State-Space Modelling

"Vectors and matrices are the very language of state-space methods"

– Friedland

METR 4202: Advanced Control & **Robotics**

Dr Surya Singh -- Lecture # 10

October 8, 2014

metr4202@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~metr4202/

Schedule

Week	Date	Lecture (W: 11:10-12:40, 24-402)
1	30-Jul	Introduction
2	6-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	13-Aug	Robot Kinematics (& Ekka Day)
4	20-Aug	Robot Dynamics & Control
5	27-Aug	Robot Motion
6	3-Sep	Robot Sensing: Perception & Multiple View Geometry
7	10-Sep	Robot Sensing: Features & Detection using Computer Vision
8	17-Sep	Navigation (+ Prof. M. Srinivasan)
9	24-Sep	Localization & Motion Planning + Control
	1-Oct	*Study break*
10	8-Oct	**State-Space Modelling**
11	15-Oct	Shaping the Dynamic Response
12	22-Oct	Linear Observers & LQR
13	29-Oct	Applications in Industry & Course Review

Announcements: Lab 2 Extended

- **Feedback from Lab 1!**
- **Lab 2 Report:**
  - Same as last semester
  - This time it's public
  - "Private" (PAF) comments are in the **online Lab2 PAF**
- **Lab 3:**
  - Cup stack/unstacking
  - Hope to get it written by this afternoon.
  - Might be a "live document" like last year
- Cool Robotics Share Site
  ➔ http://metr4202.tumblr.com/

Cool
Robotics
Video
Share

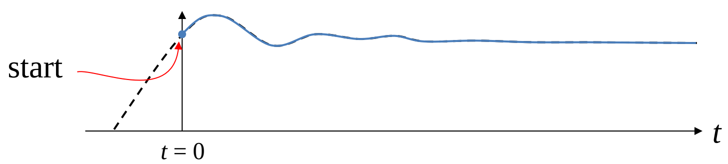
State-Space Modelling (from 2013 – Sorry!)

("Hear Ye! It be stated")

Affairs of state

- Introductory brain-teaser:
  - If you have a dynamic system model with history (ie. integration) how do you represent the instantaneous state of the plant?

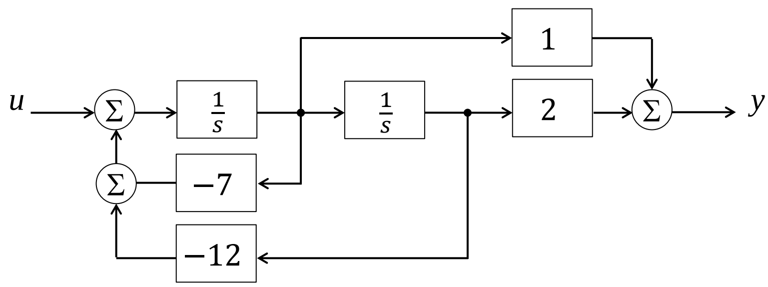
Eg. how would you setup a simulation of a step response, mid-step?



Introduction to state-space

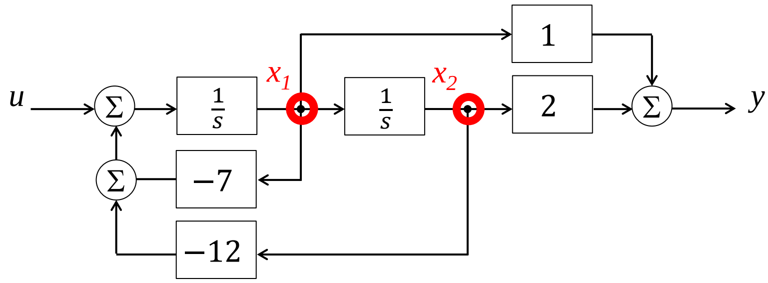
- Linear systems can be written as networks of simple dynamic elements:

$$H = \frac{s+2}{s^2+7s+12} = \frac{2}{s+4} + \frac{-1}{s+3}$$



Introduction to state-space

- We can identify the nodes in the system
  - These nodes contain the integrated time-history values of the system response
  - We call them "states"



