Table of Contents I

notes

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

1D convolution in discrete time

notes

-

Contents map

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

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

notes

Main ILO of sub-module "1D convolution in discrete time"

Compute the convolution between two single dimensional discrete time signals

notes

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

Why convolution?

because for a LTI system with impulse response $h[k]$
it follows that $y_{\text{forced}}[k] = u * h[k]$

notes

- and indeed formally we get this result, that we do not prove for now

extremely important result for LTI systems:

$$y_{\text{forced}}[k] = h * u[k] = u * h[k] :=$$

$$:= \sum_{\kappa=-\infty}^{+\infty} u[\kappa]h[k-\kappa] = \sum_{-\infty}^{+\infty} h[\kappa]u[k-\kappa]$$

. . . and this module = what that formula actually means from graphical perspectives

notes

-

Additional material

Videos:

- https://www.youtube.com/watch?v=KuXjwB4LzSA
- https://www.youtube.com/watch?v=acAw5WGtzuk
- https://www.youtube.com/watch?v=IaSGqQa5O-M (for connections with probability)
- https://www.youtube.com/playlist?list=PL4iThgVpN7hmbIhHnCa7SDOOgLMoNwED_
- https://www.youtube.com/playlist?list=PL4mJLdGEHNvhCuPXsKFrnD7AaQBlMEB6a

Animations:

- https://lpsa.swarthmore.edu/Convolution/CI.html
- https://phiresky.github.io/convolution-demo/

notes

- the world is full of material on convolution - check also this stuff and not only what we do in class

Towards decomposing this formula in pieces

$$y_{\text{forced}}[k] = h * u[k] = \sum_{-\infty}^{+\infty} u[\kappa] h[k-\kappa] = \sum_{-\infty}^{+\infty} h[\kappa] u[k-\kappa]$$

better focus on

$$\sum_{-\infty}^{+\infty} u[\kappa] h[k-\kappa]$$

or on

$$\sum_{-\infty}^{+\infty} h[\kappa] u[k-\kappa] \, ?$$

in automatic control typically better the second

notes

- in the first option one has to "flip" u
- in the second option one has to "flip" h
- the second gives us more intuitions about the system, makes the impulse response directly interpretable
- they though give the same results

Towards decomposing this formula in pieces, small change of notation

$$y_{\text{forced}}[k] = \sum_{-\infty}^{+\infty} h[\kappa]u[k-\kappa] \quad \mapsto \quad y_{\text{forced}}[\text{now}] = \sum_{-\infty}^{+\infty} h[\kappa]u[\text{now}-\kappa]$$

notes

- better doing this change of notation, so to avoid some confusion

Decomposing this formula in pieces

$$y_{\text{forced}}[\text{now}] = \sum_{-\infty}^{+\infty} h[\kappa]u[\text{now}-\kappa]$$

$\Longrightarrow$ constituent pieces =

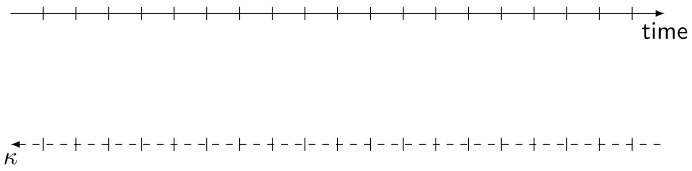
- $u[\text{now}-\kappa]$
- $h[\kappa]$
- $u[\text{now}-\kappa]h[\kappa]$
- $\sum u[\text{now}-\kappa]h[\kappa]$

notes

- the main pieces / steps of computing a convolution are these ones

Visualizing the various pieces

$$u[\text{now} - \kappa] \qquad h[\kappa] \qquad u[\text{now} - \kappa]h[\kappa] \qquad \sum_{-\infty}^{+\infty} h[\kappa]u[\text{now} - \kappa]$$

