Lecture 7- CTC, linear control for nonlinear systems

Objectives:

- Error dynamics
- CTC
- CTC stability
- Linearization of non-linear dynamics

Error dynamics

Before the Error dynamics, it's important to understand **Control error**. The Control error ***e*** is the difference between the desired/reference ***q**** and the output ***q***. Mathematically,

$$e = q^* - q \tag{1}$$

Now we can define the Error dynamics with controller gains K_d and K_p to be

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \tag{2}$$

Briefly on Robot dynamics

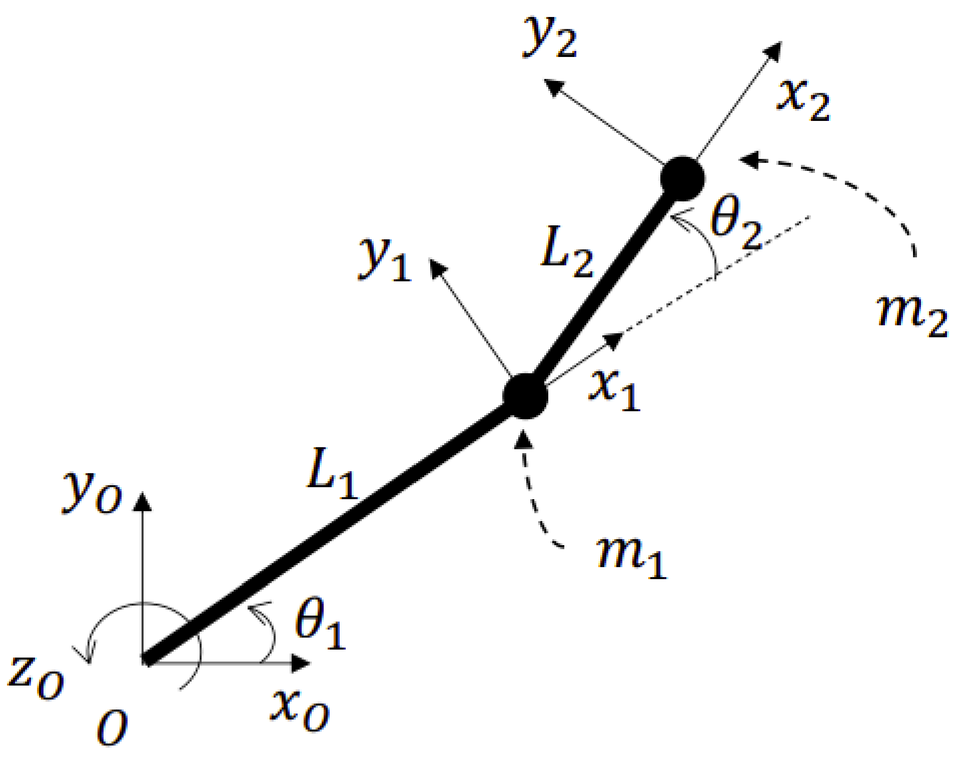
If we were to analyze a robots movement, its dynamics would be expressed as

$$\begin{gathered} H(q)\ddot{q} + c(q, \dot{q}) = \tau \\ \tau = Bu \end{gathered} \tag{3}$$

In this equation, $q, \dot{q}, \ddot{q}$ and τ are vectors of joint position, velocity, acceleration and force variables, respectively, and they are called the joint-space position, velocity, acceleration and force vectors. Each is an n-dimensional coordinate vector, where n is the number of (independent) joint variables in the mechanism.

H is called the **joint-space inertia matrix**, and it is an *n*×*n* symmetric, positive-definite matrix. **c** is called the **joint-space bias force**, which is the value of the joint-space force that must be applied to the system in order to produce zero acceleration.

You can read more on this on your own.



> In the diagram above, the θ corresponds to our q

Assuming we want to control a robot arm. We know the Error and Robot dynamics, hence

- First recall the Error dynamic
  $\ddot{q}^* - \ddot{q} + K_d \dot{e} + K_p e = 0$
- Then add H

> Note that **H** is *invertible* [1]

$H\ddot{q}^* - H\ddot{q} + H(K_d \dot{e} + K_p e) = 0$

- $H\ddot{q} + c = H\ddot{q}^* + H(K_d \dot{e} + K_p e) + c$
- Since we know that $H\ddot{q} + c = Bu$, we get
  $Bu = \tau = H\ddot{q}^* + H(K_d \dot{e} + K_p e) + c$

And this is the **Computed Torque Controller** which we will look at just below.

