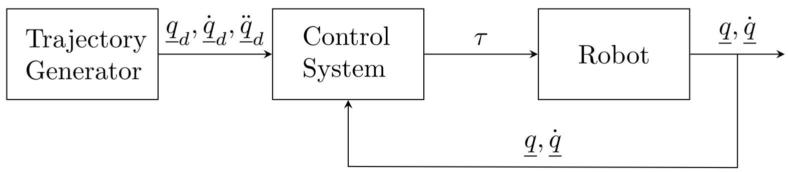
Joint–Based Linear Control

Glen Henshaw
Craig Carignan

October 28, 2023

1 Intro

Here's a block diagram of our robotic system:



And remember from last time that we can write the dynamics of our robot in the form

$$\underline{\tau} = M(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{q} + E(\underline{q}, \dot{\underline{q}})$$

We want to figure out the design of a "control law" – an equation for $\underline{\tau}$ that causes the robot to track the desired trajectory $\underline{q}_d, \dot{\underline{q}}_d, \ddot{\underline{q}}_d$. The problem is that the robot dynamics are pretty complicated. Ideally, we could use an "open loop controller" of the form

$$\underline{\tau} = M(\underline{q}_d)\ddot{\underline{q}}_d + C(\underline{q}_d, \dot{\underline{q}}_d)\dot{q}_d + E(\underline{q}_d, \dot{\underline{q}}_d)$$

but in practice this doesn't work very well because we don't know M, C, or E exactly, and plus there are always unmodelled dynamics present – we've treated our robot as if it is completely rigid, which it isn't, we've neglected the dynamics of the gears and motors in the actuator, there are always air currents or (in the SSL) water effects, and the occasional human who wanders through our robot's workspace and pushes on it.

In general, control systems make use of the actual trajectory to compute the input torques, This is known as "closed–loop" control.

$$\begin{aligned}\tilde{\underline{q}} &= \underline{q}_d - \underline{q} \\ \dot{\tilde{\underline{q}}} &= \dot{\underline{q}}_d - \dot{\underline{q}}\end{aligned}$$

$(\tilde{\underline{q}}, \dot{\tilde{\underline{q}}})$ is called "servo error". The controller computes the torque based on servo error. Linear controllers use linear combinations of the servo errors.

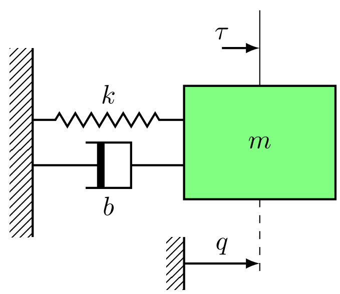
There are several performance criteria we want out of controllers:

- stability (the requirement that the errors be bounded as long as the disturbances are bounded)
- bandwidth (the ability of the robot to quickly track rapidly changing desired trajectories)
- damping (no oscillations in the output – we'll talk about this in a minute)

The easiest and most common approach to controlling robot arms is INDEPENDENT JOINT CONTROL, where we assume that each joint of the arm is dynamically independent of all the others, e.g that the dynamics really look like

$$\tau_i = m_i \ddot{q}_i + b_i \dot{q}_i + e_i(qi) \tag{1}$$

Notice that this looks like the dynamics of a simple spring–mass–damper system:



...only with no spring term in the dynamics. So, really a mass–damper system with some environmental terms. So what if we design a controller that explicitly turns this into a spring–mass–damper system? In that case we want dynamics that look something like:

$$\tau = m\ddot{q} + b\dot{q} + kq$$

One way to approximate this is to design a controller that does the following:

$$\tau_c = -k_v \dot{q} - k_p(q - q_d)$$

So let's substitute this τ_c in for τ in Equation 1:

$$m\ddot{q} + b\dot{q} + e(q) = -k_v \dot{q} - k_p(q_d - q)$$

or

$$m\ddot{q} + (b + k_v)\dot{q} + k_p(q_d - q) + e(q, \dot{q}) = 0$$

if we assume, for the moment, that $q_d = 0$ and if we neglect e, we can see that this looks precisely like a spring–mass–damper system with a mass, damping constant, and spring stiffness that we have specified. Assuming all of these coefficients are positive, the system will have a stable equilibrium point at the origin.

