EE 16B Lecture Notes

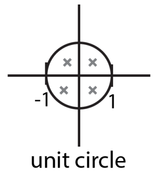
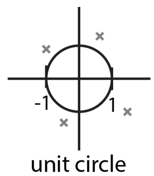
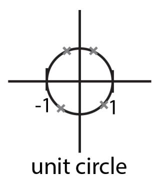
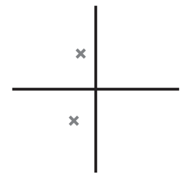
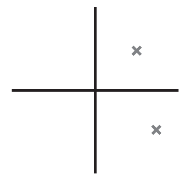
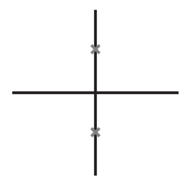
Observability

April 13, 2016

Today, we cover the differences between discrete time and continuous time in regards to stability, and then we discuss them in relation to observability.

An analysis of stability and controllability of discrete time systems and continuous time systems is included in the table on the next page.

Table 1: Differences between discrete time and continuous

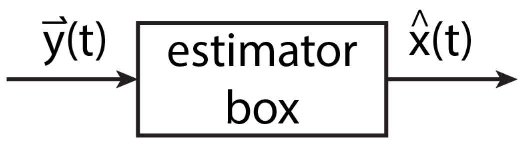
	Discrete Time	Continuous Time
Model	$\vec{x}(t+1) = A\vec{x}(t) + B\vec{u}(t) + \vec{w}$ $\vec{y}(t) = C\vec{x}(t)$	$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) + \vec{w}$ $\vec{y}(t) = C\vec{x}(t)$
Stability	Stable iff. all $\lambda < 1$  Unstable if $\exists \lvert\lambda\rvert > 1$  Marginally unstable if not unstable but $\exists \lambda, \lvert\lambda\rvert = 1$ 	Stable iff. all $Re(\lambda) < 0$  Unstable if $\exists Re(\lambda) > 0$  Marginally unstable if not unstable but $\exists \lambda, Re(\lambda) = 0$ 
Controllability	$\text{rank}\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} = n$ observe: $\text{rank}\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = n$	$\longleftarrow$same $\longleftarrow$same
Controllable Canonical Form	$A = \begin{bmatrix} \vec{0} & & I & \\ a_0 & a_1 & \dots & a_{n-1} \end{bmatrix}, \vec{b} = \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$ (A+Bk)	$\longleftarrow$same $\frac{d}{dt}\vec{x}(t) = (A + Bk)\vec{x}(t)$ $u(t) = K\vec{x}(t)$

Observers:

The following analysis will be done in discrete time, however it all holds for continuous time as well.

So we ask the question: Is it possible to track a state? We have the discrete time system below:

Assume that at first $u(t) = 0$ and $\vec{w}(t) = 0$.



We observe $\vec{y}(t)$, and we build a "estimator box" that pops out an estimate, $\hat{x}(t)$. So how do we build this box? Ideally, we want some computational structure for which we can understand it's evolution through time. So the key idea of what we're doing is that in the real world, there is the physical state x and we have observations of these states, and I want to take this and create an internal model in the "box" of what's happening in the real world. We create this internal model by using states that are computational. We "make a copy" of the system,$\hat{x}$, and apply an internal control of that system $\hat{u}(t)$:

$\hat{x}(t+1) = A\hat{x}(t) + \vec{u}(t)$

We want to make to make the input a function of our $\vec{y}(t)$. So given we have an internal model, we're going to make the internal model of what the observation should be:

$\hat{y}(t) = C\hat{x}(t)$

Since the internal model we have of the state might be wrong, what evidence do we have that it's wrong? The evidence is when there is a difference in the observation we expect and the observation we get. So we set the control to be a function of the difference of the observations:

$\hat{u}(t) = R(\vec{y}(t) - \hat{y}(t))$

We then use this difference to change our estimator. What happens when we check this?

$\hat{x}(t+1) = A\hat{x}(t) + K(\vec{y}(t) - \hat{y}(t)) = A\hat{x}(t) + KC[\vec{x}(t) - \hat{x}(t)]$

My estimator $\hat{x}$ should be close to $\vec{x}$, so I'm mainly interested in the difference, so I care about how the difference of these values evolves, or the error, $\vec{e}$.

$\vec{e}(t) = \vec{x}(t) - \hat{x}(t)$

So how does $\vec{e}(t)$ evolves?

$\vec{e}(t+1) = \vec{x}(t+1) - \hat{x}(t+1) = A(\vec{x}(t) - \hat{x}(t)) - KC[\vec{x}(t) - \hat{x}(t)]$
$= A\vec{e}(t) - KC\vec{e}(t)$
$= (A - KC)\vec{e}(t)$

So notice, the evolution of the error is only in terms of the last error, so we can just focus on the $A - KC$ term.