\theta} \vec{r}_a
\end{bmatrix}_{\mathcal{R}_a}.$$  \hspace{1cm} (2.13)

From (2.12) and (2.13), one can write the augmented direct model $(7 \times 7)$ of the appendage $[P_A^B]_{\mathcal{R}_a}(s)$ at point $P$ and expressed in frame $\mathcal{R}_a$:

$$\begin{bmatrix}
\vec{F}_{B/A} \\
\vec{T}_{B/A,P} \\
C_m
\end{bmatrix}_{\mathcal{R}_a} = \begin{bmatrix} I_6 & 0 & 0 & 0 & x_{ra} & y_{ra} & z_{ra} \end{bmatrix}_{\mathcal{R}_a} [M_A^P]_{\mathcal{R}_a}(s) \begin{bmatrix} I_6 \\
0 \\
0 \\
x_{ra} \\
y_{ra} \\
z_{ra}
\end{bmatrix}_{\mathcal{R}_a} \begin{bmatrix}
\vec{d}_P \\
\vec{\omega} + \vec{\theta} \vec{r}_a
\end{bmatrix}_{\mathcal{R}_a}.$$  \hspace{1cm} (2.14)

The block diagram of Figure 2.6 represents this operation. It also shows the connection of the first six inputs and outputs between $[P_A^B]_{\mathcal{R}_a}(s)$ and the hub’s direct model $D_B^B$ in order to get the assembly model $P_{A+B}^B(s)$, expressed in frame $\mathcal{R}_b$.  

Figure 2.5: Assembly of the base $B$ and the appendage $A$ linked with a revolute joint along $\vec{z}_a$. 

Because of the revolute joint, the projection of the torque $\vec{T}_{B/A,P}$, exerted by the base on the appendage at point $P$, along $\vec{r}_a$ axis is either: null in case of a free revolute joint or equal to $C_m$ in case of an actuated joint.
Taking into account a revolute joint between the hub and an appendage with an embedded angular momentum then allows for the modelling of CMGs (Control Moment Gyros), the axis of the joint being the precession axis of the CMG.

\[
\begin{bmatrix}
T_{ba} & 0_{3 \times 3} \\
0_{3 \times 3} & T_{ba}
\end{bmatrix}
\]

For control laws synthesis, one can use the following inverse model:

\[
\begin{bmatrix}
\frac{\vec{a}_B}{\omega} \\
\frac{\vec{d}_B}{\omega} \\
\dot{\theta}
\end{bmatrix}_{\mathcal{R}_b} = \left[ P_{B+A}^{A+B} \right]_{\mathcal{R}_b}^{-1} \begin{bmatrix}
\vec{T}_{ext} \\
\vec{T}_{ext,B}
\end{bmatrix}_{\mathcal{R}_b}
\]

(2.15)

that allows the user to introduce, between the seventh input to the seventh output, a local model of the joint mechanism or controller \(K(s)\) according to Figure 2.7.

### 2.6 Generalisation

One can generalise the previous approach:

- to take \(N_a\) appendages \(\mathcal{A}_i, i = 1, \ldots, N_a\) linked with the hub at points \(P_i\) into account,
2.6 Generalisation

\[ \begin{bmatrix} \overrightarrow{P}_{ext} \\ T_{ext,B} \end{bmatrix} \xrightarrow{\mathcal{R}_b} \begin{bmatrix} \overrightarrow{d}_B \\ \overrightarrow{\omega} \end{bmatrix} \xrightarrow{\mathcal{R}_b} \left[ P_B^{A+B} \right]^{-1} (s) \xrightarrow{C_m} \frac{1}{2} \dot{\theta} \xrightarrow{\mathcal{R}_b} \theta \]

\[ K(s) \]

Figure 2.7: Inverse dynamics model of the assembly hub + appendage with a revolute joint and a local mechanism or controller model \( K(s) \), expressed in frame \( \mathcal{R}_b \).

- to take into account that an appendage \( A_i \) can itself be the support (the parent) of another appendage \( A_{ij} \) at point \( P_{ij} \). Thus, one can model any open kinematic chain of bodies that can be described through a connection tree. To do this, one has to build the global model, starting from the tree leaves and using the previous approach where the body \( A_i \) is considered as a base (hub) or the parent of appendage \( A_{ij} \). However, because of assumption [H1], each intermediate body must be rigid. Flexible bodies are allowed only at the end of the kinematic chain.

The intrinsic formulae (they do not depend on a projection frame) that enable transport (2.3) or addition (2.4) of direct dynamics models allow to build the global model and to use it recursively. The SDT toolbox was developed on this principle.

For example, one can write the global direct dynamics model of the system shown on Figure 2.8 as follows:

\[
M_B^B \sum A_i + B(s) = D_B^B + \tau_{P_{1,B}} \left( D_{P_1}^{A_1} + \tau_{P_{1,B}}^T \tau_{P_{1,A_1}} M_{P_{1,A_1}}^B (s) \tau_{P_{1,B}} \right) \tau_{P_{1,B}} + \tau_{P_{2,B}}^T \left( M_{P_{2,B}}^{A_2} (s) \right) \tau_{P_{2,B}}.
\]

One can also use this generalization to model revolute joints between elements, as detailed in the previous section. Thus, it allows for the modelling of a gyroscopic actuator (CMG) fitted on a hub using three bodies as shown on Figure 2.9:

- the hub \( B \),
- the fork \( A_i \) connected through a revolute joint (precession axis \( \vec{r}_{ui} = \vec{z}_{ui} \)) with \( B \) at point \( P_i \),
- the spinning top \( A_{ii} \equiv W_i \) linked with \( A_i \) at point \( P_{ii} = G_i \) and with its angular momentum along \( \vec{z}_{w_i} \) (according to the assumptions [H6]).
Figure 2.8: Example of an open kinematic chain

Figure 2.9: Illustration of a CMG.
Chapter 3

Implementation with Matlab®

The MATLAB® package SDT_V1.3.zip to implement the Satellite Dynamics Toolbox can be downloaded from:
This toolbox requires a user defined data file describing the structure of the spacecraft. The variables to be declared by such a script file are described in section 3.2. The MATLAB® package contains also several examples of such a script file detailed in sections 3.4 to 3.9 and the associated demo file demoSTD.m. It is advised to the user to run step by step this demo file and to use one of the data file examples as a guide to create its own data file. The next section describes the various functions included in the toolbox in relation with the general notations used in the document and summarized in the nomenclature. The inline help of these functions are given in appendix A.