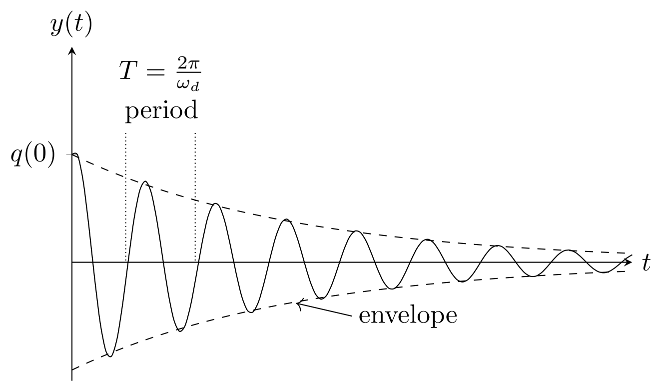
A common way of writing this equation is the "normal form":

$$\ddot{q} + \underbrace{\frac{b+k_v}{m}}_{2\zeta\omega_n}\dot{q} + \underbrace{\frac{k_p}{m}}_{\omega_n^2} q = 0$$

where

$$\begin{aligned} \omega_n &= \sqrt{\frac{k_p}{m}} \quad \text{"natural frequency"} \\ \zeta &= \frac{b+k_v}{2m\omega_n} \quad \text{"damping ratio"} \end{aligned}$$

The "unforced response" of this system looks like:



The envelope has the form

$$e^{-\zeta\omega_n t}$$

and ω_d is the "damped natural frequency",

$$\omega_d = \sqrt{1-\zeta^2}\omega_n$$

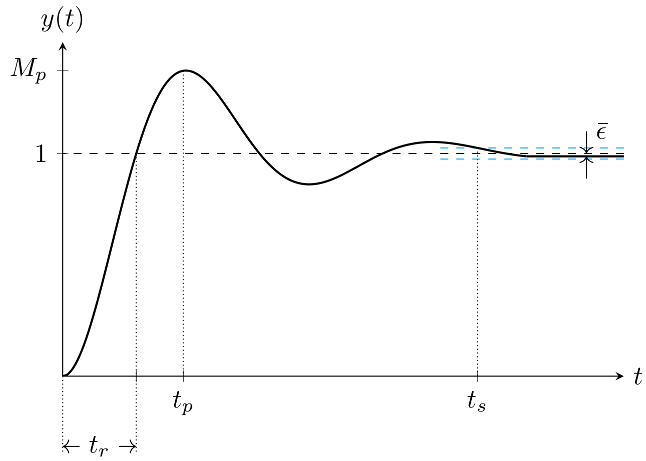
This is the step response:
where:

$$\begin{aligned} M_p &= e^{\frac{-\zeta\omega_n}{\omega_d}}\pi \times 100\% && \text{"Peak Overshoot"} \\ t_p &= \tfrac{\pi}{\omega_d} && \text{"Peak time (1/2 period)"} \\ t_r &= \tfrac{1}{\omega_d}\tan^{-1}\left(\tfrac{\omega_d}{\zeta\omega_n}\right) && \text{"rise time"} \\ t_s &= \tfrac{4}{\zeta\omega_n} && \text{"settling time (2\%)"} \end{aligned}$$

2 Solving 2nd–Order Differential Equations Via the LaPlace Transform

We can transform an ODE into an algebraic equation via the Laplace Transform:

$$\mathcal{L}\left\{\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = 0\right\}$$



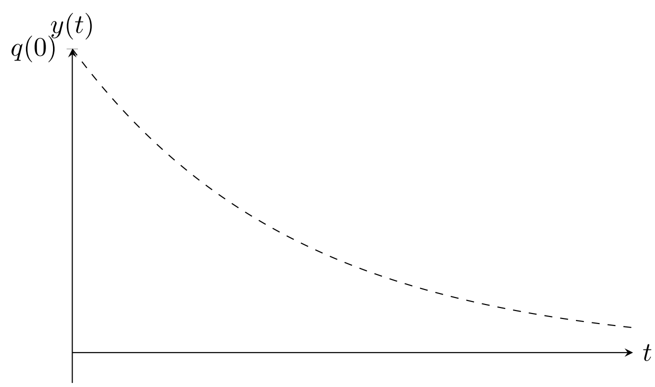
$$s^2 X(s) + 2\zeta\omega_n s X(s) + \omega_n^2 X(s) = 0$$
$$\underbrace{\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)}_{\text{“characteristic equation”}} X(s) = 0$$

The roots of the characteristic equation:

1. $\zeta > 1$ (overdamped)

$$s_{1,2} = -\zeta\omega_n \pm \sqrt{\zeta^2 - 1}\omega_n \Rightarrow x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

This is a pure decaying exponential:



2. $\zeta < 1$ (underdamped)

$$s_{1,2} = -\zeta\omega_n \pm \sqrt{1 - \zeta^2}\omega_n j \Rightarrow x(t) = e^{-\zeta\omega_n t}\left(c_1 \cos(\omega_d t) + c_2 \sin(\omega_d t)\right)$$

