Introduction to State-Space

or

"Why didn't you cover this earlier??"

Paul Pounds

28 May 2012

University of Queensland

Previously on Digital Controls…

- We discussed discretised state-space systems.
- Which few people understood or cared about!
- (oh, and a while lot of other stuff related to practical digital control)

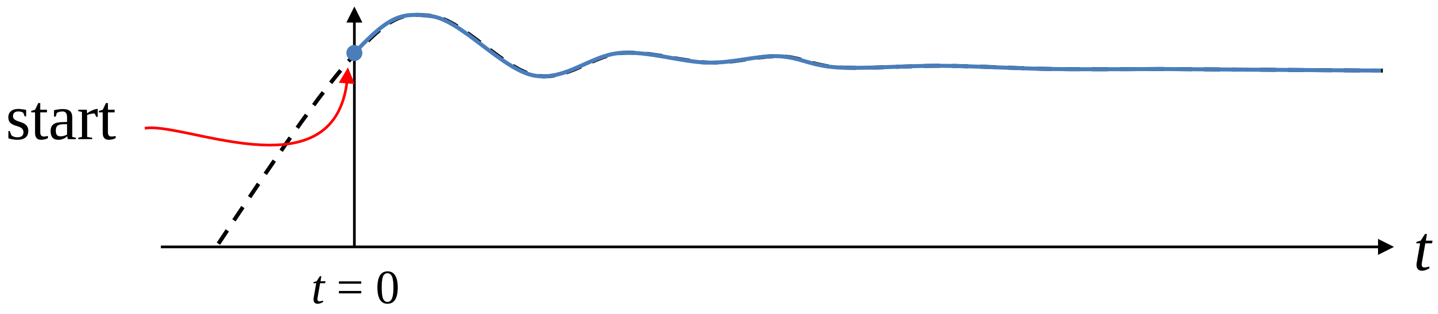
State-space lolwut?

- A ‘clean’ way of representing systems
- Easy implementation in matrix algebra
- Simplifies understanding Multi-Input-Multi-Output (MIMO) systems

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?

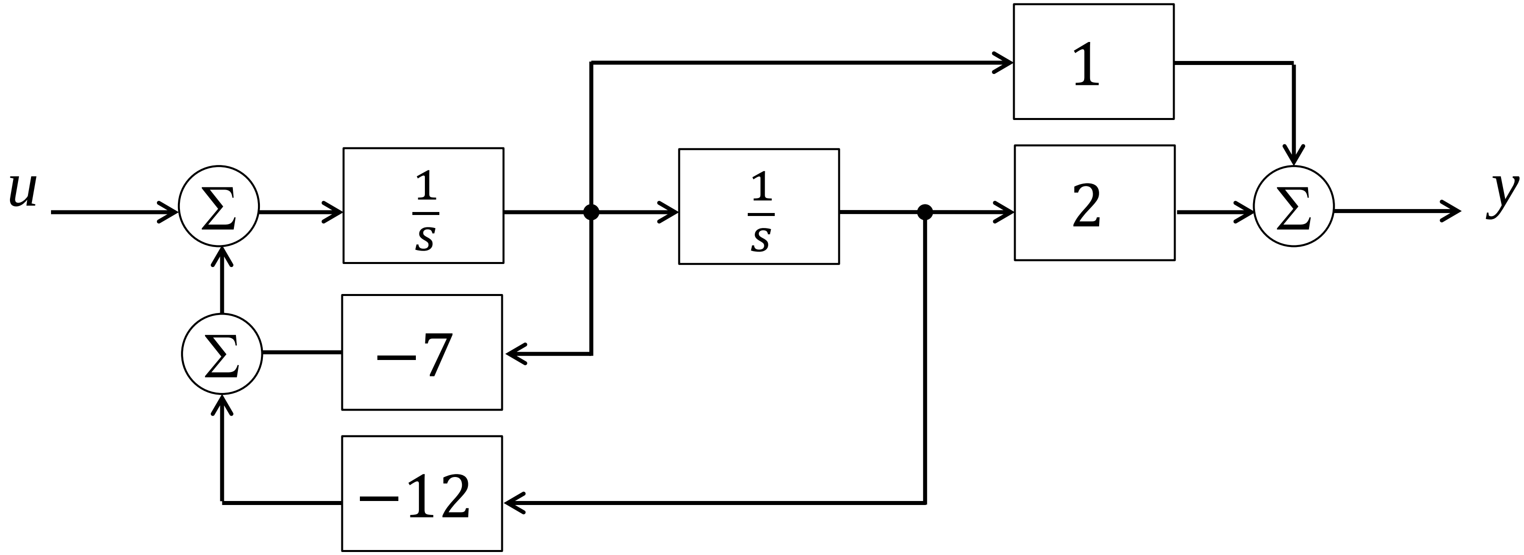
Eg. how would you setup a simulation of a step response, mid-step?



