Systems Laboratory, Spring 2025

Damiano Varagnolo – CC-BY-4.0

notes

- welcome to the course!
- on this side of this document you will find notes that accompany the text typically visualized in class
- these notes are meant to convey the messages that are not displayed in the text on the side, and basically constitute what the teacher intends to say in class

Table of Contents I

notes

- this is the table of contents of this document; each section corresponds to a specific part of the course how to linearize an ODE

notes

Contents map

developed content units	**taxonomy levels**
linearization	u1, e1

prerequisite content units	**taxonomy levels**
ODE	u1, e1

notes

Main ILO of sub-module "how to linearize an ODE"

Linearize a nonlinear ODE around an equilibrium point

notes

- by the end of this module you shall be able to do this

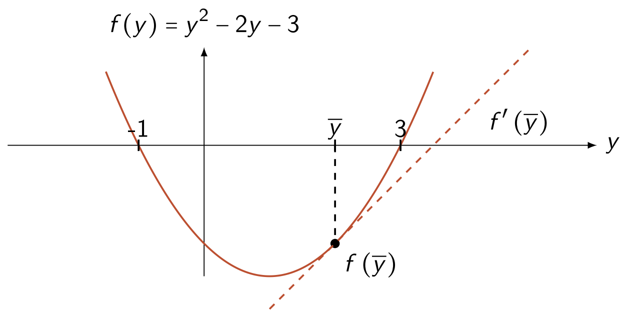
The path towards linearizing a model

- what does linearizing a function mean?
- what does linearizing a model mean?
- how shall we linearize a model?

notes

- so now we will enter into the new content from this module
- we will proceed following this mental order in answering questions

What does linearizing a scalar function mean?



$$f(y) \approx f(\overline{y}) + \left.\frac{\partial f}{\partial y}\right|_{\overline{y}} (y - \overline{y})$$

(but the approximation is valid only close to the linearization point)

notes

- graphically, it means substituting the function with an approximation that is a valid one in the neighborhood of a specific point
- in formulas we get this, that should be well known
- try to do by yourself the example of linearizing this function around $\overline{y} = 1$. You should get $f(y) = -4$ (constant)

Obvious requirement

(but sometimes people forget about it . . .)

to compute the approximation

$$f(y) \approx f(\overline{y}) + \left.\frac{\partial f}{\partial y}\right|_{\overline{y}} (y - \overline{y})$$

the derivative of f at $\overline{y}$ must be defined. *(notation: $f \in C^n$ means that f has all its derivatives up to order n defined in $\mathbb{R}$. $f \in C^n(\mathcal{X})$ means defined in $\mathcal{X} \subseteq \mathbb{R}$)*

notes

- in other words, you cannot linearize a function on a discontinuity point
- note: if you did not see this notation keep it in mind

What does linearizing a vectorial function mean?

$$\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^m, \quad \boldsymbol{f} \in C^1 \quad \text{enables computing} \quad \boldsymbol{f}(\boldsymbol{y}) \approx \boldsymbol{f}(\boldsymbol{y}_0) + \nabla_{\boldsymbol{y}} \boldsymbol{f}\big|_{\boldsymbol{y}_0} (\boldsymbol{y} - \boldsymbol{y}_0)$$

linearize $\Longrightarrow$ *approximate each component!*

Discussion: then $\nabla_{\boldsymbol{y}} \boldsymbol{f}\big|_{\boldsymbol{y}_0}$ must be a matrix. Of which dimensions?

notes

- if we have a vectorial function we can think that this means linearizing each component
- $\boldsymbol{f}(\boldsymbol{y}_0)$ transforms a n-dimensional vector to a m-dimensional one. $(\boldsymbol{y} - \boldsymbol{y}_0)$ is a n-dimensional vector, thus that matrix must be m rows and n columns

