Lecture 2

Today's Goal

- Linearization
- Stability definitions
- Simulation in Matlab/Simulink

Material

- Glad& Ljung Ch. 11, 12.1,
 (Khalil Ch 2.3, part of 4.1, and 4.3)
 (Slotine and Li, pp 40-57)
- Lecture slides

Example - Linearization around equilibrium point

The linearization of

$$\ddot{x}(t) = f(x(t)) = \frac{g}{t} \sin x(t)$$

around the equilibrium $x_0 = n\pi$ is given by

$$\ddot{x}(t) = f'(x_0)\tilde{x}(t) = \frac{g}{t}(-1)^n \tilde{x}(t)$$

2 minute exercise: Linearize

$$\dot{x}^2 = 4x^3 + u$$

around the solution

$$x_0(t) = \frac{1}{t^2}$$
 $u_0(t) = 0$

Hint: Plug-in $x(t) = t^{-2} + \tilde{x}(t)$, expand the expressions, and finally remove higher order terms (≥ 2) of \tilde{x} .

State-space form

Hence, for small (\tilde{x}, \tilde{u}) , approximately

$$\dot{\tilde{x}}(t) = A(x_0(t), u_0(t))\tilde{x}(t) + B(x_0(t), u_0(t))\tilde{u}(t)$$

where (if dim x = 2, dim u = 1)

$$\begin{split} A(x_0(t), u_0(t)) &= \frac{\partial f}{\partial x}(x_0(t), u_0(t)) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \Big|_{(x_0(t), u_0(t))} \\ B(x_0(t), u_0(t)) &= \frac{\partial f}{\partial u}(x_0(t), u_0(t)) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \\ \frac{\partial f_2}{\partial u_1} \end{bmatrix} \Big|_{(x_0(t), u_0(t))} \end{split}$$

Note that *A* and *B* are **time dependent**! However, if we don't linearize around a trajectory but linearize around an equilibrium point $(x_0(t), u_0(t)) \equiv (x_0, u_0)$ then *A* and *B* are constant.

To be able to

- linearize, both around equilibria and trajectories,
- explain definitions of local and global stability,
- check local stability and local controllability at equilibria
- simulate in Simulink,

Linearization Around a Trajectory

Idea: Make Taylor-expansion around a known solution $\{x_0(t), u_0(t)\}$ Neglect small terms (*i.e.*, keep the linear terms, as these will locally dominate over the higher order terms).

Let $dx_0/dt = f(x_0(t), u_0(t))$ be a known solution. How will a small deviation $\{\tilde{x}, \tilde{u}\}$ from this solution behave?

$$\frac{d(x_0+\tilde{x})}{dt} = f(x_0(t)+\tilde{x}(t),u_0(t)+\tilde{u}(t))$$



Linearization Around a Trajectory, cont.

Let $(x_0(t), u_0(t))$ denote a solution to $\dot{x} = f(x, u)$ and consider another solution $(x(t), u(t)) = (x_0(t) + \tilde{x}(t), u_0(t) + \tilde{u}(t))$:

$$\begin{split} \dot{x}(t) &= f(x_0(t) + \tilde{x}(t), u_0(t) + \tilde{u}(t)) \\ &= f(x_0(t), u_0(t)) + \frac{\partial f}{\partial x}(x_0(t), u_0(t))\tilde{x}(t) \\ &+ \frac{\partial f}{\partial u}(x_0(t), u_0(t))\tilde{u}(t) + O(\|\tilde{x}, \tilde{u}\|^2) \\ &\underbrace{(x_0(t), u_0(t))}_{\left[\tilde{x}(t)\right]} \\ &\underbrace{(x_0(t) + \tilde{x}(t), u_0(t) + \tilde{u}(t))} \end{split}$$

Linearization, cont'd

The linearization of the output equation

y(t) = h(x(t), u(t))

around the nominal output $y_0(t) = h(x_0(t), u_0(t))$ is given by

$$(y(t) - y_0(t)) = C(t)(x(t) - x_0(t)) + D(t)(u(t) - u_0(t))$$

where (if dim $y = \dim x = 2$, dim u = 1)

$$\begin{split} C(t) &= \left. \frac{\partial h}{\partial x} \right|_{(x_0,u_0)} = \left[\frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} \end{array} \right] \Big|_{(x_0(t),u_0(t))} \\ D(t) &= \left. \frac{\partial h}{\partial u} \right|_{(x_0,u_0)} = \left[\frac{\partial h_1}{\partial u_1} \\ \frac{\partial h_2}{\partial u_1} \right] \Big|_{(x_0(t),u_0(t))} \end{split}$$

