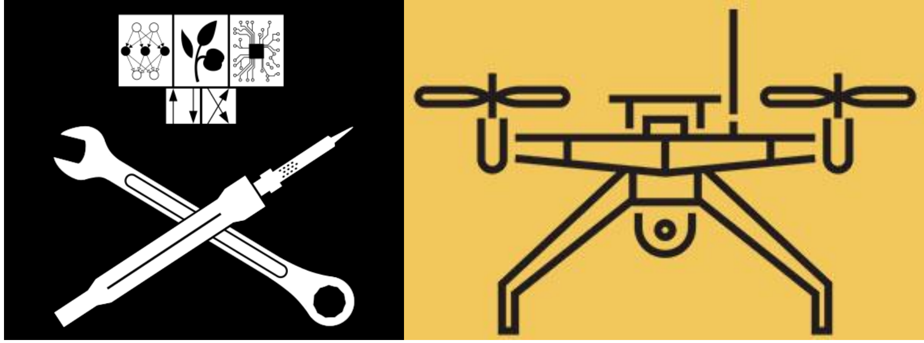


Stability and state-feedback control

Lecture 4

SDU

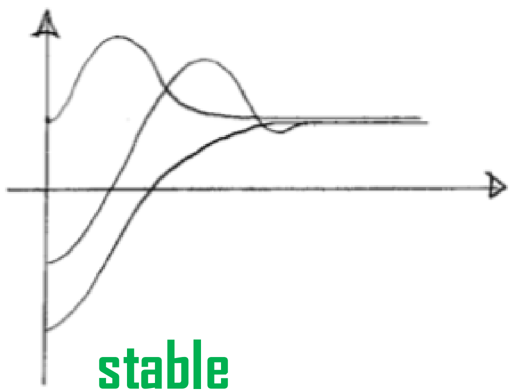
Jerome Jouffroy, Professor
jerome@sdu.dk

Today's lecture

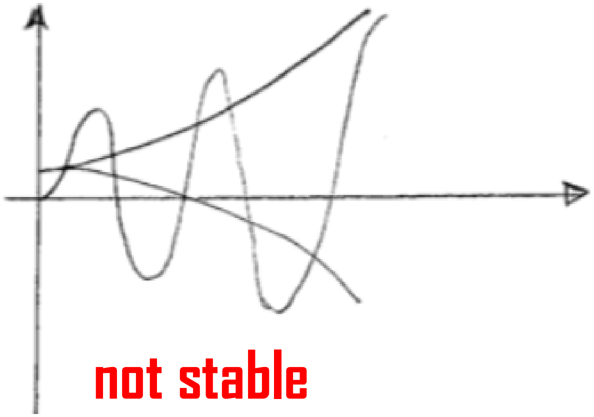
- Stability in the state-space context
- Linear state-feedback control
- The Linear Quadratic Regulator
- Stabilization of linear systems around any equilibrium point

Stability for SS. rep in a nutshell:

Question: what is stability in the context of state-space representations?



stable



not stable

final behavior independent from initial conditions

Trivial first-order example:

$$\dot{x}(t) = -x(t), \qquad x(0) = x_0$$

whose solution is $x(t) = x_0 e^{-t}$

so that we have $\lim_{t\to\infty} x(t) = \lim_{t\to\infty} x_0 e^{-t} = 0$ system is stable

(final behavior is independent from initial conditions)

Stability and state-feedback on the trivial example

start again with a scalar example

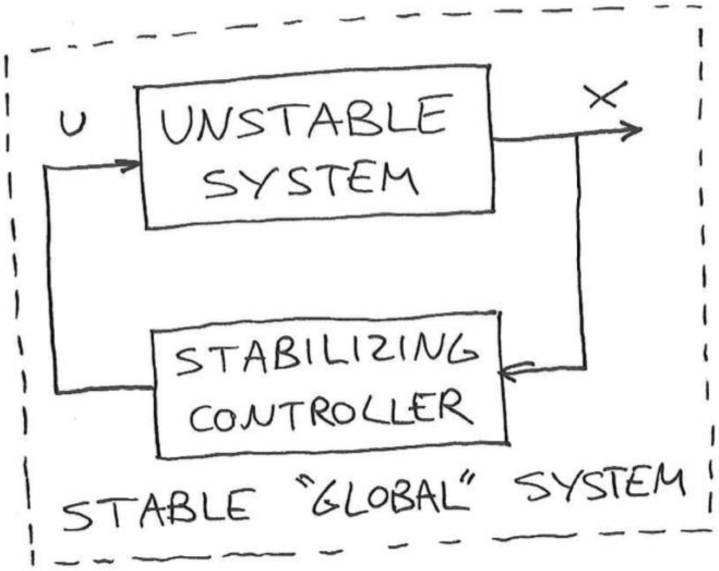
but with an input: $\dot{x}(t) = x(t) + u(t), \qquad x(0) = x_0$