3.1 Satellite Dynamics Toolbox functions

Once the user defined data script file is created, 5 functions can be used: main, translate_dynamic_model, rotate_dynamic_model, kinematic_model and antisym. Let us called nom_fich.m the user defined data script file:

• main.m: using the syntax mod=main(‘nom_fich’), the function outputs a structure variable mod with 5 fields:
  – mod.TotalMass: the total mass of the satellite ($\sum_{i=1}^{N_a} m^A_i + m^B$, in kg),
  – mod.TotalCg: the coordinates of the global centre of mass expressed in the reference frame $\mathcal{R}_b$: ($\overrightarrow{OG}$)$_{\mathcal{R}_b}$, in m),
  – mod.DynamicModel: the direct dynamics model expressed at the global
3. Implementation with Matlab®

centre of mass \( (G) \) in the reference frame \( R_b \): 
\[
\left[ M_G \sum_{i=1}^{N_a} A_i + B \right]_{R_b} (s) \text{ in SI units, meaning that the accelerations in input of the model are in m/s}^2 \text{ and in rad/s}^2 \text{ and that the forces and torques outputted by the model are in N and Nm),}
\]

- mod.InverseTotalModel: the above inverse model (inputs are forces, outputs are accelerations)
- mod.liste_IOs: a string of chars that describes the different channels of the (square) model mod.DynamicModel and built from the names of the bodies written in nom_fich.m. The first 6 channels are always relative to the 3 translations and the 3 rotations of the hub (base). The optional 7th and beyond channels are the transfers between the torques \( C_m \) and accelerations \( \ddot{\theta} \) local to each revolute joint linking two bodies, if activated.

- **translate_dynamic_model.m**: the syntax \( MB=\text{translate_dynamic_model}(A,B,MA) \) translates (or transports) the direct dynamics model \( MA \) projected in a frame \( R_i \) from point A (vector of the 3 components of the point A in the frame \( R_i \)) to point B (vector of the 3 components of the point B in the frame \( R_i \)). Inputs and output arguments are expressed in the same frame \( R_i \):
\[
[M_B]_{R_i} (s) = \left[ \begin{array}{ccc} \tau_{AB}^T & 0_{6 \times N_r} & 0_{6 \times N_r} \\ 0_{N_r \times 6} & I_{N_r} & 0_{N_r \times 6} \end{array} \right] [M_A]_{R_i} (s) \left[ \begin{array}{ccc} \tau_{AB} & 0_{6 \times N_r} & 0_{6 \times N_r} \\ 0_{N_r \times 6} & I_{N_r} & 0_{N_r \times 6} \end{array} \right],
\]
where \( N_r \) is the number of revolute joints in the structures.

- **rotate_dynamic_model.m**: the syntax \( MB=\text{rotate_dynamic_model}(MA,ANG,AXIS) \) computes the direct or the inverse dynamics model (input argument \( MA \)) projected in a frame \( R_i \) after a rotation of \( ANG \) (deg) around the axis \( AXIS \) (unit vector of the 3 components of the rotation axis \( \vec{r}_a \) in the frame \( R_i \)). Inputs and output arguments are expressed in the same frame \( R_i \). This function computes the direction cosine matrix \( T \) associated to the rotation and then:
\[
[M_B]_{R_i} (s) = \left[ \begin{array}{ccc} T & 0_{3 \times 3} & 0_{3 \times N_r} \\ 0_{3 \times N_r} & T & 0_{3 \times N_r} \\ 0_{N_r \times 3} & 0_{N_r \times 3} & I_{N_r} \end{array} \right] [M_A]_{R_i} (s) \left[ \begin{array}{ccc} T^T & 0_{3 \times 3} & 0_{3 \times N_r} \\ 0_{3 \times N_r} & T^T & 0_{3 \times N_r} \\ 0_{N_r \times 3} & 0_{N_r \times 3} & I_{N_r} \end{array} \right],
\]
where \( N_r \) is the number of revolute joints in the structures.

- **kinematic_model.m**: the syntax \( TAU=\text{kinematic_model}(A,B) \) computes the kinematic model \( TAU \) between points A and B (vectors of the 3 components of points A and B in a frame \( R_i \)):
\[
[\tau_{AB}]_{R_i} = \left[ \begin{array}{ccc} I_3 & \left( \overrightarrow{AB} \right)_{R_i} \\ 0_{3 \times 3} & I_3 \end{array} \right].
\]
• **antisym.m**: the syntax $M = \text{antisym}(V)$ computes the skew symmetric matrix $M = \left(\begin{array}{c}0 \\
abla V^{\top}
abla\end{array}\right)_{R_i}$ associated to the vector $V = \overrightarrow{V}$ (3 components) projected in the frame $R_i$ (see nomenclature).

### 3.2 User defined data file nom_fich.m

From a reference frame $R_b$, the user-defined data file `nom_fich.m` describes the geometry, the dynamics parameters of the main body (hub) and of the different appendages $A_i$. Running this file through:

```matlab
>> nom_fich
```

must creates 3 variables:

- **MB**: structure describing the main body $B$. The different required fields of this structure are detailed in table 3.1,
- **nappend**: the number of appendages $N_a$ (integer),
- **SA**: vector of $N_a$ cells. Each cell $SA\{i\} = 1:nappend$, describes the appendage $A_i$. The different required fields of the structure $SA\{i\}$ are detailed in table 3.2.

### 3.2.1 Remarks

One can also define appendage $SA\{i\}$ in an independent data file `nom_fich_i.m` (see table 3.2, line 6), that creates the same variables $MB$, nappend and SA as `nom_fich.m`. It allows the user:

- to use the recursivity of `main.m` in order to model chains of rigid bodies (see example data files `SpaceRoboticArm.m` and files `Segment1.m` to `Segment6.m` provided with the toolbox. They model a robotic manipulator on a space platform).
- to use only one data file when the platform is fitted with several identical appendages. We can meet this case when modelling a platform fitted with a cluster of gyroscopic actuators (see example data file presented in section 3.7).
### 3. Implementation with Matlab

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Notation</th>
<th>Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB.Name</td>
<td>char</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB.cg</td>
<td>[(\overrightarrow{OB})]<em>{R</em>{B}} m^3</td>
<td>double (3 \times 1)</td>
<td>m</td>
</tr>
<tr>
<td>MB.m</td>
<td>double</td>
<td></td>
<td>kg</td>
</tr>
<tr>
<td>MB.Ixx</td>
<td>([I^B_{B,R_{B}}(1,1)])</td>
<td>double</td>
<td>Kgm^2</td>
</tr>
<tr>
<td>MB.Iyy</td>
<td>([I^B_{B,R_{B}}(2,2)])</td>
<td>double</td>
<td>Kgm^2</td>
</tr>
<tr>
<td>MB.Izz</td>
<td>([I^B_{B,R_{B}}(3,3)])</td>
<td>double</td>
<td>Kgm^2</td>
</tr>
<tr>
<td>MB.Ixy</td>
<td>([I^B_{B,R_{B}}(1,2)])</td>
<td>double</td>
<td>Kgm^2</td>
</tr>
<tr>
<td>MB.Ixz</td>
<td>([I^B_{B,R_{B}}(1,3)])</td>
<td>double</td>
<td>Kgm^2</td>
</tr>
<tr>
<td>MB.Iyz</td>
<td>([I^B_{B,R_{B}}(2,3)])</td>
<td>double</td>
<td>Kgm^2</td>
</tr>
<tr>
<td>nappend</td>
<td>(N_a)</td>
<td>integer</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Description of the fields of the structure MB.