Example: linearize $\boldsymbol{f}$ around $\boldsymbol{y}_0$

$$\boldsymbol{f}(\boldsymbol{y}(t)) = \begin{bmatrix} \sin\big(y_1(t)\big) + \cos\big(y_2(t)\big) \\ \exp\big(y_1(t) y_2(t)\big) \end{bmatrix} \qquad \boldsymbol{y}_0 = \boldsymbol{y}(0) = [0, \pi]$$

notes

- if you do not feel able of doing this linearization then you should definitely refresh how to do derivatives
- The linearization involves computing the Jacobian matrix of $\boldsymbol{f}$ at $\boldsymbol{y}_0$. The Jacobian matrix is given by:

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial}{\partial y_1}(\sin(y_1) + \cos(y_2)) & \frac{\partial}{\partial y_2}(\sin(y_1) + \cos(y_2)) \\ \frac{\partial}{\partial y_1}\exp(y_1 y_2) & \frac{\partial}{\partial y_2}\exp(y_1 y_2) \end{bmatrix}.$$

Evaluating the Jacobian at $\boldsymbol{y}_0 = [0, \pi]$ gives

$$\boldsymbol{J}(\boldsymbol{y}_0) = \begin{bmatrix} \cos(0) & -\sin(\pi) \\ \pi \cdot \exp(0) & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \pi & 0 \end{bmatrix}.$$

Thus, the linearized system around $\boldsymbol{y}_0$ is

$$\boldsymbol{J}(\boldsymbol{y}_0)\Delta\boldsymbol{y}$$

And what if the vectorial function depends on more than one variable?

Assuming $\boldsymbol{f}$ differentiable in $\boldsymbol{y}_0, \boldsymbol{u}_0$,

$$\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) \approx \boldsymbol{f}(\boldsymbol{y}_0, \boldsymbol{u}_0) + \nabla_{\boldsymbol{y}} \boldsymbol{f}\big|_{\boldsymbol{y}_0, \boldsymbol{u}_0} (\boldsymbol{y} - \boldsymbol{y}_0) + \nabla_{\boldsymbol{u}} \boldsymbol{f}\big|_{\boldsymbol{y}_0, \boldsymbol{u}_0} (\boldsymbol{u} - \boldsymbol{u}_0)$$

with both $\nabla_{\boldsymbol{y}} \boldsymbol{f}\big|_{\boldsymbol{y}_0, \boldsymbol{u}_0}$ and $\nabla_{\boldsymbol{u}} \boldsymbol{f}\big|_{\boldsymbol{y}_0, \boldsymbol{u}_0}$ matrices of opportune size. Alternative notation:

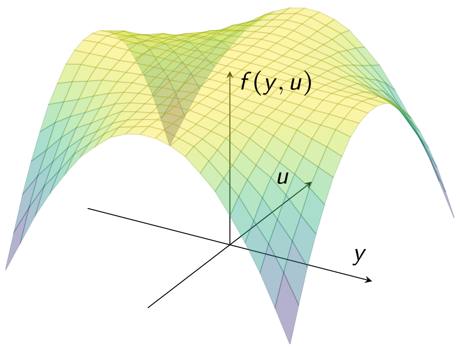
$$\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) \approx \boldsymbol{f}(\boldsymbol{y}_0, \boldsymbol{u}_0) + \nabla \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u})\Big|_{\boldsymbol{y}_0, \boldsymbol{u}_0} \begin{bmatrix} \Delta \boldsymbol{y} \\ \Delta \boldsymbol{u} \end{bmatrix}$$

notes

- if we have more variables the same rules apply, just that we have more elements in the stack
- to save space it is convenient to write things in this way. This means defining

Graphical example with a $\mathbb{R}^2 \mapsto \mathbb{R}$ function

$$f(y, u) \approx f(y_0, u_0) + \nabla f(y, u)\Big|_{y_0, u_0} \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix}$$



if $\boldsymbol{f} = [f_1, f_2]$ then have two distinct plots, but the concept is the same

notes

- see the plotting in the video of the course
- and if you got the point then this last sentence is obvious. If it does not seem obvious then this is a sign that actually better for you if you re-go through the material

