ELEC 221

Ruiheng Su 2020

Fundamentals

$$a^b = 10^{b \log_{10}(a)}$$

Complex numbers

Let $z = x + iy$ be a complex number, and $z^* = x - iy$. The magnitude of z is

$$r = |z| = \sqrt{zz^*} = \sqrt{x^2 + y^2}$$

Complex exponentials

Euler's formula:

$$e^{j\theta} = \cos(\theta) + i\sin(\theta)$$

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2} \qquad \sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

Polar coordinates

For $r^2 = x^2 + Y^2$, there is an angle $0 \leq \theta < 2\pi$ such that $\cos(\theta) = x/r$, and $\sin(\theta) = y/r$. The polar representation is

$$z = |z|\left(\frac{x}{|z|} + i\frac{y}{|z|}\right) = |z|(\cos(\theta) + i\sin(\theta)) = |z|e^{i\theta}$$

For any integer n,

$$re^{i(2\pi n + \theta)} = re^{i\theta}$$

Trigonometric Identities

$$\begin{aligned}
\cos(-\alpha) &= \cos(\alpha)\\
\sin(-\alpha) &= -\sin(\alpha)\\
\cos(\omega t) &= \sin(\omega t + \pi/2)\\
\sin(\omega t) &= \cos(\omega t - \pi/2)\\
\cos(\omega t) &= -\cos(\omega t \pm \pi)\\
\sin(\omega t) &= -\sin(\omega t \pm \pi)\\
\cos(\alpha \pm \beta) &= \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta)\\
\sin(\alpha \pm \beta) &= \sin(\alpha)\cos(\beta) \pm \cos(\alpha)\sin(\beta)\\
\cos(\alpha)\cos(\beta) &= \cos(\alpha+\beta)/2 + \cos(\alpha-\beta)/2\\
\sin(\alpha)\cos(\beta) &= \sin(\alpha+\beta)/2 + \sin(\alpha-\beta)/2\\
\cos^2(\theta) &= 1 - \sin^2(\theta)\\
\sec^2(\theta) &= \tan^2(\theta) + 1\\
\tan^2(\theta) &= \sec^2(\theta) - 1
\end{aligned}$$

Combination of $\cos(\omega t)$ and $\sin(\omega t)$

$$A\sin(\omega t) + B\cos(\omega t) = \sqrt{A^2 + B^2}\cos\left[\omega t - \tan(B/A)\right]$$

Set notation

Sets cannot have repeated members; $\{1, 2, 1\}$ is not a set.

$$A = \{1, 2, \dots, 100\} \subset \mathbb{N} \subset \mathbb{N}_0$$

A is a subset of the set of natural numbers, $\mathbb{N}$, and $\mathbb{N}$ is a subset of the set of non-negative integers, $\mathbb{N}_0 = \{0, 1, 2, \dots\}$.

$$\emptyset \subset A$$

for any set A.

$:=$	assignment
$=$	assertion

$\mathcal{P}(X)$ is the set of all subsets of X including X and $\emptyset$ called the power set.

Predicates and prototypes

Given the value a variable the predicate is true of false (e.g. given x the predicate $x \leq 5$ is true or false).
Prototype is a expression for defining sets.

$$A = \{x \in \mathbb{N} | x < 5\}$$

is a prototype.

Quantification over sets

$\forall$	for all
$\exists$	there exists

$$\forall x \in A, \text{Pred}(x)$$

is true if Pred(x) is true for all $x \in A$.

$$\exists x \in A, \text{Pred}(x)$$

is true if Pred(x) is true for at least one value of $x \in A$.

Common sets

$\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$	integers
$\mathbb{N} = \{1, 2, \dots\}$	natural numbers
$\mathbb{N}_0 = \{0, 1, 2, \dots\} = \mathbb{Z}_+$	non-negative integers
$\mathbb{R} = (-\infty, +\infty)$	real numbers
$\mathbb{R}_+ = [0, \infty)$	non-negative real numbers
$\mathbb{C} = \{x + iy \| x, y \in \mathbb{R}\}$	complex numbers
Binary $= \{0, 1\}$	binary values
Bools $= \{$true, false$\}$	truth values
Char $= \{a, b, 1, 2, \dots\}$	all alphanumeric characters
ImNum $= \left\{iy \| y \in \mathbb{R}, i = \sqrt{-1}\right\}$	imaginary numbers

Interval notation also denote sets.

$$(\alpha, \beta] = \{x \in \mathbb{R} | \alpha < x \leq \beta\}$$

Set operations

$\cup$	union
$\cap$	intersection
$\wedge$	logical AND
$\vee$	logical OR
$\backslash$	set subtraction

AND of two predicates is a conjunction. OR of two predicates is a disjunction.

$$A^c = X \backslash A, A \subset X$$

A^c is the complement of A (all $x \in X \wedge x \notin A$), where $A^c \cup A = X$

Predicate operations

The counter part of the complement is the negation.

$\neg$	negation of predicate

$$\{x \in X | \neg\text{Pred}(x)\} = x \backslash \{x \in X | \text{Pred}(x)\}$$

$$\begin{aligned}
\{x \in X | \neg(P(x) \wedge Q(x))\} &= \{x \in X | \neg P(x)\} \cap \{x \in X | \neg Q(x)\}\\
\{x \in X | \neg(P(x) \vee Q(x))\} &= \{x \in X | \neg P(x)\} \cup \{x \in X | \neg Q(x)\}
\end{aligned}$$

de Morgan's rules:

$$\begin{aligned}
X \backslash (Y \cap Z) &= (X \backslash Y) \cup (X \backslash Z)\\
X \backslash (Y \cup Z) &= (X \backslash Y) \cap (X \backslash Z)\\
(Y \cap Z)^c &= Y^c \cup Z^c\\
(Y \cup Z)^c &= Y^c \cap Z^c
\end{aligned}$$

Combinations and permutation

The number of unordered subsets of length $m \leq n$ we can construct from a set of n elements is

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

is the number of combinations of m elements from n.

$$\frac{n!}{(n-m)!}$$

is the number of ordered subsets of length m we can construct from a set of length n.
The number of permutations is $\geq$ number of combinations.

Product sets

For sets $X_1, X_2, \dots, X_n$, an $n-$tuple is an order collection of one element from each set

$$(x_1, \dots x_n), x_i \in X_i \forall i \in \{1, 2, \dots, N\}$$

The product of N sets is the set consisting of $N-$tuples

$$\prod_{i=1}^{N} X_i = \{(x_1, \dots, X_n) | x_i \in X_i, i = 1, \dots, N\}$$

Functions

f is a rule that assigns a value in Y to each $x \in X$. $f(x)$ is an element in Y.

$$f : X \to Y$$

f is *one-to-one* if

$$\forall x_1 \in X \text{ and } \forall x_2 \in X, x_1 \neq x_2 \implies f(x_1) \neq f(x_2)$$

f is *onto* if

$$\forall y \in Y, \exists x \in X | f(x) = y$$

Unless specified, a function $\{f(x) | x \in X\} \cap Y$ does not have to equal $\emptyset$.

Declarative assignment

Define we define $f : X \to Y$ by

$$f(x) = \text{expression in x } \forall x \in X$$

Is a declarative assignment. It gives the properties of the function without explaining how to construct the function.

Graphs

We can define a function by defining a its graph.

$$\text{graph}(f) = \{(x, f(x)) | x \in X\}$$

Tables

When $f : X \to Y$ has a finite domain, then graph$(f) \subset X \times Y$. We can define f by a table of all its inputs and outputs.

Procedure

Unlike declarative assignments, procedures gives a constructive method ot determine some $y \in Y$ for some $x \in X$. This is called imperative assignment.
The statement `y = Math.sin(x)/x` defines a function

$$\text{Statement} : (\text{doubles} \times \text{doubles}) \to (\text{doubles} \times \text{doubles})$$

since the program state is specified by the value of `x` and `y` which are doubles.

Composition

Given that the connection restriction holds for $f_1 : X \to Y$ and $f_2 : X' \to Y'$:

$$Y' \subset X'$$

then the composition $f_3 : X \to Y'$ is such that

$$f_3(x) = (f_2 \circ f_1)(x) = f_2(f_1(x)) \forall x \in X$$

Induction

Given the proposition: "statements $S_1, S_2, S_3, \ldots$" are all true. Mathematical induction follows:

1. Proof the first statement, S_1, is true (base case)
2. Given any integer $k \geq 1$, prove that the statement S_k tp S_{k+1} is true (inductive step)

by induction, every S_n is true.

