Lecture 2

- Linearization
- Stability definitions
- Stability and controllability from linearization
- Simulation in Matlab/Simulink

Material

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

Today's Goal

To be able to

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

Linearization Around a Trajectory

Idea: Make Taylor-expansion around a known solution $\{x^*(t), u^*(t)\}$.

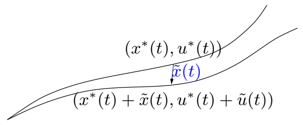
Let

$$\frac{dx^*}{dt} = f(x^*(t), u^*(t))$$

be a known solution.

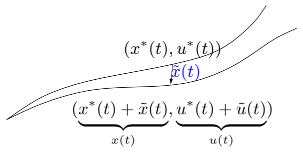
How will a small deviation $\{\tilde{x}, \tilde{u}\}$ from this solution behave?

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



Linearization Around a Trajectory

$$\begin{cases} \frac{dx}{dt} = f(x, u) \\ y = h(x, u) \end{cases} \qquad \begin{cases} \frac{d\tilde{x}}{dt} = A(t)\tilde{x} + B(t)\tilde{u} \\ \tilde{y} = C(t)\tilde{x} + D(t)\tilde{u} \end{cases}$$

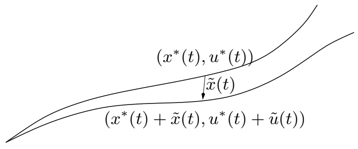


Linearization Around a Trajectory, cont.

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

$$\begin{aligned} \dot{x}(t) &= f(x^*(t) + \tilde{x}(t), u^*(t) + \tilde{u}(t)) \\ &= f(x^*(t), u^*(t)) + \frac{\partial f}{\partial x}(x^*(t), u^*(t))\tilde{x}(t) + \frac{\partial f}{\partial u}(x^*(t), u^*(t))\tilde{u}(t) + \mathcal{O}(\|\tilde{x}, \tilde{u}\|^2) \end{aligned}$$

$$\dot{\tilde{x}}(t) = \frac{\partial f}{\partial x}(x^*(t), u^*(t))\tilde{x}(t) + \frac{\partial f}{\partial u}(x^*(t), u^*(t))\tilde{u}(t) + \mathcal{O}(\|\tilde{x}, \tilde{u}\|^2)$$



State-space form

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

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

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

$$A(t) = \frac{\partial f}{\partial x}(x^*(t), u^*(t)) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} (x^*(t), u^*(t))$$

$$B(t) = \frac{\partial f}{\partial u}(x^*(t), u^*(t)) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \\ \frac{\partial f_2}{\partial u_1} \end{bmatrix} (x^*(t), u^*(t))$$

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

Linearization, cont'd

The linearization of the output equation

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

around the nominal output $y^*(t) = h(x^*(t), u^*(t))$ is given by

$$\tilde{y}(t) = C(t)\tilde{x}(t) + D(t)\tilde{u}(t)$$

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

$$\begin{aligned} C(t) &= \frac{\partial h}{\partial x}(x^*(t), u^*(t)) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} \end{bmatrix} (x^*(t), u^*(t)) \\ D(t) &= \frac{\partial h}{\partial u}(x^*(t), u^*(t)) = \begin{bmatrix} \frac{\partial h_1}{\partial u_1} \\ \frac{\partial h_2}{\partial u_1} \end{bmatrix} (x^*(t), u^*(t)) \end{aligned}$$

Example - Linearization around equilibrium point

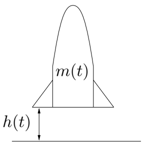
The linearization of

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

around the equilibrium $x^* = n\pi$ is given by

$$\ddot{\tilde{x}}(t) = \frac{g}{l}\sin(n\pi + \tilde{x}(t)) \approx \frac{g}{l}(-1)^n\tilde{x}(t)$$

Example: Rocket



$$\begin{aligned}\dot{h}(t) &= v(t)\\ \dot{v}(t) &= -g + \tfrac{v_e u(t)}{m(t)}\\ \dot{m}(t) &= -u(t)\end{aligned}$$

Let $u^*(t) \equiv u^* > 0$; $x^*(t) = \begin{bmatrix} h^*(t) \\ v^*(t) \\ m^*(t) \end{bmatrix}$; $m^*(t) = m^* - u^* t$.

Linearization: $\dot{\tilde{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{-v_e u^*}{m^*(t)^2} \\ 0 & 0 & 0 \end{bmatrix} \tilde{x}(t) + \begin{bmatrix} 0 \\ \frac{v_e}{m^*(t)} \\ -1 \end{bmatrix} \tilde{u}(t)$

Outline

- Linearization
- **Stability definitions**
- Stability and controllability from linearization
- Simulation in Matlab/Simulink

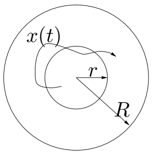
Local Stability

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

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

$$\|x(0) - x^*\| < r \quad \Longrightarrow \quad \|x(t) - x^*\| < R, \quad \text{for all } t \geq 0$$

Otherwise the equilibrium point x^* is **unstable**.