assume first that $u(t) = 0$

so that we have $x(t) = x_0 e^t$

then $\begin{cases} \text{if} \quad x_0 > 0 \Rightarrow \lim_{t\to\infty} x(t) = +\infty \\ \text{if} \quad x_0 = 0 \Rightarrow \lim_{t\to\infty} x(t) = 0 \\ \text{if} \quad x_0 < 0 \Rightarrow \lim_{t\to\infty} x(t) = -\infty \end{cases}$

➡ system with u=0 is NOT stable



introduce the <u>state-feedback control law</u>

$$u(t) = -2x(t)$$

which gives $\dot{x}(t) = x(t) - 2x(t) = -x(t)$ (stable!)

➡ the unstable system was stabilized by state-feedback

Stability of linear systems: from TFs to state-space (1/2)

Remember Transfer Functions?

$$\frac{y(s)}{u(s)} = \frac{b_m s^m + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0}$$

The above system is stable if its poles are in the left half-plane

What about state-space representations? Idea: use the result available for TFs

start with $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0$

and get the Laplace transform $s.\mathbf{x}(s) - \mathbf{x}_0 = \mathbf{A}\mathbf{x}(s)$ (with initial conditions)

or $(s\mathbf{I} - \mathbf{A})\mathbf{x}(s) = \mathbf{x}_0$

so that we have $\mathbf{x}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}_0$

which can be rewritten as $\boxed{\mathbf{x}(s) = \frac{\mathrm{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})}\mathbf{x}_0}$

(TF between x_0 and $x(s)$)

Stability of linear systems: from TFs to state-space (2/2)

Hence $\mathbf{x}(s) = \dfrac{\mathrm{adj}(s\mathbf{I}-\mathbf{A})}{\det(s\mathbf{I}-\mathbf{A})}\mathbf{x}_0$ represents a TF, with $\mathbf{x}_0$ an impulse (input)

this TF is stable if the roots of its den. are in the left half-plane,

ie if the solutions λ of the equation $\det(\lambda\mathbf{I}-\mathbf{A}) = 0$ are in the left half-plane

solutions of this eq. are call eigenvalues of matrix A

Main result: The system represented by $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t)$ is stable if and only if the real part of each eigenvalue of $\mathbf{A}$ is strictly negative. □

Small examples:

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}\mathbf{x} \Rightarrow \det(\lambda\mathbf{I}-\mathbf{A}) = (\lambda+1)^2 = 0 \xrightarrow{\text{eigenvalue}} \lambda = -1$$

(system stable!)

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}\mathbf{x} \xrightarrow{\text{eigenvalues}} \begin{vmatrix} \lambda_1 = -1 + i \\ \lambda_2 = -1 - i. \end{vmatrix}$$

(system stable!)

SDU

Linear state-feedback: the general case

start with the system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \qquad (1)$$

and introduce the control law

$$\boxed{\mathbf{u}(t) = -\mathbf{K}.\mathbf{x}(t)} \qquad (2)$$

with $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{K} \in \mathbb{R}^{m \times n}$

(matrix gain)

putting (2) into (1) gives

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}(-\mathbf{K}\mathbf{x}(t))$$

or $$\boxed{\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(t)} \qquad (3)$$ closed-loop dynamics

if the eigenvalues of A-BK are in the left half-plane, then the closed-loop dynamics are stable and x(t) converges to 0

SDU

State-feedback on a multi-input example

consider the example

$$\dot{\mathbf{x}} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & -1 & 2 \\ 0 & 2 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} \mathbf{u}$$

$\dim(\mathbf{x}) = 3$ and $\dim(\mathbf{u}) = 2$ $\mathbf{K} \in \mathbb{R}^{2\times 3}$

exercise: check that this system is not stable when $\mathbf{u} = 0$.

state-feedback law?