Thus, linearization = stopping the Taylor series at order one

$$f \in C^M(\mathbb{R}) \qquad \Longrightarrow \qquad f(y) \approx \sum_{m=0}^{M} \frac{f^{(m)}(y_0)}{m!}(y - y_0)^m$$

multivariable extension = less neat formulas, but the concept is the same. The most

important case for our purposes:

$$\boldsymbol{f} \in C^1(\mathbb{R}^n, \mathbb{R}^m) \qquad \Longrightarrow \qquad \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) \approx \boldsymbol{f}(\boldsymbol{y}_0, \boldsymbol{u}_0) + \nabla_{\boldsymbol{y}} \boldsymbol{f}\big|_{\boldsymbol{y}_0}(\boldsymbol{y} - \boldsymbol{y}_0) + \nabla_{\boldsymbol{u}} \boldsymbol{f}\big|_{\boldsymbol{u}_0}(\boldsymbol{u} - \boldsymbol{u}_0)$$

notes

- actually what we are doing is using a Taylor expansion, and stopping at order one
- and what we eventually have is this thing

What does linearizing an ODE mean?

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) \qquad \approx \qquad \dot{\widetilde{\boldsymbol{y}}} = A\widetilde{\boldsymbol{y}} + B\widetilde{\boldsymbol{u}}$$

linearize $\Longrightarrow$ *approximate the dynamics!*

notes

- eventually if we think that a state-space model is a couple of functions, linearizing it means linearizing these functions
- as for the solution to the question, the fact is that $\boldsymbol{u}$ is the input, and in a simulation we may think that this variable is fixed. However if $\boldsymbol{u}$ is computed in closed loop, then the model itself will affect this variable, if the controller is model based. In this case we should write $\widetilde{\boldsymbol{u}}$ in the approximated model

Discussion: what is the simplest way to make this linear?

$$\dot{y} = ay + bu^{2/3}$$

Another discussion: can we apply the same "linearization trick" to $\dot{y} = a\sqrt{y} + bu$?

notes

- note that sometimes one may do some simple tricks to do a linearization that is actuall a non-linearization
- for example here we may say that $\overline{u}(t) = u(t)^{2/3}$, and get a linear model in $\overline{u}$
- this basically would say just putting a nonlinear function in the block scheme, and with this "renaming" we may use linear control theory for this nonlinear system
- this trick though does not work always. It works only if we have the autonomous version of the system that is linear, and then the inputs appear as a sum of independent inputs each with its own nonlinearity

Discussion: why do we linearize nonlinear systems?

notes

- remember: we linearize a model because it may be more meaningful to do linear control than nonlinear control
- moreover in any case the linear approximation may be a good description, if the curvature of $\boldsymbol{f}$ is not too big and if we consider a sufficiently small neighborhood of the linearization point

Discussion: where do we linearize nonlinear systems?

notes

- it has no sense to linearize in a point that is not an equilibrium, because we would then consider a system for which the trajectories by default go away from the neighborhood where the approximation is meaningful
- moreover we want to do controllers typically around plant operation points, and operation points tend to be equilibria
- formally thus one may linearize everywhere (assuming that the maps are differentiable in that points) but in practice one does linearizations only around equilibria

Linearization procedure - continuous time systems

$$\left(\boldsymbol{y}_{\mathrm{eq}}, \boldsymbol{u}_{\mathrm{eq}}\right) \text{ equilibrium } \implies \boldsymbol{f}\left(\boldsymbol{y}_{\mathrm{eq}}, \boldsymbol{u}_{\mathrm{eq}}\right) = 0$$

Procedure (assuming that the Taylor expansion exists):

