CheatSheet: SE(3) operations

Rigid transformation

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

Transformation from B to A:

$${}^Am_B : {}^Bp \in \mathcal{R}^3 \cong \mathcal{E}(3) \; \rightarrow \; {}^Ap = {}^Am_B({}^Bp) = {}^AM_B \; {}^Bp$$

$${}^Ap = {}^AR_B \, {}^Bp + {}^AAB$$

$${}^AM_B = \begin{bmatrix} {}^AR_B & {}^AAB \\ 0 & 1 \end{bmatrix}$$

Transformation from A to B:

$${}^Bp = {}^AR_B^T \, {}^Ap + {}^BBA, \quad \text{with } {}^BBA = - {}^AR_B^T \, {}^AAB$$

$${}^BM_A = \begin{bmatrix} {}^AR_B^T & - {}^AR_B^T \, {}^AAB \\ 0 & 1 \end{bmatrix}$$

For Featherstone, $E = {}^BR_A = {}^AR_B^T$ and $r = {}^AAB$. Then:

$${}^BM_A = \begin{bmatrix} {}^BR_A & - {}^BR_A \, {}^AAB \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} E & -Er \\ 0 & 1 \end{bmatrix}$$

$${}^AM_B = \begin{bmatrix} {}^BR_A^T & {}^AAB \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} E^T & r \\ 0 & 1 \end{bmatrix}$$

Composition

$${}^AM_B \, {}^BM_C = \begin{bmatrix} {}^AR_B \, {}^BR_C & {}^AAB + {}^AR_B \, {}^BBC \\ 0 & 1 \end{bmatrix}$$

$${}^AM_B^{-1} \, {}^AM_C = \begin{bmatrix} {}^AR_B^T \, {}^AR_C & {}^AR_B^T({}^AAC - {}^AAB) \\ 0 & 1 \end{bmatrix}$$

Motion Application

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

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

$${}^AX_B = \begin{bmatrix} {}^AR_B & {}^AAB_\times \, {}^AR_B \\ 0 & {}^AR_B \end{bmatrix}$$

$${}^AX_B^{-1} = {}^BX_A = \begin{bmatrix} {}^AR_B^T & - {}^AR_B^T \, {}^AAB_\times \\ 0 & {}^AR_B^T \end{bmatrix}$$

For Featherstone, $E = {}^BR_A = {}^AR_B^T$ and $r = {}^AAB$. Then:

$${}^BX_A = \begin{bmatrix} {}^BR_A & - {}^BR_A \, {}^AAB_\times \\ 0 & {}^BR_A \end{bmatrix} = \begin{bmatrix} E & -Er_\times \\ 0 & E \end{bmatrix}$$

$${}^AX_B = \begin{bmatrix} {}^BR_A^T & {}^AAB_\times \, {}^BR_A^T \\ 0 & {}^BR_A^T \end{bmatrix} = \begin{bmatrix} E^T & r_\times E^T \\ 0 & E^T \end{bmatrix}$$

Force Application

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

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

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

$$ {}^AX_B^* = {}^AX_B^{-T} = \begin{bmatrix} {}^AR_B & 0 \\ {}^AAB_\times \, {}^AR_B & {}^AR_B \end{bmatrix} $$

(because ${}^AAB_\times^T = -{}^AAB_\times$).

$$ {}^AX_B^{-*} = {}^BX_A^* = \begin{bmatrix} {}^AR_B^T & 0 \\ -{}^AR_B^T \, {}^AAB_\times & {}^AR_B^T \end{bmatrix} $$

For Featherstone, $E = {}^BR_A = {}^AR_B^T$ and $r = {}^AAB$. Then:

$$ {}^BX_A^* = \begin{bmatrix} {}^BR_A & 0 \\ -{}^BR_A \, {}^AAB_\times & {}^BR_A \end{bmatrix} = \begin{bmatrix} E & 0 \\ -Er_\times & E \end{bmatrix} $$

$$ {}^AX_B^* = \begin{bmatrix} {}^BR_A^T & 0 \\ {}^AAB_\times \, {}^BR_A^T & {}^BR_A^T \end{bmatrix} = \begin{bmatrix} E^T & 0 \\ r_\times E^T & E^T \end{bmatrix} $$