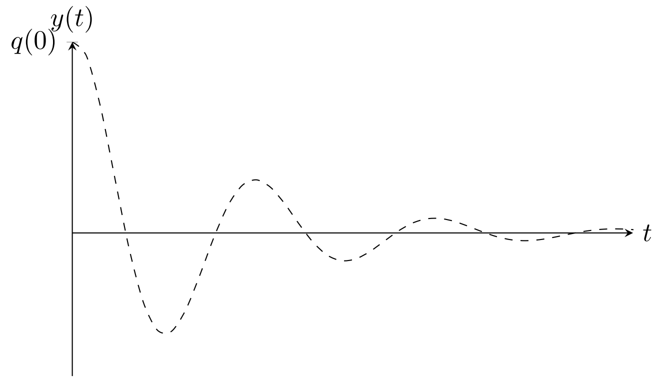
where $j = \sqrt{-1}$ and $\omega_d = \sqrt{1 - \zeta^2}\omega_n$ or

$$x(t) = r\ e^{-\zeta\omega_n t} \cos(\omega_d t - \underbrace{\delta}_{\text{phase}})$$

where

$$
\begin{aligned}
r &= \sqrt{c_1^2 + c_2^2} \\
\delta &= \mathrm{atan2}(c_2, c_1)
\end{aligned}
$$

This is (hopefully!) decaying but oscillatory:



2.1 Example

Assume we have a linear dynamic system

$$m\ddot{x} + b\dot{x} + kx = f$$

1. Find $x(t)$ for the ICs $x(0) = -1$, $\dot{x}(0) = 0$ Characteristic equation:

$$s^2 + \underbrace{1 \cdot s}_{2\zeta\omega_n} + \underbrace{1}_{\omega_n^2} = 0$$

$$
\begin{aligned}
\Rightarrow \omega_n^2 = 1 &\quad \rightarrow \quad \omega_n = 1 \\
2\zeta\omega_n = 1 &\quad \rightarrow \quad \zeta = 0.5
\end{aligned}
$$

so the time solution is

$$x(t) = e^{-\zeta\omega_n t}\left(c_1 \cos(\omega_d T) + c_2 \sin(\omega_d t)\right)$$

where $\omega_d = \omega_n\sqrt{1-\zeta^2} = \sqrt{1-(0.5)^2} = \sqrt{3}/2$.
Apply the ICs to find c_1 and c_2:

$$
\begin{aligned}
x(0) &= c_1 = -1 \quad \Rightarrow c_1 = -1 \\
\dot{x}(0) &= \omega_d c_2 - \zeta\omega_n c_1 = 0 \quad \Rightarrow c_2 = -1/\sqrt{3}
\end{aligned}
$$

so

$$x(t) = e^{-1/2\ t}\left[\cos(\frac{\sqrt{3}}{2}t - \frac{1}{\sqrt{3}}\sin\frac{\sqrt{3}}{2}t\right]$$

or

$$
\begin{aligned}
r &= \sqrt{(-1)^2 + (\frac{-1}{\sqrt{3}})^2} = \frac{2}{\sqrt{3}} \\
\delta &= \text{atan2}\left(-1, -\frac{1}{\sqrt{3}}\right) = 120^\circ
\end{aligned}
$$

2. Now apply the P.D. controller

$$f_c = -k_p x - k_v \dot{x}$$

what gains do we need to achieve critical damping and stiffness=16?

$$
\begin{aligned}
\text{Dynamics:} &\qquad f = m\ddot{x} + b\dot{x} + kv \\
\text{Control:} &\qquad f_c = -k_p x - k_v \dot{x}
\end{aligned}
$$

The closed–loop dynamics are

$$
\begin{aligned}
m\ddot{x} + b\dot{x} + kx &= -k_p x - k_v \dot{x} \\
m\ddot{x} + (b + k_v)\dot{x} + (k + k_p)x &= 0
\end{aligned}
$$

To find the stiffness:

$$
\begin{aligned}
k_{total} &= k + k_p \\
16 &= 1 + k_p \\
\Rightarrow k_p &= 15
\end{aligned}
$$

to find the damping ($\zeta = 1$):

$$
\begin{aligned}
2\zeta\omega_n &= \frac{b + k_v}{m} \Rightarrow k_v = \underbrace{2\zeta}_{1}\underbrace{\omega_n}_{?}\underbrace{m}_{1} - \underbrace{b}_{1} \\
\omega_n^2 &= \frac{k + k_p}{m} = \frac{1 + 15}{1} = 16 \Rightarrow \omega_n = 4 \\
\Rightarrow k_v &= 2(1)(4)(1) - 1 \\
\Rightarrow k_v &= 7
\end{aligned}
$$