Table of Contents I

- how to linearize a RR
  - Most important python code for this sub-module
  - Self-assessment material

notes

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

how to linearize a RR

notes

Contents map

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

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

notes

Main ILO of sub-module "how to linearize a RR"

Linearize a nonlinear RR around an equilibrium point

notes

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

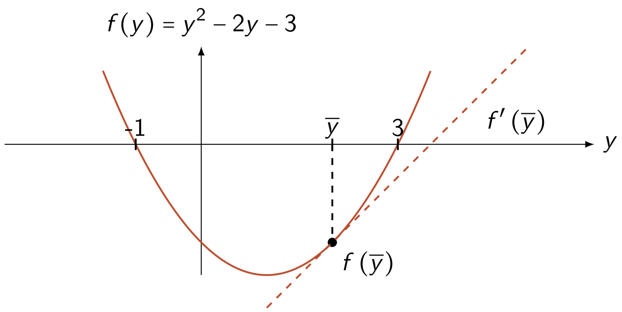
The path towards linearizing a model

- 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?

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

linearize $\Longrightarrow$ *approximate each component!*

Discussion: then $\left.\nabla_{\boldsymbol{y}} \boldsymbol{f}\right|_{\boldsymbol{y}_0}$ 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 $\boldsymbol{y}_0$

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

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 $\boldsymbol{y}_0, \boldsymbol{u}_0$,

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

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:

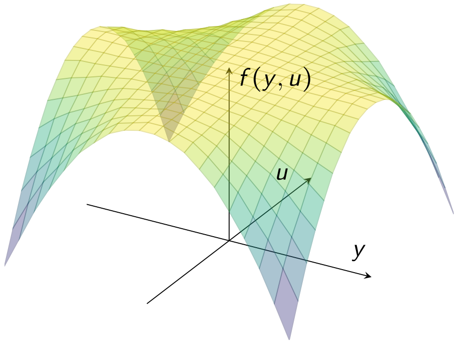
$$\boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}\right) \approx \boldsymbol{f}\left(\boldsymbol{y}_0, \boldsymbol{u}_0\right) + \nabla\boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}\right)\Big|_{\boldsymbol{y}_0, \boldsymbol{u}_0} \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{u} \end{bmatrix}$$

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

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

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

important case for our purposes:

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

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?

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

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?

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

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

notes

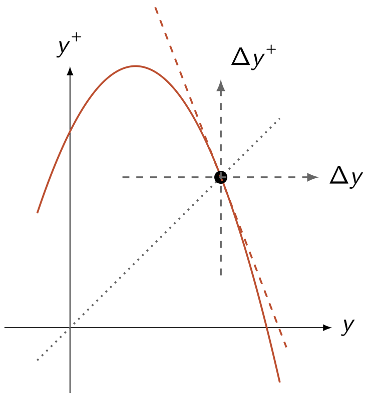
- note that sometimes one may do some simple tricks to do a linearization that is actuall a non-linearization
- for example here we may say that $\overline{u}[k] = u[k]^{2/3}$, and get a linear model in $\overline{u}$
- this basically would say just putting a nonlinear function in the block scheme, and 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? Note: main difference w.r.t. CT systems = equilibria are on the bisectors:



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
- in DT systems, the equilibria are given by the condition $y = f(y)$, that means on the bisectors
- and as in CT systems, saying that one wants to work with the linearized system means that one will have to introduce a change of coordinates, and go from the coordinate y to that of Δy, whose meaning is "offset w.r.t. the equilibrium"

Linearization procedure - discrete time systems

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

Procedure (assuming that the Taylor expansion exists):

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

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

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

$$\implies \quad \boldsymbol{y}^+ = \left(\boldsymbol{y}_{\text{eq}}^+ + \Delta\boldsymbol{y}^+\right) = \boldsymbol{y}_{\text{eq}} + \Delta\boldsymbol{y}^+ \approx \boldsymbol{f}(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}) + \nabla \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u})\Big|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}} \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{u} \end{bmatrix}$$

note then that $\boldsymbol{y}_{\text{eq}} = \boldsymbol{f}(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}})$, so that we can simplify the previous expression

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 $\boldsymbol{y}_{\text{eq}}$ is constant and given the fact that we are on an equilibrium, this follows

Linearization procedure - discrete time systems

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

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

and, since

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

it follows that the approximated system is

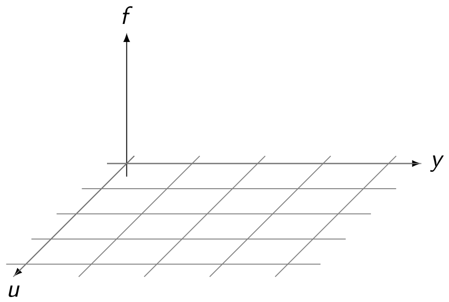
$$\Delta\widetilde{\boldsymbol{y}}^+ = A\Delta\widetilde{\boldsymbol{y}} + B\Delta\boldsymbol{u}$$

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?

