Table of Contents I

notes

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

computing free evolutions and forced responses of LTI systems

notes

-

Contents map

developed content units	taxonomy levels
free evolution	u1, e1
forced response	u1, e1

prerequisite content units	taxonomy levels
LTI RR	u1, e1
convolution	u1, e1
partial fraction decomposition	u1, e1

notes

▪

Main ILO of sub-module "computing free evolutions and forced responses of LTI systems"

Compute free evolutions and forced responses of LTI systems using Z-transforms based formulas (but only as procedural tools)

notes

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

Disclaimer

the formulas introduced in this module shall be taken as "ex machina"

notes

- in other words, they are given and to be assumed as true
- in other courses or modules they will be derived from other principles

Focus in this module = on ARMA models

$$y^{[n]} = a_{n-1} y^{[n-1]} + \ldots + a_0 y + b_m u^{[m]} + \ldots + b_0 u$$

with $^{[i]}$ meaning the i-th step ahead sample (e.g, $y^{[3]} = y^{+++}$). *Discussion:* why is the LHS $y^{[n]}$ and not $a_n y^{[n]}$? *Discussion:* and which initial conditions shall we consider?

notes

- generalizing the LTIs we saw until now, we can arrive at these models, and in this module we will treat only these models (there may be other generalizations but you will see them in other modules)
- the $a_{n-1} y^{[n-1]} + \ldots + a_0 y$ part is called Auto-Regressive
- the $b_m u^{[m]} + \ldots + b_0 u$ part is called Moving-Average
- these names make more sense in discrete time systems of the type $y(k+n) = a_{n-1} y(k+n-1) + \ldots + a_0 y(k) + b_m u^(k+m) + \ldots + b_0 u(k)$ and k a discrete time index. Here we see that the a's correspond to an autoregression, and the b's to the coefficients of a moving average. In any case we use ARMA for both continuous and discrete dynamics of these types
- note that in mechanical systems like motors, the derivatives of u are meaningful because they capture the system's response to changes in the input signal, accounting for physical constraints like inertia
- this is because if we were having $a_n y^{[n]}$ on the left hand side then we could divide all the a's and b's on the right hand side and get the same dynamics
- so we prefer to work with monic polynomials (i.e., in which the leading coefficient, that is the nonzero coefficient of highest degree, is equal to 1) because we have less numbers to carry around (plus it will be convenient for other purposes that we will see later on in the course)
- as for the initial conditions that one shall consider, we typically assume all the conditions on the u equal to zero, while on the y they may be different from zero

Z transforms - links for who would like to get more info about them

Z transforms = discretization of Laplace transforms; interesting material:

- `https://www.youtube.com/watch?v=XJRW6jamUHk`
- `https://www.youtube.com/watch?v=acQecd6dmxw`
- `https://www.youtube.com/watch?v=4PV6ikgBShw`
- `https://www.youtube.com/watch?v=7Gl4kJUjp4c`

notes

- note that this module treats the formulas as "given", so who wants to look at these links shall do only for self-interest, not for preparing oneself for exercises at the exam related to this module

Laplace transforms - links for who would like to get more info about them

Laplace transforms = extension of Fourier transforms; interesting material:

- `https://www.youtube.com/watch?v=r6sGWTCMz2k` (Fourier series)
- `https://www.youtube.com/watch?v=spUNpyF58BY` (Fourier transforms)
- `https://www.youtube.com/watch?v=nmgFG7PUHfo` (on the historical importance of Fast Fourier Transforms)
- `https://www.youtube.com/watch?v=7UvtU75NXTg` (Laplace Transforms, in math)
- `https://www.youtube.com/watch?v=n2y7n6jw5d0` (Laplace Transforms, graphically)

notes

- note that this module treats the formulas as "given", so who wants to look at these links shall do only for self-interest, not for preparing oneself for exercises at the exam related to this module

Main usefulness: convolution in time transforms into multiplication in Z-domain, and viceversa