$\begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} h(t) = v(t) \\ v(t) = -g + \frac{v_e u(t)}{m(t)} \\ \dot{w}(t) = -u(t) \end{array} \end{array} \end{array} \end{array}$ $\begin{array}{c} \begin{array}{c} \begin{array}{c} h(t) = v(t) \\ v(t) = -g + \frac{v_e u(t)}{m(t)} \\ \dot{w}(t) = -u(t) \end{array} \end{array}$ $\begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} h_0(t) \\ v_0(t) \\ m_0(t) \end{array} \end{array} \end{array} ; m_0(t) = m_0 - u_0 t.$ $\begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \\ \\ \\ \end{array} \end{array}$ $\begin{array}{c} \begin{array}{c} \end{array} \\ \begin{array}{c} \begin{array}{c} \\ \\ \\ \end{array} \end{array} \end{array}$ $\begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \\ \\ \end{array} \end{array} = \left[\begin{array}{c} \begin{array}{c} 0 & 1 & 0 \\ 0 & 0 & \frac{-v_e u_0}{m_0(t)^2} \\ 0 & 0 & 0 \end{array} \right] \tilde{x}(t) + \left[\begin{array}{c} 0 \\ \frac{v_e}{m_0(t)} \\ \frac{v_e}{m_0(t)} \\ -1 \end{array} \right] \tilde{u}(t)$

Definition: A norm function $||x|| : x \in \mathbb{R}^n \to \mathbb{R}$ satisfies the following three properties:

- ▶ ||x|| = 0 if and only if (iff) x = 0, ||x|| > 0 otherwise.
- ► ||ax|| = a||x||, for any positive *a* and any signal vector *x*.
- (The triangle inequality) $||x + y|| \le ||x|| + ||y||$





Definition: (Euclidean norm)

$$\|x\| = (x_1^2 + x_2^2 + \ldots + x_n^2)^{1/2}$$
 x_3

Asymptotic Stability

Definition The equilibrium x = 0 is locally asymptotically stable (LAS) if it

1) is stable

2) there exists r > 0 so that if ||x(0)|| < r then

 $x(t) \longrightarrow 0 \text{ as } t \longrightarrow \infty.$

(PhD-exercise: Show that 1) does not follow from 2))

Part II: Stability definitions

Local Stability

Consider $\dot{x} = f(x)$ where f(0) = 0

Definition The equilibrium x = 0 is **stable** if, for any R > 0, there exists r > 0, such that

 $||x(0)|| < r \implies ||x(t)|| < R$, for all $t \ge 0$

Otherwise the equilibrium point is unstable.



Global Asymptotic Stability

Definition The equilibrium is said to be **globally asymptotically stable (GAS)** if it is LAS and for all x(0) one has

 $x(t) \rightarrow 0 \text{ as } t \rightarrow \infty.$

Lyapunov's Linearization Method

Theorem Assume

has the linearization

$$\frac{d}{dt}(x(t) - x_0) = A(x(t) - x_0)$$

 $\dot{x} = f(x)$

around the equilibrium point x_0 and put

 $\alpha(A) = \max \operatorname{Re}(\lambda(A))$

• If $\alpha(A) < 0$, then $\dot{x} = f(x)$ is LAS at x_0 ,

• If $\alpha(A) > 0$, then $\dot{x} = f(x)$ is unstable at x_0 ,

• If $\alpha(A) = 0$, then no conclusion can be drawn.

(Proof in Lecture 4)

2

Part III: Check local stability and controllability

Example

The linearization of

$$\dot{x}_1 = -x_1^2 + x_1 + \sin(x_2) \dot{x}_2 = \cos(x_2) - x_1^3 - 5x_2$$

at $x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ gives $A = \begin{pmatrix} -1 & 1 \\ -3 & -5 \end{pmatrix}$

Eigenvalues are given by the characteristic equation

$$0 = \det(\lambda I - A) = (\lambda + 1)(\lambda + 5) + 3$$

This gives $\lambda = \{-2, -4\}$, which are both in the left half-plane, hence the *nonlinear system* is LAS around x_0 .

