Lecture 5:

Control: Method of Virtual Constraints

Recall:

Model (kinematics and dynamcis) $\rightarrow$ control design $\rightarrow$ simulate (loop of control desing and simulate)

In the last session we talked about the modeling (kinematics and dynamics) of a multi-body system. In particular, we derived the following equation for the double pendulum:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = 0 \tag{1}$$

using the Lagrangian Mechanics method.

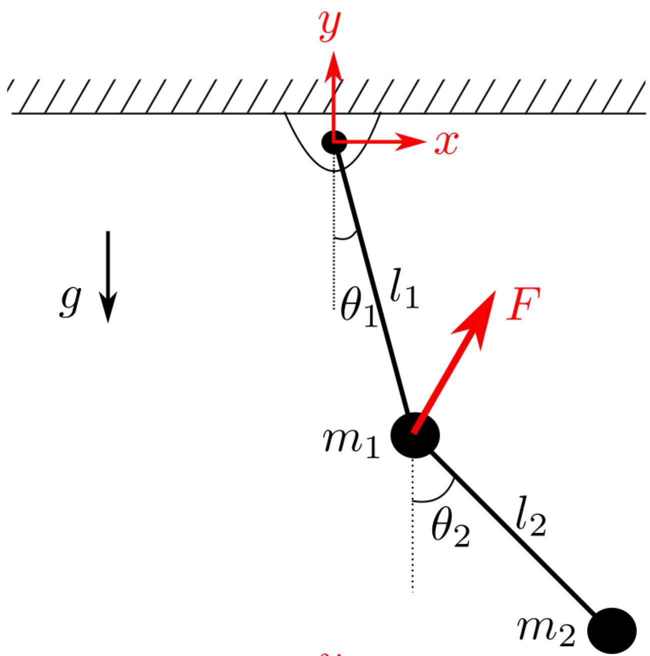
Question: Why the right hand side of this equation is zero? Is it always zero? No. The right hand was zero in our double pendulum example, because there existed no external force other than gravity. Then, the question is, what if we have external forces/torques? How would that change the equation above? This is what we will talk about today. I show how you model the control, and I present to you the method of virtual constraints as a control method.

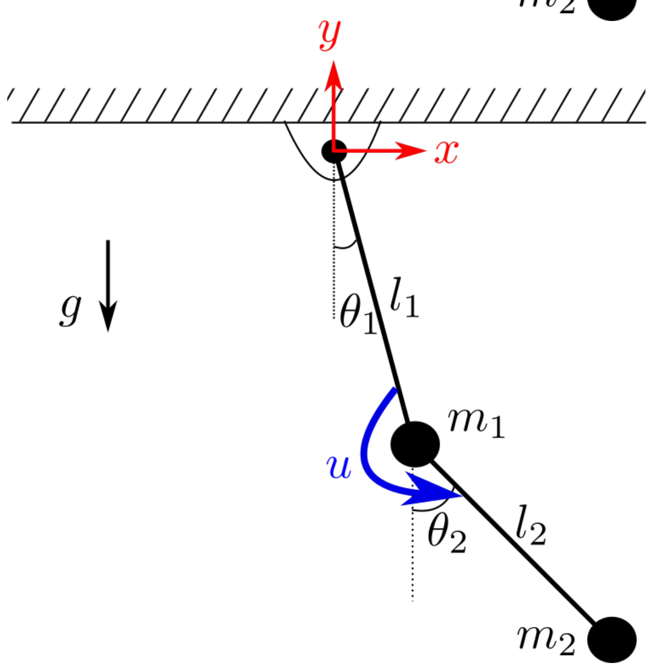
Modeling external forces/torques:

Without external forces other than gravity, the Lagrangian equations of motion were:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} = 0 \tag{2}$$

(figures of double pendulum with external force and torque).





With external forces/torques the Lagrangian equations are written as:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} = G_i \tag{3}$$

where G_is are called **generalized forces**. Intuitively, G_i is the projection of the external force in the direction of q_i. Now, I show how we calculate generalized forces.

Example: Double pendulum with external force

We cacluate G_i with the notion of **Virtual Work** δW done by the external force. The virtual work done by force F is the work done by the force F for a virtual displacement $(\delta\theta_1, \delta\theta_2)$.

