Signals, Systems, and Control

Dr. Edmund Lam

with help from Dr. Hayden So and Prof. Y.S. Hung

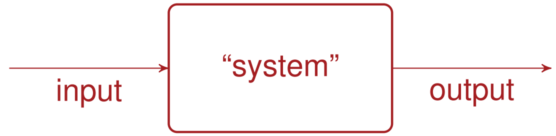
Department of Electrical and Electronic Engineering
The University of Hong Kong

ENGG1015: Introduction to Electrical and Electronic Engineering
(First Semester, 2012–13)

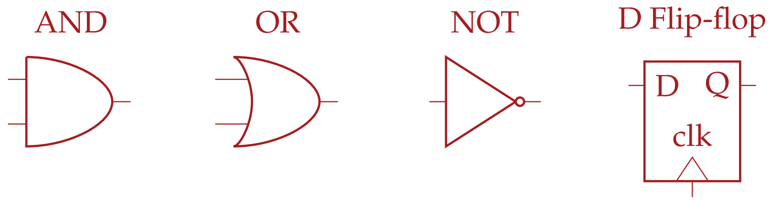
http://www.eee.hku.hk/~engg1015

Review: Digital Systems

1. System Diagram



2. Digital Logic Primitives

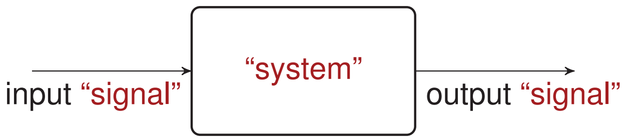


3. Schematics $\Longleftrightarrow$ Boolean expressions

$$y = a \cdot \bar{b} + \bar{a} \cdot b$$

Signals and Systems

The system diagram applies not only to digital logic *(input/output not necessarily binary)*.



- Also called a **block diagram**
- A **system** maps an **input signal** to an **output signal**
- A "signals and systems" abstraction
- Examples: Combinational logic system, computer system, . . .

Examples

Communication system

- **signal**: digitized voice (or image or video or text)
- **system**: consists of antenna, base stations, etc
- output = input? e.g. background noise filtering

Imaging system

- **signal**: light intensity
- **system**: consists of lenses, photodetectors
- output = input? e.g. lens distortion

Financial system

- **signal**: $$
- **system**: consists of lots of computers
- output = input? A complicated function!

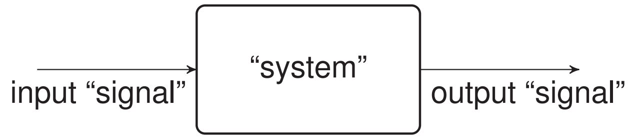
Biomedical system

- e.g. Electroencephalography, or EEG
- **signal**: electrical (voltage)
- **system**: consists of electrodes and other components
- output = input?

A "Signals and Systems" Viewpoint

- Very diverse applications
- A similar viewpoint: flow of information
- Abstraction: same "language" (mathematics) to deal with signals and systems
- **Good news**: what you learn here is very useful
  (or, why many electrical and electronic engineers are in Wall Street, Medicine, Business, Music, Psychology, Linguistics, . . .)

Discrete-time Systems



- A system can be very complicated
- Goal is to describe the system with simple "behaviors" or "rules"
- We restrict ourselves to discrete-time systems, i.e., input and output signals are a sequence of numbers
- *Conceptually, such discrete systems can be built by representing the numbers in binary and using lots and lots of combinational and sequential logic gates*

Four Different Representations

Objective: Develop different ways to represent and understand a discrete-time system

"Time-domain" methods:

- Signal flow graph
- Linear constant-coefficient difference equation (LCCDE)

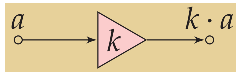
"Transform" methods:

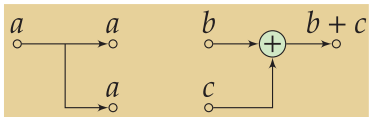
- Operator notation and arithmetics
- z-transform and transfer function

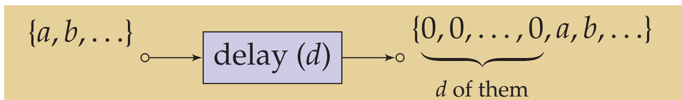
They are equivalent representations, but differ in their use.