#### 3.2.2 A first tutorial (on inertia tensor specified at any reference point)

In the user defined data file `nom_fich.m`, the tensor of inertia of each body (fields `MB.I**` and `SA{i}.I**`) must be defined w.r.t. to its centre of mass. In some applications, data are provided at a reference point \(O\) different from the centre of mass \(B\), then Huygens thereom (1.15) can be used in the script file `nom_fich.m` by using the following Matlab® sequence based on the function `antisym`:

```matlab
MB.Name='Platform'; % Name of the body
MB.cg = [0.0063;-0.0200;1.7744]; % position of gravity center in \((0,X,Y,Z)\) (m)
MB.m = 905.6290; % mass (kg)

MB.IO=[3.8991*1.0e+03 0.0086*1.0e+03 -0.0265*1.0e+03;...
0.0086*1.0e+03 4.0907*1.0e+03 0.0114*1.0e+03;...
-0.0265*1.0e+03 0.0114*1.0e+03 0.4606*1.0e+03]; % (Kg m^2)

MB.I=MB.IO + MB.m*antisym(MB.cg)^2; % Huygens theorem
MB.Ixx=MB.I(1,1); % Main Inertia in \((0,X,Y,Z)\) (kgm^2)
MB.Iyy=MB.I(2,2); % (CG1,X1,Y1,Z1) (kgm^2)
MB.Izz=MB.I(3,3); % (CG1,X1,Y1,Z1) (kgm^2)
MB.Ixy=MB.I(1,2); % Cross Inertia in \((0,X,Y,Z)\) (kgm^2)
MB.Ixz=MB.I(1,3); % (CG1,X1,Y1,Z1) (kgm^2)
MB.Iyz=MB.I(2,3);
```
In this example the Huygens theorem is applied to the main body MB but the same procedure can be applied to any appendage SA\{i\}.

3.3 FAQ

- *The field liste_1Os returned by the function main is wrong (?)*: the function main.m is called recursively and uses persistent variables. In a nominal use, the persistent variables are managed inside the function main.m but if the function returned on an error message (due to wrong data in nom_fich.m for instance), it is then required to clear persistent variables for the next call by the command:

  clear main

- *The fields DynamicModel or InverseTotalModel returned by the function main contains some NaN (?)*: the mass or inertia seen from one of the d.o.fs is null. There is a mistake in the file nom_fich.m.

- *The ss model returned by the function main in the field DynamicModel or InverseTotalModel contains unstable dynamics (?)*: the mass or inertia of one or more bodies is not definite positive. There is a mistake in the file nom_fich.m.
<table>
<thead>
<tr>
<th>Variable name</th>
<th>Notation</th>
<th>Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SA{i}.Name</td>
<td>char</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SA{i}.P</td>
<td>double $3 \times 1$</td>
<td>$m$</td>
</tr>
<tr>
<td>3</td>
<td>SA{i}.TM</td>
<td>$T_{ba}$</td>
<td>double $3 \times 3$</td>
</tr>
<tr>
<td>4(*)</td>
<td>SA{i}.pivot</td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td>5(*)</td>
<td>SA{i}.pivotdir</td>
<td>$[\vec{r}<em>a]</em>{Ra} = [x_{ra} \ y_{ra} \ z_{ra}]^T$</td>
<td>double $3 \times 1$</td>
</tr>
<tr>
<td>6(**)</td>
<td>SA{i}.angle</td>
<td>$\theta$</td>
<td>double</td>
</tr>
<tr>
<td>7(***))</td>
<td>SA{i}.filename</td>
<td>char</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SA{i}.cg</td>
<td>double $3 \times 1$</td>
<td>$m$</td>
</tr>
<tr>
<td>9(#)</td>
<td>SA{i}.omega</td>
<td>$\Omega$</td>
<td>double</td>
</tr>
<tr>
<td>10</td>
<td>SA{i}.m</td>
<td>$m^A$</td>
<td>double</td>
</tr>
<tr>
<td>11</td>
<td>SA{i}.Ixx</td>
<td>$I_{Aa}^A R_a$ (1,1)</td>
<td>double</td>
</tr>
<tr>
<td>12</td>
<td>SA{i}.Iyy</td>
<td>$I_{Aa}^A R_a$ (2,2)</td>
<td>double</td>
</tr>
<tr>
<td>13</td>
<td>SA{i}.Izz</td>
<td>$I_{Aa}^A R_a$ (3,3)</td>
<td>double</td>
</tr>
<tr>
<td>14</td>
<td>SA{i}.Ixy</td>
<td>$I_{Aa}^A R_a$ (1,2)</td>
<td>double</td>
</tr>
<tr>
<td>15</td>
<td>SA{i}.Ixz</td>
<td>$I_{Aa}^A R_a$ (1,3)</td>
<td>double</td>
</tr>
<tr>
<td>16</td>
<td>SA{i}.Iyz</td>
<td>$I_{Aa}^A R_a$ (2,3)</td>
<td>double</td>
</tr>
<tr>
<td>17(##)</td>
<td>SA{i}.flex</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>SA{i}.L</td>
<td>$[L_P]<em>{Ra}$ ou $[L_A]</em>{Ra}$</td>
<td>double $N \times 6$</td>
</tr>
<tr>
<td>19</td>
<td>SA{i}.wi</td>
<td>$[\omega_1 \ \omega_2 \ \cdots \ \omega_N]$</td>
<td>double $1 \times N$</td>
</tr>
<tr>
<td>20</td>
<td>SA{i}.z</td>
<td>$[\xi_1 \ \xi_2 \ \cdots \ \xi_N]$</td>
<td>double $1 \times N$</td>
</tr>
</tbody>
</table>

Table 3.2: Description of the fields of the structure in cell `SA{i}`.

(*) if $SA{i}.pivot=1$ then there is a revolute joint between the appendage and the hub at point $P$. The direction $\vec{r}_a$ of the revolute joint axis can be specified in the field # 5 $SA{i}.pivotdir$, otherwise the default value is $SA{i}.pivotdir=[0;0;1]$. In case of a revolute joint, the direct dynamics model will be augmented on the outputs, by the joint torque $C_m$ and on the inputs, by the joint acceleration $\ddot{\theta}$. If $SA{i}.pivot=0$ then the appendage is cantilevered to the hub $B$ at point $P$ and the fields # 5 and 6 are optional or disregarded.

(**) if $SA{i}.pivot=1$ then is possible to take into account a tilt of the appendage by $SA{i}.angle$ degrees along the revolute joint axis.

(***) if the appendage is described by a data file, then the fields from line 8 to 20 are optional or disregarded.

(#) if the field $SA{i}.omega$ exists, then the appendage is considered as an onboard angular momentum along $\vec{z}_a$. Then, $SA{i}.omega$ and $SA{i}.Izz$ are the angular rate and the main inertia of the spinning top, $SA{i}.Ixx$ is its radial inertia, fields # 12, 14, 15 and 16 are optional or disregarded and the appendage must be rigid: $SA{i}.flex=0$.