Recall: $W = \int_a^b \boldsymbol{F} \cdot \boldsymbol{dr}$. (Look at the figure, path from a to b)

$\boldsymbol{r} = (l_1 \sin(\theta_1), -l_1 \cos(\theta_1))$

$\boldsymbol{F} = (F_x, F_y)$

$\delta\boldsymbol{r} = (l_1 \cos(\theta_1)\delta\theta_1, l_1 \sin(\theta_1)\delta\theta_1)$

In the cartesian interial coordiantes (x-y), the virtual work done by the force $\boldsymbol{F}$ is

$\delta W = \boldsymbol{F} \cdot \delta\boldsymbol{r} = (F_x l_1 \cos(\theta_1) + F_y l_1 \sin(\theta_1))\delta\theta_1 + (0)\delta\theta_2$

On the other hand, in the generalized coordinate system $q = (\theta_1, \theta_2)$,

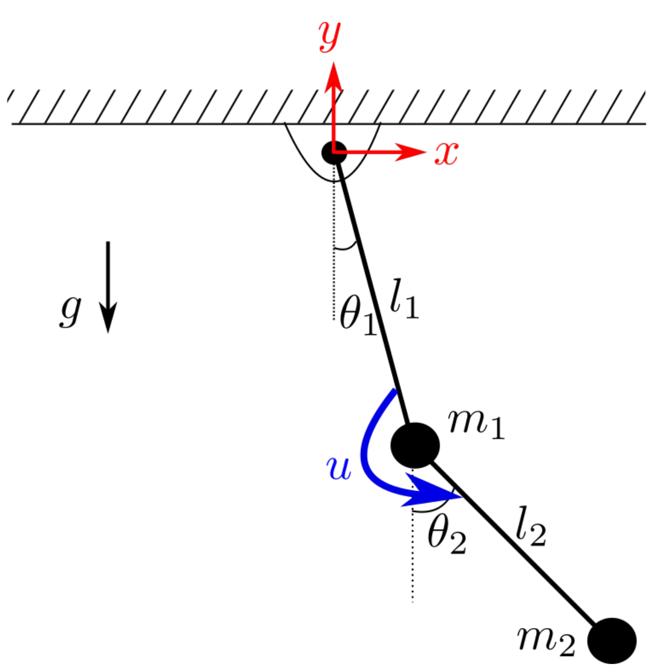
$$\delta W = G_1 \delta\theta_1 + G_2 \delta\theta_2 \tag{4}$$

Thus, $G_1 = F_x l_1 \cos(\theta_1) + F_y l_1 \sin(\theta_1)$ and $G_2 = 0$:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} &= F_x l_1 \cos(\theta_1) + F_y l_1 \sin(\theta_1) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} &= 0 \end{aligned}$$

Example: Double pendulum with external torque

Now, assume that there is an actuator that derives the angle between link 1 and 2 of the double pendulum (figure below). We want to write the equations of motion, assuming that the toque provided by the actuator is u . Note that the torque can derive the angle **between** the links and not the absolute angle (in reality the motor is installed on link 1 and it derives the the orientation of link 2 with respect to link 1.).



$\delta W = u\delta\beta$ where β is the anlge between the two links: verify that $\beta = \pi + \theta_2 - \theta_1$. Therefore,

$$\delta W = u(\delta\theta_2 - \delta\theta_1) = -u\delta\theta_1 + u\delta\theta_2 \tag{5}$$

Thus,

$$\frac{d}{dt}(\frac{\partial L}{\partial\dot{\theta}_1}) - \frac{\partial L}{\partial\theta_1} = -u$$
$$\frac{d}{dt}(\frac{\partial L}{\partial\dot{\theta}_2}) - \frac{\partial L}{\partial\theta_2} = u$$

So, from last session, and looking at the right hand side of the equation above:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu \tag{6}$$

where $B_{2\times 1}$ is:

$$B = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{7}$$

When we say "control" the double pendulum we mean determine an algorithm (function) for u to get a desired behavior. In general, a lot of multi-body systems with control (many robotic platforms) have the following form of equations of motion:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu \tag{8}$$

If q is $n \times 1$, and u is $m \times 1$, then the system has n degrees of freedom with m controllers. *Degrees of underactuation is defined as $n - m$.*

So, what is the degrees of underactuation for the double pendulum example above? Yes, 1.