Systems

A system is a function that transforms or generates signals.

$$S : [X \to Y] \to [G \to F]$$

$[X \to Y] = \{x | x : X \to Y\}$ is the space of all functions x that maps from the set X to Y. $S(x) \in [G \to F], \forall x \in [X \to Y]$.
The number of elements in $[X \to Y]$, where X has m elements, and Y has n elements is

$$n^m$$

Behaviour

For $S : [D \to R] \to [X \to Y]$, suppose $x \in [D \to R]$, and $y = S(x)$. (x, y) a behaviour of the system. The Set of all behaviours defines the system

$$\text{Behaviours}(S) = \{(x, y) | x \in [D \to R] \text{ and } y = S(x)\}$$

Memoryless systems

The output of the memoryless system at time t only depends on the input at time t.
A system $S : [X \to Y] \to [X \to Y']$ is memoryless iff $\exists$ a function $f : Y \to Y'$ such that $\forall t \in X$ and $\forall x \in [X \to Y]$

$$[F(x)](t) = f(x(t))$$

Composition of graphs

For systems S_1 and S_2 expressed as their graphs,

$$\text{graph}(S_1) = \{(x, y_1) \in X \times Y | y_1 = S_1(x)\}$$
$$\text{graph}(S_1) = \{(y_2, z) \in Y \times Z | z = S_2(x)\}$$

The graph of $S = S_2 \circ S_1$ is

$$\text{graph}(S) = \{(x, z) \in X \times Z | \exists y_1, \exists y_2,$$
$$(x, y_1) \in \text{graph}(S_1) \wedge (y_2, z) \in \text{graph}(S_2) \wedge y_1 = y_2\}$$

State machines

State machines describe a system by giving a sequence of step by steps operations for the evolution (or state transition) of the system. The state contains all information needed by the system to produce an output.

Structure of state machines

Input signals and output signals maps from non-negative natural number to a arbitrary set. Each natural number represents ordering, but not time. Time between state transitions need not be even.

$$\text{EventStream} : \mathbb{N}_0 \to \text{Symbols}$$

A state machine is a 5-tuple:

$$\text{StateMachine} = (\text{States}, \text{Inputs}, \text{Outputs}, \text{update}, \text{initialState})$$

This is called a "sets and functions" model.

Trace and state response

The state response of a system are the series of state transitions that a system goes through, given a set of inputs.
The trace of a system shows the state response and the input. For example:

$$\text{state1} \overset{\text{input1}}{\to} \text{state2} \overset{\text{input2}}{\to} \text{state3} \ldots$$

Update

Given $s(n) \in$ State at step n, and the input signal $x(n)$, the update function

$$(s(n+1), y(n)) = \text{update}(s(n), x(n))$$

We can decompose the update function into

$$\text{nextState} : \text{States} \times \text{Inputs} \to \text{States}$$
$$\text{output} : \text{States} \times \text{Inputs} \to \text{Outputs}$$

such that

$$(s(n+1), y(n)) = (\text{nextState}(s(n), x(n)), \text{output}(s(n), x(n)))$$

When there are a finite set of states and inputs, we can express the update function as a table. The first column lists the current state.

```
             | input 1              | ... |
---------------------------------------------
state 1      | (nextState 1, output 1) | ... |
---------------------------------------------
state 2      | (nextState 2, output 2) | ... |
```

Stuttering

By convention, we input the symbol "absent" within our input and output sets, such that

$$\text{update}(s, \text{absent}) = (s, \text{absent})$$

State transition diagrams

Finite state machines are be represented as a collection of nodes and arcs. Each node is labelled with a state. Each node is connected with arcs. Each arc is labelled with

$$\{\text{input symbol}\}/\text{output symbol}$$

Some arcs are labelled with "else". "else" is the set of input symbols that are not already specified by other arcs emerging from the current node.

Mealy machine

Current output is directly determined by current input.

Moore machine

Current output is the current state.

Composing state machines

Systems are functions, so composition of functions are function compositions. But since state machines are a procedural description of systems, they cannot be written a single function composition.

Defining a state machine

To define a composite machine requires defining the sets and function model for the entire system, and the

- update functions for each component machines

Synchrony

Composing components involves feeding the outputs of one component into another. Synchrony is a style of composition where every component of the composite machine reacts simultaneously and instantaneously.

Input symbols are simultaneous with output symbols

Side by side composition

Defines a single statemachine representing the synchronous operation of two component state machines. This naturally leads to product form input/outputs. For component systems A and B:

$$\begin{aligned}
\text{States} &= \text{States}_A \times \text{States}_B \\
\text{Inputs} &= \text{Inputs}_A \times \text{Inputs}_B \\
\text{Outputs} &= \text{Outputs}_A \times \text{Outputs}_B \\
\text{InitialState} &= (\text{initialState}_A, \text{InitialState}_B)
\end{aligned}$$

The update function of the composed system is

$$\begin{aligned}
&((s_A(n+1), s_B(n+1)), (y_A(n), y_B(n))) = \\
&\text{update}((s_A(n), s_B(n)), (x_A(n), xs_B(n))),
\end{aligned}$$

where

$$\begin{aligned}
(s_A(n+1), y_A(n)) &= \text{update}_A(s_A(n), x_A(n)) \\
(s_B(n+1), y_B(n)) &= \text{update}_B(s_B(n), x_B(n))
\end{aligned}$$

Cascade composition

For 2 component state machines A and B, cascade composition involves feeding the outputs of A as the input of B. This requires

$$\text{Outputs}_A \subset \text{Inputs}_B$$

And

$$\begin{aligned}
\text{States} &= \text{States}_A \times \text{States}_B \\
\text{Inputs} &= \text{Inputs}_A \\
\text{Outputs} &= \text{Outputs}_B \\
\text{InitialState} &= (\text{initialState}_A, \text{initialState}_B)
\end{aligned}$$

The update function is

$$\begin{aligned}
&((s_A(n+1), s_B(n+1)), y_B(n)) = \\
&\text{update}((s_A(n), s_B(n)), x(n)),
\end{aligned}$$

where

$$\begin{aligned}
(s_A(n+1), y_A(n)) &= \text{update}_A(s_A(n), x(n)) \\
(s_B(n+1), y_B(n)) &= \text{update}_B(s_B(n), y_A(n))
\end{aligned}$$

The output y_A is instantaneously available to update$_B$. But we must first determine y_A before passing it to update$_B$.

Product-from inputs and outputs

Product form inputs and outputs help us represent a state machine that takes input from multiple places in a system, and sends its output to multiple places. Each input is called a port, and the set of inputs at that port is the port alphabet.

$$\begin{aligned}
\text{Inputs} &= \text{Inputs}_A \times \text{Inputs}_B \\
\text{Outputs} &= \text{Outputs}_A \times \text{Outputs}_B
\end{aligned}$$

This is different from side-by-side systems since its it does not require product-form states.
Its important to include a *stutter* self-loop at each state. We might compose this system with others, we might not want it undergo a state transition via an *else* path.

Feedback

The output symbol in a feedback state machine depends on the input which dependent on its output. The solutions to a feedback equation are called fixed points.

$$x = f(x)$$

For a reachable state, a system with no fixed points, or multiple fixed points is ill-formed. A system with a single unique fixed point in each reachable state is well-formed.

Feedback composition with no inputs

We provide an artificial input alphabet:

$$\text{Inputs} = \{\text{react}, \text{absent}\}$$

react is a symbol to tell system to undergo the next state transition.
The update function is

$$\begin{cases} \text{update}(s(n), y(n)) & \text{if } x(n) = \text{react} \\ (s(n), x(n)) & \text{if } x(n) = \text{absent} \end{cases}$$

For the state machine to be well-formed there must be a single unique non-stuttering fixed point to the output function

$$y(n) = \text{ouput}(s(n), y(n))$$

for every state.

State-determined output

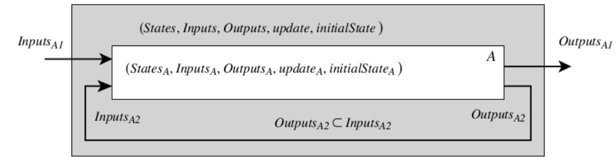
A state machine has state-determined output if in every reachable state, $s(n) \in$ States, there is a unique output symbol independent of the non-stuttering input.