Linear system equations

- We can represent the dynamic relationship between the states with a linear system:

$$\dot{x}_1 = -7x_1 - 12x_2 + u$$
$$\dot{x}_2 = x_1 + 0x_2 + 0u$$

$$y = x_1 + 2x_2 + 0u$$

State-space representation

- We can write linear systems in matrix form:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -7 & 12 \\ 1 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$
$$\boldsymbol{y} = \begin{bmatrix} 1 & 2 \end{bmatrix} \boldsymbol{x} + 0u$$

Or, more generally:

$$\left.\begin{aligned} \dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x} + \mathbf{B}u \\ y &= \mathbf{C}\boldsymbol{x} + Du \end{aligned}\right\} \text{“State-space equations”}$$

State-space representation

- State-space matrices are not necessarily a unique representation of a system
  - There are two common forms
- <u>Control canonical form</u>
  - Each node – each entry in $\boldsymbol{x}$ – represents a state of the system (each order of s maps to a state)
- <u>Modal form</u>
  - Diagonals of the state matrix **A** are the poles (“modes”) of the transfer function

State variable transformation

- Important note!
  - The states of a control canonical form system are not the same as the modal states
  - They represent the same dynamics, and give the same output, but the vector values are different!

- However we can convert between them:
  - Consider state representations, $\boldsymbol{x}$ and $\boldsymbol{q}$ where

$$\boldsymbol{x} = \mathbf{T}\boldsymbol{q}$$

$\mathbf{T}$ is a "transformation matrix"

State variable transformation

- Two homologous representations:

$$\begin{matrix} \dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}u \\ y = \mathbf{C}\boldsymbol{x} + Du \end{matrix} \quad \text{and} \quad \begin{matrix} \dot{\boldsymbol{q}} = \mathbf{F}\boldsymbol{q} + \mathbf{G}u \\ y = \mathbf{H}\boldsymbol{q} + Ju \end{matrix}$$

We can write:

$$\dot{\boldsymbol{x}} = \mathbf{T}\dot{\boldsymbol{q}} = \mathbf{AT}\boldsymbol{z} + \mathbf{B}u$$
$$\dot{\boldsymbol{q}} = \mathbf{T}^{-1}\mathbf{AT}\boldsymbol{z} + \mathbf{T}^{-1}\mathbf{B}u$$

Therefore, $\mathbf{F} = \mathbf{T}^{-1}\mathbf{AT}$ and $\mathbf{G} = \mathbf{T}^{-1}\mathbf{B}$

Similarly, $\mathbf{C} = \mathbf{HT}$ and $D = J$

Controllability matrix

- To convert an arbitrary state representation in **F**, **G**, **H** and *J* to control canonical form **A**, **B**, **C** and *D*, the "controllability matrix"

$$\mathcal{C} = [\mathbf{G} \quad \mathbf{FG} \quad \mathbf{F}^2\mathbf{G} \quad \cdots \quad \mathbf{F}^{n-1}\mathbf{G}]$$

must be nonsingular.

Why is it called the "controllability" matrix?

Example:
(Back To) Robot Arms

Slides 17-27 Source: R. Lindeke, ME 4135, "Introduction to Control"

Remembering the Motion Models:

- Recall from Dynamics, the Required Joint Torque is:

$$\tau_i = D_i\left(q\right)\ddot{q}_i + C_i\left(q,\dot{q}_i\right) + h\left(q\right) + b\left(\dot{q}_i\right)$$

Dynamical Manipulator Inertial Tensor – a function of position and acceleration

Coupled joint effects (centrifugal and coriolis) issues due to multiple moving joints