Inertia

Inertia application

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

Coordinate transform:

$$ {}^BY = {}^BX_A^* \, {}^AY \, {}^BX_A^{-1} $$

since:

$$ {}^B\phi = {}^BX_A^* \, {}^B\phi = {}^BX_A^* \, {}^AI \, {}^AX_B \, {}^B\nu $$

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

$$ {}^cY = \begin{bmatrix} m & 0 \\ 0 & {}^CI \end{bmatrix} $$

Expressed in any non-centered coordinate system A:

$$ {}^AY = {}^AX_C^* \, {}^CI \, {}^AX_C^{-1} = \begin{bmatrix} m & m \, {}^AAC_\times^T \\ m \, {}^AAC_\times & {}^AI + m \, {}^AAC_\times \, {}^AAC{\times}^T \end{bmatrix} $$

Changing the coordinates system from B to A:

$$ {}^AY = {}^AX_B^* \, {}^BX_C^* \, {}^CI \, {}^BX_C^{-1} \, {}^AX_B^{-1} $$

$$ = \begin{bmatrix} m & m[{}^AAB + {}^AR_B \, {}^BBC]_\times^T \\ m[{}^AAB + {}^AR_B \, {}^BBC]_\times & {}^AR_B \, {}^BI \, {}^AR_B^T - m[{}^AAB + {}^AR_B \, {}^BBC]_\times^2 \end{bmatrix} $$

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

$$ {}^Am_B : {}^BY = (m, {}^BBC, {}^BI) \to {}^AY = (m, {}^AAB + {}^AR_B \, {}^BBC, {}^AR_B \, {}^BI \, {}^AR_B^T) $$

Similarly, the inverse action is:

$$ {}^Am_B^{-1} : {}^AY \to {}^BY = (m, {}^AR_B^T({}^AAC - {}^AAB), {}^AR_B^T \, {}^AI \, {}^AR_B) $$

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.

Inertia addition

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

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

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:

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

$$= \begin{bmatrix} m\omega \times v - \omega \times (mc \times \omega) \\ \omega \times (mc \times v) + \omega \times (I\omega) - \omega \times (c \times (mc \times \omega)) - v \times (mc \times \omega) \end{bmatrix}$$

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.

Joint

We denote by 1 the coordinate system attached to the parent (predecessor) body at the joint input, and 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

NB: 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.

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 group, 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 considers first order explicit Euler. More general integrators (e.g. Runge-Kutta in Lie groups) remains to be written. Adequate references are welcome.

RNEA

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 $^{\lambda(i)}M_{i_0}$.

Forward loop

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

$$^{\lambda(i)}M_i = \,^Bfj.M_0 \,^Bfj.M$$
$$^0M_i = \,^0M_{\lambda(i)} \,^{\lambda(i)}M_i$$
$$^i\nu_i = \,^{\lambda(i)}X_i^{-1} \,^{\lambda(i)}\nu_{\lambda(i)} + \,^Bfj.\nu$$
$$^i\alpha_i = \,^{\lambda(i)}X_i^{-1} \,^{\lambda(i)}\alpha_{\lambda(i)} + \,^Bfj.S\dot{v}_q + \,^Bfj.c + \,^i\nu_i \times \,^Bfj.\nu$$
$$^i\phi_i = \,^iY_i \,^i\alpha_i + \,^i\nu_i \times \,^iY_i \,^i\nu_i - \,^0X_i^{-*} \,^0\phi_i^{ext}$$

Backward loop

For each joint i from leaf to root, do:

$$\tau_i = \,^Bfj.S^T \,^i\phi_i$$
$$^{\lambda(i)}\phi_{\lambda(i)} \mathrel{+}= \,^{\lambda(i)}X_i^* \,^i\phi_i$$

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 ${}^{i}m_0$.