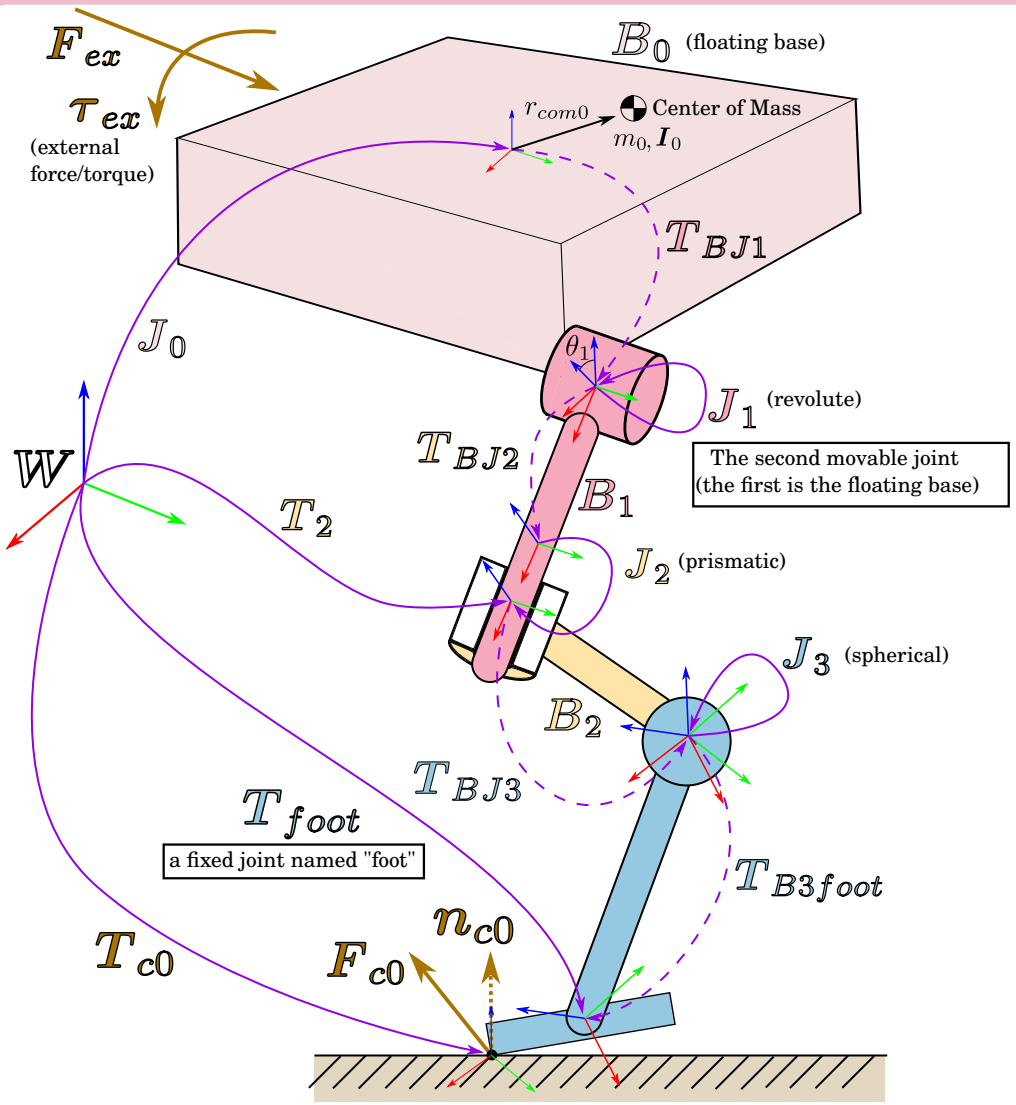
RaiSim Cheatsheet: Articulated system

by Jemin Hwangbo

floating-base system



Joint state representation

$\mathbb{J}_0$: $\psi_0 = r_0, q_0$ (Base position and quaternion); $u_0 = v_0, \omega_0^W$

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

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

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

$\mathbb{J}_3$: $\psi_3 = q_3$ (joint rotation relative to the parent, expressed in quaternion); $u_3 = \omega_3$ (angular velocity relative to the parent)