Gravitational Effects

Frictional Effect due to Joint/Link movement

Lets simplify the model

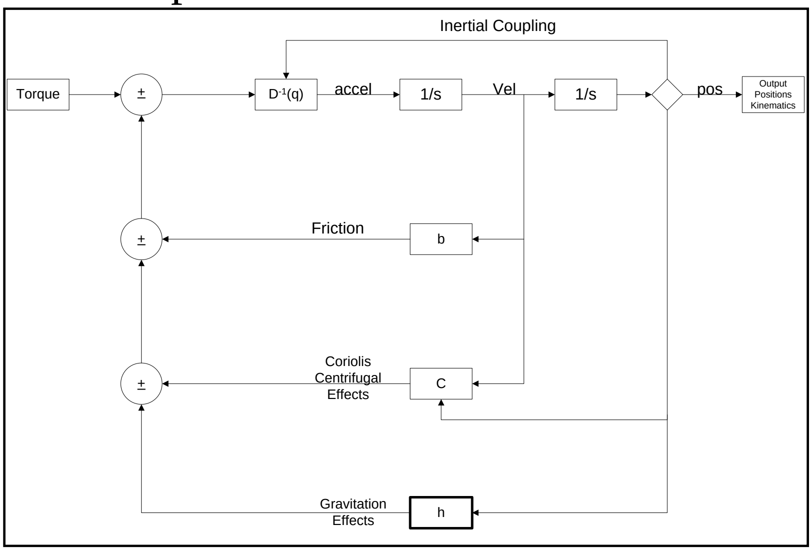
- This torque model is a 2nd order one (in position) lets look at it as a velocity model rather than positional one then it becomes a system of highly coupled 1st order differential equations
- We will then isolate Acceleration terms (acceleration is the 1st derivative of velocity)

$$a = \dot{v} = \ddot{q} = D_i^{-1}(q)\left(\tau_i - C_i(q,\dot{q}_i) - h(q) - b\left(\dot{q}_1\right)\right)$$

Considering Control:

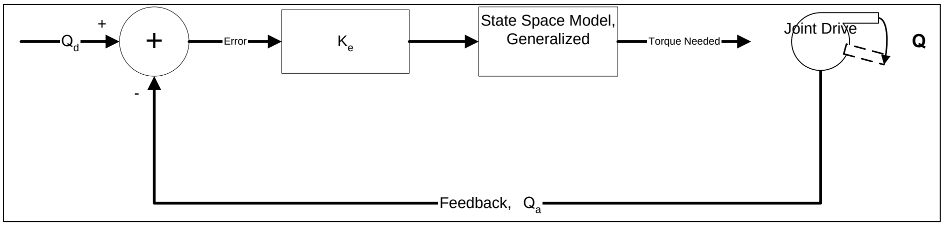
- Each Link's torque is influenced by each other links motion
  - We say that the links are highly coupled
- Solution then suggests that control should come from a simultaneous solution of these torques
- We will model the solution as a "State Space" design and try to balance the torque-in with *positional control*-out – the most common way it is done!
  - But we could also use 'force control' to solve the control problem!

The State-Space Control Model:



Setting up a Real Control

- We will (start) by using positional error to drive our torque devices



- This simple model is called a PE (proportional error) controller

PE Controller:

- To a 1st approximation, $\tau = K_m * I$
  - Torque is proportional to motor current

- And the Torque required is a function of 'Inertial' (Acceleration) and 'Friction' (velocity) effects as suggested by our L-E models

$$\tau_m \simeq J_{eq}\ddot{q} + F_{eq}\dot{q}$$

→ Which can be approximated as:

$$K_m I_m = J_{eq}\ddot{q} + F_{eq}\dot{q}$$

Setting up a "Control Law"

- We will use the positional error (as drawn in the state model) to develop our torque control
- We say then for PE control:

$$\tau \propto k_{pe}(\theta_d - \theta_a)$$

- Here, k_{pe} is a "gain" term that guarantees sufficient current will be generated to develop appropriate torque based on observed positional error

Using this Control Type:

- It is a representation of the physical system of a mass on a spring!
- We say after setting our target as a 'zero goal' that:

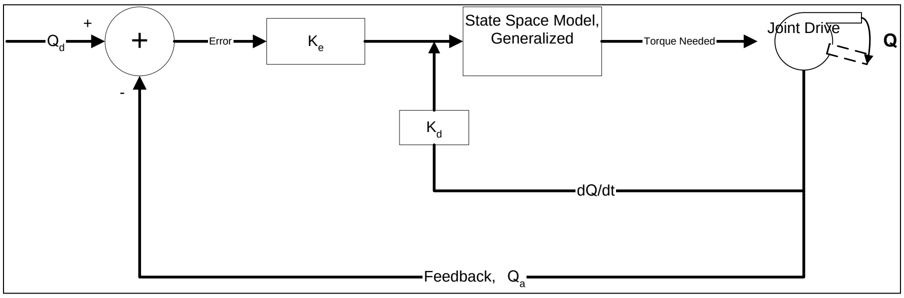
$$-k_{pe} * \theta_a = J\ddot{\theta} + F\dot{\theta}$$

the solution of which is:

