SPART Tutorial – Dynamics

Equations of motion and inertia matrices

The equations of motion of a multibody system take the following form:

$$\mathbf{H}\dot{\mathbf{u}}+\mathbf{C}\mathbf{u}=\tau$$

with $\mathbf{H}\left(\mathcal{Q}\right)\in\mathbb{R}^{(6+n)\times(6+n)}$ being the symmetric, positive-definite Generalized Inertia Matrix (GIM), $\mathbf{C}\left(\mathcal{Q},\mathbf{u}\right)\in\mathbb{R}^{(6+n)\times(6+n)}$ the Convective Inertia Matrix (CIM), and $\tau\in\mathbb{R}^{6+n}$ the generalized forces (joint-space forces).

The contributions of the base-link and the manipulator can be made explicit when writing the equations of motion.

$$\begin{bmatrix}\mathbf{H}_0 & \mathbf{H}_{0m}\\ \mathbf{H}_{0m}^T & \mathbf{H}_m\end{bmatrix}\begin{bmatrix}\dot{\mathbf{u}}_0\\ \dot{\mathbf{u}}_m\end{bmatrix}+\begin{bmatrix}\mathbf{C}_0 & \mathbf{C}_{0m}\\ \mathbf{C}_{m0} & \mathbf{C}_m\end{bmatrix}\begin{bmatrix}\mathbf{u}_0\\ \mathbf{u}_m\end{bmatrix}=\begin{bmatrix}\tau_0\\ \tau_m\end{bmatrix}$$

These GIM and CIM are computed as follows:

```
%Inertias projected in the inertial frame
[I0,Im]=I_I(R0,RL,robot);
%Mass Composite Body matrix
[M0_tilde,Mm_tilde]=MCB(I0,Im,Bij,Bi0,robot);
%Generalized Inertia Matrix
[H0, H0m, Hm] = GIM(M0_tilde,Mm_tilde,Bij,Bi0,P0,pm,robot);
%Generalized Convective Inertia Matrix
[C0, C0m, Cm0, Cm] = CIM(t0,tL,I0,Im,M0_tilde,Mm_tilde,Bij,Bi0,P0,pm,robot);
```

Although the equations of motion can be used to solve the forward dynamic problem (determining the motion of the system given a set of applied forces $\tau\rightarrow\dot{\mathbf{u}}$) and the inverse dynamic problem (determining the forces required to produce a prescribe motion $\dot{\mathbf{u}}\rightarrow\tau$) there are more computationally efficient ways of doing so.

Forward dynamics

To solve the forward dynamics, the forces acting on the multibody system are specified as an input. The generalized forces τ are the forces acting on the joints $\tau_m\in\mathbb{R}^n$ and on the base-link $\tau_{\mathbf{0}}\in\mathbb{R}^6$. Specifically, the generalized forces τ act upon the generalized velocities $\mathbf{u}$.

In τ_0, as in the twist vector, the torques $\mathbf{n}_0^{\{\mathcal{L}_0\}} \in \mathbb{R}^3$, projected in the base-link body-fixed CCS, come first and are followed by forces $\mathbf{f}_0 \in \mathbb{R}^3$, applied to the base-link center-of-mass.

$$\tau_0 = \begin{bmatrix} \mathbf{n}_0^{\{\mathcal{L}_0\}} \\ \mathbf{f}_0 \end{bmatrix}$$

The wrench applied to the ith link, $\mathbf{w}_i \in \mathbb{R}^6$, encapsulates the torques $\mathbf{n}_i \in \mathbb{R}^3$ and forces $\mathbf{f}_i \in \mathbb{R}^3$, projected in the inertial CCS, applied to the center-of-mass of each link.

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix}$$

Here is an example of how to define them:

```
%Wrenches
wF0=zeros(6,1);
wFm=zeros(6,robot.n_links_joints);

%Generalized forces
tau0=zeros(6,1);
taum=zeros(robot.n_q,1);
```

After these forces are defined, a forward dynamic solver is available.

```
%Forward dynamics
[u0dot_FD,umdot_FD] = FD(tau0,taum,wF0,wFm,t0,tL,P0,pm,I0,Im,Bij,Bi0,u0,um,robot);
```

As an example, if you need to incorporate the weight of the links (with the z-axis being the vertical direction), set the wrenches as follows:

```
%Gravity
g=9.8; %[m s-2]

%Wrenches (includes gravity and assumes z is the vertical direction)
wF0=zeros(6,1);
wF0(6)=-robot.base_link(i).mass*g;
wFm=zeros(6,robot.n_links);
for i=1:robot.n_links
        wFm(6,i)=-robot.links(i).mass*g;
end
```

Inverse dynamics

For the inverse dynamics, the acceleration of the base-link $\dot{\mathbf{u}}_0$ and of the joints $\dot{\mathbf{u}}_m$ are specified, then the `ID` function computes the inverse dynamics, providing the required forces to obtain these accelerations.

```
%Generalized accelerations
u0dot=zeros(6,1);
umdot=zeros(robot.n_q,1);

%Oprational-space accelerations
[t0dot,tLdot]=Accelerations(t0,tL,P0,pm,Bi0,Bij,u0,um,u0dot,umdot,robot);

%Inverse Dynamics - Flying base
[tau0,taum] = ID(wF0,wFm,t0,tL,t0dot,tLdot,P0,pm,I0,Im,Bij,Bi0,robot);
```

If the base-link is left uncontrolled $\dot{\tau}_0 = \mathbf{0}$ (floating-base case) and thus the base-link acceleration is unknown, the `Floating_ID` function is available.

```
%Accelerations
umdot=zeros(robot.n_q,1);

%Inverse Dynamics - Floating Base
[taum_floating,u0dot_floating] =
Floating_ID(wF0,wFm,Mm_tilde,H0,t0,tL,P0,pm,I0,Im,Bij,Bi0,u0,um,umdot,robot);
```

Finding more information

The Function Reference provides more documentation on the SPART functions. If you don't find what you need you can always Get in touch.