The goal of control is to determine an algorithm for u to get a desired behavior. For legged robots, this desired behavior can be **walking in a stable way** or **keep balance** when standing. Or more precisely, desired behaviors such as obtaining **stable periodic walking with a given walking speed**.

In a more general term, this is called **feedback control**, that is evantually $u = u(q, \dot{q}, \dots)$ or more generally $u = u(sensory\ data)$. Recall the famous feedback loop diagram.

Anyways, there is a lot to control and a few courses are needed to cover only part of the control theory. Here, I present a control method, which later you will be using to control the 3-link biped in the mini-project.

Conrol Design: Method of virtual constraints

If we set $u = 0$ in the double pendulum example, the pendulum oscillates freely. Imagine that our desired behavior is that we want θ_2 **to follow** θ_1. How would you go about doing it?

Virtual constraints: Nonlinear control (Hard way)

How would you present the desired behavior of "θ_2 follows θ_1 " mathematically? Answer: $y = \theta_2 - \theta_1$ and we want $y \to 0$. Let's differentiate y until a control input appears:

$$\begin{aligned} y &= \theta_2 - \theta_1 \\ \dot{y} &= \dot{\theta}_2 - \dot{\theta}_1 \\ \ddot{y} &= \ddot{\theta}_2 - \ddot{\theta}_1 \end{aligned}$$

Now we subtitute for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ from the equations of motion:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = M^{-1}(Bu - C\dot{q} - G(q))$$

where $q = [\theta_1; \theta_2]$. Therefore,

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = (M^{-1}B)u - M^{-1}(C\dot{q} - G(q))$$

For simplicity, let $P_{2\times 1} = M^{-1}B$ and $Q = M^{-1}(C\dot{q} - G(q))$. So,

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} P_1 u \\ P_2 u \end{bmatrix} - \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

Thus,

$$\ddot{\theta}_2 - \ddot{\theta}_1 = (P_2 - P_1)u - (Q_2 - Q_1) \tag{9}$$

Consequently,

$$\ddot{y} = (P_2 - P_1)u - (Q_2 - Q_1) \tag{10}$$

You see that u is explicitely present on the right hand side of the equation. Let's not forget our goal. Our goal was to determine u such that $y = \theta_1 - \theta_2 \to 0$. From differential equations (DE), recall that if $K_p > 0$ and $K_d > 0$ then the solution to the following DE converges to zero as time goes to infinity:

$$\ddot{y} + K_d\dot{y} + K_p y = 0 \tag{11}$$

Do you agree? Recall the characteristic equation from DE ($r^2 + K_d r + K_p = 0$). Thus, if we set u such that $\ddot{y} = -K_d\dot{y} - K_p y$ we have solved the problem!

From equation \eqref{eqn_ddy},

$$
\begin{aligned}
(P_2 - P_1)u - (Q_2 - Q_1) &= -K_d\dot{y} - K_p y \\
&= -K_d(\dot{\theta}_2 - \dot{\theta}_1) - K_p(\theta_2 - \theta_1)
\end{aligned}
$$

Thus,

$$
u = \frac{1}{P_2 - P_1}(-K_d(\dot{\theta}_2 - \dot{\theta}_1) - K_p(\theta_2 - \theta_1) + (Q_2 - Q_1)) \tag{12}
$$

This is our control design using the method of virtual constraints. Question, is this a nonlinear control or a linear control? Why?

The resulting dynamics after the virtual constraint is satisfied is called the **zero dynamics** or **reduced dynamics**. To get the equations of **zero dynamics** all you need to do is to substitue $\theta_2 - \theta_1 = 0, \dot{\theta}_2 - \dot{\theta}_1 = 0, \ddot{\theta}_2 - \ddot{\theta}_1 = 0$ into equations of motion of the double pendulum, with the above control input.

Virtual constraints: Linear control (Easy way)

It turns out that in many occasions, simple linear version of \eqref{eqn_u_nonlinear} works as well. With a good tuning of K_d and K_p we can get $y \to 0$ with the linear PD control:

$$
u = -K_d(\dot{\theta}_2 - \dot{\theta}_1) - K_p(\theta_2 - \theta_1) \tag{13}
$$

Note that the `-` sign behind K_p > 0 and K_d > 0 here is very important. Why is it negative here? Look at it this way, if $\theta_2 > \theta_1$ and $\dot{\theta}_2 > \dot{\theta}_1$ do you need a positive u or a negative u?