(##) if $SA{i}.flex=0$ then the appendage is supposed rigid and the fields from line 18 to 20 are optional or disregarded. If $SA{i}.flex=1$ then the appendage is flexible and $SA{i}.L$ is the list of the modal participation factors $L_A$ expressed at centre of mass $A$ of the appendage. If $SA{i}.flex=2$ then the appendage is flexible and $SA{i}.L$ is the list of the modal participation factors $L_P$ at anchoring point $P$. We remind that $L_P = L_A T_A P$.  

3.4 Example 1: Spacecraft1.m

This file describes a platform (MB.Name='Platform') fitted with 2 (nappend=2) flexible appendages (SA{1} and SA{2}) cantilevered in 2 different points of the platform:

- a solar array (SA{1}.Name = 'Solar Array') described by 3 flexible modes,
- an antenna (SA{2}.Name = 'Antenna') described by 3 flexible modes.

The frequencies $\omega_i$, $i = 1, 2, 3$ of the 3 modes are the same for both appendages. Because of the symmetry of the two modal participation factors along the z axis of each appendage, one flexible mode is uncontrollable. It is reduced when computing a minimal realization of the model.

The following MATLAB® session (see also the script file demoSTD.m) shows how to use this file with the SDT and perform complementary analyses.

```matlab
>> mod=main('Spacecraft1');
2 states removed.
>> % 1 flexible mode removed (due to symmetry of flexible appendage)
>> mod.TotalCg % the position of the total centre of mass G.
ans =
   0.1909
   0.3909
   0

>> mod.liste_IOs % list of channels
ans =
Trans. X Platform
Trans. Y Platform
Trans. Z Platform
Rot. X Platform
Rot. Y Platform
Rot. Z Platform

>> Md=mod.DynamicModel; % direct dynamic model.

>> % Translation of the direct dynamic model from the global centre of mass G
>> % to the hub’s centre of mass B:
>> Spacecraft1 % to get all the data in the workspace.
>> MD_B=translate_dynamic_model(mod.TotalCg,MB.cg,Md);
```
>> % Inverse Dynamic Model:
>> Mi=mod.InverseTotalModel;
>> damp(Mi) % The 5=6-1 flexible modes.

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.03e-02 + 8.03e+00i</td>
<td>5.02e-03</td>
<td>8.03e+00</td>
</tr>
<tr>
<td>-4.03e-02 - 8.03e+00i</td>
<td>5.02e-03</td>
<td>8.03e+00</td>
</tr>
<tr>
<td>-5.07e-02 + 9.03e+00i</td>
<td>5.61e-03</td>
<td>9.03e+00</td>
</tr>
<tr>
<td>-5.07e-02 - 9.03e+00i</td>
<td>5.61e-03</td>
<td>9.03e+00</td>
</tr>
<tr>
<td>-9.64e-02 + 1.70e+01i</td>
<td>5.67e-03</td>
<td>1.70e+01</td>
</tr>
<tr>
<td>-9.64e-02 - 1.70e+01i</td>
<td>5.67e-03</td>
<td>1.70e+01</td>
</tr>
<tr>
<td>-1.09e-01 + 1.80e+01i</td>
<td>6.02e-03</td>
<td>1.80e+01</td>
</tr>
<tr>
<td>-1.09e-01 - 1.80e+01i</td>
<td>6.02e-03</td>
<td>1.80e+01</td>
</tr>
<tr>
<td>-1.11e-01 + 2.09e+01i</td>
<td>5.29e-03</td>
<td>2.09e+01</td>
</tr>
<tr>
<td>-1.11e-01 - 2.09e+01i</td>
<td>5.29e-03</td>
<td>2.09e+01</td>
</tr>
</tbody>
</table>

(Frequencies expressed in rad/seconds)

>> figure
>> sigma(Mi) % Frequency domain response.

The frequency response is presented on Figure 3.1.

![Figure 3.1: Frequency response (Singular Values) of the inverse dynamics model.](image-url)
3.5 Example 2: Spacecraft2.m

This file describes the same platform as example 1 but now, the link between the solar array and the hub is a revolute joint along $\vec{e}_{a_1} = \vec{z}_{a_1}$. The following MATLAB® session (see also the script file demoSTD.m) shows how to use this file with the SDT and perform complementary analyses, more particularly study the response of the revolute joint’s angle to a force impulse applied at the global centre of mass. One will notice that thanks to the revolute joint, the 6 flexible modes are now both controllable and observable: computing a minimal realization of the model will not reduce the number of modes.

```matlab
>> clear all,close all,
>> mod=main('Spacecraft2');
>> mod.liste_I0s % Now the dynamic model is 7x7
ans =

Trans. X Platform
Trans. Y Platform
Trans. Z Platform
Rot. X Platform
Rot. Y Platform
Rot. Z Platform
Joint: Solar Array/Platform

>> Md=mod.DynamicModel; % direct dynamic model.
>> Mi=mod.InverseTotalModel; % inverse dynamic model
>> Mi71=Mi(7,1); % Transfer between a force applied on the
>> % platform along X axis and the revolute joint’s
>> % relative acceleration.
>> ddint=tf(1,[1 0 0]); % double integrations
>> % impulse response of Mi71/s/s :
>> figure
>> impulse(Mi71*ddint,3) % response of the revolute joint angle to
>> % an impulse on the X thruster.
```

The impulse response is shown on Figure 3.2.

Remark: The file Spacecraft3.m describes the same assembly but uses file APPENDAGE1.m to define the solar array. File Spacecraft4.m shows the usefulness of the recursivity used in function main.m: it describes the same assembly as Spacecraft3.m, with a third (more complex) appendage which is the assembly described in Spacecraft3.m...
3. Implementation with Matlab®

Figure 3.2: Impulse response of the transfer between \( \theta \), the angle of the revolute joint, and the force applied at the global centre of mass along \( X \)

3.6 Example 3: Spacecraft5.m

This file describes the same platform as example 1 but now, it is fitted with 3 more appendages (\texttt{nappend=5}) modelling embedded angular momentums along the 3 directions (\( \vec{x}_b, \vec{y}_b, \vec{z}_b \)) in the hub’s frame. Such devices can model reaction wheels (RWA), that have a very high speed (close to their saturation).

The following Matlab® session (see also the script file \texttt{demoSTD.m}) shows how to use this file with the SDT and perform complementary analyses. One will notice the following facts:

- The reduction of one uncontrollable mode (same as example 1)
- The reduction of a double pair of integrators. Indeed, to model the 3 embedded angular momentum (see the remarks in section 2.3.3), we introduced three pairs of integrators in the direct dynamics model. However, only 2 integrators are required to model the gyroscopic couplings due to the total angular momentum (i.e. the sum of the 3 embedded angular momentums).
- The gyroscopic mode in the inverse dynamics model which is represented by a pair of poles on the imaginary axis, at low frequency (around 0.1 \( rd/s \))
- The reduction of a pair of integrators when introducing the 3 integrators between the 3 angular accelerations and speeds of the hub. All the integrators added into the dynamics model to take into account gyroscopic couplings disappeared in the reduction, as expected.
>> clear all, close all,
>> mod=main('Spacecraft5');
2 states removed.
2 states removed.
2 states removed.
>> % 1 flexible mode removed (due to symmetry of flexible appendages)
>> % + 2 double integrators removed.
>> Md=mod.DynamicModel; % direct dynamic model.
>> damp(Md) % only 2 integrators to represent the 3 angular momentums.