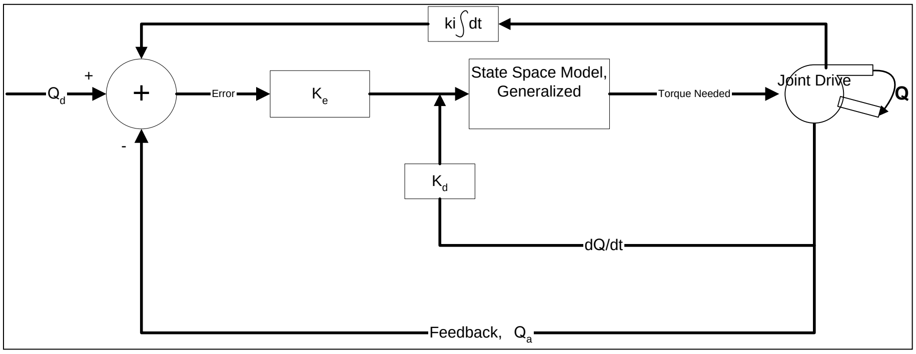
θ_a is a function of the servo feedback as a function of time!

$$\Big(\theta_a = e^{-\left({}^{F}\!/_{2J}\right)t}\left[C_1 e^{\left({}^{1}\!/_{2}\right)\omega t} + C_2 e^{-\left({}^{1}\!/_{2}\right)\omega t}\right]$$

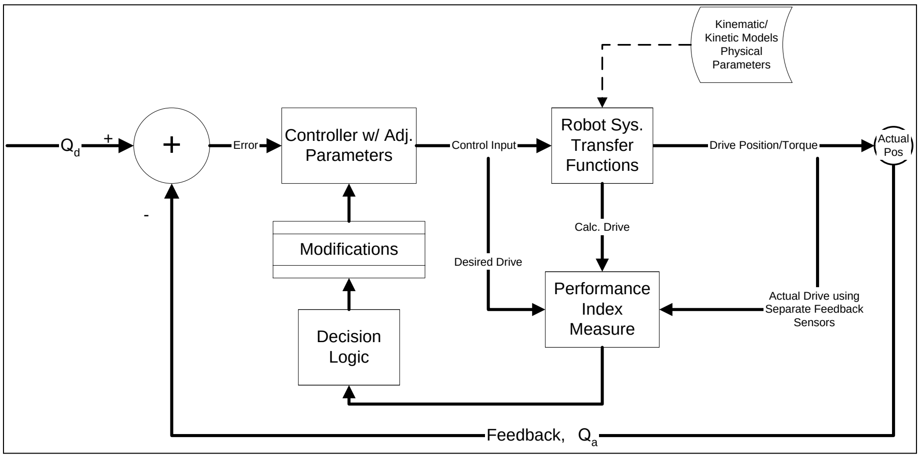
State Space Model of PD:



PID State Space Model:



State Model of Adjustable Controller



Controllability

Controllability matrix

- If you can write it in CCF, then the system equations must be linearly independent.
- Transformation by any nonsingular matrix preserves the controllability of the system.
- Thus, a nonsingular controllability matrix means $\boldsymbol{x}$ can be driven to any value.