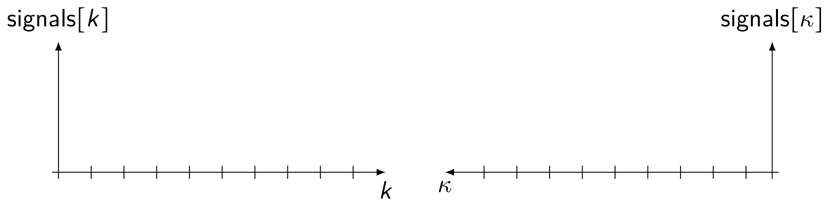


notes

- likely better watch the video in class for this animation, or try to refer to `https://lpsa.swarthmore.edu/Convolution/CI.html`

Example

$$u[k] = \begin{cases} 1 & \text{for } k \in \{1,2\} \\ 0 & \text{otherwise} \end{cases} \qquad h[k] = \begin{cases} 2 & \text{for } k \in \{0,1\} \\ 1 & \text{for } k \in \{2\} \\ 0 & \text{otherwise} \end{cases}$$



notes

- see the video of the lesson for the example
- note though that with the second option the meaning of h is "how important the past is to create the current output"

Another Example:

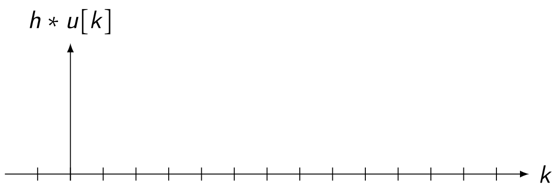
$$u[k] = \begin{cases} 1 & \text{for } k \in \{0,1,3\} \\ 0 & \text{otherwise} \end{cases} \qquad h[k] = \begin{cases} 2 & \text{for } k \in \{0,1,2\} \\ 0 & \text{otherwise} \end{cases} \qquad ?$$

notes

- see the video of the lesson

Paramount message

h in $y_{\text{forced}}[k] = \sum_{-\infty}^{+\infty} h[\kappa] u[k-\kappa]$ represents how much the past u's contribute to the current y_{forced}:



notes

- this is a very important interpretation, thus better to fix it
- for example, $h(7)$ says how much $u(t-7)$ will enter into $y[k]$ for any k
- so somehow $h[\kappa]$ has an important graphical meaning
- if $h[\kappa]$ decays fast, then somehow the system "forgets" the past inputs fast, and viceversa

Refreshing what we are doing and why

Dynamics of a cart: $v[k+1] = \alpha v[k] + \beta F[k]$ with:

- **control input:** $u[k]$ (actuation from the motor, in this case $= F[k]$)
- **system output:** $y[k]$ (cart velocity, in this case $= v[k]$)
- **impulse response:** $h[k]$ (output corresponding to the input $\delta[k]$ assuming $y[0] = 0$)
- **free evolution:** $y_{\text{free}}[k]$ (output in time corresponding to no input, i.e., $u[k] = 0$, and initial condition $y[0]$ whatever it is)
- **forced response:** $y_{\text{forced}}[k] = u * h[k]$ (output in time corresponding to null initial condition, i.e., $y[0] = 0$, and input $u[k]$ whatever it is)
- **total response:** $y[k] = y_{\text{free}}[k] + y_{\text{forced}}[k]$

notes

- just to be sure: what we are doing here is essential
- we want to understand how to map the input into the output
- if we want to do model predictive control indeed we need to know what a certain input is going to cause to the output

Quiz time!

$$h * u[k] := \sum_{-\infty}^{+\infty} h[\kappa] u[k - \kappa]$$

- is $h * u[k] = u * h[k]$?
- is $(\alpha h_1 + \beta h_2) * u[k] = \alpha (h_1 * u[k]) + \beta (h_2 * u[k])$?
- if both $h[\kappa] = 0$ and $u[k] = 0$ if $t < 0$, how can we simplify $y[k] = \sum_{-\infty}^{+\infty} h[\kappa] u[k - \kappa]$?

notes

-
- obviously yes, since we can do the change of variables
-
- yes because the integrals are linear. This implies that the convolution operator is linear
-
- yes, it becomes $y[k] = \sum_{0}^{t} h[\kappa] u[k - \kappa]$

Summarizing

Compute the convolution between two single dimensional continuous time signals

- take one of the two signals
- translate it to the "current k"
- flip it
- multiply the two signals in a pointwise fashion
- compute the discrete integral of the result

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

Methods implementing (discrete) convolutions

- https://numpy.org/doc/2.1/reference/generated/numpy.convolve.html
- https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve.html

notes

- note that these methods take discrete arrays, also multidimensional

Self-assessment material

notes

Question 1

What does the convolution integral $y_{\text{forced}}[k] = \sum_{-\infty}^{+\infty} h[\kappa]u[k-\kappa]$ represent in the context of LTI systems?

