SE(3) operations

N. Mansard

1 Rigid transformation

$$m : p \in \mathcal{E}(3) \to m(p) \in E(3)$$

Transformation from B to A:

$${}^{A}m_B : \ {}^{B}p \in \mathcal{R}^3 \cong \mathcal{E}(3) \ \to \ {}^{A}p = \ {}^{A}m_B(\ {}^{B}p) = \ {}^{A}M_B \ \ {}^{B}p$$

$${}^{A}p = \ {}^{A}R_B \ {}^{B}p + \ {}^{A}AB$$

$${}^{A}M_B = \begin{bmatrix} {}^{A}R_B & {}^{A}AB \\ 0 & 1 \end{bmatrix}$$

Transformation from A to B:

$${}^{B}p = \ {}^{A}R_B^T \ {}^{A}p + \ {}^{B}BA, \quad \text{with} \ \ {}^{B}BA = - \ {}^{A}R_B^T \ {}^{A}AB$$

$${}^{B}M_A = \begin{bmatrix} {}^{A}R_B^T & - \ {}^{A}R_B^T \ {}^{A}AB \\ 0 & 1 \end{bmatrix}$$

For Featherstone, $E = \ {}^{B}R_A = \ {}^{A}R_B^T$ and $r = \ {}^{A}AB$. Then:

$${}^{B}M_A = \begin{bmatrix} {}^{B}R_A & - \ {}^{B}R_A \ {}^{A}AB \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} E & -Er \\ 0 & 1 \end{bmatrix}$$

$${}^{A}M_B = \begin{bmatrix} {}^{B}R_A^T & {}^{A}AB \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} E^T & r \\ 0 & 1 \end{bmatrix}$$

2 Composition

$${}^{A}M_B \ {}^{B}M_C = \begin{bmatrix} {}^{A}R_B \ {}^{B}R_C & {}^{A}AB + \ {}^{A}R_B \ {}^{B}BC \\ 0 & 1 \end{bmatrix}$$

$${}^{A}M_B^{-1} \ {}^{A}M_C = \begin{bmatrix} {}^{A}R_B^T \ {}^{A}R_C & {}^{A}R_B^T(\ {}^{A}AC - \ {}^{A}AB) \\ 0 & 1 \end{bmatrix}$$

3 Motion Application

$$ {}^A\nu = \begin{bmatrix} {}^A v \\ {}^A\omega \end{bmatrix} $$

$$ {}^B\nu = {}^B X_A \, {}^A\nu $$

$$ {}^A X_B = \begin{bmatrix} {}^A R_B & {}^A AB_\times \, {}^A R_B \\ 0 & {}^A R_B \end{bmatrix} $$

$$ {}^A X_B^{-1} = {}^B X_A = \begin{bmatrix} {}^A R_B^T & - \, {}^A R_B^T \, {}^A AB_\times \\ 0 & {}^A R_B^T \end{bmatrix} $$

For Featherstone, $E = {}^B R_A = {}^A R_B^T$ and $r = {}^A AB$. Then:

$$ {}^B X_A = \begin{bmatrix} {}^B R_A & - \, {}^B R_A \, {}^A AB_\times \\ 0 & {}^B R_A \end{bmatrix} = \begin{bmatrix} E & -E r_\times \\ 0 & E \end{bmatrix} $$

$$ {}^A X_B = \begin{bmatrix} {}^B R_A^T & {}^A AB_\times \, {}^B R_A^T \\ 0 & {}^B R_A^T \end{bmatrix} = \begin{bmatrix} E^T & r_\times E^T \\ 0 & E^T \end{bmatrix} $$

4 Force Application

$$ {}^A\phi = \begin{bmatrix} {}^A f \\ {}^A\tau \end{bmatrix} $$

$$ {}^B\phi = {}^B X_A^* \, {}^A\phi $$

For any ϕ, ν, $\phi\dot{\nu} = {}^A\phi^T \, {}^A\nu = {}^B\phi^T \, {}^B\nu$ and then:

$$ {}^A X_B^* = {}^A X_B^{-T} = \begin{bmatrix} {}^A R_B & 0 \\ {}^A AB_\times \, {}^A R_B & {}^A R_B \end{bmatrix} $$