A system with state-determined outputs are always well-formed

State-determined output machines combined with other state machines always have well-formed feedback compositions

Non-state determined machines can still be well-formed in a feedback composition.

Feedback composition with inputs



Is a state machine with a hidden feedback input.

$$\begin{aligned}
\text{Inputs}_A &= \text{Inputs}_{A_1} \times \text{Inputs}_{A_2} \\
\text{Outputs}_A &= \text{Outputs}_{A_1} \times \text{Outputs}_{A_2}
\end{aligned}$$

We require

$$\text{Outputs}_{A_2} \subset \text{Inputs}_{A_2}$$

The output function is

$$\begin{aligned}
\text{output}_A &: \text{States}_A \times \text{Inputs}_A \to \text{Outputs}_A \implies \\
\text{output}_{A_1} &: \text{States}_{A_1} \times \text{Inputs}_{A_1} \to \text{Outputs}_{A_1} \\
\text{output}_{A_2} &: \text{States}_{A_2} \times \text{Inputs}_{A_2} \to \text{Outputs}_{A_2}
\end{aligned}$$

We want to find the unknown output symbols $(y_1(n), y_2(n))$ at reaction n given $s(n)$, such that

$$\begin{aligned}
\text{output}_A(s(n), (x_1(n), y_2(n))) = (y_1(n), y_2(n)) \implies \\
\text{output}_{A_1}(s(n), (x_1(n), y_2(n))) = y_1(n) \\
\text{output}_{A_2}(s(n), (x_1(n), y_2(n))) = y_2(n)
\end{aligned}$$

The state machine is well-formed if for every reachable state $\exists$ a single non-stuttering output $y_2(n)$ that solves

$$\text{output}_{A_2}(s(n), (x_1(n), y_2(n))) = y_2(n)$$

Constructive procedure for feedback composition

1. Set all unspecified signals as unknown
2. Using what is given, try each state machine to determine something about the output symbols
3. Update what you know about unknowns
4. Repeat

Linear systems

We require

- State space, input, output are numeric sets $\implies$

$$\begin{aligned}
\text{States} &= \mathbb{R}^N \\
\text{Inputs} &= \mathbb{R}^M \\
\text{Outputs} &= \mathbb{R}^K
\end{aligned}$$

- The update function is linear

For $M, K > 1$, we have a multiple input multiple output system (MIMO). When $M = K = 1$, we have a single input single output system (SISO).

Time

We associate n with the notion of time. Given a even time interval of δ, step n occurs at time $n\delta$. Since a physical system must take on some physical value at each n, absent is disallowed.

$$\text{InputSignals} = [\mathbb{Z}_+ \to \mathbb{R}^M]$$
$$\text{OutputSignals} = [\mathbb{Z}_+ \to \mathbb{R}^K]$$

Linear functions

A function is linear it has the superposition property:

$$\forall u, v \in \mathbb{R}^N, \forall a, b \in \mathbb{R}, f(au + bv) = af(u) + bf(v)$$

A linear function $f : \mathbb{R}^N \to \mathbb{R}^M$ acting on a vector can be written as matrix vector multiplication.

The $[A, B, C, D]$ representation

For a linear system, we require

$$\underline{s}(0) = \underline{0}$$

$\underline{s}(0)$ is also called the rest state (equilibrium) of the system. We can write the nextState and output functions as

$$\underline{s}(n+1) = \underline{A}\underline{s}(n) + \underline{B}\underline{x}(n),$$
$$\underline{y}(n) = \underline{C}\underline{s}(n) + \underline{D}\underline{x}(n),$$

where $\underline{A}$ is a $N \times N$ matrix, $\underline{B}$ is a $N \times M$ matrix, $\underline{C}$ is a $K \times N$ matrix, and $\underline{D}$ is a $K \times M$ matrix.

Impulse response

The impulse response is a complete description of LTI systems when $M = 1$ (Inputs $\in \mathbb{R}^1$). The state response is to a signal

$$\delta_{n0} = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

is

$$\begin{aligned} \underline{s}(0) &= \underline{0} \\ \underline{s}(1) &= \underline{B} \\ \underline{s}(2) &= \underline{AB} \\ &\vdots \\ \underline{s}(n) &= \underline{A}^{n-1}\underline{B} \end{aligned}$$

The response of the system (denoted as $\underline{h}$ instead of $\underline{y}$) is

$$\underline{h}(n) = \begin{cases} 0 & n < 0 \\ \underline{D} & n = 0 \\ \underline{CA}^{n-1}\underline{B} & n \geq 1 \end{cases}$$

1-D SISO systems

$[A, B, C, D] \to [a, b, c, d]$ are constants in this case. Repeatedly applying $s(n+1) = as(n) + bx(n)$, the state response is

$$s(n) = a^n s(0) + \sum_{m=0}^{n-1} a^{n-1-m} bx(m)$$

Using $y = cs + dx$, we have

$$y(n) = ca^n s(0) + \left[\sum_{m=0}^{n-1} ca^{n-1-m} bx(m)\right] + dx(n)$$

Zero-state and zero-input response

- When the initial state $s(0) = 0$, the state response of the system is called the **zero-state state response**.
- When $x(n) = 0 \forall n \in \mathbb{Z}$, its called the **zero-input state response**.
- When $s(0) = 0$, the **zero-state output response** is

  $$y(n) = \sum_{-\infty}^{\infty} h(n-m)x(m),$$

  where h is the impulse response, $h(n < 0) = 0$. This is

  $$y(n) = h * x = x * h$$

- The **zero-input state response** is

  $$s(n) = a^n s(0)$$

  which converges to 0 for $n \to \infty$ iff $|a| < 1$.

Multidimensional SISO systems

A SISO system is multidimensional when States = $\mathbb{R}^N$ for some $N > 1$.

$$s(n+1) = \underbrace{\underline{A}}_{N\times N} s(n) + \underbrace{\underline{b}}_{1\times N} x(n)$$
$$y(n) = \underbrace{\underline{c}^\intercal}_{1\times N} s(n) + dx(n)$$

The state is

$$s(n) = \underline{A}^n s(0) + \sum_{m=0}^{n-1} \underline{A}^{n-1-m} bx(m)$$

The impulse response is

$$\underline{h}(n) = \begin{cases} 0 & n < 0 \\ \underline{D} & n = 0 \\ \underline{c^\intercal A}^{n-1} B & n \geq 1 \end{cases}$$

A system whose impulse response has a finite extent is a finite impulse response system (FIR).

Multidimensional MIMO systems

$$\underline{s}(n) = \underline{A}^n \underline{s}(0) + \sum_{m=0}^{n-1} \underline{A}^{n-1-m} \underline{Bx}(m)$$

There is a useful identity to compute power series. For $a \neq 1$:

$$\sum_{m=0}^{M} a^m = \frac{a^{1+M} - 1}{a - 1}$$

Continuous-time state-space models

For an LTI SISO system, the continuous-time state space model has the form: $\forall t \in \mathbb{R}_+$,

$$\dot{\underline{z}}(t) = \underline{A}\underline{z}(t) + \underline{b}v(t)$$
$$w(t) = \underline{c}^\intercal \underline{z}(t) + dv(t),$$

where

- $z : \mathbb{R}_t \to \mathbb{R}^N$ is the state response
- $\dot{z}(t)$ is vector of the time derivative of each component of z
- $v : \mathbb{R}_+ \to \mathbb{R}$ is the input signal
- $w : \mathbb{R}_+ \to \mathbb{R}$ is the output signal

We can approximate a continuous time system using finite difference quotients.

Difference equations

A general difference equation is the following: $\forall n \in \mathbb{N}_0$

$$y(n) + \sum_{i=1}^{N} a_i y(n-i) = \sum_{i=0}^{N} b_i x(n-i)$$

with initial conditions (when $n = 0$):

$$y(-1), y(-2), \ldots, y(-N)$$

> A difference equation system is LTI when
>
> $$x(n) = 0 \qquad \forall n < 0$$
>
> And all initial conditions $y(-1), \ldots, y(-N)$ are 0.