#1: Signal flow graph

We *restrict* ourselves to three "primitives":

1. Multiplication (gain): 

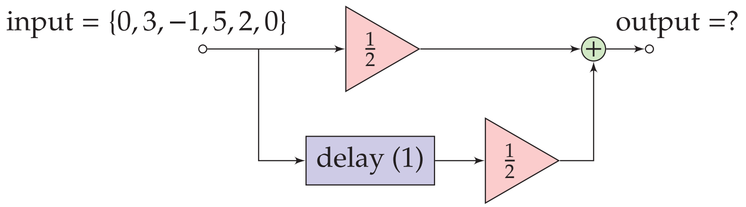
   (k can be integer, fraction, negative number. . .)
2. Split/add (adder): 

   (A value becomes two **identical** copies)
   (Two values added together)
3. Delay: 

   (The sequence is delayed by d **integer** units)

#1: Signal Flow Graph Example

Example (A):



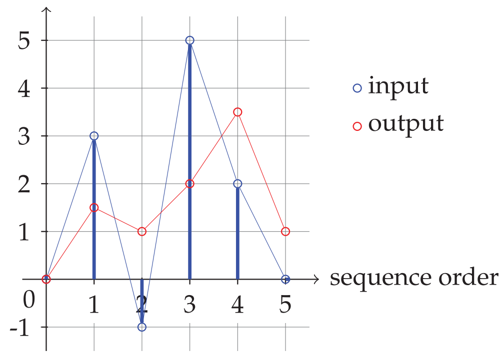
Assume: The system has no signal before the input.

$$0 \to \tfrac{1}{2}\cdot(0) = 0$$
$$3 \to \tfrac{1}{2}\cdot(3) + \tfrac{1}{2}\cdot(0) = \tfrac{3}{2}$$
$$-1 \to \tfrac{1}{2}\cdot(-1) + \tfrac{1}{2}\cdot(3) = 1$$
$$5 \to \tfrac{1}{2}\cdot(5) + \tfrac{1}{2}\cdot(-1) = 2$$
$$2 \to \tfrac{1}{2}\cdot(2) + \tfrac{1}{2}\cdot(5) = \tfrac{7}{2}$$
$$0 \to \tfrac{1}{2}\cdot(0) + \tfrac{1}{2}\cdot(2) = 1$$

Hence, output is $\{0, \frac{3}{2}, 1, 2, \frac{7}{2}, 1\}$.

#1: Signal Flow Graph Example

Example (A): (cont.)

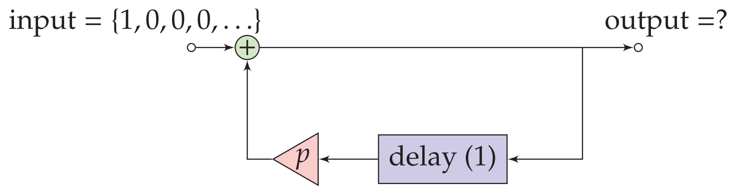


Observation: This output is a smoothed version of this input

Deduction: This discrete-time system achieves smoothing

#1: Signal Flow Graph Example

Example (B):



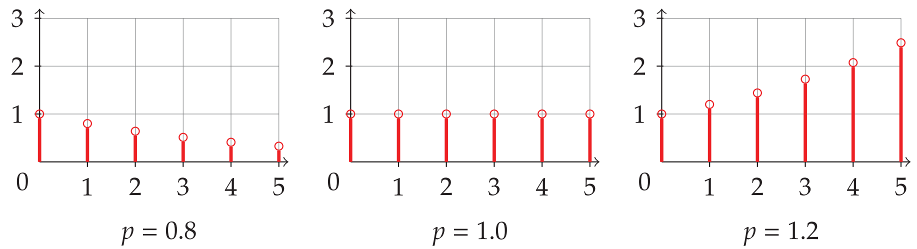
Assume: The system has no signal before the input.

$$1 \to 1 = 1$$
$$0 \to 0 + p\cdot(1) = p$$
$$0 \to 0 + p\cdot(p) = p^2$$
$$0 \to 0 + p\cdot(p^2) = p^3$$
$$0 \to 0 + p\cdot(p^3) = p^4$$

Hence, output is $\{1, p, p^2, p^3, \ldots\}$.