Asymptotic Stability

Definition The equilibrium x^* is **locally asymptotically stable (LAS)** if it

1) is stable
2) there exists $r > 0$ so that if $\|x(0) - x^*\| < r$ then

$$x(t) \longrightarrow x^* \quad \text{as} \quad t \longrightarrow \infty.$$

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

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

$$x(t) \to x^* \text{ as } t \to \infty.$$

- Linearization
- Stability definitions
- **Stability and controllability from linearization**
- Simulation in Matlab/Simulink

Lyapunov's Linearization Method

Theorem Assume

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

has the linearization

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

around the equilibrium point x^* and put

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

- If $\alpha(A) < 0$, then $\dot{x} = f(x)$ is LAS at x^*,
- If $\alpha(A) > 0$, then $\dot{x} = f(x)$ is unstable at x^*,
- If $\alpha(A) = 0$, then no conclusion can be drawn.

(Proof in Lecture 4)

Example

The linearization of

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

at $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ 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^*.

Local Controllability

Theorem Assume
$$\dot{x} = f(x, u)$$
has the linearization
$$\frac{d\tilde{x}}{dt} = A\tilde{x} + B\tilde{u}$$
around the equilibrium (x^*, u^*) then the nonlinear system is *locally controllable* provided that (A, B) controllable.

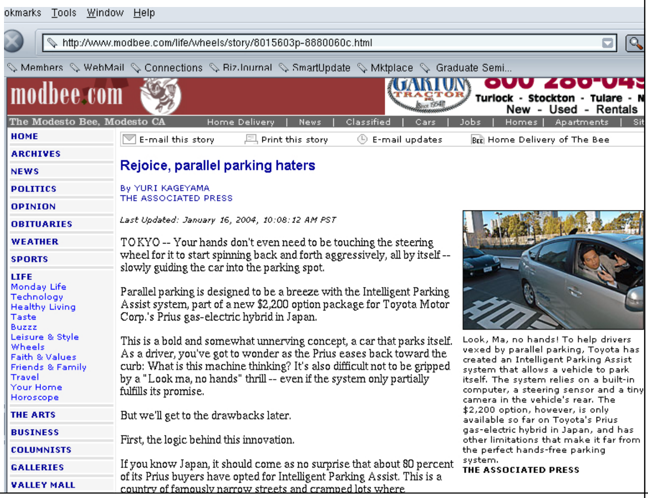
Here local controllability is defined as follows:

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

5 minute exercise:

Is the ball and beam
$$\ddot{x} = x\dot{\phi}^2 + g\sin\phi + \frac{2r}{5}\ddot{\phi}$$
nonlinearly locally controllable around
$\dot{\phi} = \phi = x = \dot{x} = 0$ (with $\ddot{\phi}$ as input)?

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



Rejoice, parallel parking haters

By YURI KAGEYAMA
THE ASSOCIATED PRESS

Last Updated: January 16, 2004, 10:08:12 AM PST

TOKYO -- Your hands don't even need to be touching the steering wheel for it to start spinning back and forth aggressively, all by itself -- 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.

This is a bold and somewhat unnerving concept, a car that parks itself. As a driver, you've got to wonder as the Prius 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.

But we'll get to the drawbacks later.

First, the logic behind this innovation.

If you know Japan, it should come as no surprise that about 80 percent of its Prius buyers have opted for Intelligent Parking Assist. This is a country of famously narrow streets and cramped lots where

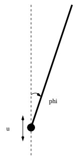
Look, Ma, no hands! To help drivers vexed by parallel parking, Toyota has created an Intelligent Parking Assist system that allows a vehicle to park itself. The system relies on a built-in computer, a steering sensor and a tiny camera in the vehicle's rear. The $2,200 option, however, is only available so far on Toyota's Prius gas-electric hybrid in Japan, and has other limitations that make it far from the perfect hands-free parking system.
THE ASSOCIATED PRESS

2016 IEEE Intelligent Vehicles Symp., Gothenburg

[Evestedt, Ljungqvist, Axehill] More parking in lecture 12.

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
$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{l}(g + u)\sin(x_1)\end{aligned}$$

Example, cont.

The linearization around $x_1 = x_2 = 0, u = 0$ is given by
$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g}{l}x_1\end{aligned}$$

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.

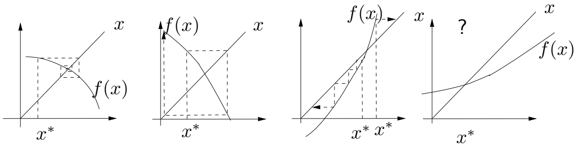
Bonus — Discrete Time

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

Example: The difference equation
$$x_{k+1} = f(x_k)$$
is asymptotically stable at x^* if the linearization
$$\left.\frac{\partial f}{\partial x}\right|_{x^*} \text{ has all eigenvalues in } |\lambda| < 1$$
(that is, within the unit circle).