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.36e-15</td>
<td>-1.00e+00</td>
<td>1.36e-15</td>
</tr>
<tr>
<td>-5.68e-15</td>
<td>1.00e+00</td>
<td>5.68e-15</td>
</tr>
<tr>
<td>-4.00e-02 + 8.00e+00i</td>
<td>5.00e-03</td>
<td>8.00e+00</td>
</tr>
<tr>
<td>-4.00e-02 - 8.00e+00i</td>
<td>5.00e-03</td>
<td>8.00e+00</td>
</tr>
<tr>
<td>-4.00e-02 + 8.00e+00i</td>
<td>5.00e-03</td>
<td>8.00e+00</td>
</tr>
<tr>
<td>-4.00e-02 - 8.00e+00i</td>
<td>5.00e-03</td>
<td>8.00e+00</td>
</tr>
<tr>
<td>-7.50e-02 + 1.50e+01i</td>
<td>5.00e-03</td>
<td>1.50e+01</td>
</tr>
<tr>
<td>-7.50e-02 - 1.50e+01i</td>
<td>5.00e-03</td>
<td>1.50e+01</td>
</tr>
<tr>
<td>-7.50e-02 + 1.50e+01i</td>
<td>5.00e-03</td>
<td>1.50e+01</td>
</tr>
<tr>
<td>-7.50e-02 - 1.50e+01i</td>
<td>5.00e-03</td>
<td>1.50e+01</td>
</tr>
<tr>
<td>-1.00e-01 + 2.00e+01i</td>
<td>5.00e-03</td>
<td>2.00e+01</td>
</tr>
<tr>
<td>-1.00e-01 - 2.00e+01i</td>
<td>5.00e-03</td>
<td>2.00e+01</td>
</tr>
</tbody>
</table>

(Frequencies expressed in rad/seconds)

>> Mi=mod.InverseTotalModel;
>> damp(Mi)

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.14e-11 + 9.21e-02i</td>
<td>2.32e-10</td>
<td>9.21e-02</td>
</tr>
<tr>
<td>-2.14e-11 - 9.21e-02i</td>
<td>2.32e-10</td>
<td>9.21e-02</td>
</tr>
<tr>
<td>-4.03e-02 + 8.03e+00i</td>
<td>5.02e-03</td>
<td>8.03e+00</td>
</tr>
<tr>
<td>-4.03e-02 - 8.03e+00i</td>
<td>5.02e-03</td>
<td>8.03e+00</td>
</tr>
<tr>
<td>-5.06e-02 + 9.02e+00i</td>
<td>5.61e-03</td>
<td>9.02e+00</td>
</tr>
<tr>
<td>-5.06e-02 - 9.02e+00i</td>
<td>5.61e-03</td>
<td>9.02e+00</td>
</tr>
<tr>
<td>-9.62e-02 + 1.70e+01i</td>
<td>5.67e-03</td>
<td>1.70e+01</td>
</tr>
<tr>
<td>-9.62e-02 - 1.70e+01i</td>
<td>5.67e-03</td>
<td>1.70e+01</td>
</tr>
<tr>
<td>-1.09e-01 + 1.80e+01i</td>
<td>6.02e-03</td>
<td>1.80e+01</td>
</tr>
<tr>
<td>-1.09e-01 - 1.80e+01i</td>
<td>6.02e-03</td>
<td>1.80e+01</td>
</tr>
<tr>
<td>-1.11e-01 + 2.09e+01i</td>
<td>5.29e-03</td>
<td>2.09e+01</td>
</tr>
<tr>
<td>-1.11e-01 - 2.09e+01i</td>
<td>5.29e-03</td>
<td>2.09e+01</td>
</tr>
</tbody>
</table>
3. Implementation with Matlab

(Frequencies expressed in rad/seconds)

```matlab
>> % ==> in addition to flexible modes, one can see a very low frequency
>> % (around 0.1 rd/s) mode corresponding to the gyroscopic mode
>> figure
>> sigma(Mi) % ==> a very low frequency resonance.
>> % Angular model:
>> Mi=Mi(4:6,4:6);
>> % Integrations between angular acceleration and rates:
>> int=tf(1,[1 0]);
>> Mi_int=minreal(Mi*(int*eye(3)));
2 states removed.
>> % ==> 2 states removed: extra integrators to model gyroscopic couplings
>> % are removed, as wanted.
```

The frequency-domain response is shown on Figure 3.3.

![Frequency-domain response](image)

Figure 3.3: Frequency-domain response (Singular Values) of the inverse dynamics model.

3.7 Example 4: FOUR_CMGs.m

This file describes the example of a platform fitted with a cluster of 4 identical gyroscopic actuators, in a pyramidal configuration. The geometries, weights and inertias of the different bodies are very close to those of the experimental platform TETRAGYRE developed by ONERA/DCSD (see [http://www.onera.fr/dcsd/gyrodynes/](http://www.onera.fr/dcsd/gyrodynes/))
and Figure 3.4). The default configuration is obtained when the spinning tops axes are horizontal and when the sum of the angular momentums is zero.

The main data file is `FOUR_CMGs.m`. It describes the platform (hub) $B$ and the nominal configuration of the 4 CMGs. One difference with the example files detailed above is that we add a global variable $\text{angles} (4 \times 1)$ that represents the current configuration of the precession axes. It also enables the computation of the dynamics model in any configuration without modifying the data file. This variable is initialized with the nominal angular configuration ($\text{angles}=[0;0;0;0]$). The file `dataCMG.m` describes one CMG (see Figure 2.9) and is called 4 times by the file `FOUR_CMGs.m`. The CMG is modelled as a fork $A_i$ for the main part of the appendage and a spinning top $W_i$ as the appendage’s appendage.

The following MATLAB® session (see also the script file `demoSTD.m`) shows how to use this file with the SDT, performs complementary analyses and proposes a 3-axes attitude control law. This control law has the following structure, also depicted in Figure 3.5:

- a inner control loop feedbacking the precession speeds $\dot{\sigma}_i$. It is a pure proportional control and it is the same on the 4 precession axes:
  \[ C_{m_i} = 0.06(\dot{\sigma}_{i,\text{ref}} - \dot{\sigma}_i), \forall i = 1, \ldots, 4, \]

- an outer control loop feedbacking the 3 hub attitudes. It is a pure proportional control and it is the same on the hub’s 3 rotation axes:
  \[ \begin{bmatrix} \mathbf{T}_{\text{ref}} \end{bmatrix}_{R_b} = 3[(\phi_{\text{ref}} - \phi) \ (\theta_{\text{ref}} - \theta) \ (\psi_{\text{ref}} - \psi)]^T, \]
• a guidance law for the cluster of gyroscopic actuators which is the pseudo-inverse of the Jacobian $J_0 = J(\sigma_i)|_{\sigma_i=0}$ ($3 \times 4$) computed from the 4 precession speeds to the hub’s 3 rotation speeds when in nominal configuration:

$$
\begin{bmatrix}
\dot{\sigma}_{1,\text{ref}} & \dot{\sigma}_{2,\text{ref}} & \cdots & \dot{\sigma}_{4,\text{ref}}
\end{bmatrix}^T = J_0^T (J_0 J_0^T)^{-1} \begin{bmatrix}
\vec{T}_{\text{ref}}_R
\end{bmatrix}_{\text{hub}}.
$$