CTC

Computed Torque Controller (CTC) is a powerful nonlinear controller which it widely used in control robot manipulator. It is based on Feed-back linearization and computes the required arm torques using the nonlinear feedback control law. The equation for the Computed Torque Controller from the calculation above is

$$\tau = (H\ddot{q}^* + c) + H(K_d \dot{e} + K_p e) \tag{4}$$

And this equation above can be divided into two: Feedback and Feedforward parts.

The Feedback part (for tracking a reference input) is

$$\tau_{FB} = H(K_d \dot{e} + K_p e) \tag{5}$$

While the Feedforward part (for eliminating nonlinear terms of the system) is

$$\tau_{FF} = H\ddot{q}^* + c \tag{6}$$

CTC Stability

In order to find the stability of a CTC, first let's the Error dynamics $\ddot{e} + K_d \dot{e} + K_p e = 0$. After recalling that, the stability criteria is pretty straightforward.

If K_d and K_p are negative definite [2] then it is stable.

Linearization

Most systems we model are nonlinear but it's easier to identify the properties and relationship between variables if the system is linear, hence the need to linearize.

Simple Linearization

For a given nonlinear system

$$\dot{x} = f(x, u) \tag{7}$$

We simply compute

$$A := \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x(0)\\u=u(0)}}, \qquad B := \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x(0)\\u=u(0)}} \tag{8}$$

Affine system

If the system is affine [3], the resulting linear system c will be

$$c = f(x(0), u(0)) - \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x(0)\\u=u(0)}} - \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x(0)\\u=u(0)}} \tag{9}$$

Jacobian Linearization of non-linear dynamics

For a given nonlinear system

$$\dot{x}(t) = f(x(t), u(t)) \tag{10}$$

where f is a function mapping $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. We describe the Jacobian Linearization of the original nonlinear system, about the equilibrium point $(\bar{x}, \bar{u})$ as follows.

First what is the equilibrium point? A point $\bar{x} \in \mathbb{R}^n$ is called an **equilibrium point** if there is a specific $\bar{u} \in \mathbb{R}^m$ (called the equilibrium input) such that

$$f(\bar{x}, \bar{u}) = 0_n \tag{11}$$

Also, we include a notion called deviation variables to measure the difference if we start a little bit away from $\bar{x}$, and we apply a slightly different input from $\bar{u}$.

$$\begin{aligned} \delta_x(t) &:= x(t) - \bar{x} \\ \delta_u(t) &:= u(t) - \bar{u} \end{aligned} \tag{12}$$

Now we can substitute using the equilibrium points and deviation variables, we get

$$\dot{\delta_x}(t) = f(\bar{x} + \delta_x(t), \bar{u} + \delta_u(t)) \tag{13}$$

Then we do a Taylor expansion and get

$$\dot{\delta_x}(t) \approx f(\bar{x}, \bar{u}) + \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x}\\u=\bar{u}}} \delta_x(t) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x}\\u=\bar{u}}} \delta_u(t) \tag{14}$$

However, recalling from our explanation of equilibrium point $f(\bar{x}, \bar{u}) = 0$, we get

$$\dot{\delta_x}(t) \approx \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x} \\ u=\bar{u}}} \delta_x(t) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x} \\ u=\bar{u}}} \delta_u(t) \tag{15}$$

So we define constant matrices A and B such that

$$A := \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x} \\ u=\bar{u}}} \in \mathbb{R}^{nxn}, \qquad B := \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x} \\ u=\bar{u}}} \in \mathbb{R}^{nxm} \tag{16}$$

Finally, we get a linear system

$$\dot{\delta_x}(t) = A\delta_x(t) + B\delta_u(t) \tag{17}$$

References

1. Jacobian Linearization

   https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/pph02-ch19-23.pdf

2. Robot dynamics

   http://www.scholarpedia.org/article/Robot_dynamics

---

1. is a square **matrix** that the product of the **matrix** and its inverse is the identity **matrix** ↩

2. A *negative definite matrix* is a Hermitian *matrix* all of whose eigenvalues are *negative*. ↩

3. **Affine systems** are nonlinear **systems** that are linear in the input ↩