- consider $\boldsymbol{y} = \boldsymbol{y}_{\mathrm{eq}} + \Delta \boldsymbol{y}$, and $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{eq}} + \Delta \boldsymbol{u}$
- compute

$$\boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}\right) \approx \boldsymbol{f}\left(\boldsymbol{y}_0, \boldsymbol{u}_0\right) + \nabla_{\boldsymbol{y}} \boldsymbol{f}\big|_{\boldsymbol{y}_0} \left(\boldsymbol{y} - \boldsymbol{y}_0\right) + \nabla_{\boldsymbol{u}} \boldsymbol{f}\big|_{\boldsymbol{u}_0} \left(\boldsymbol{u} - \boldsymbol{u}_0\right)$$

setting though $\boldsymbol{y}_0 = \boldsymbol{y}_{\mathrm{eq}}$

$$\implies \qquad \frac{\partial\left(\boldsymbol{y}_{\mathrm{eq}} + \Delta \boldsymbol{y}\right)}{\partial t} \approx \boldsymbol{f}\left(\boldsymbol{y}_{\mathrm{eq}}, \boldsymbol{u}_{\mathrm{eq}}\right) + \nabla \boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}\right)\Big|_{\boldsymbol{y}_{\mathrm{eq}}, \boldsymbol{u}_{\mathrm{eq}}} \begin{bmatrix} \Delta \boldsymbol{y} \\ \Delta \boldsymbol{u} \end{bmatrix}$$

note then that $\dfrac{\partial\left(\boldsymbol{y}_{\mathrm{eq}} + \Delta \boldsymbol{y}\right)}{\partial t} = \Delta \dot{\boldsymbol{y}}$ and that $\boldsymbol{f}\left(\boldsymbol{y}_{\mathrm{eq}}, \boldsymbol{u}_{\mathrm{eq}}\right) = \boldsymbol{0}$

notes

- to see the whole procedure, let's start considering that by definition this happens
- then one may compute this
- since we are working around the equilibrium, this happens
- but then given that $\boldsymbol{y}_{\mathrm{eq}}$ is constant and given the fact that we are on an equilibrium, this follows

Linearization procedure - continuous time systems

$$(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}) \text{ equilibrium } \implies$$

$$\Delta\dot{\boldsymbol{y}} \approx \nabla_{\boldsymbol{y}} \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u})\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}} \Delta\boldsymbol{y} + \nabla_{\boldsymbol{u}} \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u})\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}} \Delta\boldsymbol{u}$$

and, since

- the two ∇'s are matrices, and
- this is an approximate dynamics,

it follows that the approximated system is

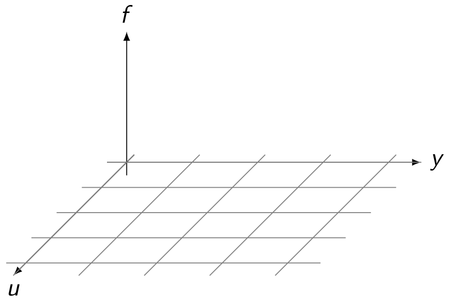
$$\Delta\dot{\widetilde{\boldsymbol{y}}} = A\Delta\widetilde{\boldsymbol{y}} + B\Delta\boldsymbol{u}$$

notes

- summarizing, we get this
- note that we are thus getting some equations that refer to an approximated system, and thus we should use different letters to indicate the different variables

What does this mean graphically?

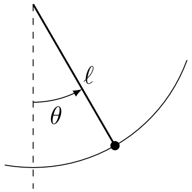
$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) \qquad \text{vs.} \qquad \Delta\dot{\widetilde{\boldsymbol{y}}} = A\Delta\widetilde{\boldsymbol{y}} + B\Delta\widetilde{\boldsymbol{u}}$$



notes

- see the drawings made during the video of the lesson

A from-start-to-end example: the pendulum



First step: equations of motion:

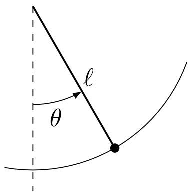
- gravity: $F_g = -mg\sin(\theta)$
- friction: $F_f = -f\ell\dot{\theta}$
- input torque: $F_u = u/\ell$

$$\text{resulting dynamics: } m\ell\ddot{\theta} = -mg\sin(\theta) - f\ell\dot{\theta} + \frac{u}{\ell}$$

notes

- let's consider a whole example, from start to end
- let's consider the pendulum example
- the system dynamics are these ones
- we can see that the system is nonlinear