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



notes

- see the drawings made during the video of the lesson

Summarizing the procedure

- linearizing $\boldsymbol{y}^+ = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u})$ 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 $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) = \boldsymbol{y}$
- each equilibrium will lead to its "own" corresponding linear model $\boldsymbol{y}^+ = A\boldsymbol{y} + B\boldsymbol{u}$, where A and B thus depend on $(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}})$ and $\boldsymbol{y}$, $\boldsymbol{u}$ in $\boldsymbol{y}^+ = A\boldsymbol{y} + B\boldsymbol{u}$ have actually the meaning of $\Delta\boldsymbol{y}$, $\Delta\boldsymbol{u}$ with respect to the equilibrium
- each linearized model $\boldsymbol{y}^+ = A\boldsymbol{y} + B\boldsymbol{u}$ 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 Z-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

Summarizing

Linearize a nonlinear RR 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 is the primary purpose of linearizing a nonlinear system around an equilibrium point?

Potential answers:

I: **(correct)** To approximate the system's behavior in a small neighborhood of the equilibrium.
II: **(wrong)** To completely replace the nonlinear system with a linear one globally.
III: **(wrong)** To eliminate all nonlinearities in the system.
IV: **(wrong)** To make the system unstable for control purposes.
V: **(wrong)** I do not know

Solution 1:

The primary purpose of linearizing a nonlinear system around an equilibrium point is to approximate the system's behavior in a small neighborhood of the equilibrium. This allows us to use linear control techniques and analysis tools, which

notes

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

Question 2

Which of the following is a necessary condition for linearizing a function $f(y)$ around a point $\overline{y}$?

Potential answers:

I: **(correct)** The function $f(y)$ must be differentiable at $\overline{y}$.
II: **(wrong)** The function $f(y)$ must be discontinuous at $\overline{y}$.
III: **(wrong)** The function $f(y)$ must be constant in a neighborhood of $\overline{y}$.
IV: **(wrong)** The function $f(y)$ must be linear globally.
V: **(wrong)** I do not know

Solution 1:

A necessary condition for linearizing a function $f(y)$ around a point $\overline{y}$ is that the function must be differentiable at $\overline{y}$. This ensures that the Taylor series expansion can be applied to approximate the function.

notes

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

Question 3

What is the dimension of the Jacobian matrix $\nabla_{\boldsymbol{y}} \boldsymbol{f}$ for a vectorial function $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$?

Potential answers:

I: **(correct)** $m \times n$
II: **(wrong)** $n \times m$
III: **(wrong)** $n \times n$
IV: **(wrong)** $m \times m$
V: **(wrong)** I do not know

Solution 1:

The Jacobian matrix $\nabla_{\boldsymbol{y}} \boldsymbol{f}$ for a vectorial function $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$ has dimensions $m \times n$. This is because it consists of the partial derivatives of each of the m components of $\boldsymbol{f}$ with respect to each of the n variables in $\boldsymbol{y}$.

notes

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

Question 4

In the context of discrete-time systems, where are the equilibria located?

Potential answers:

I: **(correct)** On the bisector where $y = f(y)$.
II: **(wrong)** On the bisector where $y = u$.
III: **(wrong)** On the bisector where $f(y) = u$.
IV: **(wrong)** On the bisector where $y = 0$.
V: **(wrong)** I do not know

Solution 1:

In discrete-time systems, the equilibria are located on the bisector where $y = f(y)$. This is because, by definition, an equilibrium point satisfies $\boldsymbol{y}_{\text{eq}} = \boldsymbol{f}(\boldsymbol{y}_{\text{eq}}, \boldsymbol{u}_{\text{eq}})$.

notes

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

Question 5

What is the main limitation of linearizing a nonlinear system around an equilibrium point?

Potential answers:

I: **(correct)** The approximation is only valid in a small neighborhood of the equilibrium.
II: **(wrong)** The linearized system is always unstable.
III: **(wrong)** The linearized system cannot be used for control purposes.
IV: **(wrong)** The linearized system is always globally accurate.
V: **(wrong)** I do not know

Solution 1:

The main limitation of linearizing a nonlinear system around an equilibrium point is that the approximation is only valid in a small neighborhood of the equilibrium. Outside this neighborhood, the linear approximation may significantly deviate from the actual nonlinear behavior.

notes

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

Recap of sub-module "how to linearize a RR"

- 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