For the case $b_0 = 1, b_{i\neq 0} = 0$, we have a multidimensional SISO system:

$$y(n) + \sum_{i=1}^{N} a_i y(n-i) = x(n)$$

The state space representation of the system is

$$\underline{s}(n) = s_i \underline{e}_i = \begin{bmatrix} y(n-N) \\ \vdots \\ y(n-1) \end{bmatrix}$$

And (“controllable/controller canonical form”)

$$\underline{s}(n+1) = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & & 1 & 0 \\ -a_N & -a_{N-1} & -a_{N-2} & \ldots & -a_2 & -a_1 \end{bmatrix} \underline{s}(n) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} x(n)$$

with output function:

$$y(n) = \begin{bmatrix} -a_n & \cdots - a_1 \end{bmatrix} \underline{s}(n) + x(n)$$

LTI systems obey the superposition. Scaling and shifting the input scales and shifts the output by the same amount. The output for a superposition of inputs is the superposition of outputs for each input. Let

$$z(n) = x(n) - a_1 z(n-1) - \cdots - a_n z(n-N),$$

z is the output for a single $x(n)$, then for a superposition of scaled and shifted x:

$$b_0 x(n) + \cdots + b_N x(n-N) \to \underbrace{b_0 z_n + \cdots + b_N z(n-N)}_{y(n)}$$

$y(n)$ is the system output. Since each $z(n)$ is defined as

$$z(n) = \begin{bmatrix} -a_N & \cdots & -a_1 \end{bmatrix} \underline{s}(n) + x(n)$$

and each component of s is

$$s_i = z(n-(N+1)+i)$$

then

$$\begin{aligned} y(n) &= \underbrace{b_0 \begin{bmatrix} -a_N & \dots & a_1 \end{bmatrix} \underline{s}(n) + b_0 x(n)}_{\text{output for input } b_0 x(n)} + \begin{bmatrix} b_N & \cdots & b_1 \end{bmatrix} \underline{s}(n) \\ &= b_0 (\begin{bmatrix} -a_N & \dots & a_1 \end{bmatrix} \underline{s}(n) + x(n)) + \begin{bmatrix} b_N & \cdots & b_1 \end{bmatrix} \underline{s}(n) \\ &= \begin{bmatrix} b_N - b_0 a_N & \dots & b_1 - b_0 a_1 \end{bmatrix} \underline{s}(n) + b_0 x(n) \end{aligned}$$

Similarity transform

The state space $\underline{s}(n)$ can be thought of as being mapped from $\underline{\xi}(n)$ through a transformation $\underline{T}$

$$\underline{s}(n) = \underline{T}\,\underline{\xi}(n)$$

$[\underline{A}, \underline{B}, \underline{C}, \underline{D}]$ corresponding to $\underline{s}(n)$ is $[\underline{T}^{-1}\underline{AT}, \underline{T}^{-1}\underline{B}, \underline{CT}, \underline{D}]$ to $\underline{\xi}(n)$.

Solving state space equations

Solving the state space equations requires us to compute matrix powers. We can do this via spectral decomposition, or via the Cayley-Hamilton theorem.

Spectral decomposition

We factor a matrix in terms of it eigenvalue and eigenvectors. Let $\underline{A}$ be a $n \times n$ matrix with n linearly independent eigenvectors. Then

$$\underline{A} = \underline{PDP}^{-1},$$

where $\underline{P}$ is an $n \times n$ matrix whose i^{th} column corresponds to the i^{th} eigenvector of $\underline{A}$; $\underline{D}$ is a diagonal matrix such that the D_{ii} entry is the i^{th} eigenvalue of $\underline{A}$.

$$\underline{A} = \underline{PDP}^{-1} \implies \underline{A}^n = \underline{PDP}^{-1}\underline{PDP}^{-1} \dots \underline{PDP}^{-1}$$

We can collect $\underline{P}$ and its inverse and find that

$$\underline{A}^n = \underline{PD}^n \underline{P}^{-1},$$

where

$$\underline{D}^n = \begin{bmatrix} \lambda_1^n & 0 & & 0 \\ 0 & \lambda_2^n & & 0 \\ & & \ddots & \\ 0 & 0 & & \lambda_N^n \end{bmatrix}$$

Applying similarity transform do not change the eigenvalues ($\underline{D}$ does not change)

$$\underline{A} \to \underline{T}^{-1}\underline{AT} \implies \underbrace{\underline{T}^{-1}\underline{P}}_{\tilde{\underline{P}}}\,\underline{D}^n\,\underbrace{\underline{P}^{-1}\underline{T}}_{\tilde{\underline{P}}^{-1}}$$

Cayley Hamilton theorem

An $n \times n$ matrix $\underline{A}$ satisfies its own characteristic equation; if

$$(-1)^n \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1 \lambda + c_0 = 0$$

is the characteristic equation of $\underline{A}$ (from $\det(\underline{A} - \underline{I}\lambda) = 0$), then the following also holds:

$$(-1)^n \underline{A}^n + c_{n-1}\underline{A}^{n-1} + \cdots + c_1 \underline{A} + c_0 = \underline{0}$$

Any m^{th} power of a $n \times n$ matrix ($m > n$) $\underline{A}$ can be expresses as a linear combination of $\underline{A}^k$ for $k \le n$:

$$\underline{A}^m = c_0 \underline{I} + c_1 \underline{A} + c_2 \underline{A}^2 + \cdots + c_{n-1}\underline{A}^{n-1},$$

where the coefficients depend on m. Knowing n eigenvalues $(\lambda_1, \dots, \lambda_n)$ of $\underline{A}$, then we can solve a system of $n \times n$ equations for the coefficients:

$$\begin{aligned} \lambda_1^m &= c_0 + c_1 \lambda_1 + \cdots + c_{n-1}\lambda_1^{n-1} \\ &\vdots \\ \lambda_{n-1}^m &= c_0 + c_1 \lambda_{n-1} + \cdots + c_{n-1}\lambda_{n-1}^{n-1} \\ \lambda_n^m &= c_0 + c_1 \lambda_n + \cdots + c_{n-1}\lambda_n^{n-1} \end{aligned}$$

Controllability

Is the ability to drive the system from any initial state to any other final state in a finite time interval.

$$\begin{aligned} s(n) &= \underline{A}^n s(0) + \sum_{m=0}^{n-1} \underline{A}^{n-1-m} \underline{B}x(m) \\ &= \underbrace{[\underline{B}, \underline{AB}, \dots, \underline{A}^{n-1}\underline{B}]}_{\underline{H}} \begin{bmatrix} x(n-1) \\ \vdots \\ x(0) \end{bmatrix} \end{aligned}$$

The system with inputs $\underline{x} \in \mathbb{R}^r$ and states $\underline{s} \in \mathbb{R}^N$ is controllable iff

$$\text{rank}(\underline{H}) = N, \underline{H} \in \mathbb{R}^{N \times rN}$$

For a single input system, $\underline{B} \to b$, system is controllable iff

$$\det(\underline{H}) \neq 0$$

Observability

Is the ability to infer initial state from observing input and output in a finite time interval. By definition,

$$\underline{y}(n) = \underline{CA}^n \underline{s}(0) + \sum_{m=0}^{n-1} \underline{CA}^{n-1-m} \underline{Bx}(m) + \underline{Dx}(n)$$

Since we know the inputs, each observation is

$$\sum_{m=0}^{n-1} \underline{CA}^{n-1-m} \underline{Bx}(m) + \underline{Dx}(n)$$

so

$$\Delta \underline{y} = y(n) - \sum_{m=0}^{n-1} \underline{CA}^{n-1-m} \underline{Bx}(m) - \underline{Dx}(n) = \underline{CA}^n \underline{s}(0)$$

A linear system with states $\underline{s} \in \mathbb{R}^n$, and outputs $\underline{y} \in \mathbb{R}^p$, is observable iff the $np \times n$ matrix has rank n:

$$\text{rank}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\right) = n$$

For single output system, $\underline{C} \to c^\intercal$, the system is observable iff

$$\det\left(\begin{bmatrix} c^\intercal \\ c^\intercal A \\ \vdots \\ c^\intercal A^{n-1} \end{bmatrix}\right) = n$$

Stability

Things do not grow without bound. **When there is zero input and arbitrary initial state:**

The state/output of a system is marginally stable iff

$$\forall n \in \mathbb{N}_0 \qquad \|s(n)\| \le T_s < \infty$$

We require all eigenvalues of $\underline{A}$ are distinct and $|\lambda_i| \le 1$. If they are not distinct, then $|\lambda_i| < 1$.