5 minute exercise:

Is the ball and beam

$$\frac{7}{5}\ddot{x} = x\dot{\phi}^2 + g\sin\phi + \frac{2r}{5}\ddot{\phi}$$

nonlinearly locally controllable around $\dot{\phi} = \phi = x = \dot{x} = 0$ (with $\ddot{\phi}$ as input)?

Remark: This is a bit bit more detailed model of the ball and beam than we saw in Lecture 1.

However...

And now for the major limitation: The system works only in situations where the car can continuously back up into a space — not for those tight spots where you must inch your way into a space by going back and forth, wrestling with the wheel.

Unfortunately, such spots are quite common in Japan. And that's precisely when you wish you had a smart car that would graciously help you park.

For me, the parking system also took some getting used to.

You can't turn the car too much before you start parking because the car will get confused and tell you to start over. You must decisively glide straight into pre-parking position before the car will let you begin jigging the arrows on the panel.

When I tried the system in our tiny parking lot at home, the system kept flashing warnings on the screen that the car was too close or too far from where I wanted to park.

Bosch 2008 (Automatic parking assistance)

- Multiple turns
- parking lot > car length + 80 cm

More parking in lecture 12

Example, cont.

The linearization around $x_1 = x_2 = 0, u = 0$ is given by

$$\dot{x}_1 = x_2$$

 $\dot{x}_2 = rac{g}{l} x_1$

It is not controllable, hence no conclusion can be drawn about nonlinear controllability

However, simulations show that the system is stabilized by

$$u(t) = \varepsilon \omega^2 \sin(\omega t)$$

if ω is large enough !

Demonstration We will come back to this example later.

Local Controllability

Theorem Assume

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

has the linearization

$$\frac{d}{dt}(x(t) - x_0) = A(x(t) - x_0) + B(u(t) - u_0)$$

around the equilibrium (x_0, u_0) then

• (A, B) controllable $\Rightarrow f(x, u)$ nonlinear locally controllable

Here nonlinear locally controllable is defined as:

For every T > 0 and $\varepsilon > 0$ the set of states x(T) that can be reached from $x(0) = x_0$, by using controls satisfying $||u(t) - u_0|| < \varepsilon$, contains a small ball around x_0 .

okmarks <u>T</u>ools <u>W</u>indow <u>H</u>elp

🔊 🕓 http://ww	w.modbee.com/life/wheels/story/8015603p-8880060c.html	D 🔍
S Members S Wel	Mail 🛇 Connections 🛇 Riz-Inumal 🛇 SmartUpdate 🛇 Mktplace 🛇 Grad	luate Semi
modbee c	om 🧐 🧐	Turlock - Stockton - Tulare - New - Used - Rentals
The Modesto Bee,	Modesto CA Home Delivery News Classified Cars	Jobs Homes Apartments Sit
HOME	E-mail this story 🖳 Print this story 🕒 E-mail updates	Box Home Delivery of The Bee
ARCHIVES		
NEWS	Rejoice, parallel parking haters	
POLITICS	By YURI KAGEYAMA	
OPINION	THE ASSOCIATED PRESS	
OBITUARIES	Last Updated: January 16, 2004, 10:08:12 AM PST	
WEATHER	TO KYO Your hands don't even need to be touching the steering	
SPORTS	wheel for it to start spinning back and forth aggressively, all by itself	
LIFE Monday Life Technology Healthy Living Taste Buzzz	slowly guiding the car into the parking spot. Parallel parking is designed to be a breeze with the Intelligent Parking Assist system, part of a new \$2,200 option package for Toyota Motor Corp.'s Prius gas-electric hybrid in Japan.	400
Leisure & Style Wheels Faith & Values Friends & Family Travel Your Home Horoscope	This is a hold and somewhat unnerving concept, a car that parks itself. As a driver, poive got to wonder as the Prints eases back toward the curb: What is this machine thinking? It's also difficult not to be gripped by a "Look ma, no hands" thrill — even if the system only partially fulfills its promise.	Look, Ma, no hands! To help drivers vexed by parallel parking, Toyota has created an Intelligent Parking Assist system that allows a vehicle to park itself. The system relies on a bult-in computer, a steering sensor and a ting camera in the vehicle' rear. The
THE ARTS	But we'll get to the drawbacks later.	\$2,200 option, however, is only available so far on Toyota's Prius gas-electric hybrid in Japan, and has
BUSINESS	First, the logic behind this innovation. First, the logic behind this innovation. If you know Japan, it should come as no surprise that about 80 percent the perfect hands free structures. A sourcitated pression of the structure of the struc	
COLUMNISTS		the perfect hands-free parking
GALLERIES		system. THE ASSOCIATED PRESS