Figure 3.5: Attitude control using 4 CMGs.

```matlab
>> clear all, close all,
>> mod=main(‘FOUR_CMGs’);
Angular configuration: 0 0 0 0
2 states removed.
>> mod.liste_I0s  % Definition of the various channels
ans =

Trans. X Platform
Trans. Y Platform
Trans. Z Platform
Rot. X Platform
Rot. Y Platform
Rot. Z Platform
Joint: CMG #1/Platform
Joint: CMG #2/Platform
Joint: CMG #3/Platform
Joint: CMG #4/Platform

>> Md=mod.DynamicModel;  % direct dynamic model
>> % Model restricted to 3 axes angular motion + CMGs channels:
>> Md=Md(4:10,4:10);
```
3.7 Example 4: FOUR_CMGs.m

>> % One can check that the total angular momentum is null:
>> M3x3=minreal(Md(1:3,1:3))
6 states removed.

M3x3 =

\[
\begin{array}{ccc}
\text{d} &=& \begin{bmatrix} u1 & u2 & u3 \\ y1 & 5.315 & 0 & -0.005015 \\ y2 & 0 & 5.315 & -0.005015 \\ y3 & -0.005015 & -0.005015 & 10.61 \end{bmatrix} \\
\end{array}
\]

Static gain.

>> % ==> The 3x3 model is static: no more gyroscopic couplings between platform
>> % and inertial frame.

>> Mi=inv(Md); % Inverse dynamic model.
>> damp(Mi)

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.55e-15 + 4.85e+01i</td>
<td>7.33e-17</td>
<td>4.85e+01</td>
</tr>
<tr>
<td>-3.55e-15 - 4.85e+01i</td>
<td>7.33e-17</td>
<td>4.85e+01</td>
</tr>
<tr>
<td>-5.33e-15 + 4.85e+01i</td>
<td>1.10e-16</td>
<td>4.85e+01</td>
</tr>
<tr>
<td>-5.33e-15 - 4.85e+01i</td>
<td>1.10e-16</td>
<td>4.85e+01</td>
</tr>
<tr>
<td>-7.25e-17 + 8.40e+01i</td>
<td>8.63e-19</td>
<td>8.40e+01</td>
</tr>
<tr>
<td>-7.25e-17 - 8.40e+01i</td>
<td>8.63e-19</td>
<td>8.40e+01</td>
</tr>
</tbody>
</table>

(Frequencies expressed in rad/seconds)

>> % ==> the pulsations of the 3 gyroscopic modes.

>> % Dynamic model for a new angular configuration:
>> global angles
>> angles=[10;20;-5;-25];
>> mod_bis=main('FOUR_CMGs');
Angular configuration: 10 20 -5 -25
2 states removed.
>> damp(mod_bis.InverseTotalModel)

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.24e-16 + 3.13e+01i</td>
<td>3.95e-18</td>
<td>3.13e+01</td>
</tr>
<tr>
<td>-1.24e-16 - 3.13e+01i</td>
<td>3.95e-18</td>
<td>3.13e+01</td>
</tr>
</tbody>
</table>
3. Implementation with Matlab®

\[
\begin{align*}
7.11\times10^{-15} + 6.32\times10^1i & \quad -1.12\times10^{-16} \quad 6.32\times10^1 \\
7.11\times10^{-15} - 6.32\times10^1i & \quad -1.12\times10^{-16} \quad 6.32\times10^1 \\
8.29\times10^{-17} + 8.59\times10^1i & \quad -9.65\times10^{-19} \quad 8.59\times10^1 \\
8.29\times10^{-17} - 8.59\times10^1i & \quad -9.65\times10^{-19} \quad 8.59\times10^1
\end{align*}
\]

(Frequencies expressed in rad/seconds)

\[
\begin{align*}
\text{>>} & \quad \% \implies \text{angular frequencies of gyroscopic modes have changed.} \\
\text{>>} & \quad \% \text{First integrators (angular rates on outputs on nominal model } M_i): \\
\text{>>} & \quad \text{int} = \text{tf}(1,[1 \ 0]); \\
\text{>>} & \quad Gv = \text{minreal}((\text{int} \times \text{eye}(7)) \times M_i); \\
\text{3 states removed.}
\end{align*}
\]

\[
\begin{align*}
\text{>>} & \quad \% \text{Precession rate servo-loop (inner loop):} \\
\text{>>} & \quad \text{bf2} = \text{feedback}(Gv \times \text{diag([1 1 1 0.06 0.06 0.06 0.06])},\text{eye}(4),4:7,4:7); \\
\text{>>} & \quad \text{bf2} = \text{minreal}(	ext{bf2}); \quad 3 \text{ states removed: OK}!!
\end{align*}
\]

\[
\begin{align*}
\text{>>} & \quad \% \text{Platform attitude servo-loop:} \\
\text{>>} & \quad \% \text{The jacobian is defined as the } \text{DCgain} \text{ of the transfer between the 4} \\
\text{>>} & \quad \% \text{precession rates reference inputs and the 3 platform angular rates:} \\
\text{>>} & \quad \text{Jacob} = \text{dcgain(bf2(1:3,[4:7]))}
\end{align*}
\]

\[
\begin{align*}
\text{Jacob} = \\
\begin{bmatrix}
0.0120 & -0.0120 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0120 & -0.0120 \\
0.0035 & 0.0035 & 0.0035 & 0.0035
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{>>} & \quad \% \text{Second integrators: platform angular positions and rates on outputs:}
\end{align*}
\]
Example 5: SpaceRoboticArm.m

This file describes a spatial vehicle fitted with a manipulator (arm) with 6 degrees of freedom and a high aspect ratio, as shown on Figure 3.7. We suppose that the segments are all rigid. The obtained model has 12 channels: 6 for the degrees of freedom of the hub and 6 for the manipulator. The files Segment1.m, Segment2.m, ..., Segment6.m describe the different segments. Each segment is composed of a main body (hub) and one appendage which is the next segment. The global variable config (6×1) can be used to set the angular configuration of the manipulator (same as in the previous example).