The state/output of a system is asymptotically stable iff

$$\lim_{n \to \infty} \|s(n)\| = 0$$

We require all eigenvalues of $\underline{A}$ are $|\lambda_i| < 1$.

A discrete-time LTI system is BIBO stable iff its impulse response is absolutely summable.

- An asymptotically stable state-space system is BIBO stable
- If the system is controllable and observable, BIBO stability and asymptotic stability are equivalent

Continuous-time state-space models

Frequency Domain

- The field of communications if about synthesizing signals to match a channel, and analyzing signals to extract information.
- In frequency domain, signals are represented by sums of sinusoidal signals.

Frequency decomposition

We define *angular frequency* in radians per second:

$$\omega = 2\pi f$$

An *Octave* is a factor of two change in frequency (e.g. 440 Hz to 880 Hz is one octave, 880 to 1760 is another). Each frequency in the scale is $2^{1/12}$ times the frequency below it.
To a first approximation, the Timbre of musical instrument is due a linear combination of the fundamental frequency and its harmonics. Coefficients that determine the weight of each frequency is dependent on the instrument.

Phase

For the pure tone below, ϕ is the phase of signal:

$$g(t) = \sin(2\pi f t + \phi)$$

Changing the phase changes the relative starting point of the signal with respect to other signals.

Spatial frequency

Images have spatial frequency [cycles per unit distance]. The amplitude of the variation is intensity of the image.

$$\text{Image} : X \times Y \to \text{Intensity}$$

An $V \times H$ sized image that varies sinusoidally along x and y can be represented as

$$\forall x \in X, \forall y \in Y \ \text{Image}(x, y) = \sin(2\pi x/V) \times \sin(2\pi y/H)$$

Periodic and finite signals

A continuous time periodic signal with period p is defined as

$$x(t) = x(t + np), \ \forall t, p \in \mathbb{R}, \forall n \in \mathbb{Z}$$

When we restrict the domain of the signal of a subset of $\mathbb{R}$, we get a finite signal:

$$x'(t) = \begin{cases} x(t) & t \in [a, b] \\ 0 & t \notin [a, b] \end{cases}$$

We can define a periodic signal of period $p = b - a$ from a finite signal by a “shift-and-add summation":

$$x(t) = \sum_{m=-\infty}^{\infty} y'(t - mp)$$

The sum of two periodic signals is periodic iff there exists $k_1, k_2 \in \mathbb{Z}_+$ such that

$$k_1 p_1 = k_2 p_2$$

Fourier series

A Fourier series expansion can be used to approximate periodic signals.

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \cos(k\omega_0 t + \phi_k)$$

ω_0 is the fundamental frequency of the signal: $2\pi/p$. The terms with $k \geq 2$ are the harmonics of the series: they have an integer multiple of the fundamental frequency.

Discrete-time signals

A discrete-time signal is periodic with period $p > 0$, $p \in \mathbb{Z}$ if

$$x(n + p) = x(n) \ \forall n \in \underline{Z}$$

Not all sinusoidal signals in discrete time are periodic.

$$x(n) = \cos(2\pi f n) \implies x(n + p) = \cos(2\pi f n + 2\pi f p)$$

which is periodic only if $f p$ is an integer. So we require f to be a rational number (f is a fraction of two integers). In general,

$$p = K/f,$$

for the smallest integer $K > 0, K \in \underline{Z}$ such that K/f is an integer.
The discrete-time Fourier series is

$$x(n) = A_0 + \sum_{k=1}^{K} A_k \cos(k\omega_0 n + \phi_k)$$

for

$$K = \begin{cases} (p-1)/2 & p \text{ is odd} \\ p/2 & p \text{ is even} \end{cases}$$

Frequency response

Given a sinusoidal input, LTI systems output a sinusoid with the same frequency, but possibly altered phase and amplitude.

LTI systems

Time-domain signals are functions of a variable representing time (regardless of whether it is continuous or discrete).

Time invariance

Let's define the delay system, D_τ, such that

$$\forall t \in \mathbb{R}, y(t) = D_\tau[x(t)] = x(t - \tau)$$

A system S is time-invariant if

$$\forall \tau \in \mathbb{R}, S \circ D_\tau = D_\tau \circ S$$

This is equivalent to

$$S[D_\tau[x]] = D_\tau[S[x]]$$

For any input x that produces an output y, a delayed input $D_\tau(x)$ produces an output $D_\tau(y)$.

Linearity

A system, S, is linear if

$$S(ax_1 + bx_2) = aS(x_1) + bS(x_2)$$

Linearity and time invariance

Consider a continuous-time signal given by the complex exponential:

$$\forall t \in \mathbb{R}, x(t) = e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$$

From the property of exponentials, a delayed complex exponential is a scaled complex exponential:

$$D_\tau[x] = e^{-i\omega\tau} e^{i\omega t} = ax$$

We can show complex exponentials are eigenfunctions to LTI systems:

$$S(e^{i\omega t}) = H(\omega) e^{i\omega t},$$

where $H : \mathbb{R} \to \mathbb{C}$ is a property of the system called the frequency response.
A discrete time exponential is given by

$$\forall n \in \mathbb{Z}, x(n) = e^{i\omega n}$$

But since $\forall n \in \mathbb{Z}, \exp(i\theta) = \exp(i(2\pi n + \theta))$,

$$\forall \omega \in \mathbb{R}, H(\omega) = H(\omega + 2K\pi)$$

for $K \in \mathbb{Z}$.

Finding and using the frequency response

Consider a general linear difference equation: $\forall n \in \mathbb{Z}$,

$$\begin{aligned} &a_0 y(n) + a_1 y(n-1) + \cdots + a_N y(n-N) \\ &= b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) \end{aligned}$$

We know that for all $k \in [0, N]$,

$$y(n-k) = H(\omega)\exp(i\omega(n-k))$$

Substituting in:

$$\begin{aligned} &a_0 H(\omega) e^{i\omega n} + \cdots + a_N H(\omega) e^{i\omega(n-N)} \\ &= b_0 e^{i\omega n} + \cdots + b_M e^{i\omega(n-M)} \end{aligned}$$

Factoring out $H(\omega)$ and cancelling $\exp(i\omega n)$, we get

$$H(\omega) = \frac{b_0 + b_1 e^{i\omega} + \cdots + b_M e^{-i\omega M}}{a_0 + a_1 e^{i\omega} + \cdots + a_N e^{-i\omega N}}$$

Consider a general linear differential equation: $\forall t \in \mathbb{R}$

$$\begin{aligned} &a_N \frac{d^N y}{dt^N}(t) + \cdots + a_1 \frac{dy}{dt}(t) + a_0 y(t) \\ &= b_M \frac{d^M x}{dt^M}(t) + \cdots + b_1 \frac{dx}{dt}(t) + b_0 x(t) \end{aligned}$$

The coefficients are real, or complex constants. Given $x(t) = \exp(i\omega t)$, the output for all t is $y(t) = H(\omega) e^{i\omega t}$. Since

$$\frac{d^k}{dt^k} e^{i\omega t} = (i\omega)^k e^{i\omega t}$$

Substituting in:

$$\begin{aligned} &a_N (i\omega)^N H(\omega) e^{i\omega t} + \\ &\cdots + a_1 (i\omega) H(\omega) e^{i\omega t} + a_0 H(\omega) e^{i\omega t} \\ &= b_M (i\omega)^M e^{i\omega t} + \cdots + b_1 (i\omega) e^{i\omega t} + b_0 e^{i\omega t} \end{aligned}$$

Factoring cancelling $\exp(i\omega t)$ on both sides and isolating:

$$H(\omega) = \frac{b_M (i\omega)^M + \cdots + b_1 (i\omega) + b_0}{a_N (i\omega)^N + \cdots + a_1 (i\omega) + a_0}$$

Real valued system

For a real valued LTI system (system that can only produce real outputs), given the input $x = \cos(x)$, we have conjugate symmetry in the frequency response, and

$$H(\omega) = H^*(-\omega)$$

and the output is

$$y(t) = \text{Re}\{H(\omega)e^{i\omega t}\}$$

We can express $H(\omega)$ as

$$H(\omega) = |H(\omega)|e^{i\angle H(\omega)}$$

$|H(\omega)|$ is the magnitude response, and $\angle H(\omega)$ is the phase response. The output is

$$\forall t \in \mathbb{R},\, y(t) = |H(\omega)|\cos(\omega t + \angle H(\omega))$$

Fourier series with complex exponentials

For a general complex valued periodic signal $x : \mathbb{R} \to \mathbb{C}$, of period p, in continuous time:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{ik\omega_0 t},$$

and $\omega_0 = 2\pi/p$. The coefficients are given by

$$X_k = \frac{1}{p}\int_p x(t)e^{jk\omega_0 t}\, dt$$

For a general complex valued periodic signal $x : \mathbb{Z} \to \mathbb{C}$, of period p, in discrete time:

$$x(n) = \sum_{k=0}^{p-1} X_k e^{ik\omega_0 n}$$

Frequency response of composite systems

For a cascade connection,

$$\forall \omega \in \mathbb{R} \qquad H(\omega) = H_2(\omega)H_1(\omega)$$

For a feedback connection,

$$H(\omega) = \frac{H_1(\omega)}{1 - H_1(\omega)H_2(\omega)}$$

Filtering

LTI systems are often considered filters. They may attenuate some frequencies while enhancing others. (But they do not introduce new frequencies.)