State evolution

- Consider the system matrix relation:

$$\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$$
$$y = \mathbf{H}\boldsymbol{x} + Ju$$

The time solution of this system is:

$$\boldsymbol{x}(t) = e^{\mathbf{F}(t-t_0)}\,\boldsymbol{x}(t_0) + \int_{t_0}^{t} = e^{\mathbf{F}(t-\tau)}\,\mathbf{G}u(\tau)d\tau$$

If you didn't know, the matrix exponential is:

$$e^{\mathbf{K}t} = \mathbf{I} + \mathbf{K}t + \frac{1}{2!}\mathbf{K}^2t^2 + \frac{1}{3!}\mathbf{K}^3t^3 + \cdots$$

Stability

- We can solve for the natural response to initial conditions $\boldsymbol{x_0}$:

$$\boldsymbol{x}(t) = e^{p_i t}\boldsymbol{x}_0$$
$$\therefore \dot{\boldsymbol{x}}(t) = p_i e^{p_i t}\boldsymbol{x}_0 = \mathbf{F}e^{p_i t}\boldsymbol{x}_0$$

Clearly, a system will be stable provided
$\mathrm{eig}(\mathbf{F}) < 0$

Characteristic polynomial

- From this, we can see $\mathbf{F}\boldsymbol{x}_0 = p_i\boldsymbol{x}_0$

or, $(p_i\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$

which is true only when $\det(p_i\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$

Aka. the characteristic equation!

- We can reconstruct the CP in s by writing:

$$\det(s\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

Great, so how about control?

- Given $\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$, if we know $\mathbf{F}$ and $\mathbf{G}$, we can design a controller $u = -\mathbf{K}\boldsymbol{x}$ such that

$$\text{eig}(\mathbf{F} - \mathbf{GK}) < 0$$

- In fact, if we have full measurement and control of the states of $\boldsymbol{x}$, we can position the poles of the system in arbitrary locations!

(Of course, that never happens in reality.)

Example: PID control

- Consider a system parameterised by three states:
  - x_1, x_2, x_3
  - where $x_2 = \dot{x}_1$ and $x_3 = \dot{x}_2$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & -2 \end{bmatrix} \boldsymbol{x} - \mathbf{K}u$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \boldsymbol{x} + 0u$$

x_2 is the output state of the system;

x_1 is the value of the integral;

x_3 is the velocity.

- We can choose **K** to move the eigenvalues of the system as desired:

$$\det \begin{bmatrix} 1 - K_1 & & \\ & 1 - K_2 & \\ & & -2 - K_3 \end{bmatrix} = \mathbf{0}$$

All of these eigenvalues must be positive.

It's straightforward to see how adding derivative gain K_3 can stabilise the system.

Just scratching the surface

- There is a lot of stuff to state-space control

- One lecture (or even two) can't possibly cover it all in depth

Go play with Matlab and check it out!

Discretisation FTW!

- We can use the time-domain representation to produce difference equations!

$$\boldsymbol{x}(kT+T) = e^{\mathbf{F}T}\,\boldsymbol{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+T-\tau)}\mathbf{G}u(\tau)d\tau$$

Notice $\boldsymbol{u}(\tau)$ is not based on a discrete ZOH input, but rather an integrated time-series.

We can structure this by using the form:

$$u(\tau) = u(kT), \qquad kT \le \tau \le kT + T$$

Discretisation FTW!

- Put this in the form of a new variable:

$$\eta = kT + T - \tau$$

Then:

$$\boldsymbol{x}(kT+T) = e^{\boldsymbol{F}T}\boldsymbol{x}(kT) + \left(\int_{kT}^{kT+T} e^{\boldsymbol{F}\eta}d\eta\right)\boldsymbol{G}u(kT)$$

Let's rename $\boldsymbol{\Phi} = e^{\boldsymbol{F}T}$ and $\boldsymbol{\Gamma} = \left(\int_{kT}^{kT+T} e^{\boldsymbol{F}\eta}d\eta\right)\boldsymbol{G}$

Discrete state matrices

So,

$$\boldsymbol{x}(k+1) = \boldsymbol{\Phi x}(k) + \boldsymbol{\Gamma}u(k)$$
$$y(k) = \mathbf{H}\boldsymbol{x}(k) + \mathbf{J}\boldsymbol{u}(k)$$

Again, $\boldsymbol{x}(k+1)$ is shorthand for $\boldsymbol{x}(kT+T)$

Note that we can also write $\boldsymbol{\Phi}$ as:

$$\boldsymbol{\Phi} = \mathbf{I} + \mathbf{F}T\boldsymbol{\Psi}$$

where

$$\boldsymbol{\Psi} = \mathbf{I} + \frac{\mathbf{F}T}{2!} + \frac{\mathbf{F}^2T^2}{3!} + \cdots$$

