RaiSim Cheatsheet: Articulated system
by Jemin Hwangbo

Legend

— Varying transformation (a function of the generalized coordinate)

-- Constant transformation (independent of the generalized coordinate)

floating-base system

$\mathbb{B}_0$ (floating base)

r_{com0} — Center of Mass — $m_0, \boldsymbol{I}_0$

$\mathbb{T}_{BJ1}$

θ_1

$\mathbb{J}_0$

$\mathbb{J}_1$ (revolute)

The second movable joint (the first is the floating base)

$\mathbb{T}_{BJ2}$

$\mathbb{B}_1$

W

$\mathbb{T}_2$

$\mathbb{J}_2$ (prismatic)

$\mathbb{B}_2$

$\mathbb{J}_3$ (spherical)

$\mathbb{T}_{foot}$

a fixed joint named "foot"

$\mathbb{T}_{BJ3}$

$\mathbb{T}_{B3foot}$

$\boldsymbol{n}_{c0}$

$\boldsymbol{F}_{c0}$

$\boldsymbol{T}_{c0}$

Joint state representation

$\mathbb{J}_0$
$$\psi_0 = r_0, q_0 \quad \text{[Base position and quaternion]}$$
$$u_0 = v_0, \omega_0^W$$

[angular velocity defined in the world frame!!!]

$\mathbb{J}_1$
$$\psi_1 = \theta_1$$
$$u_1 = \dot{\theta}_1$$

$\mathbb{J}_2$
$$\psi_2 = d_2$$
$$u_2 = \dot{d}_2$$

$\mathbb{J}_3$
$$\psi_3 = q_3 \quad \text{[joint rotation relative to the parent, expressed in quaternion]}$$
$$u_3 = \omega_3 \quad \text{[angular velocity relative to the parent]}$$

Robot state representation
$$\boldsymbol{\psi} = [r_0, q_0, \theta_1, d_2, q_3]^T$$
$$\boldsymbol{u} = [v_0, \omega_0, \dot{\theta}_1, v_2, \omega_3]^T$$

fixed-base system

θ_3

θ_2

$\mathbb{J}_3$

$\mathbb{J}_2$

θ_4

$\mathbb{J}_4$

$\mathbb{J}_1$

θ_1

Note! the first moving joint is joint 1, not joint 0

$\mathbb{J}_0$ (the fixed base)

Robot state representation
$$\boldsymbol{\psi} = [\theta_1, \theta_2, \theta_3, \theta_4]$$
$$\boldsymbol{u} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]$$

How to get ?

Transformations

$$\begin{aligned} v^W &= J_p \boldsymbol{u} \\ \omega^W &= J_r \boldsymbol{u} \end{aligned}$$

its derivatives

associated jacobians, that satisfy →

$\mathbb{T}_{foot}$

getFramePosition("foot", position_ref)

getFrameOrientation("foot", rotation_matrix_ref)

getFrameVelocity("foot", velocity_ref)

getFrameAngularVelocity("foot", ang_vel_ref)

getDenseFrameJacobian("foot", jaco_ref)

getDenseFrameRotationalJacobian("foot", jaco_ref)

$\mathbb{T}_2$

getPosition(2, position_ref)

getOrientation(2, rotation_matrix_ref)

getFrameVelocity(2, velocity_ref)

getFrameAngularVelocity(2, ang_vel_ref)

Call these methods with the joint name. All joints are converted to frames

$\mathbb{B}_1$ — Body doesn't have a frame of its own (by the URDF convention). It is attached to the associated joint frame

Robot definition (non-const ref's)

$\mathbb{T}_{BJ2}$

getJointPos_P()

getJointAxis_P()

getJointOrientation_P()

r_{com0} — getLinkCOM()

m_0 — getMass()

$\boldsymbol{I}_0$ — getInertia()

Robot state

$\boldsymbol{\psi}$ — getGeneralizedCoordinate()

$\boldsymbol{u}$ — getGeneralizedVelocity()

Contacts (identical to single body methods)

$\boldsymbol{n}_{c0}$ (contact normal) — getContacts()[0].getNormal()

$\boldsymbol{T}_{c0}$

getContacts()[0].getPosition()

getContacts()[0].getContactFrame()

getContacts()[0].impulse() / dt

'dt' can be obtained by raisim::World::getTimeStep()

$\boldsymbol{F}_{c0}$ — This contact impulse is defined in the contact frame. To get impulse/force in the world frame, premultiply it by the contact frame rotation matrix