The following MATLAB® session (see also the script file demoSTD.m in the STD_V1.3 sub-directory) shows how to use this file with the SDT.
Figure 3.6: Step response of the CMG platform with attitude control, on the 3 axes

Figure 3.7: A manipulator arm fitted on a spatial vehicle

>> clear all, close all,
>> global config
>> mod=main('SpaceRoboticArm');
Angular configuration: 0 30 120 0 0 0
>> mod.liste_IOs % The 12 channels (d.o.f. order).

ans =

Trans. X Main_body
3.8 Example 5: \textsc{SpaceroBoticArm.m}

Trans. Y Main_body
Trans. Z Main_body
Rot. X Main_body
Rot. Y Main_body
Rot. Z Main_body
Joint: Link#6/Link#5
Joint: Link#5/Link#4
Joint: Link#4/Link#3
Joint: Link#3/Link#2
Joint: Link#2/Link#1
Joint: Link#1/Main_body

\texttt{>> mod.TotalCg} \quad \% \text{Position of the global centre of mass.}

\text{ans =}

\begin{verbatim}
  0.0270
-0.3741
  1.4893
\end{verbatim}

\texttt{>> mod.DynamicModel}

\text{ans =}

\text{d =}

\begin{verbatim}
  \begin{array}{cccc}
  y1 & 1400 & -6.311e-30 & 6.311e-30 & -7.272e-31 \\
  y2 & -6.311e-30 & 1400 & 3.864e-46 & 4.547e-13 \\
  y3 & 6.311e-30 & 3.864e-46 & 1400 & -3.411e-13 \\
  y5 & 0 & 3.088e-30 & -2.132e-14 & -57.02 \\
  y6 & 3.411e-13 & 2.132e-14 & -2.361e-30 & -266.3 \\
  y7 & 0 & 0 & 0 & 0.3375 \\
  y8 & 8.472e-16 & -32.74 & -18.9 & 202.1 \\
  y9 & -4.216e-15 & 25.2 & -43.65 & -241.7 \\
  y10 & -1.002e-13 & 598.8 & -1037 & -4377 \\
  y11 & -5.978e-14 & 1500 & 523.7 & -1.07e+04 \\
  y12 & 523.7 & 37.8 & 0 & -266.3 \\
  \end{array}
\end{verbatim}

\text{u5 =}

\begin{verbatim}
  \begin{array}{cccc}
  y1 & 0 & 3.411e-13 & 0 & 8.472e-16 \\
  y2 & 3.088e-30 & 2.132e-14 & 0 & -32.74 \\
  y3 & -2.132e-14 & -2.361e-30 & 0 & -18.9 \\
  \end{array}
\end{verbatim}
3. Implementation with Matlab®

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y4</td>
<td>-57.02</td>
<td>-266.3</td>
<td>0.3375</td>
<td>202.1</td>
</tr>
<tr>
<td>y5</td>
<td>1.141e+04</td>
<td>1269</td>
<td>4.133e-17</td>
<td>15.53</td>
</tr>
<tr>
<td>y6</td>
<td>1269</td>
<td>2934</td>
<td>-5.646e-17</td>
<td>-26.91</td>
</tr>
<tr>
<td>y7</td>
<td>4.133e-17</td>
<td>-5.646e-17</td>
<td>0.3375</td>
<td>2.067e-17</td>
</tr>
<tr>
<td>y8</td>
<td>15.53</td>
<td>-26.91</td>
<td>2.067e-17</td>
<td>32.09</td>
</tr>
<tr>
<td>y9</td>
<td>19.45</td>
<td>11.23</td>
<td>-0.3375</td>
<td>2.562e-15</td>
</tr>
<tr>
<td>y10</td>
<td>223.7</td>
<td>129.2</td>
<td>-0.3375</td>
<td>2.562e-15</td>
</tr>
<tr>
<td>y11</td>
<td>57.02</td>
<td>225.4</td>
<td>-0.3375</td>
<td>-208.9</td>
</tr>
<tr>
<td>y12</td>
<td>1269</td>
<td>2267</td>
<td>-5.646e-17</td>
<td>-27.79</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>u9</td>
<td>u10</td>
<td>u11</td>
<td>u12</td>
</tr>
<tr>
<td>y1</td>
<td>-4.216e-15</td>
<td>-1.002e-13</td>
<td>-5.978e-14</td>
</tr>
<tr>
<td>y2</td>
<td>25.2</td>
<td>598.8</td>
<td>1500</td>
</tr>
<tr>
<td>y3</td>
<td>-43.65</td>
<td>-1037</td>
<td>523.7</td>
</tr>
<tr>
<td>y4</td>
<td>-241.7</td>
<td>-4377</td>
<td>-1.07e+04</td>
</tr>
<tr>
<td>y5</td>
<td>19.45</td>
<td>223.7</td>
<td>57.02</td>
</tr>
<tr>
<td>y6</td>
<td>11.23</td>
<td>129.2</td>
<td>225.4</td>
</tr>
<tr>
<td>y7</td>
<td>-0.3375</td>
<td>-0.3375</td>
<td>-0.3375</td>
</tr>
<tr>
<td>y8</td>
<td>2.562e-15</td>
<td>2.562e-15</td>
<td>-208.9</td>
</tr>
<tr>
<td>y9</td>
<td>30.01</td>
<td>385.8</td>
<td>225.1</td>
</tr>
<tr>
<td>y10</td>
<td>385.8</td>
<td>7804</td>
<td>3983</td>
</tr>
<tr>
<td>y11</td>
<td>225.1</td>
<td>3983</td>
<td>1.088e+04</td>
</tr>
<tr>
<td>y12</td>
<td>11.91</td>
<td>145.3</td>
<td>265.9</td>
</tr>
</tbody>
</table>

Static gain.

>> % This model is static since all links are rigid

3.9 Example 6: Spacecraft1u.m

This version of the toolbox (SDT Version 1.3) also allows to take parametric variations of parameters into account, and more particularly the main structural parameters such as the mass, the inertias, the geometry (position of the centre of mass and position of the anchoring points), the angular frequencies, dampings and modal participation factors. To sum up, all the parameters defined by the fields of type double of the structures MB and SA{i} (see table 3.1 and 3.2) except the angular parameters (SA{i}.TM, SA{i}.pivotdir and SA{i}.angle) can be defined with parametric uncertainties. The varying parameters can be defined by the function ureal from the Robust Control Toolbox (RCT). Then, the output variables of the SDT are uncertain matrices or state-space representations and are fully compatible with the analysis tools of the RCT.

The file Spacecraft1u.m is based on example 2 and adds parametric variations on dynamics and geometric parameters of the hub and of the appendage. The following
MATLAB® session shows how to perform a sensitivity analysis of the frequency response of the inverse dynamics model (see Figure 3.8).

```matlab
>> clear all, close all, 
>> mod = main('Spacecraft1u'); 
>> mod.TotalCg % the position of the global centre of mass.
ans =

Uncertain matrix with 3 rows and 1 columns.
The uncertainty consists of the following blocks:
MB_m: Uncertain real, nominal = 100, variability = [-10, 10]%, 1 occurrences
SA1_CGx: Uncertain real, nominal = 1, variability = [-10, 10]%, 1 occurrences
SA1_Px: Uncertain real, nominal = 0, range = [-0.01, 0.01], 1 occurrences
SA1_Py: Uncertain real, nominal = 1, variability = [-10, 10]%, 1 occurrences
SA1_m: Uncertain real, nominal = 50, variability = [-10, 10]%, 1 occurrences
SA2_CGy: Uncertain real, nominal = 1, variability = [-10, 10]%, 1 occurrences
SA2_Py: Uncertain real, nominal = -1, variability = [-10, 10]%, 1 occurrences
SA2_Pz: Uncertain real, nominal = 0, range = [-0.01, 0.01], 1 occurrences
SA2_m: Uncertain real, nominal = 20, variability = [-10, 10]%, 1 occurrences

Type "ans.NominalValue" to see the nominal value, "get(ans)" to see all properties, and "ans.Uncertainty" to interact with the uncertain elements.

>> Mi = mod.InverseTotalModel; % inverse dynamic model
>> figure
>> sigma(Mi)
>> % sensitivity of the frequency domain response to parametric variations.
```