First step: transforming this in a state space system



thus from

$$m\ell\ddot{\theta} = -mg\sin(\theta) - f\ell\dot{\theta} + \frac{u}{\ell}$$

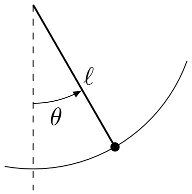
into

$$\dot{y}_1 = y_2$$
$$\dot{y}_2 = -\frac{g}{\ell}\sin(y_1) - \frac{f}{m}y_2 + \frac{1}{m\ell^2}u$$

notes

- then we can rewrite the system into its state space form
- obviously here the states will be angular position and angular velocity

Second step: finding the equilibria (assuming $u = 0$)



thus from

$$\dot{y}_1 = y_2$$
$$\dot{y}_2 = -\frac{g}{\ell}\sin(y_1) - \frac{f}{m}y_2 + \frac{1}{m\ell^2}u$$

to

$$\begin{cases} 0 = y_2 \\ 0 = -\frac{g}{\ell}\sin(y_1) - \frac{f}{m}y_2 \end{cases} \implies \boldsymbol{y}_{\text{eq.inst}} = \begin{bmatrix} \pi + 2k\pi \\ 0 \end{bmatrix}, \qquad \boldsymbol{y}_{\text{eq.st}} = \begin{bmatrix} 0 + 2k\pi \\ 0 \end{bmatrix}$$

notes

- then the equilibria follow immediately
- as intuition drives, we have several equilibria (all the upwards, and all the downwards). From practical perspectives we may just consider these two ones: $y_{\text{eq.st}} = [0, 0]^T$ and $y_{\text{eq.inst}} = \pi$

Linearizing around the first equilibrium

$$\dot{y}_1 = y_2$$
$$\dot{y}_2 = -\frac{g}{\ell}\sin(y_1) - \frac{f}{m}y_2 + \frac{1}{m\ell^2}u$$

linearizing around $\boldsymbol{y}_{\text{eq.st}} = [0, 0]^T$, $u = 0$ implies

$$A = \left.\begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix}\right|_{\boldsymbol{y}_{\text{eq}\alpha}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} & -\frac{f}{m} \end{bmatrix}$$

$$B = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix}\right|_{\boldsymbol{y}_{\text{eq}\alpha}} = \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix}$$

notes

- then doing the computations we get these equivalences

Linearizing around the second equilibrium

$$\dot{y}_1 = y_2$$
$$\dot{y}_2 = -\frac{g}{\ell}\sin(y_1) - \frac{f}{m}y_2 + \frac{1}{m\ell^2}u$$

linearizing around $\boldsymbol{y}_{\text{eq}\beta} = [\pi, 0]^T$, $u = 0$ implies

$$A = \left.\begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix}\right|_{\boldsymbol{y}_{\text{eq}\beta}} = \begin{bmatrix} 0 & 1 \\ \frac{g}{\ell} & -\frac{f}{m} \end{bmatrix}$$

$$B = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix}\right|_{\boldsymbol{y}_{\text{eq}\beta}} = \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix}$$

notes

- and for the second case we get this

The two linearized systems

Around the stable equilibrium: $\begin{bmatrix} \Delta\dot{y}_1 \\ \Delta\dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} & -\frac{f}{m} \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix} u$

Around the unstable equilibrium: $\begin{bmatrix} \Delta\dot{y}_1 \\ \Delta\dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{\ell} & -\frac{f}{m} \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix} u$

the trajectories starting close to the stable equilibrium "stay around there", while the trajectories starting close to the unstable equilibrium "run away". This is because of the inner structure of the two state update matrices – another reason why we shall study linear algebra

notes

- summarizing, we get this
- and note then how just changing one sign in the A matrices we have to get two systems that behave completely differently
- we definitely have to learn linear algebra