rewrite the system in component form

$$\begin{cases} \dot{x}_1 = \underline{2x_1} + u_1 \\ \dot{x}_2 = x_1 - x_2 + \underline{2x_3} + 2u_2 \\ \dot{x}_3 = 2x_2 - x_3 \end{cases}$$

it would be great if we could have a cascade of 3 stable systems

let $\begin{vmatrix} u_1 = -3x_1 \\ u_2 = -x_3 \end{vmatrix}$

(closed-loop dynamics)

so that we have

$$\begin{cases} \dot{x}_1 = -x_1 \\ \dot{x}_2 = x_1 - x_2 \\ \dot{x}_3 = 2x_2 - x_3 \end{cases} \qquad \dot{\mathbf{x}} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 2 & -1 \end{bmatrix} \mathbf{x}$$

exercise: check that this system is stable when $\mathbf{K} = \begin{bmatrix} -3 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

$\mathbf{K} \in \mathbb{R}^{2\times 3}$

SDU

Towards a more "systematic" way to tune K

The previous example was useful but not very systematic:

we had to guess the values of K

<u>Feedback control objective</u>:

we would like the poles/eigenvalues of the system in closed-loop to be $\boldsymbol{\lambda}_{cl}$

hence we need to find the feedback gain K such that

$$\boxed{\boldsymbol{\lambda}_{cl} = \mathrm{eig}(\mathbf{A} - \mathbf{BK})}$$

(matrix algebraic equation)

many different algorithms/techniques exist so that, given matrices A, B and desired eigenvalues, K is calculated...

examples (also in Matlab): `K = acker(A,B,lambda_cl)`

(the "Ackerman" method)

`place` (more stable numerically than "Ackerman")

The Linear Quadratic Regulator

Optimal control: find signal $\mathbf{u}(t)$ on $[0, T]$

such that $J = \int_0^T \left[\mathbf{x}^T(\tau)\mathbf{Q}\mathbf{x}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}\mathbf{u}(\tau)\right] d\tau$ is minimized

with $\mathbf{Q}, \mathbf{R}$ tuning matrices

Q, R used to balance the relative importance between:

- distance between x and 0: $\mathbf{x}^T\mathbf{Q}\mathbf{x}$, (generalization of $q||\mathbf{x}||^2 = q\mathbf{x}^T\mathbf{x}$)
- minimize energy consumption: $\mathbf{u}^T\mathbf{R}\mathbf{u}$

when $T \to \infty$ ie when $\boxed{J = \int_0^\infty \left[\mathbf{x}^T(\tau)\mathbf{Q}\mathbf{x}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}\mathbf{u}(\tau)\right] d\tau}$

control signal u(t) can be expressed as $\boxed{\mathbf{u} = -\mathbf{K}\mathbf{x}}$

where K is given $\boxed{\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}}$

Linear Quadratic Regulator

where P is the solution of $\boxed{\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0}$

(matrix Riccati equation, solved on a computer)

SDU

Tuning the LQR: the Bryson's rule

Question: How to choose/tune matrices Q and R?

Bryson's "rule": choose Q and R as diagonal matrices with entries

$$\boxed{Q_{ii} = 1/\text{maximum acceptable value of } x_i^2}$$

and

$$\boxed{R_{ii} = 1/\text{maximum acceptable value of } u_i^2}$$

Remark: only a guideline, improvement by further tweaking values afterwards (like ZN for PID)

LQR: Stability result

> Closed-loop system is stable as long as Q and R are both strictly positive definite

SDU

LQR: the discrete-time case

For discrete-time systems
described by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

the cost function at T= ∞ is written as

$$\boxed{J = \sum_{0}^{\infty} \left[\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k))\right]}$$

...and the rest is the same!

(ie K calculation, matrix Riccati equation)

Stabilization of a linear system around an equilibrium point (1/3)

<u>Equilibrium points and linear state-space representations</u>:

consider the linear ss rep.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

which is controlled by state-feedback:

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

by applying this control law, we also have that $\mathbf{x}^* = 0$.

indeed: $0 = \mathbf{A}'\mathbf{x}^* = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}^*$

What about stabilizing around something different than 0?

For linear systems represented by $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$

equilibrium points have to satisfy $\boxed{0 = \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^*}$

SDU

Stabilization of a linear system around an equilibrium point (2/3)

<u>Calculating the error dynamics</u>:

define
$$\left|\begin{array}{l}\Delta\mathbf{x}(t) := \mathbf{x}(t) - \mathbf{x}^* \\ \Delta\mathbf{u}(t) := \mathbf{u}(t) - \mathbf{u}^*\end{array}\right.$$

control objective: we want x(t) to go to x*.