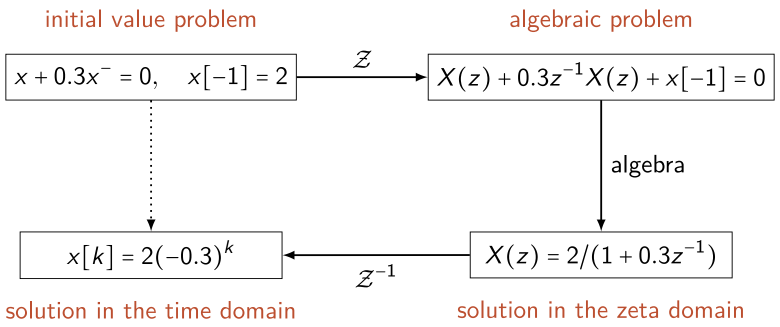
$$\begin{cases} H(z) = \mathcal{Z}\{h[k]\} \\ U(z) = \mathcal{Z}\{u[k]\} \end{cases} \implies \mathcal{Z}\{h * u[k]\} = H(z)U(z)$$

Noticeable name: *transfer function* $(= H(z) = \mathcal{Z}\{\text{impulse response}\})$

notes

- this is by far one of the most important properties of Zeta transforms for our purposes: convolution in one of the domains will be multiplication in the other
- this implies that instead of computing $u * y$, if computing H and U is fast, and if inverting HU is fast, that way is preferrable
- the name "transfer function" is an important one and you will hear about it quite often

An intuitive explanation of the usefulness of the Zeta transform in automatic control



notes

- this means that we can follow this scheme
- in other words, for complicated differential equations Zeta transform allow us to solve the problem algebraically. This is often much easier than solving the ODE directly

First set of formulas to memorize: Zeta-transforming derivatives

(these will be motivated in other courses)

$$\mathcal{Z}\left\{x^{-}\right\} = z^{-1}X(z) + x[-1]$$

$$\mathcal{Z}\left\{x^{--}\right\} = z^{-2}X(z) + x[-2] + z^{-1}x[-1]$$

$$\mathcal{Z}\left\{x^{---}\right\} = z^{-3}X(z) + x[-3] + z^{-1}x[-2] + z^{-2}x[-1]$$

$$\mathcal{Z}\left\{x^{[-m]}\right\} = \ldots$$

notes

- these formulas shall be remembered by heart

Example: discretized spring mass system

$$y + \alpha_1 y^{-} + \alpha_2 y^{--} + \beta u^{--}$$

$$\Downarrow$$

$$Y(z) + \left(\alpha_1 z^{-1} Y(z) + \alpha_1 y[-1]\right) + \left(\alpha_2 z^{-2} Y(z) + \alpha_2 y[-2] + \alpha_2 z^{-1} y[-1]\right) = \beta z^{-2} U(z)$$

$$\Downarrow$$

$$Y(z) + \alpha_1 z^{-1} Y(z) + \alpha_2 z^{-2} Y(z) = \alpha_1 y[-1] + \alpha_2 y[-2] + \alpha_2 z^{-1} y[-1] + \beta z^{-2} U(z)$$

$$\Downarrow$$

$$Y(z) = \frac{\alpha_1 y[-1] + \alpha_2 y[-2] + \alpha_2 z^{-1} y[-1]}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}} + \frac{z^{-2}\beta}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}} U(z)$$

notes

- let's then start this path building on top of previous results
- more precisely, from the fact that using Zeta transforms we were able to characterize the free evolution of second order LTI systems
- and $Y(z) \neq 0$ happens when the initial conditions of the system are not null

And what shall we do once we get this?

generalizing the previous slide: $Y(z) = \frac{M(z)}{A(z)} + \frac{B(z)}{A(z)} U(z)$

with

- $\frac{M(z)}{A(z)}$ = Zeta transform of the free evolution
- $\frac{B(z)}{A(z)} U(z)$ = Zeta transform of the forced response

$\Longrightarrow$ we shall anti-transform; how? Main 2 cases:

- either $U(z) = \frac{\text{polynomial in } z}{\text{polynomial in } z}$
- or $U(z)$ = something else

notes

- now we have this first result, where we note that the total signal is the sum of the two individual signals “free evolution” plus “forced response”, but in the Zeta domain
- for the sake of this module we consider that $U(z)$ may be rational or not

first case: rational $U(z)$

notes

How to do if $U(z) = \dfrac{\text{polynomial in } z}{\text{polynomial in } z}$

$$Y(z) = \frac{M(z)}{A(z)} + \frac{B(z)}{A(z)} U(z) \quad \mapsto \quad Y(z) = \frac{M(z)}{A(z)} + \frac{C(z)}{D(z)}$$

write each of the two parts of the signal as

$$\frac{N(z)}{(z-\lambda_1)(z-\lambda_2)(z-\lambda_3)\cdots}$$

notes

- in this case we have a situation for which we can write both elements as polynomial over polynomial

Next step: partial fraction decomposition

- **case single poles:** if $\dfrac{N(z)}{(z-\lambda_1)(z-\lambda_2)(z-\lambda_3)\cdots}$ is s.t. $\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \cdots$ then there exist $\alpha_1, \alpha_2, \alpha_3, \ldots$ s.t.

$$\frac{N(z)}{(z-\lambda_1)(z-\lambda_2)(z-\lambda_3)\cdots} = \frac{\alpha_1 z}{z-\lambda_1} + \frac{\alpha_2 z}{z-\lambda_2} + \frac{\alpha_3 z}{z-\lambda_3} + \cdots \tag{1}$$

- **case repeated poles:** if some poles are repeated, then there exist $\alpha_{1,1}, \ldots, \alpha_{1,n1}, \alpha_{2,1}, \ldots, \alpha_{2,n2}, \ldots,$ s.t.

$$\frac{N(z)}{(z-\lambda_1)^{n1}(z-\lambda_2)^{n2}\cdots} = \frac{\alpha_{1,1}(z)}{z-\lambda_1} + \ldots + \frac{\alpha_{1,n1}(z)}{(z-\lambda_1)^{n1}} + \frac{\alpha_{2,1}(z)}{z-\lambda_2} + \ldots + \frac{\alpha_{2,n2}(z)}{(z-\lambda_2)^{n2}} + \ldots \tag{2}$$

"But how do I compute α_1, α_2, etc.?" $\mapsto$
`en.wikipedia.org/wiki/Partial_fraction_decomposition`
(tip: start from `en.wikipedia.org/wiki/Heaviside_cover-up_method`)

notes

- let's remember that the partial fraction decomposition concept helps us factorizing ratios of polynomials in a sum of simpler ratios
- and let's also remember that there is the possibility of having multiple poles (something that, as we will see very soon, connects with the concept of non-trivial Jordan structure of the A expressing this LTI system)
- in case somebody does not remember how to do it, there is a couple of resources that may help re-gaining knowledge on this tool

Anti-transforming in the rational $U(z)$ and simple poles case

$$\text{if } Y(z) = \frac{\alpha_1 z}{z - \lambda_1} + \frac{\alpha_2 z}{z - \lambda_2} + \ldots \text{ then use}$$

$$\mathcal{Z}\left\{\alpha \lambda^k\right\} = \frac{\alpha z}{z - \lambda} \quad \leftrightarrow \quad \mathcal{Z}^{-1}\left\{\frac{\alpha z}{z - \lambda}\right\} = \alpha \lambda^k$$

notes

- given this transform, $y[k]$ is then immediately a sum of terms of the type $t^n e^{\lambda t}$ for opportune n's that depend on the specific λ
- we see that this must connect with the structure of the Jordan form of the A expressing this LTI
- we will reinforce this connection later on – the important for now is to realize that it exists
- now either all the terms are simple, or there are some repeated lambda's

Anti-transforming in the rational $U(z)$ case

$$\text{if } Y(z) = \frac{\alpha_{1,1} z}{z - \lambda_1} + \ldots + \frac{\alpha_{1,n1}(z)}{(z - \lambda_1)^{n1}} + \frac{\alpha_{2,1} z}{z - \lambda_2} + \ldots + \frac{\alpha_{2,n2}(z)}{(z - \lambda_2)^{n2}} + \ldots$$

then use some software suite!