Summarizing the procedure

- linearizing $\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u})$ is meaningful only around an equilibrium $(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}})$
- to find the equilibria of a system we need to solve $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) = \boldsymbol{0}$
- each equilibrium will lead to its "own" corresponding linear model $\dot{\boldsymbol{y}} = A\boldsymbol{y} + B\boldsymbol{u}$, where A and B thus depend on $(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}})$ and $\boldsymbol{y}$, $\boldsymbol{u}$ in $\dot{\boldsymbol{y}} = A\boldsymbol{y} + B\boldsymbol{u}$ have actually the meaning of $\Delta\boldsymbol{y}$, $\Delta\boldsymbol{u}$ with respect to the equilibrium
- each linearized model $\dot{\boldsymbol{y}} = A\boldsymbol{y} + B\boldsymbol{u}$ is more or less valid only in a neighborhood of $(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}})$. Moreover the size of this neighborhood depends on the curvature of $\boldsymbol{f}$ around that specific equilibrium point

notes

- so, the first step to linearize a system is remembering that we do linearizations around equilibria
- then we should also remember how to compute equilibria
- as shown in the pendulum example above, each equilibrium leads to its own dynamics
- and we also have that the approximation is more or less valid, depending on different factors

Recapping the rationale behind linearization

- linear systems are easier to analyze than nonlinear systems
- modal analysis and rational Laplace-transforms call for linear systems
- many advanced control techniques are based on linear systems

linearization = a very useful tool to do
analysis and design of control systems

notes

- and let's also spend a minute on saying again why we do this
- first of all, this is true
- moreover we will see soon that linear systems enable doing a lot of very useful analyses
- and you can get controllers performing very well with linear systems
- so, remember this

Linearization - Another example

electrostatic microphone:

- q = capacitor charge
- h = distance of armature from its natural equilibrium
- $\boldsymbol{y} = \left[q, h, \dot{h}\right]$
- R = circuit resistance
- E = voltage generated by the generator (constant)
- C = capacity of the capacitor
- m = mass of the diaphragm + moved air
- k = mechanical spring coefficient
- β = mechanical dumping coefficient
- u_1 = incoming acoustic signal

notes

- this is another example that you may try by yourself

Linearization - Another example

a physics-driven model:

$$\begin{cases} \dot{y}_1 &= -\dfrac{1}{Ra} y_1 \left(L + y_2\right) + \dfrac{E}{R} \\ \dot{y}_2 &= y_3 \\ \dot{y}_3 &= -\dfrac{\beta}{m} y_3 - \dfrac{k}{m} y_2 - \dfrac{y_1^2}{2am} + \dfrac{1}{m} u_1 \end{cases}$$

notes

- without entering too much into details (you can also take a look at the corresponding example in the Fornasini Marchesini), this is a meaningful model of this system

Linearization - Example

1-st step: compute the equilibria

$$\begin{cases} \dot{y}_1 & = -\frac{1}{Ra} y_1 \left(L + y_2 \right) + \frac{E}{R} \\ \dot{y}_2 & = y_3 \\ \dot{y}_3 & = -\frac{\beta}{m} y_3 - \frac{k}{m} y_2 - \frac{y_1^2}{2am} + \frac{1}{m} u_1 \end{cases}$$

2-nd step: compute the matrices

$$A = \nabla_{\boldsymbol{y}} \boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}\right)\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}} \quad B = \nabla_{\boldsymbol{u}} \boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}\right)\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}} \quad C = \nabla_{\boldsymbol{y}} \boldsymbol{g}\left(\boldsymbol{y}, \boldsymbol{u}\right)\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}} \quad D = \nabla_{\boldsymbol{u}} \boldsymbol{g}\left(\boldsymbol{y}, \boldsymbol{u}\right)\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}}$$

notes

- try to do this by yourself!