Example

An inverted pendulum with vertically moving pivot point



$$\ddot{\phi}(t) = \frac{1}{l} \left(g + u(t) \right) \sin(\phi(t)),$$

where u(t) is acceleration, can be written as

$$\dot{x}_1 = x_2$$

 $\dot{x}_2 = \frac{1}{l}(g+u)\sin(x_1)$

Bonus — Discrete Time

Many results are parallel (observability, controllability,...)

Example: The difference equation

 $x_{k+1} = f(x_k)$

is asymptotically stable at x^* if the linearization

 $\left. \frac{\partial I}{\partial x} \right|_{x^*}$ has all eigenvalues in $|\lambda| < 1$

(that is, within the unit circle).

Example (cont'd): Numerical iteration

 $x_{k+1} = f(x_k)$

 $x^* = f(x^*)$

to find fixed point

When does the iteration converge?



Simulink



Choose Simulation Parameters



Don't forget "Apply"

How To Get Better Accuracy

Modify Refine, Absolute and Relative Tolerances, Integration method

Refine adds interpolation points:



Part IV: Simulation

Often the only method

- $\dot{x} = f(x)$
 - ACSLSimnon
 - Similar
 Simulink
- $F(\dot{x}, x) = 0$

► Omsim

- http://www.control.lth.se/~cace/omsim.html
- Dymola http://www.dynasim.se/
 Modelica
 - http://www.dynasim.se/Modelica/index.html

Special purpose

- Spice (electronics)
- EMTP (electromagnetic transients)
- Adams (mechanical systems)

Simulink, An Example

File -> New -> Model
Double click on Continuous
Transfer Fcn
Step (in Sources)
Scope (in Sinks)
Connect (mouse-left)
Simulation->Parameters



Save Results to Workspace



Check "Save format" of output blocks ("Array" instead of "Structure") >> plot(t,y)

(or use "Structure" which also contains the time information.)

Points were correct, only the plot was bad



Use Scripts to Document Simulations

Submodels, Example: Water tanks

Equation for one water tank:

If the block-diagram is saved to stepmodel.mdl, the following Script-file simstepmodel.m simulates the system:

open_system('stepmodel') set_param('stepmodel','RelTol','1e-3') set_param('stepmodel','AbsTol','1e-6') set_param('stepmodel','Refine','1') tic sim('stepmodel',6) toc subplot(2,1,1),plot(t,y),title('y') subplot(2,1,2),plot(t,u),title('u')

Linearization in Simulink

Use the command trim to find e.g., stationary points to a system >> A=2.7e-3;a=7e-6,g=9.8;

>> % Example to find input u for desired states/output >> [x0,u0,y0]=trim('flow',[0.1 0.1]',[],0.1) x0 = 0.1000 0.1000 u0 = 8.3996e-06 y0 =

0.1000

Linearization in Simulink; Alternative

By right-clicking on a signal connector in a Simulink model you can add "Linearization points" (inputs and/or outputs).



Start a "Control and Estimation Tool Manager" to get a linearized model by Tools -> Control Design ->Linear analysis ...

where you can set the operating points, export linearized model to Workspace (Model-> Export to Workspace) and much more.

Today

- Linearization, both around equilibria and trajectories,
- Definitions of local and global stability,
- How to check local stability and local controllability at equilibria
- Simulation tool: Simulink,

Next Lecture

- Phase plane analysis
- Classification of equilibria

$\dot{h} = (u-q)/A$

 $q = a\sqrt{2g\sqrt{h}}$

Corresponding Simulink model:



Make a subsystem and connect two water tanks in series.



Linearization in Simulink, cont.

Use the command linmod to find a linear approximation of the system around an operating point:

>> [aa,bb,cc,dd]=linmod('flow',x0,u0);

>> sys=ss(aa,bb,cc,dd);

>> bode(sys)

Computer exercise

Simulation of JAS 39 Gripen



- Simulation
- Analysis of PIO using describing functions
- Improve design