

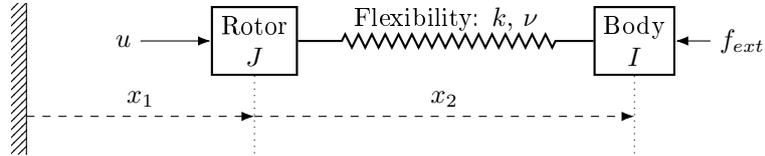
A note on the model used by Jiminy to simulate flexibilities

Matthieu Vigne

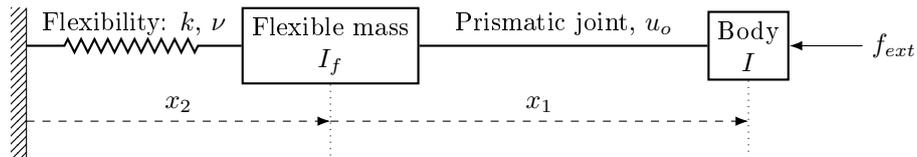
25th March 2020

Jiminy supports the addition of joint flexibilities to an otherwise rigid model, as well as rotor inertia. The goal of this document is to clarify how this is performed internally, and why this is justified.

The flexible joint model supported by Jiminy consists of a simple spring damper. The corresponding system, displayed in Figure 1a, consists in placing, between a rotor of equivalent inertia J and the output body, of inertia I , a spring-damper of stiffness k and damping ν . We call u the motor torque, and f_{ext} the external forces applied on the output body. Note that the figure is done on a prismatic joint for simplicity only: in practice, flexibilities are rotations meant to be applied onto revolute axes - but the equations are the same between both systems.



(a) Representation of a flexible joint (prismatic): a spring-damper is placed between the rotor and the output body



(b) The system actually being simulated by Jiminy: an extra mass is added to attach the spring. u_o is the torque at the output of the transmission, and is equal to $u - J\ddot{x}_2$.

Figure 1: A representation, for prismatic joints, of both systems under study.

However, such system cannot be easily represented in *pinocchio*: indeed, *pinocchio* implements rigid body dynamics as having a set of joints connecting rigid bodies: as such, it has no notion of transmission *per se*. The rotor does not correspond to a physical part of inertia J (the armature inertia, J , is computed for simple transmission by multiplying the real inertia by the square of the reduction ratio) - thus, rotor inertia is implemented in *Jiminy* by modifying the corresponding diagonal term of the generalized inertia matrix (note that this approach works for simple, 1DoF joints, but is not generic). As such, it is no

possible to add a degree of freedom to place the spring between the rotor and the output, as in the case in Figure 1a.

Instead, the flexibility is placed before rotor of the flexible joint being simulated. Since each joint must be connected to a body, this requires the addition of an extra body, connected to the flexibility: this body has its own inertia, I_f . In practice we will want I_f close to zero, to avoid modifying the dynamical properties of the overall system. The resulting system is shown in Figure 1b

We now need to show that both systems are indeed equivalent, when taking the limit $I_f \rightarrow 0$. Consider the first, SEA system: applying Newton's law of motion to the rotor, then the output body, yields:

$$\begin{cases} J\ddot{x}_2 = u + kx_1 + \nu\dot{x}_1 \\ I(\ddot{x}_1 + \ddot{x}_2) = -kx_1 - \nu\dot{x}_1 + f_{ext} \end{cases} \quad (1)$$

which equivalently writes:

$$\begin{cases} I(\ddot{x}_1 + \ddot{x}_2) = -kx_1 - \nu\dot{x}_1 + f_{ext} \\ I(\ddot{x}_1 + \ddot{x}_2) + J\ddot{x}_2 = u + f_{ext} \end{cases} \quad (2)$$

The system simulated by Jiminy writes in turn:

$$\begin{cases} I_f\ddot{x}_1 = -kx_1 - \nu\dot{x}_1 - u_o \\ I(\ddot{x}_1 + \ddot{x}_2) = u_o + f_{ext} \end{cases} \quad (3)$$

where u_o is the torque at the output of the transmission. Applying Euler's law to the rotor gives us: $J\ddot{x}_2 = u - u_o$. Thus, (3) rewrites:

$$\begin{cases} I_f\ddot{x}_1 = -kx_1 - \nu\dot{x}_1 - (u - J\ddot{x}_2) \\ I(\ddot{x}_1 + \ddot{x}_2) = u - J\ddot{x}_2 + f_{ext} \end{cases} \quad (4)$$

add by adding both lines, we obtain:

$$\begin{cases} (I + I_f)\ddot{x}_1 + I\ddot{x}_2 = -kx_1 - \nu\dot{x}_1 + f_{ext} \\ I(\ddot{x}_1 + \ddot{x}_2) + J\ddot{x}_2 = u + f_{ext} \end{cases} \quad (5)$$

Thus, from (2) and (5), it is now clear that both systems are equivalent when $I_f \ll I$. So while Jiminy does not exactly simulates a SEA as depicted in Figure 1a, it remains sufficiently close, in practice and for reasonable values of the output inertia I , to this system.