(because ${}^A AB_\times^T = - \, {}^A AB_\times$).

$$ {}^A X_B^{-*} = {}^B X_A^* = \begin{bmatrix} {}^A R_B^T & 0 \\ - \, {}^A R_B^T \, {}^A AB_\times & {}^A R_B^T \end{bmatrix} $$

For Featherstone, $E = {}^B R_A = {}^A R_B^T$ and $r = {}^A AB$. Then:

$$ {}^B X_A^* = \begin{bmatrix} {}^B R_A & 0 \\ - \, {}^B R_A \, {}^A AB_\times & {}^B R_A \end{bmatrix} = \begin{bmatrix} E & 0 \\ -E r_\times & E \end{bmatrix} $$

$$ {}^A X_B^* = \begin{bmatrix} {}^B R_A^T & 0 \\ {}^A AB_\times \, {}^B R_A^T & {}^B R_A^T \end{bmatrix} = \begin{bmatrix} E^T & 0 \\ r_\times E^T & E^T \end{bmatrix} $$

5 Inertia

5.1 Inertia application

$$ {}^{A}Y : {}^{A}\nu \to {}^{A}\phi = {}^{A}Y \, {}^{A}\nu $$

Coordinate transform:

$$ {}^{B}Y = {}^{B}X_{A}^{*} \, {}^{A}Y \, {}^{B}X_{A}^{-1} $$

since:

$$ {}^{B}\phi = {}^{B}X_{A}^{*} \, {}^{B}\phi = {}^{B}X_{A}^{*} \, {}^{A}I \, {}^{A}X_{B} \, {}^{B}\nu $$

Cannonical form. The inertia about the center of mass c is:

$$ {}^{c}Y = \begin{bmatrix} m & 0 \\ 0 & {}^{C}I \end{bmatrix} $$

Expressed in any non-centered coordinate system A:

$$ {}^{A}Y = {}^{A}X_{C}^{*} \, {}^{C}I \, {}^{A}X_{C}^{-1} = \begin{bmatrix} m & m \, {}^{A}AC_{\times}^{T} \\ m \, {}^{A}AC_{\times} & {}^{A}I + m \, {}^{A}AC_{\times} \, {}^{A}AC{\times}^{T} \end{bmatrix} $$

Changing the coordinates system from B to A:

$$ {}^{A}Y = {}^{A}X_{B}^{*} \, {}^{B}X_{C}^{*} \, {}^{C}I \, {}^{B}X_{C}^{-1} \, {}^{A}X_{B}^{-1} $$

$$ = \begin{bmatrix} m & m[\, {}^{A}AB + {}^{A}R_{B} \, {}^{B}BC]_{\times}^{T} \\ m[\, {}^{A}AB + {}^{A}R_{B} \, {}^{B}BC]_{\times} & {}^{A}R_{B} \, {}^{B}I \, {}^{A}R_{B}^{T} - m[\, {}^{A}AB + {}^{A}R_{B} \, {}^{B}BC]_{\times}^{2} \end{bmatrix} $$

Representing the spatial inertia in B by the triplet $(m, \, {}^{B}BC, \, {}^{B}I)$, the expression in A is:

$$ {}^{A}m_{B} : {}^{B}Y = (m, \, {}^{B}BC, \, {}^{B}I) \to {}^{A}Y = (m, \, {}^{A}AB + {}^{A}R_{B} \, {}^{B}BC, \, {}^{A}R_{B} \, {}^{B}I \, {}^{A}R_{B}^{T}) $$

Similarly, the inverse action is:

$$ {}^{A}m_{B}^{-1} : {}^{A}Y \to {}^{B}Y = (m, \, {}^{A}R_{B}^{T}({}^{A}AC - {}^{A}AB), \, {}^{A}R_{B}^{T} \, {}^{A}I \, {}^{A}R_{B}) $$