Simplifying calculation

- We can also use $\boldsymbol{\Psi}$ to calculate $\boldsymbol{\Gamma}$
  - Note that:

$$\Gamma = \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T^k}{(k+1)!} T\mathbf{G}$$
$$= \boldsymbol{\Psi}T\mathbf{G}$$

$\boldsymbol{\Psi}$ itself can be evaluated with the series:

$$\boldsymbol{\Psi} \cong \mathbf{I} + \frac{\mathbf{F}T}{2}\left\{\mathbf{I} + \frac{\mathbf{F}T}{3}\left[\mathbf{I} + \cdots \frac{\mathbf{F}T}{n-1}\left(\mathbf{I} + \frac{\mathbf{F}T}{n}\right)\right]\right\}$$

State-space z-transform

We can apply the z-transform to our system:

$$(z\mathbf{I} - \mathbf{\Phi})\boldsymbol{X}(z) = \mathbf{\Gamma}U(k)$$
$$Y(z) = \mathbf{H}\boldsymbol{X}(z)$$

which yields the transfer function:

$$\frac{Y(z)}{\boldsymbol{X}(z)} = G(z) = \mathbf{H}(z\mathbf{I} - \mathbf{\Phi})^{-\mathbf{1}}\mathbf{\Gamma}$$

State-space control design

- Design for discrete state-space systems is just like the continuous case.
  - Apply linear state-variable feedback:

$$u = -\mathbf{K}\boldsymbol{x}$$

such that $\det(z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K}) = \alpha_c(z)$

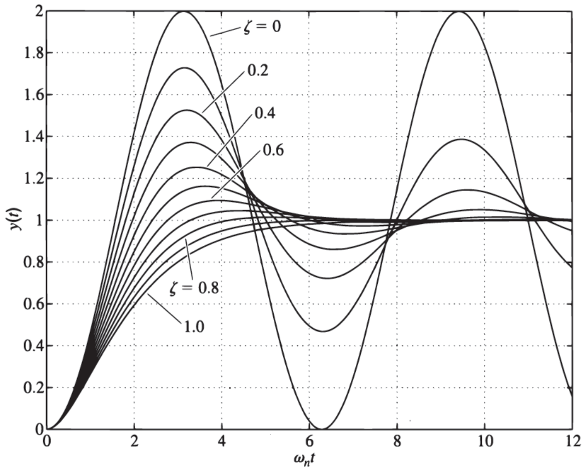
where $\alpha_c(z)$ is the desired control characteristic equation

Predictably, this requires the system controllability matrix $\boldsymbol{\mathcal{C}} = [\mathbf{\Gamma} \quad \mathbf{\Phi}\mathbf{\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}]$ to be full-rank.

2nd Order System Response

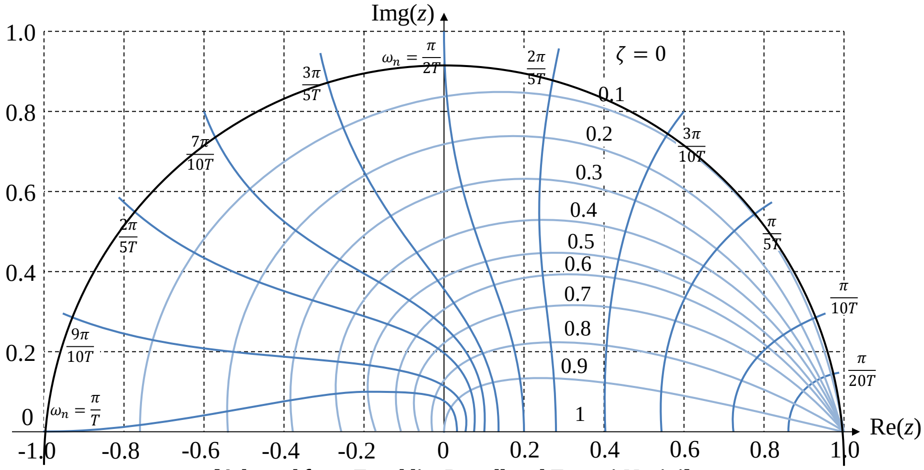
2nd Order System Response

- Response of a 2nd order system to increasing levels of damping:



Damping and natural frequency

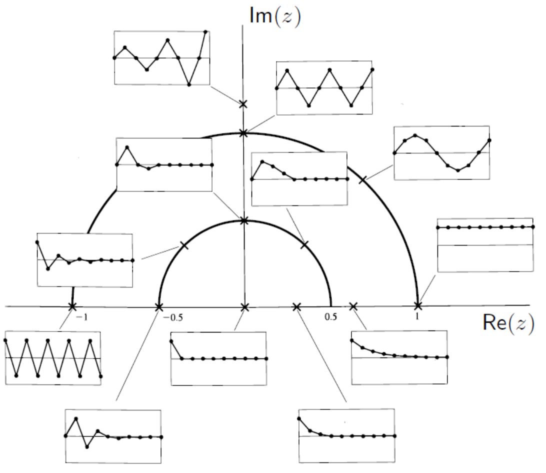
$$z = e^{sT} \text{ where } s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$



[Adapted from Franklin, Powell and Emami-Naeini]

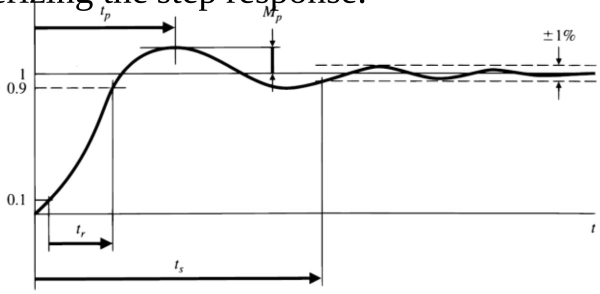
Pole positions in the z-plane

- Poles inside the unit circle are **stable**
- Poles outside the unit circle **unstable**
- Poles on the unit circle are oscillatory
- Real poles at 0 < z < 1 give exponential response
- Higher frequency of oscillation for larger
- Lower apparent damping for larer and r



2^{nd} Order System Specifications

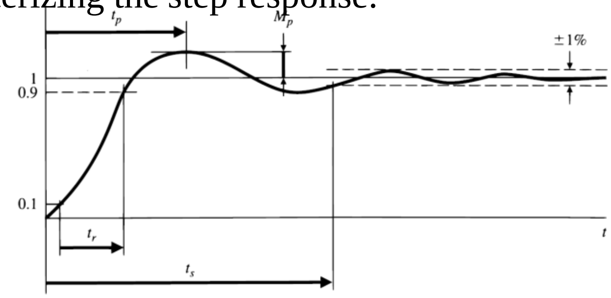
Characterizing the step response:



- Rise time (10% → 90%): $t_r \approx \frac{1.8}{\omega_0}$
- Overshoot: $M_p \approx \frac{e^{-\pi\zeta}}{\sqrt{1-\zeta^2}}$
- Settling time (to 1%): $t_s = \frac{4.6}{\zeta\omega_0}$
- Steady state error to unit step: e_{ss}
- Phase margin:
  $\phi_{PM} \approx 100\zeta$

2^{nd} Order System Specifications

Characterizing the step response:



- Rise time (10% → 90%) & Overshoot:
  t_r, M_p → ζ, ω_0 : Locations of dominant poles
- Settling time (to 1%):
  t_s → radius of poles: $|z| < 0.01^{\frac{T}{t_s}}$
- Steady state error to unit step:
  e_{ss} → final value theorem $e_{ss} = \lim_{z\to 1} \{(z-1)\,F(z)\}$

Ex: System Specifications → Control Design [1/4]

Design a controller for a system with:

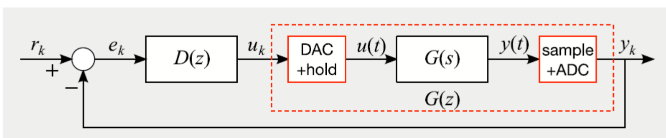
- A continuous transfer function: $G(s) = \dfrac{0.1}{s(s+0.1)}$
- A discrete ZOH sampler
- Sampling time (T_s): T_s= 1s
- C $u_k = -0.5u_{k-1} + 13\left(e_k - 0.88e_{k-1}\right)$