Example (cont'd): Numerical iteration
$$x_{k+1} = f(x_k)$$
to find fixed point
$$x^* = f(x^*)$$

When does the iteration converge?



Outline

- Linearization
- Stability definitions
- Stability and controllability from linearization
- **Simulation in Matlab/Simulink**

Simulation

Often the only method

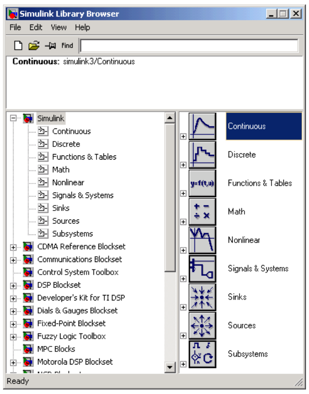
$\dot{x} = f(x)$
- ACSL
- Simnon
- Simulink

$F(\dot{x}, x) = 0$
- Omsim
- Dymola
- Modelica (`www.modelica.org`)

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

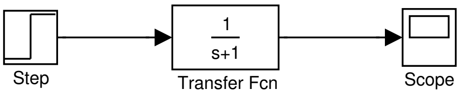
Simulink

```
> matlab
>> simulink
```

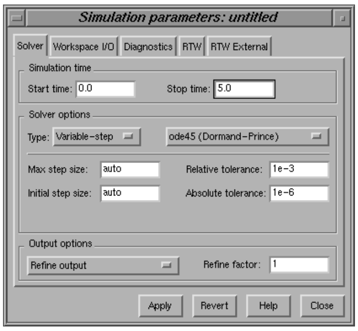


Simulink, An Example

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



Choose Simulation Parameters



Simulation time

Start time: 0.0

Stop time: 5.0

Solver options

Type: Variable−step

ode45 (Dormand−Prince)

Max step size: auto

Relative tolerance: 1e−3

Initial step size: auto

Absolute tolerance: 1e−6

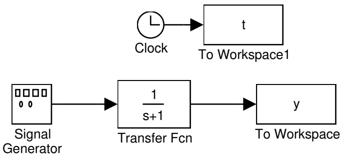
Output options

Refine output

Refine factor: 1

Don't forget "Apply"

Save Results to Workspace



Check "Save format" of output blocks ("Array" instead of "Structure")

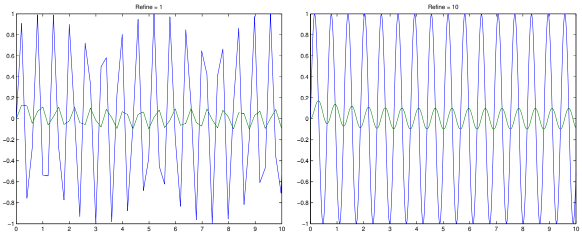
```
>> plot(t,y)
```

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

How To Get Better Accuracy

Modify Refine, Absolute and Relative Tolerances, Integration method

Refine adds interpolation points:



Use Scripts to Document Simulations

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

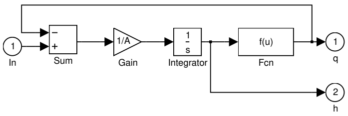
```
open_system('stepmodel')
set_param('stepmodel','RelTol','1e-3')
set_param('stepmodel','AbsTol','1e-6')
set_param('stepmodel','Refine','1')
tic
sim('stepmodel',6)
toc
subplot(2,1,1),plot(t,y),title('y')
subplot(2,1,2),plot(t,u),title('u')
```

Submodels, Example: Water tanks

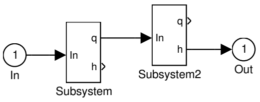
Equation for one water tank:

$$\begin{aligned} \dot{h} &= (u-q)/A \\ q &= a\sqrt{2g}\sqrt{h} \end{aligned}$$

Corresponding Simulink model:



Make a subsystem and connect two water tanks in series.



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
```

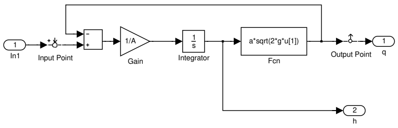
Linearization in Simulink, cont.

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)
```

Linearization in Simulink; Alternative

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

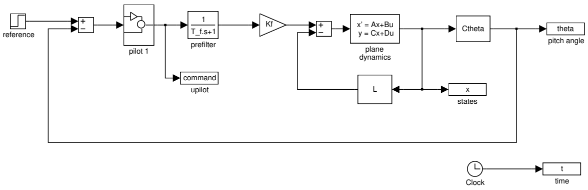


Start a “Control and Estimation Tool Manager” to get a linearized model by
`Tools -> Control Design ->Linear analysis ...`

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

Computer exercise

Simulation of JAS 39 Gripen



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

Summary

- Linearization, both around equilibria and trajectories
- Definitions of local and global stability
- Check local stability and local controllability at equilibria
- Simulation tool in this course: Simulink

Next: Lecture 3

- Phase plane analysis
- Classification of equilibria