Robot state representation

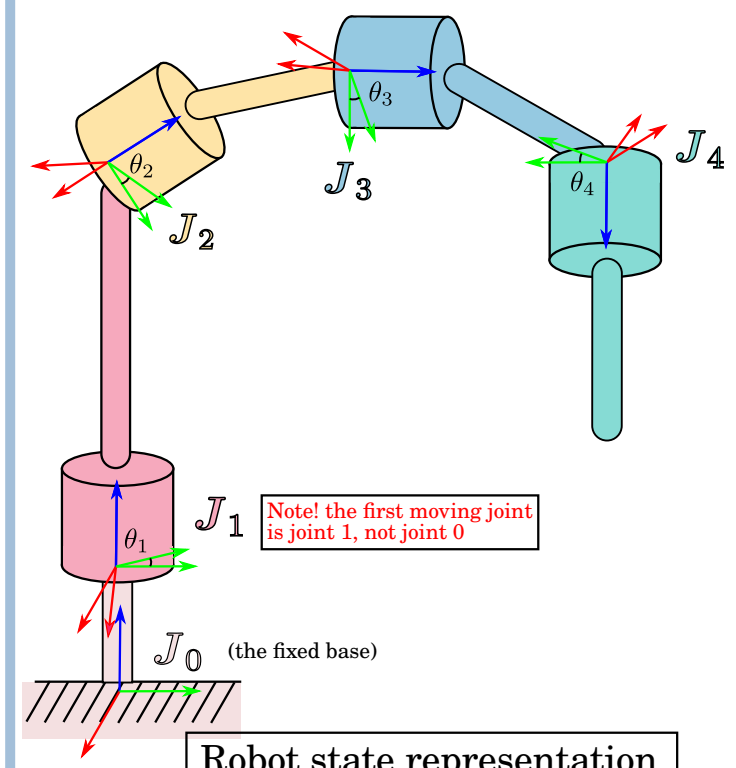
$$\boldsymbol{\psi} = [r_0, q_0, \theta_1, d_2, q_3]^T$$
$$\boldsymbol{u} = [v_0, \omega_0, \dot{\theta}_1, v_2, \omega_3]^T$$

Legend

Varying transformation (a function of the generalized coordinate)

Constant transformation (independent of the generalized coordinate)

fixed-base system



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

		its derivatives	associated jacobians, that satisfy $v^W = J_p \boldsymbol{u}$, $\omega^W = J_r \boldsymbol{u}$
$\mathbb{T}_{foot}$	getFramePosition("foot", position_ref)	getFrameVelocity("foot", velocity_ref)	getDenseFrameJacobian("foot", jaco_ref)
	getFrameOrientation("foot", rotation_matrix_ref)	getFrameAngularVelocity("foot", ang_vel_ref)	getDenseFrameRotationalJacobian("foot", jaco_ref)
$\mathbb{T}_2$	getPosition(2, position_ref)	getFrameVelocity(2, velocity_ref)	
	getOrientation(2, rotation_matrix_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 of the list)

$\mathbb{T}_{BJ1}$	getJointPos_P()	r_{com0}	getLinkCOM()
$\mathbb{T}_{BJ2}$	getJointAxis_P()	m_0	getMass()
$\mathbb{T}_{BJ3}$	getJointOrientation_P()	I_0	getInertia()

Robot state

$\boldsymbol{\psi}$	getGeneralizedCoordinate()
$\boldsymbol{u}$	getGeneralizedVelocity()

Energy

getKineticEnergy()

getPotentialEnergy({0,0,-9.81})

Potential energy is a function of gravity and measure relative to the {0,0,0} point.

Contacts (identical to single body methods)

n_{c0} (contact normal)	getContacts()[0].getNormal()
T_{c0}	getContacts()[0].getPosition()
	getContacts()[0].getContactFrame()
F_{c0}	getContacts()[0].impulse() / dt

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

This contact impulse is defined in the contact frame T_{c0}. To get impulse/force in the world frame, premultiply it by the contact frame rotation matrix

external force/torque

F_{ex}	setExternalForce(bodyId, force)
	setExternalForce(bodyId, frameId, frame, force)
τ_{ex}	setExternalTorque(bodyId, force)

This method applies force at the center of mass of the body