Summarizing

Linearize a nonlinear ODE around an equilibrium point

- find the equilibria
- select an equilibrium
- compute the derivatives around that equilibrium
- use the formulas
- don't forget that you are also changing the coordinate system!

notes

- you should now be able to do this, following the pseudo-algorithm in the itemized list

Most important python code for this sub-module

notes

This will do everything for you

https://python-control.readthedocs.io/en/latest/generated/control.linearize.html

though it is dangerous to use tools without knowing how they work

notes

- be always wary of using code without knowing the effects and meaning of the operations they do

Self-assessment material

notes

Question 1

What does it mean to linearize a nonlinear ordinary differential equation (ODE)?

Potential answers:

I: **(correct)** It means approximating the nonlinear ODE with a linear model around an equilibrium point.
II: **(wrong)** It means replacing the ODE with a completely unrelated linear system.
III: **(wrong)** It means integrating the ODE analytically to find a closed-form solution.
IV: **(wrong)** It means ignoring all nonlinear terms in the system dynamics.
V: **(wrong)** I do not know

Solution 1:

Linearizing an ODE means approximating it with a linear model around an equilibrium point using a first-order Taylor series expansion. This allows for easier analysis and control design.

notes

- see the associated solution(s), if compiled with that ones :)

Question 2

What is the primary requirement for performing a valid linearization of a function?

Potential answers:

I: **(wrong)** The function must be polynomial.
II: **(correct)** The function must be differentiable at the point of linearization.
III: **(wrong)** The function must be bounded over the entire real line.
IV: **(wrong)** The function must have a second derivative at all points.
V: **(wrong)** I do not know

Solution 1:

A function must be differentiable at the point of linearization to compute its first-order Taylor series expansion, which is the basis for linearization.

notes

- see the associated solution(s), if compiled with that ones :)

Question 3

Why do we typically linearize a nonlinear system around an equilibrium point?

Potential answers:

I: **(wrong)** Because equilibrium points always yield globally valid linear models.
II: **(wrong)** Because nonlinear systems have no real solutions.
III: **(correct)** Because an equilibrium point ensures the validity of the local linear approximation.
IV: **(wrong)** Because linearization eliminates all system dynamics.
V: **(wrong)** I do not know

Solution 1:

Linearizing around an equilibrium point ensures that the approximation is meaningful in a small neighborhood, as the system is at rest or has steady-state behavior there.

notes

- see the associated solution(s), if compiled with that ones :)

Question 4

In a state-space representation of an ODE, what do the matrices A and B represent in the linearized system?

Potential answers:

I: **(wrong)** A and B are arbitrary matrices chosen for stability.
II: **(wrong)** A represents the second derivative of the state, and B represents the system's damping.
III: **(wrong)** A and B are obtained by solving the system for eigenvalues and eigenvectors.
IV: **(correct)** A is the Jacobian of the system dynamics with respect to the state, and B is the Jacobian with respect to the input.
V: **(wrong)** I do not know

Solution 1:

The matrices A and B in a linearized state-space model come from computing the Jacobian matrices of the system dynamics with respect to the state and input, respectively, at the equilibrium point.

notes

- see the associated solution(s), if compiled with that ones :)

Question 5

Which of the following is a common limitation of linearizing a nonlinear system?

Potential answers:

I: **(correct)** The linearized model is only valid in a small neighborhood around the linearization point.
II: **(wrong)** The linearized model has no practical applications in control.
III: **(wrong)** Linearization makes the system unstable.
IV: **(wrong)** Linearization eliminates all dynamic behavior of the system.
V: **(wrong)** I do not know

Solution 1:

A linearized model is typically valid only in a small neighborhood around the equilibrium point where it was derived. If the system deviates significantly from this region, the approximation may no longer be accurate.

notes

- see the associated solution(s), if compiled with that ones :)

Recap of sub-module "how to linearize an ODE"

- linearization requires following a series of steps (see the summary above)
- the model that one gets in this way is an approximation of the original model
- having a graphical understanding of what means what is essential to remember how to do things
- better testing a linear controller before a nonlinear one

notes

- the most important remarks from this sub-module are these ones