Hence we want $\Delta\mathbf{x}(t)$ to go to 0

dynamics of $\Delta x(t)$?

$$\Rightarrow \quad \frac{d}{dt}(\Delta\mathbf{x}(t)) = \frac{d}{dt}(\mathbf{x}(t) - \mathbf{x}^*)$$

and recall that we also have $\mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^* = 0$

which gives $\frac{d}{dt}(\Delta\mathbf{x}(t)) = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u} - \mathbf{B}\mathbf{u}^*$

finally $\boxed{\frac{d}{dt}(\Delta\mathbf{x}(t)) = \mathbf{A}\Delta\mathbf{x}(t) + \mathbf{B}\Delta\mathbf{u}(t)}$ error dynamics

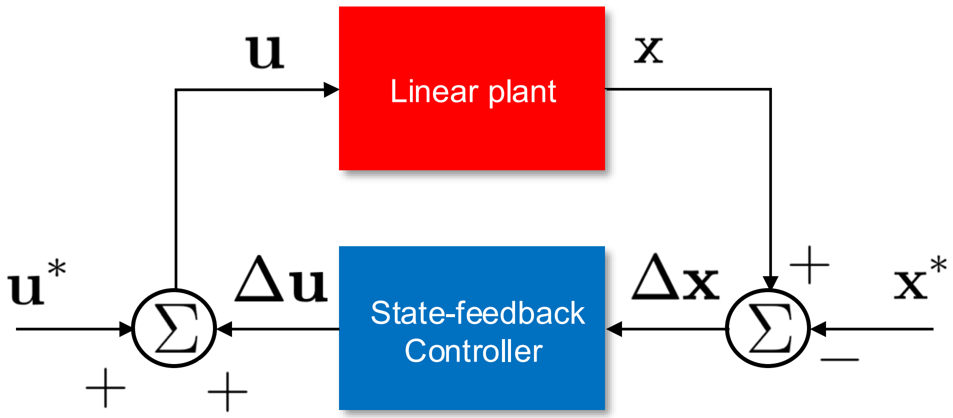
Stabilization of a linear system around an equilibrium point (3/3)

Stabilize $\frac{d}{dt}(\Delta\mathbf{x}(t)) = \mathbf{A}\Delta\mathbf{x}(t) + \mathbf{B}\Delta\mathbf{u}(t)$ around 0?

use and tune the controller $\Delta\mathbf{u}(t) = -\mathbf{K}.\Delta\mathbf{x}(t)$ (1)

For implementation, rewrite (1) as $\boxed{\mathbf{u}(t) = -\mathbf{K}(\mathbf{x}(t) - \mathbf{x}^*) + \mathbf{u}^*}$

or $\mathbf{u} = \underbrace{-\mathbf{K}\mathbf{x}}_{\text{feedback term}} + \underbrace{\mathbf{u}^* + \mathbf{K}\mathbf{x}^*}_{\text{feedforward term}}$



Following a constant reference signal r (feedforward gain)

We would like to express the previous controller in a form where the output y follows a reference r...

(ie same as PID controller...)

we would like the controller to have the form $\boxed{\mathbf{u} = -\mathbf{K}\mathbf{x} + \bar{\mathbf{N}}\mathbf{r}}$ (1)

and that $\mathbf{y}^* = \mathbf{r}$

write the equilibrium equation $\begin{cases} 0 = \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^* \\ \mathbf{y}^* = \mathbf{C}\mathbf{x}^* \end{cases}$ with the objective $\mathbf{y}^* = \mathbf{r}$

rewrite both in matrix form together $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{r} \end{bmatrix}$

invert $\begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{r} \end{bmatrix}$ and define $\mathbf{N} := \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_x \\ \mathbf{N}_{21} & \mathbf{N}_u \end{bmatrix}$

then $\begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_x \\ \mathbf{N}_{21} & \mathbf{N}_u \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_x\mathbf{r} \\ \mathbf{N}_u\mathbf{r} \end{bmatrix}$

so that $\mathbf{u} = -\mathbf{K}\mathbf{x} + \mathbf{u}^* + \mathbf{K}\mathbf{x}^*$ becomes $\mathbf{u} = -\mathbf{K}\mathbf{x} + \mathbf{N}_u\mathbf{r} + \mathbf{K}\mathbf{N}_x\mathbf{r}$

and we have (1) with the feedforward gain $\boxed{\bar{\mathbf{N}} := \mathbf{N}_u + \mathbf{K}\mathbf{N}_x}$