Introduction to state-space

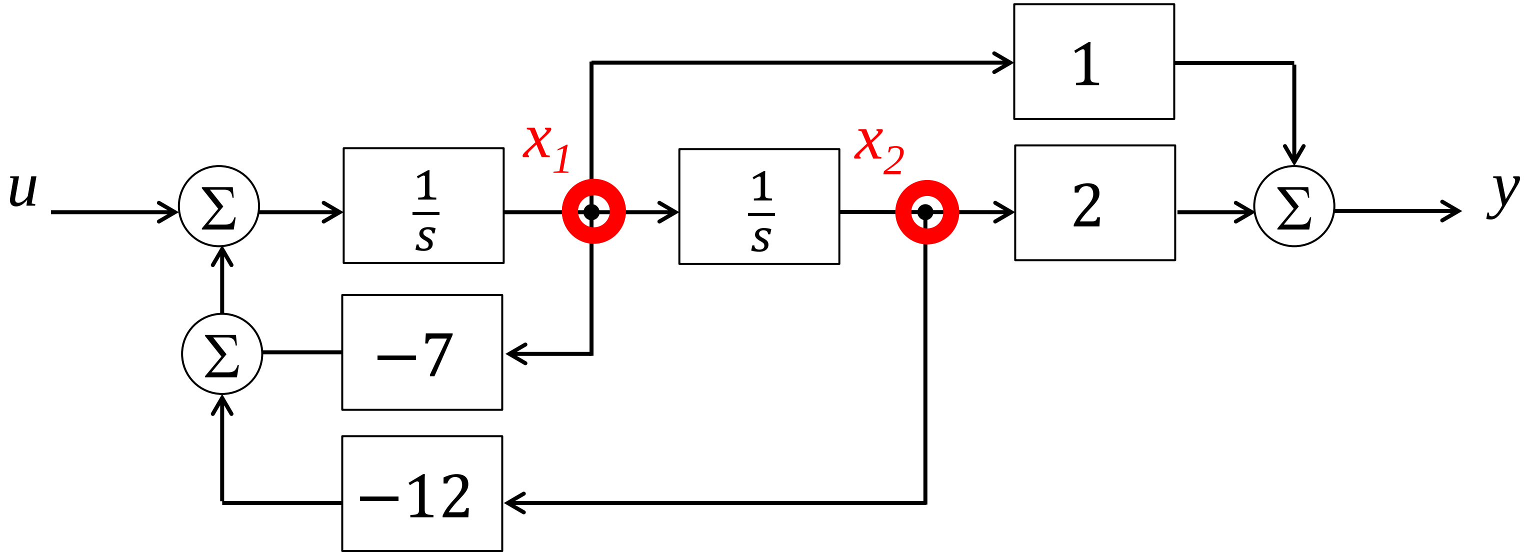
- Linear systems can be written as networks of simple dynamic elements:

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



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
- Control canonical form
  - Each node – each entry in $\boldsymbol{x}$ – represents a state of the system (each order of s maps to a state)
- Modal form
  - Diagonals of the state matrix **A** are the poles (“modes”) of the transfer function

Control canonical form

- CCF matrix representations have the following structure:

$$
\begin{bmatrix}
-a_1 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} & -a_n \\
1 & 0 & & 0 & 0 & 0 \\
0 & 1 & & & & \\
\vdots & & \ddots & & & \vdots \\
 & & & 1 & 0 & 0 \\
0 & 0 & \cdots & 0 & 1 & 0
\end{bmatrix}
$$

Pretty diagonal!

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

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

T is a "transformation matrix"

State variable transformation

- Two homologous representations:

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

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"

$$\mathcal{C} = \begin{bmatrix} \mathbf{G} & \mathbf{FG} & \mathbf{F}^2\mathbf{G} & \cdots & \mathbf{F}^{n-1}\mathbf{G} \end{bmatrix}$$

must be nonsingular.

>deep think<

Why is it called the "controllability" matrix?

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.

Kind of awesome

- The controllability of a system depends on the particular set of states you chose
- You can't tell just from a transfer function whether all the states of $\boldsymbol{x}$ are controllable
- The poles of the system are the Eigenvalues of $\mathbf{F}$, (p_i).

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

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

Characteristic polynomial

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

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

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
$$\mathrm{eig}(\mathbf{F} - \mathbf{G}\mathbf{K}) < 0$$

- 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!

And now for…

Estimation and Kalman Filtering

starring

Surya Singh!

Fun Fact: In Soviet Russia, State controls you!

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,

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

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 **Ψ** to calculate **Γ**
  - Note that:

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

$$= \mathbf{\Psi} T \mathbf{G}$$

Ψ itself can be evaluated with the series:

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

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:

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

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

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

Predictably, this requires the system controllability matrix

$\boldsymbol{\mathcal{C}} = [\boldsymbol{\Gamma} \quad \boldsymbol{\Phi\Gamma} \quad \boldsymbol{\Phi}^2\boldsymbol{\Gamma} \quad \cdots \quad \boldsymbol{\Phi}^{n-1}\boldsymbol{\Gamma}]$ to be full-rank.