Convolution

The frequency response of a system tells us what the system do, but give no information about how the system actually works.

Convolution sum and integral

The convolution operator $(*)$ have different meaning when consider discrete or continuous time signals.

- Commutativity: $x * y = y * x$
- Linearity: $x * (a_1 y_1 + a_2 y_2) = a_1(x * y_1) + a_2(x * y_2)$

Let $x, y \in [\mathbb{Z} \to \mathbb{R}]$ be two discrete-time signals. The convolution of x, y is a signal $x * y$, given by

$$\forall n \in \mathbb{Z} \qquad (x * y)(n) = \sum_{k=-\infty}^{\infty} x(k)y(n-k)$$

We can define LTI systems using convolution. A system S defined by

$$S(x) = h * x$$

Is linear since convolution is linear, and time invariant since

$$\begin{aligned} D_\tau(h * x) &= h * D_\tau(x) \\ &= \sum_{k=-\infty}^{\infty} h(k)x(n-\tau-k) \qquad \text{let} z = D_\tau(x) \\ &= \sum_{k=-\infty}^{\infty} h(k)z(n-k) \\ &= h * z \\ &= h * D_\tau(x) \end{aligned}$$

Let $x, y \in [\mathbb{R} \to \mathbb{R}]$ be two continuous time signals. The convolution of x and y is

$$\forall t \in \mathbb{R} \qquad (x * y)(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)\, dt$$

Signals as sums of weighted delta functions

Any discrete-time signal $x : \mathbb{Z} \to \mathbb{R}$ can be expressed as a sum of weighted Kronecker delta functions

$$\forall n \in \mathbb{Z} \qquad x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) = (x * \delta)(n)$$

Any continuous time signal be can written as integrals of weighted Dirac delta functions

$$\forall t \in \mathbb{R} \qquad x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)\, d\tau = (x * \delta)(n)$$

Impulse response and convolution

The impulse response of an LTI system is the output of system given an input signal δ.

$$\forall n \in \mathbb{Z} \qquad h(n) = [S(\delta)]\,(n)$$

Given an input $x(n) = (x * \delta)(n)$, the output

$$\begin{aligned} [S(x * \delta)]\,(n) &= [x * S(\delta)]\,(n) \\ &= [x * h]\,(n) \end{aligned}$$

Similarly, for a continuous time signal, we define the impulse response as

$$h(t) = [S(\delta)]\,(t)$$

And,

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\, d\tau = (x * h)(t)$$

So

the output of any LTI system is the convolution of the input signal with the impulse response.

Frequency response and impulse response

Consider an input signal $x(n) = \exp(i\omega n)$. The output can be written in two ways

$$\begin{aligned} H(\omega)e^{i\omega n} = (h * x)(n) &= \sum_{k=-\infty}^{\infty} h(k)e^{i\omega(n-k)} \\ &= e^{i\omega n}\sum_{k=-\infty}^{\infty} h(k)e^{-i\omega k} \end{aligned}$$

Clearly,

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-i\omega k}$$

This is the **discrete-time Fourier transform (DTFT)** of the impulse response.

The frequency response of a discrete-time system is the DTFT of the impulse response.

- If h is real, then $H^*(-\omega) = H(\omega)$
- DTFT is periodic with period 2π. $H(\omega + 2\pi) = H(\omega)$

In continuous time,

$$H(\omega)e^{-i\omega t} = \int_{-\infty}^{\infty} h(\tau)e^{-i\omega(t-\tau)}\, d\tau$$

Gives

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-i\omega t}\, dt$$

Is the **continuous time Fourier transform (CTFT or FT)**.

The frequency response of a continuous-time system is the CTFT of the impulse response.

- If h is real, then $H^*(-\omega) = H(\omega)$

Causality

A causal system only depends on input at the current time or before.

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

is causal when $h(k) = 0$ for all $k < 0$.

Decibels

If the gain of a filter at frequency ω is $|H(\omega)|$, then the gain is

$$G(\omega) = 20\log_{10}(|H(\omega)|) \implies |H(\omega)| = 10^{G(\omega)/20}$$

The four Fourier transforms

We denote the space of all continuous-time signals as

$$\text{ContSignals} = [\mathbb{R} \to \mathbb{C}]$$

The space of all discrete-time signals as

$$\text{DiscSignals} = [\mathbb{Z} \to \mathbb{C}]$$

Periodic signals are subsets

$$\text{ContPeriodic}_p \subset \text{ContSignals}$$
$$\text{DiscPeriodic}_p \subset \text{DiscSignals}$$

Fourier transform: A formula that gives the frequency-domain function in terms of the time-domain function.

Inverse Fourier transform: A formula that gives the time domain function in terms of the frequency domain function

The Fourier series (FS)

Describes $x \in \text{ContPeriodic}_p$ as a sum weighted sum fo complex exponentials

$$\forall t \in \mathbb{R} \qquad x(t) = \sum_{m=-\infty}^{\infty} X_m e^{im\omega_0 t}$$

where $\omega_0 = 2\pi/p$. The Fourier series coefficients are given by

$$\forall m \in \mathbb{Z} \qquad X_m = \frac{1}{p}\int_0^p x(t)e^{-im\omega_0 t}\,dt$$

$$\text{FourierSeries}_p : \text{ContPeriodic}_p \to \text{DiscSignals}$$
$$\text{InverseFourierSeries}_p : \text{DiscSignals} \to \text{ContPeriodic}_p$$

- Given x explicitly as a sum of sinusoids, the fundamental frequency f_0 is the largest common factor that divide $f_1, \ldots, f_n$

If the signal is real valued, then

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \cos(k\omega_0 t + \phi_k)$$

the coefficients are

$$X_k = \begin{cases} A_0 & k = 0 \\ A_k e^{i\phi_k}/2 & k > 0 \\ A_{-k}e^{-i\phi_{-k}}/2 & k < 0 \end{cases}$$

Coefficients are conjugate symmetric:

$$X_k = X_{-k}^*$$

The discrete Fourier transform (DFT)

The discrete Fourier series (DFS) for $x \in \text{DiscPeriodic}_p$ is

$$\forall n \in \mathbb{Z} \qquad x(n) = \sum_{k=0}^{p-1} X_k e^{ik\omega_0 n}$$

where $\omega_0 = 2\pi/p$. The Fourier series coefficients are

$$\forall k \in \mathbb{Z} \qquad X_k = \frac{1}{p}\sum_{m=0}^{p-1} x(m)e^{-imk\omega_0}$$

The discrete-time Fourier series is

$$x(n) = A_0 + \sum_{k=1}^{K} A_k \cos(k\omega_0 n + \phi_k)$$

for

$$K = \begin{cases} (p-1)/2 & p \text{ is odd} \\ p/2 & p \text{ is even} \end{cases}$$

The discrete Fourier transform (DFT) is

$$\forall n \in \mathbb{Z} \qquad x(n) = \frac{1}{p}\sum_{k=0}^{p-1} X'_k e^{ik\omega_0 n}$$

where

$$\forall k \in \mathbb{Z} \qquad X'_k = \sum_{m=0}^{p-1} x(m)e^{-imk\omega_0}$$

The DFT and DFS coefficients are related by

$$X'_k = pX_k$$

- The DFT of a signal is a signal that is it self periodic $X' = \text{DFT}_p(x)$

The expression for X'_k in terms of $x(m)$ looks like a (DFT), differing by a single negative sign.

$$\text{DFT}_p : \text{DiscPeriodic}_p \to \text{DiscPeriodic}_p$$
$$\text{InverserDFT}_p : \text{DiscPeriodic}_p \to \text{DiscPeriodic}_p$$

The discrete-Time Fourier transform (DTFT)

The discrete-Time Fourier transform (DTFT) for any $x \in \text{DiscSignals}$ is

$$\forall \omega \in \mathbb{R} \qquad X(\omega) = \sum_{m=-\infty}^{\infty} x(m)e^{-i\omega m}$$

$X \in \text{ContPeriodic}_{2\pi}$, is continuous and 2π periodic. It has an inverse given by

$$\forall n \in \mathbb{Z} \qquad x(n) = \frac{1}{2\pi}\int_a^b X(\omega)e^{i\omega n}\,d\omega,$$

where b and a can be chosen as long as $b - a = 2\pi$.