Motion-to-force map:

$$ Y\nu = \begin{bmatrix} m & mc_{\times}^{T} \\ mc_{\times} & I + mc_{\times}c_{\times}^{T} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} mv - mc \times \omega \\ mc \times v + I\omega - mc \times (c \times \omega) \end{bmatrix} $$

Nota: the square of the cross product is:

$$ \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times}^{2} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}^{2} = \begin{bmatrix} -y^2 - z^2 & xy & xz \\ xy & -x^2 - z^2 & yz \\ xz & yz & -x^2 - y^2 \end{bmatrix} $$

There is no computational interest in using it.

5.2 Inertia addition

$$Y_p = \begin{bmatrix} m_p & m_p p_\times^T \\ m_p p_\times & I_p + m_p p_\times p_\times^T \end{bmatrix}$$

$$Y_q = \begin{bmatrix} m_q & m_q q_\times^T \\ m_q q_\times & I_q + m_q q_\times q_\times^T \end{bmatrix}$$

6 Cross products

Motion-motion product:

$$\nu_1 \times \nu_2 = \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} \times \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} v_1 \times \omega_2 + \omega_1 \times v_2 \\ \omega_1 \times \omega_2 \end{bmatrix}$$

Motion-force product:

$$\nu \times \phi = \begin{bmatrix} v \\ \omega \end{bmatrix} \times \begin{bmatrix} f \\ \tau \end{bmatrix} = \begin{bmatrix} \omega \times f \\ \omega \times \tau + v \times f \end{bmatrix}$$

A special form of the motion-force product is often used:

$$\begin{aligned} \nu \times (Y\nu) &= \nu \times \begin{bmatrix} mv - mc \times \omega \\ mc \times v + I\omega - mc \times (c \times \omega) \end{bmatrix} \\ &= \begin{bmatrix} m\omega \times v - \omega \times (mc \times \omega) \\ \omega \times (mc \times v) + \omega \times (I\omega) - \omega \times (c \times (mc \times \omega)) - v \times (mc \times \omega) \end{bmatrix} \end{aligned}$$

Setting $\beta = mc \times \omega$, this product can be written:

$$\nu \times (Y\nu) = \begin{bmatrix} \omega \times (mv - \beta) \\ \omega \times (c \times (mv - \beta) + I\omega) - v \times \beta \end{bmatrix}$$

This last form cost five $\times$, four $+$ and one 3×3 matrix-vector multiplication.

7 Joint

We denote by 1 the coordinate system attached to the parent (predecessor) body at the joint input, ad by 2 the coordinate system attached to the (child) successor body at the joint output. We neglect the possible time variation of the joint model (ie the bias velocity $\sigma = \nu(q, 0)$ is null).

The joint geometry is expressed by the rigid transformation from the input to the ouput, parametrized by the joint coordinate system $q \in \mathcal{Q}$:

$$^2m_1 \cong \ ^2M_1(q)$$

The joint velocity (i.e. the velocity of the child wrt. the parent in the child coordinate system) is:

$$^2\nu_{12} = \nu_J(q, v_q) = \ ^2S(q) v_q$$

where ${}^2S$ is the joint Jacobian (or constraint matrix) that define the motion subspace allowed by the joint, and v_q is the joint coordinate velocity (i.e. an element of the Lie algebra associated with the joint coordinate manifold), which would be $v_q = \dot{q}$ when $\dot{q}$ exists.

The joint acceleration is:

$$ {}^2\alpha_{12} = S\dot{v}_q + c_J + {}^2\nu_1 \times {}^2\nu_{12} $$

where $c_J = \sum_{i=1}^{n_q} \frac{\partial S}{\partial q_i} \dot{q}_i$ (null in the usual cases) and ${}^2\nu_1$ is the velocity of the parent body with respect to an absolute (Galilean) coordinate system[1].