Potential answers:

- I: **(wrong)** The free evolution of the system output.
- II: **(correct)** The forced response of the system output due to the input $u[k]$.
- III: **(wrong)** The total response of the system, including initial conditions.
- IV: **(wrong)** The impulse response of the system.
- V: **(wrong)** I do not know.

Solution 1:

The convolution integral $y_{\text{forced}}[k] = \sum_{-\infty}^{+\infty} h[\kappa]u[k-\kappa]$ represents the forced response of the system output due to the input $u[k]$. It describes how the system responds to the input when initial conditions are zero.

notes

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

Question 2

Which of the following is true about the convolution operation $h * u[k]$?

Potential answers:

- I: **(wrong)** It is only defined for periodic signals.
- II: **(wrong)** It is only applicable to discrete-time systems.
- III: **(correct)** It is commutative, i.e., $h * u[k] = u * h[k]$.
- IV: **(wrong)** It requires both signals to be symmetric.
- V: **(wrong)** I do not know.

Solution 1:

The convolution operation is commutative, meaning $h * u[k] = u * h[k]$. This property holds for continuous-time signals in LTI systems.

notes

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

Question 3

What does the impulse response $h[k]$ of an LTI system represent?

Potential answers:

I: **(wrong)** The input signal $u[k]$ applied to the system.
II: **(wrong)** The free evolution of the system output.
III: **(wrong)** The total response of the system, including initial conditions.
IV: **(correct)** The output of the system when the input is a Dirac delta function $\delta[k]$.
V: **(wrong)** I do not know.

Solution 1:

The impulse response $h[k]$ represents the output of the system when the input is a Dirac delta function $\delta[k]$. It characterizes the system's behavior.

notes

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

Question 4

If $h[\kappa] = 0$ for $\kappa < 0$ and $u[k] = 0$ for $t < 0$, how can the convolution integral $y[k] = \sum_{-\infty}^{+\infty} h[\kappa] u[k - \kappa]$ be simplified?

Potential answers:

I: **(correct)** $y[k] = \sum_{0}^{t} h[\kappa] u[k - \kappa]$
II: **(wrong)** $y[k] = \sum_{0}^{+\infty} h[\kappa] u[k - \kappa]$
III: **(wrong)** $y[k] = \sum_{-\infty}^{+\infty} h[\kappa] u[\kappa]$
IV: **(wrong)** $y[k] = \sum_{-\infty}^{0} h[\kappa] u[k - \kappa]$
V: **(wrong)** I do not know.

Solution 1:

notes

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

Question 5

What is the graphical interpretation of $h[\kappa]$ in the convolution integral $y_{\text{forced}}[k] = \sum_{-\infty}^{+\infty} h[\kappa] u[k-\kappa]$?

Potential answers:

- I: **(wrong)** It represents the future inputs of the system.
- II: **(correct)** It represents how much past inputs contribute to the current output.
- III: **(wrong)** It represents the free evolution of the system.
- IV: **(wrong)** It represents the total energy of the system.
- V: **(wrong)** I do not know.

Solution 1:

The term $h[\kappa]$ in the convolution integral represents how much past inputs $u[k-\kappa]$ contribute to the current output $y[k]$. This is a key interpretation in LTI systems.

notes

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

Recap of sub-module "1D convolution in discrete time"

- convolution is an essential operator, since it can be used for LTI systems to compute forced responses
- its graphical interpretation aids interpreting impulse responses as how the past inputs contribute to current outputs

notes

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