The continuous-time Fourier transform

CTFT (also known as simply FT), takes $x \in \text{ContSignals}$, and outputs continuous function $X(\omega)$:

$$\forall \omega \in \mathbb{R} \qquad X(\omega) = \int_{-\infty}^{\infty} x(t)e^{i\omega t}\,dt$$

The inverse:

$$\forall t \in \mathbb{R} \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{i\omega t}\,d\omega$$

Fourier transforms v.s. Fourier series

- Fourier series only apply to periodic signals
- Fourier transforms apply to any signal

Let $y \in \text{DiscSignals}$, and $x \in \text{ContSignals}_p$, where

$$\forall n \in \mathbb{Z} \qquad x(n) = \sum_{m=-\infty}^{\infty} y'(n - mp)$$

and

$$\forall n \in \mathbb{Z} \qquad y'(n) = \begin{cases} y(n) & n \in [0, p-1] \\ 0 & \text{otherwise} \end{cases}$$

The DTFT of y' is related to the DFT of x by

$$X_k = Y'(k\omega_0)$$

The DFT of a periodic signal is equal to samples of the DTFT of a finite signal with sampling interval $\omega_0 = 2\pi/p$

When we do not have a closed form expression for a signal, the DFT reveals the structure of the DTFT

When $x \in \text{ContPeriodic}_p$, then FS gives the weights in the CTFT, within a scaling factor of 2π.

$$\forall \omega \in \mathbb{R} \qquad X(\omega) = 2\pi\sum_{m=-\infty}^{\infty} X_m\delta(\omega - m\omega_0)$$

Sampling and reconstruction

A discrete-time signal is constructed by sampling a continuous-time signal. A continuous-time signal is reconstructed by interpolating a discrete-time signal.

Sampling

Define a system, Sampler_T:

$$\text{Sampler}_T : [\mathbb{R} \to \mathbb{C}] \to [\mathbb{Z} \to \mathbb{C}]$$

1. T is the sampling period or sampling interval, with units seconds per sample
2. $f_s = 1/T$ is the sampling frequency or sample rate, with units samples per second, or Hz

For $x \in$ ContSignals, $y = \text{Sampler}_T(x)$ gives

$$\forall n \in \mathbb{Z} \qquad y(n) = x(nT)$$

The DTFT of y is related to the CTFT of x by

$$Y(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega - 2\pi k}{T}\right)$$

Relative to $X(\omega)$, the graph of $X(a\omega)$ appears

- compressed if $a > 1$
- stretch if $a < 1$

$Y(\omega)$ is the sum of CTFT of x with copies shifted by $k2\pi/T$ and normalized.

Aliasing does not occur if $X(\omega) = 0$ outside of

$$-\pi/T < \omega < \pi/T$$

This means that in the range $-\pi < \omega < \pi$,

$$Y(\omega) = \frac{1}{T} X\left(\frac{\omega}{T}\right)$$

Sampling a sinusoid

The sampled version of a real valued cosine is

$$\forall n \in \mathbb{Z} \qquad y(n) = \cos(2\pi f nT)$$

In the the sampled version, f cannot be distinguished from $f + f_s$.

$$\begin{aligned}\cos(2\pi(f + f_s)nT) &= \cos(2\pi f nT + 2\pi f_s nT)\\ &= \cos(2\pi f nT + 2\pi n)\\ &= \cos(2\pi f nT)\end{aligned}$$

The frequency is $2\pi f T$, in radians per sample. This is also the *normalized frequency* since it does not depend on T.

Avoiding aliasing ambiguities

Half of the sampling frequency is called the Nyquist frequency.

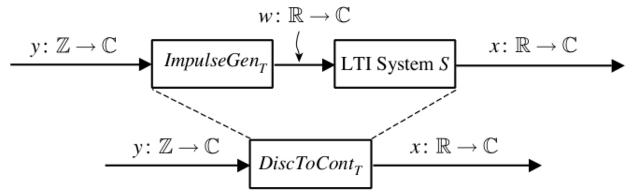
Reconstruction

Three possible definitions of a system DiscToCont_T

$$\text{DiscToCont}_T : [\mathbb{Z} \to \mathbb{C}] \to [\mathbb{R} \to \mathbb{C}]$$

1. Zero-order hold: $x(t) = y(n)$ for $t = nT$ to $t = (n+1)T$
2. Linear interpolation: straight lines connect the samples
3. Ideal interpolation: a continuous, smooth curve passes though all points in the sample

A model for reconstruction



An interpolator system takes a discrete time signal, yx, a returns a continuous time signal. y first passes through ImpulseGen, then an LTI system S

ImpulseGen_T takes a discrete-time signal y (the sampled signal) and outputs a continuous-time signal of weighted Dirac delta functions

$$w(t) = \sum_{n=-\infty}^{\infty} y(n)\,\delta(t - nT)$$

The CTFT of $w(t)$ is

$$W(\omega) = Y(\omega T)$$

The LTI system S that comes next in the cascade has impulse response:

1. Zero-order hold:

$$h(t) = \begin{cases} 1 & 0 \le t < T \\ 0 & \text{otherwise} \end{cases}$$

2. Linear interpolation:

$$h(t) = \begin{cases} 1 + t/T & -T < t < 0 \\ 1 - t/T & 0 \le t < T \\ 0 & \text{otherwise} \end{cases}$$

3. Ideal interpolation:

The impulse response

$$h(t) = \frac{\sin(\pi t/T)}{\pi t/T}$$

Frequency response

$$H(\omega) = \begin{cases} T & -\pi/T < \omega < \pi/T \\ 0 & \text{otherwise} \end{cases}$$

The Nyquist-Shannon sampling theorem

If x is a continuous-signal, with Fourier transform X, and if $X(\omega)$ is zero outside the range $-\pi/T < \omega < \pi/T$ radians per second, then

$$x = \text{IdealInterpolator}_T(\text{Sampler}_T(x))$$

$$w(t) = \text{IdealInterpolator}_T(\text{Sampler}_T(x))$$

$$W(\omega) = \begin{cases} \sum_{k=-\infty}^{\infty} X(\omega - \frac{2\pi k}{T}) & -\pi/T < \omega < \pi/T \\ 0 & \text{otherwise} \end{cases}$$

If a signal only contains frequencies less than $f_s/2$, then it can be perfected constructed by samples taken at a frequency f_s.

Stability

- **Unstable system**: output can grow without bound even if the input is bounded.
- The Z-transform allows us to analyze discrete-time unstable systems where DTFT does not exist.
- the Z-transform of the impulse response is called the **transfer function**

Boundedness and stability

A signal $x(n)$, $n \in \mathbb{Z}$ is bounded when there exists a real number M, $M < \infty$, such that

$$|x(n)| \le M \qquad \forall n \in \mathbb{Z}$$

For $x \in$ DiscSignals, if

$$\sum_{n=-\infty}^{\infty} |x(n)|$$

exists, then it's DTFT exists and is finite for all ω.

For $x \in$ ContSignals, if

$$\int_{-\infty}^{\infty} |x(n)|\, dt$$

exists, and

- x has a finite number of max and min for any finite interval
- x is discontinuous for a finite number of points

then it's CTFT exists and is finite for all ω. (**Dirichlet conditions**)

Using the triangle inequality, $\|a + b\| \le \|a\| + \|b\|$, we have

$$\forall \omega \in \mathbb{R} \qquad |X(\omega)| \le \sum_{n=-\infty}^{\infty} |x(n)|$$

Stability

A system if BIBO (bounded input bounded output) stable if the output signal is bounded for all bounded input signals.