The joint calculations take as input the joint position q and velocity v_q and should output ${}^2M_1$, ${}^2\nu_{12}$ and ${}^2c$ (this last vector being often a trivial 0_6 vector). In addition, the joint model should store the position of the joint input in the central coordinate system of the previous joint ${}^0m_1$ which is a constant value.

The joint integrator computes the exponential map associated with the joint manifold. The function inputs are the initial position q_0, the velocity v_q and the length of the integration interval t. It computes q_t as:

$$ q_t = q_0 + \int_0^t v_q dt $$

For the simple vectorial case where $v_q = \dot{q}$, we have $q_t = q_0 + tv_q$. Written in the more general case of a Lie groupe, we have $q_t = q_0 exp(tv_q)$ where exp denotes the exponential map (i.e. integration of a constant vector field from the Lie algebra into the Lie group). This integration only consider first order explicit Euler. More general integrators (e.g. Runge-Kutta in Lie groupes) remains to be written. Adequate references are welcome.

8 RNEA

8.1 Initialization

$$ {}^0\nu_0 = 0; \ {}^0\alpha_0 = -g $$

In the following, the coordinate system i is attached to the output of the joint (child body), while $lambda(i)$ is the central coordinate system attached to the parent joint. The coordinated system associated with the joint input is denoted by i_0. The constant rigid transformation from $\lambda(i)$ to the joint input is then ${}^{\lambda(i)}M_{i_0}$.

[1] The abosulte velocity ν_1 is also the relative velocity wrt. the Galilean coordinate system Ω. The exhaustive notation should be $\nu_{\Omega 1}$ but ν_1 is prefered for simplicity.

8.2 Forward loop

For each joint i, update the joint calculation $\mathbf{j}_i.\text{calc}(q, v_q)$. This compute $\mathbf{j}.M = {}^{\lambda(i)}M_{i_0}(q)$, $\mathbf{j}.\nu = {}^{i}\nu_{\lambda(i)i}(q, v_q)$, $\mathbf{j}.S = {}^{i}S(q)$ and $\mathbf{j}.c = \sum_{k=1}^{n_q} \frac{\partial^i S}{\partial q_k}\dot{q}_k$. Attached to the joint is also its placement in body $\lambda(i)$ denoted by $\mathbf{j}.M_0 = {}^{\lambda(i)}M_{i_0}$. Then:

$$
\begin{aligned}
{}^{\lambda(i)}M_i &= \mathbf{j}.M_0\ \mathbf{j}.M \\
{}^{0}M_i &= {}^{0}M_{\lambda(i)}\ {}^{\lambda(i)}M_i \\
{}^{i}\nu_i &= {}^{\lambda(i)}X_i^{-1}\ {}^{\lambda(i)}\nu_{\lambda(i)} + \mathbf{j}.\nu \\
{}^{i}\alpha_i &= {}^{\lambda(i)}X_i^{-1}\ {}^{\lambda(i)}\alpha_{\lambda(i)} + \mathbf{j}.S\dot{v}_q + \mathbf{j}.c + {}^{i}\nu_i \times \mathbf{j}.\nu \\
{}^{i}\phi_i &= {}^{i}Y_i\ {}^{i}\alpha_i + {}^{i}\nu_i \times {}^{i}Y_i\ {}^{i}\nu_i - {}^{0}X_i^{-*}\ {}^{0}\phi_i^{ext}
\end{aligned}
$$

8.3 Backward loop

For each joint i from leaf to root, do:

$$
\begin{aligned}
\tau_i &= \mathbf{j}.S^T\ {}^{i}\phi_i \\
{}^{\lambda(i)}\phi_{\lambda(i)} &\mathrel{+}= {}^{\lambda(i)}X_i^{*}\ {}^{i}\phi_i
\end{aligned}
$$

8.4 Nota

It is more efficient to apply X^{-1} than X. Similarly, it is more efficient to apply X^{-*} than X^*. Therefore, it is better to store the transformations ${}^{\lambda(i)}m_i$ and ${}^{0}m_i$ than ${}^{i}m_{\lambda(i)}$ and ${}^{i}m_0$.