notes

- if though there are some higher order denominators, then remembering the formulas becomes very easily a nightmare - in this case just use python or Matlab or whatever you want to perform this operation numerically

Something though to remember

$$\mathcal{Z}\left\{k^m \lambda^k\right\} \propto \frac{\star(z)}{(z-\lambda)^{m+1}}$$

notes

- remember though that higher order denominators lead to powers, similarly to what happens with Laplace transforms and multiple poles

Numerical Example: Inverse Zeta Transform of a Rational Function

$$Y(z) = \frac{3z}{z-2} + \frac{5z}{z+1}$$

goal = compute the inverse Zeta transform $y[k] = \mathcal{Z}^{-1}\left\{Y(z)\right\}$

notes

- now let's do this exercise. Let's assume we started from an opportune $u[k]$ and ARMA model and initial conditions such that the general formula

$$Y(z) = \frac{M(z)}{A(z)} + \frac{B(z)}{A(z)} U(z)$$

has brought us to this specific $Y(z)$

Step 1: Identify the terms

$$Y(z) = \frac{3z}{z-2} + \frac{5z}{z+1}$$

Here:

- $\lambda_1 = 2$, with coefficient $\alpha_{1,1} = 3$
- $\lambda_2 = -1$, with coefficient $\alpha_{2,1} = 5$

notes

- we get immediately this, by applying what we saw in the previous slides

Step 2: Apply the inverse Zeta transform formula

by means of

$$\mathcal{Z}^{-1}\left\{\frac{z}{(z-\lambda)}\right\} = \lambda^k$$

we compute the inverse Zeta transform of each term:

- $\mathcal{Z}^{-1}\left\{\frac{3z}{z-2}\right\} = 3 \cdot 2^k$
- $\mathcal{Z}^{-1}\left\{\frac{5z}{z+1}\right\} = 5 \cdot (-1)^k$

notes

- then we get this

Step 3: Combine the results

then we have that the inverse Zeta transform $y[k]$ is the sum of the individual transforms, i.e.,

$$y[k] = 3 \cdot 2^k + 5 \cdot (-1)^k$$

notes

- then we get this

Another thing to remember: complex conjugate poles lead to sinusoidal modes

$$\mathcal{Z}\left\{\cos(\omega k)\right\} = \frac{z\big(z - \cos(\omega)\big)}{z^2 - 2z\cos(\omega) + 1}$$

notes

- and similarly to what was happening in continuous time, denominators with complex conjugate poles will translate into oscillatory modes

Extremely important result

a LTI in free evolution behaves as a combination of terms λ^k, $k\lambda^k$, $k^2\lambda^k$, etc. for a set of different λ's and powers of k, called the *modes* of the system

Discussion: assuming that we have two modes, $(0.3)^k$ and $(-0.9)^k$, so that

$$y[k] = \alpha_1 0.3^k + \alpha_2(-0.9)^k.$$

What determines α_1 and α_2?

notes

- these signals are thus somehow describing the natural way a free evolution evolves
- we already saw them with Jordan forms, and we did not give them a name then
- but they have a specific name: they are the modes of a LTI
- these numbers are given by the initial conditions of the system

second case: irrational $U(z)$

notes

In this case we cannot use partial fractions decompositions as before

$$\text{from } Y(z) = \frac{M(z)}{A(z)} + \frac{B(z)}{A(z)} U(z) \text{ we follow the algorithm}$$

- find $y_{\text{free}}[k]$ from PFDs of $\frac{M(z)}{A(z)}$ as before
- find the impulse response $h[k]$ from PFDs of $\frac{B(z)}{A(z)}$ as before
- find $y_{\text{forced}}[k]$ as $h * u[k]$

notes

- in this case we need to find the various terms independently

Summarizing

Compute free evolutions and forced responses of LTI systems using Z-transforms based formulas (but only as procedural tools)