Figure 3.8: Frequency responses (Singular Values) of the inverse dynamics model for several parameter configurations.
3. Implementation with Matlab®

3.10 Exercise

Statement

Using the data file SpaceRoboticArm.m (see section 3.8) describing a chaser spacecraft with a robotic arm and the data file Spacecraft1.m (see section 3.4) describing a spacecraft with two symmetrical flexible appendages, the objective of this exercise is to build the data file to analyse the dynamic behavior of the chaser holding the spacecraft considered as a debris (see Figure 3.9). The interface point $P_7$ between the arm end effector and the debris is characterized by:

- its coordinate vector in the frame $\mathcal{R}_{a_6}$ attached to the end effector (Segment6.m):
  \[
  \begin{bmatrix}
  O_6 P_7 \\
  \end{bmatrix}
  \mathcal{R}_{a_6} = \begin{bmatrix} 0 & 0 & 0.55 \end{bmatrix}^T,
  \]

- its coordinates vector in the frame $\mathcal{R}_{a_7}$ attached to the debris spacecraft (Spacecraft1.m):
  \[
  \begin{bmatrix}
  O_7 P_7 \\
  \end{bmatrix}
  \mathcal{R}_{a_7} = \begin{bmatrix} -0.3 & 0 & -1.2 \end{bmatrix}^T.
  \]

The direction cosine matrix between frames $\mathcal{R}_{a_6}$ and $\mathcal{R}_{a_7}$ (see nomenclature) is:

\[
T_{a_6a_7} = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta \\
\end{bmatrix}
\]

with: $\theta = 80$ deg.

Solution

Firstly, the data file Segment6.m must be completed by the following instructions to take into account that Segment6 (the end-effector) hold an appendage described in the data file Spacecraft1.m

```matlab
nappend = 1;
%==========================================================================
% APPENDAGE 1 : SA{1}
% DEBRIS SPACECRAFT
%==========================================================================
SA{1}.Name = 'DEBRIS';       % Name of next appendage: the debris
SA{1}.P = [0;0;0.55];        % position of connection point P7 in (O_6,X,Y,Z) (m)
SA{1}.TM = [cos(80*pi/180) 0 sin(80*pi/180) 0 1 0 -sin(80*pi/180) 0 cos(80*pi/180)];
SA{1}.pivot = 0;             % cantilevered joint
SA{1}.filename = 'Spacecraft1';
```

ISAE/DMIA/ADIS
Figure 3.9: A manipulator arm fitted on a spatial vehicle holding a debris spacecraft.

Secondly, the data file `Spacecraft1.m` must be completed by the following instructions to take into account the new reference point \( P_7 \) (instead of the previous one: \( O_7 \)):

\[
\% \text{DEBRIS case: taking into account } P_7 \text{ and } O_7:
MB_c = MB_c + [0.3;0;1.2];
SA{1}.P = SA{1}.P + [0.3;0;1.2];
SA{2}.P = SA{2}.P + [0.3;0;1.2];
\]

Then the model of the assembly: chaser + robotic arm + debris can be computed using:

\[
>> \text{clear all, close all,}
>> \text{global config}
>> \text{mod=main('SpaceRoboticArm');}
\]
References


EMPTY PAGE
Appendix A

Inline help

A.1 help of function main.m

MODEL = main(FILENAME) compute the model MODEL of the spacecraft described in the file FILENAME.m

MODEL is a structure:
MODEL.TotalMass: total mass of the spacecraft,
MODEL.TotalCg: 3x1 coordinate vector of the global centre of mass (CGtot) in the reference frame attached to the main body (0, X, Y, Z),
MODEL.InverseTotalModel: [(6+NBPIVOTS)x(6+NBPIVOTS) ss] inverse dynamic model between inputs:
* 6 dof external force-torque applied on the main body,
* NBPIVOTS torques applied inside the NBPIVOTS pivot joints between main body and appendages, and outputs:
* 6 dof linear-angular accelerations of main body,
* NBPIVOTS relative angular accelerations of appendages w.r.t main body around pivots axis.
This model is written at point CGtot in frame (CGtot, X, Y, Z).
MODEL.DynamicModel: [(6+NBPIVOTS)x(6+NBPIVOTS) ss] direct dynamic model of the spacecraft (the inverse of the previous one).
MODEL.liste_IOs: Description and ordering of the (6+NBPIVOTS) inputs (outputs) used in the dynamic model.

Reference:
Alazard, D., Cumer, C., and Tantawi, K., "Linear dynamic modeling of spacecraft with various flexible appendages and on-board angular momentums".

ISAE/DMIA/ADIS  Page 55/58
A.2 help of function translate_dynamic_model.m

TDM = translate_dynamic_model(vec_A, vec_B, DM) translate the direct dynamic model from point A to point B in the same reference frame.

Inputs:
* vec_A: 3x1 coordinate vector of point A,
* vec_B: 3x1 coordinate vector of point B,
* DM: dynamic model at point A.

Output:
* TDM: dynamic model at point B.

It is assumed that the first 6x6 block of DM represent the dynamic model between the 6 dof acceleration vector and the 6 dof external force vector.

A.3 help of function rotate_dynamic_model.m

DM_OUT = rotate_dynamic_model (DM_IN , ANGLE, AXIS) computes the dynamic model after a rotation of ANGLE (deg) around the axis AXIS:
* DN_IN: dynamic model of a body in a frame R,
* ANGLE: rotation angle (deg),
* AXIS: 3 components in the frame R of the unitary vector along the rotation axis,
* DN_OUT: dynamic model of the rotated body in the frame R.

It is assumed that the first 6x6 block of DM_IN represent the dynamic model between the 6 dof acceleration vector and the 6 dof external force vector.

A.4 help of function kinematic_model.m

JACOB=kinematic_model(vec_A, vec_B) calculates the kinematic model JACOB of a body between two points A and B.

Inputs:
* vec_A: 3x1 coordinate vector of point A in a given frame,
* vec_B: 3x1 coordinate vector of point B (in the same frame).
Output:
* JACOB: 6x6 kinematic model (projected in the same frame).
\[
\begin{bmatrix}
\text{eye}(3) & (*AB) \\
\text{zeros}(3) & \text{eye}(3)
\end{bmatrix}
\]

A.5 help of function antisym.m

MAT=antisym(VEC) computes the antisymmetric matrix MAT associated with a vector VEC.
if VEC=[x y z]', then MAT=[0 -z y; z 0 -x; -y x 0].