> A discrete/continuous-time LTI system is BIBO stable iff
> - the input and output signals are bounded (the definition)
>
> Equivalently,
> - The system's impulse response is absolutely summable (integrable)
>
> A continuous-time LTI system is BIBO stable iff its impulse response is absolutely integrable.

> Useful geometric series identity. The sum converges for $a \in \mathbb{C}$, $|a| < 1$.
> $$\sum_{m=0}^{\infty} a^m = \frac{1}{1-a}$$

The Z-transform

For $r \in \mathbb{R}, r \geq 0$, and a signal x that is not absolutely summable,

$$\forall n \in \mathbb{Z} \qquad x_r(n) = x(n) r^{-n}$$

maybe absolutely summable. The DTFT of x_r exists, and $\forall \omega \in \mathbb{R}$,

$$\begin{aligned} X_r(\omega) &= \sum_{m=-\infty}^{\infty} x(m) r^{-m} e^{-i\omega m} \\ &= \sum_{m=-\infty}^{\infty} x(m) \left(r e^{i\omega}\right)^{-m} \end{aligned}$$

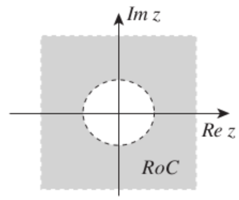
> Let $z = r\exp(i\omega)$, the Z-transform of x is
> $$\forall z \in \text{RoC}(x) \qquad \hat{X}(z) = \sum_{m=-\infty}^{\infty} x(m) z^{-m}$$
> where
> $$\begin{aligned} \text{RoC}(x) = \{&z = r\exp(i\omega) \in \mathbb{C} | \\ &x_r \text{ (or } x(n)z^{-n}) \text{ is absolutely summable}\} \end{aligned}$$

When $x_r(n)$ is absolutely summable, then

$$\begin{aligned} \sum |x(m) z^{-m}| &= \sum |x(m)||r^{-m}||e^{-i\omega m}| \\ &= \sum |x(m)||r^{-m}| < \infty \end{aligned}$$

Whether the series converges only depends on r, or $|z|$, and not on ω. So RoC are circularly symmetric.

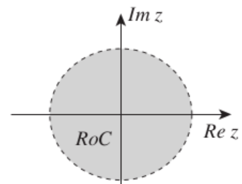
Causal signal



causal or right sided

If x is causal, then $x(n) = 0$ for all $n < 0$. The RoC of $\hat{X}(z)$ must also include $z \to \infty$.

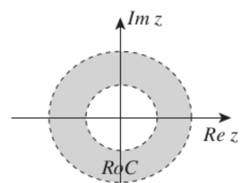
Anti-causal signal



anti-causal

A signal is anti-causal if $x(n) = 0$ for all $n > 0$.

Two-sided signal



two-sided

can always be expressed as a sum of causal and anti-causal signals. The RoC is the intersection of the two region of convergences.

Stability and the Z-transform

> When x **is absolutely summable**, then its DTFT is the Z-transform evaluated on the unit circle in the complex z plane
> $$\forall \omega \in \mathbb{R} \qquad X(\omega) = \hat{X}\left(e^{i\omega}\right)$$

> (Z-transform stability criterion): A discrete-time LTI system is stable iff its transfer function of a has a RoC that includes the unit circle.

Rational Z-transforms and poles and zeros

A **rational polynomial** is a ratio of two finite-order polynomials, $A(z)$ and $B(z)$.

$$\hat{X}(z) = \frac{A(z)}{B(z)}$$

The Z-transform of many signals are rational polynomials.

- **Zeros**: the roots of the numerator polynomial
- **Poles**: the roots of denominator polynomial

> (**Causal system stability criterion**): A causal system All the poles of the transfer function lie within the unit circle.

> Regions of convergence is always boarder by pole locations.

A proper rational polynomial has the higher order polynomial in the denominator.

> If the Z-transform of a causal signal is a rational polynomial, it must be proper.

Laplace and Z-transforms

> For the Z-transform of a real-valued signal, the poles and zeros occur in complex-conjugate pairs.
>
> If there is a pole at $z = q$, then there must be a zero at $z = q^*$.

Causal signals and the initial value theorem

> If x is a causal signal, then
> $$x(0) = \lim_{z\to\infty} \hat{X}(z)$$

Frequency response and pole-zero plots

We can estimate the magnitude response of a stable LTI system from a pole zero plot of its transfer function.

1. Begin at $\omega = 0$.
2. Trace counterclockwise around the unit circle as ω increases.
3. If you pass near a zero, then the magnitude response should dip.
4. If you pass near a pole, then the magnitude response should rise

The inverse transforms

We consider only the case where the Z-transform can be expressed as a rational polynomial.
When $\hat{X}(z)$ is some strictly proper polynomial, we can factor its denominator,

$$\hat{X}(z) = \frac{a_M z^M + \cdots + a_1 z + a_0}{(z-p_1)^{m_1}(z-p_2)^{m_2}\ldots(z-p_k)^{m_k}}$$

The partial faction expansion of $\hat{X}$ is

$$\hat{X}(z) = \sum_{i=1}^{k}\left[\frac{R_{i1}}{(z-p_i)} + \frac{R_{i2}}{(z-p_i)^2} + \cdots + \frac{R_{im_i}}{(z-p_i)^{m_i}}\right]$$

Linear difference and differential equation

Consider an LTI difference equation.

$$\begin{aligned} n \geq 0 \qquad y(n) + a_1 y(n-1) + \cdots + a_m y(n-m) \\ = b_0 x(n) + b_1 x(n-1) + \cdots + b_k x(n-k) \end{aligned}$$

We require

- system is initially at rest ($y(n<0)=0$ or $h(n<0)=0$)
- or M initial conditions

$$y(-1) = \overline{y}(-1)$$

$$\vdots$$

$$y(-M) = \overline{y}(-M)$$

We assume that the input signal starts at $n=0$, so $x(n<0)=0$.

When there are non-zero initial conditions (in the input or the output), then we need to account for the initial conditions in our Z-transform.

For example, if we know $\overline{y}(-1), \overline{y}(-2)$, then in transforming $u(n)y(n-1)$, we need to

$$\begin{aligned} \sum_{n=0}^{\infty} y(n-1)z^{-n} &= \overline{y}(-1)z^0 + z^{-1}\sum_{n=1}^{\infty} y(n-1)z^{-(n-1)} \\ &= \overline{y}(-1)z^0 + z^{-1}\hat{Y}(z) \end{aligned}$$

- The **zero-state response** is output when all initial conditions are zero. (Given by inverse Z-transform of $\hat{H}(z)\hat{X}(z)$)
- The **zero input state response** is the inverse Z-transform of the portion of the signal that involves the initial conditions but not the input signal

State-space models

We can use the Z-transform to find matrix powers.

The Z-transform of $\underline{A}^n u(n)$ is

$$\sum_{n=0}^{\infty} z^{-n}\underline{A}^n = z\left(z\underline{I} - \underline{A}\right)^{-1}$$

Starting with $\underline{A}$, we have all the information to find $z\left(z\underline{I}-\underline{A}\right)^{-1}$. We can then take its inverse Z-transform to find $\underline{A}^n u(n)$.

Since $\underline{A}^n u(n)$ is causal, the region of convergence is

$$\{z \in \mathbb{C} \mid |z| > |p|\}$$

p is the largest pole of $1/\det(z\underline{I} - \underline{A})$.

For a SISO system:

$$\begin{aligned} s(n+1) &= \underbrace{\underline{A}}_{N\times N}\ \underline{s}(n) + \underbrace{\underline{b}}_{1\times N}\ x(n) \\ y(n) &= \underbrace{\underline{c}^\intercal}_{1\times N}\ s(n) + dx(n) \end{aligned}$$

The transfer function is

$$\hat{H}(z) = \underline{c}^\intercal\left(z\underline{I} - A\right)^{-1}\underline{b} + d$$

The zero input response is

$$\hat{Y}_{zi}(z) = z\underline{c}^\intercal\left(z\underline{I} - \underline{A}\right)^{-1}s(0)$$