The closed loop system is required to have:

- M_p < 16%
- t_s < 10 s
- e_{ss} < 1

Ex: System Specifications → Control Design [2/4]

1. (a) Find the pulse transfer function of $G(s)$ plus the ZOH



$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\} = \frac{(z-1)}{z}\mathcal{Z}\left\{\frac{0.1}{s^2(s+0.1)}\right\}$$

e.g. look up $\mathcal{Z}\{a/s^2(s+a)\}$ in tables:

$$G(z) = \frac{(z-1)}{z}\,\frac{z\Big((0.1 - 1 + e^{-0.1})z + (1 - e^{-0.1} - 0.1e^{-0.1})\Big)}{0.1(z-1)^2(z - e^{-0.1})}$$

$$= \frac{0.0484(z + 0.9672)}{(z-1)(z - 0.9048)}$$

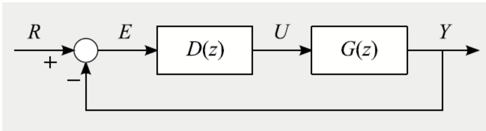
(b) Find the controller transfer function (using z = shift operator):

$$\frac{U(z)}{E(z)} = D(z) = 13\frac{(1 - 0.88z^{-1})}{(1 + 0.5z^{-1})} = 13\frac{(z - 0.88)}{(z + 0.5)}$$

Ex: System Specifications → Control Design [3/4]

2. Check the steady state error e_{ss} when r_k = unit ramp

$$e_{ss} = \lim_{k\to\infty} e_k = \lim_{z\to 1}(z-1)E(z)$$



$$\frac{E(z)}{R(z)} = \frac{1}{1+D(z)G(z)}$$

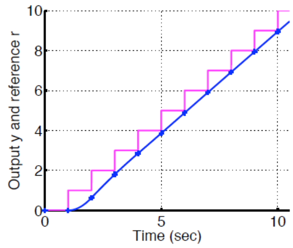
$$R(z) = \frac{Tz}{(z-1)^2}$$

so $$e_{ss} = \lim_{z\to 1}\left\{(z-1)\frac{Tz}{(z-1)^2}\frac{1}{1+D(z)G(z)}\right\} = \lim_{z\to 1}\frac{T}{(z-1)D(z)G(z)}$$

$$= \lim_{z\to 1}\frac{T}{(z-1)\dfrac{0.0484(z+0.9672)}{(z-1)(z-0.9048)}D(1)}$$

$$= \frac{1-0.9048}{0.0484(1+0.9672)D(1)} = 0.96$$

$\Longrightarrow$ $e_{ss} < 1$ (as required)



Ex: System Specifications → Control Design [4/4]

3. Step response: overshoot $M_p < 16\%$ $\Longrightarrow$ $\zeta > 0.5$

settling time $t_s < 10$ $\Longrightarrow$ $|z| < 0.01^{1/10} = 0.63$

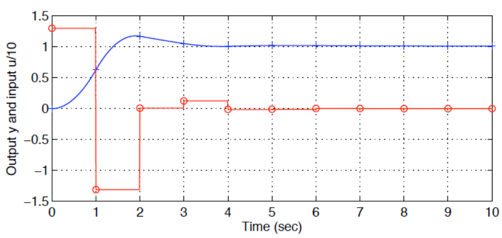
The closed loop poles are the roots of $1 + D(z)G(z) = 0$, i.e.

$$1 + 13\frac{(z-0.88)}{(z+0.5)}\frac{0.0484(z+0.9672)}{(z-1)(z-0.9048)} = 0$$

$$\Longrightarrow z = 0.88,\ -0.050 \pm j0.304$$

But the pole at $z = 0.88$ is cancelled by controller zero at $z = 0.88$, and

$$z = -0.050 \pm j0.304 = re^{\pm j\theta} \quad \Longrightarrow \begin{cases} r = 0.31,\ \theta = 1.73 \\ \zeta = 0.56 \end{cases}$$



all specs satisfied!