- Z-transform the DT ARMA
- if $u[k]$ admits a rational $U(z)$ then write $Y(z) = \frac{\text{polynomial}}{\text{polynomial}}$, do PFD, and do inverse-Zetas
- if $u[k]$ does not admit a rational $U(z)$, do similarly as before but do PFD only for the free evolution and impulse response, and find the forced response by means of convolution

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

Two essential libraries

- https://python-control.readthedocs.io/en/0.10.1/generated/control.modal_form.html
- https://docs.sympy.org/latest/modules/physics/control/lti.html

notes

- these libraries provide you the necessary tools to perform modal analysis as here

Self-assessment material

notes

Question 1

What is the primary purpose of using Z-transforms in the context of LTI systems?

Potential answers:

I: **(correct)** To convert convolution in the time domain into multiplication in the Z-domain.
II: **(wrong)** To derive the Laplace transform from the Fourier transform.
III: **(wrong)** To compute the eigenvalues of the system matrix.
IV: **(wrong)** To solve partial differential equations directly.
V: **(wrong)** I do not know

Solution 1:

The primary purpose of using Z-transforms in LTI systems is to simplify the analysis by converting convolution in the time domain into multiplication in the Z-domain. This property makes it easier to solve differential equations and analyze system behavior.

notes

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

Question 2

In the ARMA model $y^{[n]} = a_{n-1}y^{[n-1]} + \ldots + a_0 y + b_m u^{[m]} + \ldots + b_0 u$, why is the left-hand side $y^{[n]}$ and not $a_n y^{[n]}$?

Potential answers:

I: **(correct)** To work with monic polynomials, simplifying the analysis.
II: **(wrong)** To ensure the system is always stable.
III: **(wrong)** To make the system nonlinear.
IV: **(wrong)** To reduce the number of initial conditions required.
V: **(wrong)** I do not know

Solution 1:

The left-hand side is $y^{[n]}$ and not $a_n y^{[n]}$ to work with monic polynomials, which simplifies the analysis by reducing the number of coefficients to carry around.

notes

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

Question 3

What is the purpose of partial fraction decomposition in the context of Z-transforms?

Potential answers:

I: **(correct)** To break down a complex rational function into simpler terms for inverse Z-transform.
II: **(wrong)** To compute the convolution of two signals directly.
III: **(wrong)** To derive the Laplace transform from the Z-transform.
IV: **(wrong)** To solve nonlinear differential equations.
V: **(wrong)** I do not know

Solution 1:

Partial fraction decomposition is used to break down a complex rational function into simpler terms, making it easier to compute the inverse Z-transform and analyze the system's behavior.

notes

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

Question 4

What are the modes of a LTI system in free evolution?

Potential answers:

I: (**correct**) Combinations of terms like λ^k, $k\lambda^k$, $k^2\lambda^k$, etc.
II: (**wrong**) The eigenvalues of the system matrix.
III: (**wrong**) The coefficients of the ARMA model.
IV: (**wrong**) The initial conditions of the system.
V: (**wrong**) I do not know

Solution 1:

The modes of a LTI system in free evolution are combinations of terms like λ^k, $k\lambda^k$, $k^2\lambda^k$, etc., which describe the natural evolution of the system.

notes

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

Question 5

How is the forced response of a LTI system computed when $U(z)$ is not rational?

Potential answers:

I: (**correct**) By computing the convolution of the impulse response $h[k]$ with the input $u[k]$.
II: (**wrong**) By using partial fraction decomposition on $U(z)$.
III: (**wrong**) By directly inverting the Z-transform of $U(z)$.
IV: (**wrong**) By solving the system's differential equations numerically.
V: (**wrong**) I do not know

Solution 1:

When $U(z)$ is not rational, the forced response is computed by finding the impulse response $h[k]$ and then convolving it with the input $u[k]$.

notes

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

Recap of sub-module “computing free evolutions and forced responses of LTI systems”

- finding such signals require knowing a couple of formulas by heart
- partial fraction decomposition is king here, one needs to know how to do that

notes

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