#1: Signal Flow Graph Example

Example (B): (cont.)



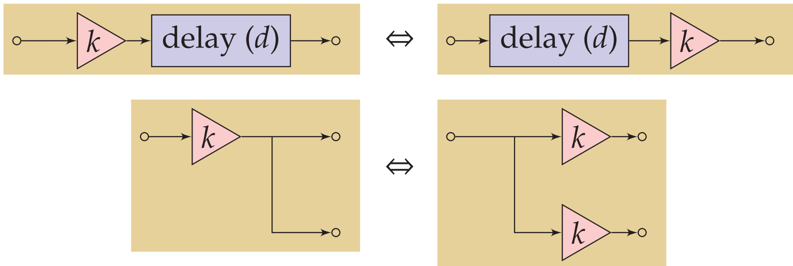
Observation: This output "decays" or "stays unchanged" or "grows without bound" for a unit input

Deduction: The behavior of this discrete-time system depends a lot on the value of p. This is extremely important for us later on!

#1: Signal Flow Graph Mini-conclusions

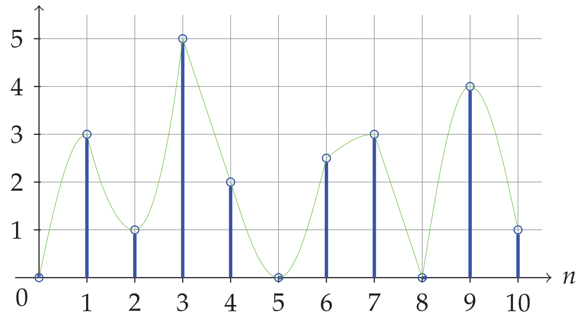
Some conclusions about flow graphs:

1. Given an input, we can "follow the flow" to deduce the output
2. Hardware implementation by putting in the appropriate components (assume we have them) according to the flow graph
3. It is *not always obvious* what the system achieves, but we can guess
4. Intuitively, some changes to the flow graphs are permitted:



#2: Difference Equations

Look at the input sequence as a discrete-time signal



- Data present at regular time intervals
- Often due to sampling of a continuous signal
- Some data are naturally discrete, e.g. daily stock price
- *For our purpose here, we'll just assume the input and output signals are discrete without worrying how they come to be*

#2: Difference Equations: The Mathematics

Conventions:

- Signal: $x[n]$ (square bracket)
- Often $n = 0, 1, \ldots N-1$ for a length-N signal.
- Assume $x[n] = 0$ outside this range.
- Use $x[n]$ for an input signal, $y[n]$ for an output signal

Alternatives:

1. Can have negative n, e.g., $n = -N, -N+1, \ldots N-1, N$ for a length $2N+1$ signal
2. Can have infinite length signal (conceptually): $n =$ all integers

#2: Difference Equations: The Mathematics

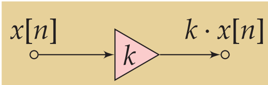
Exercise: Try plotting the following signals

1. $x[n] = C\alpha^n$, where $n = 0, \ldots, 99$.
   1. $\alpha = 0.99, C = 10$
   2. $\alpha = 1.00, C = 1$
   3. $\alpha = 1.01, C = 0.1$
2. $x[n] = A\cos(\omega n + \phi)$, where $n = 0, \ldots, 99$.
   1. $A = 1, \phi = 0, \omega = 0.01\pi$ (low frequency)
   2. $A = 1, \phi = 0, \omega = 0.1\pi$ (middle frequency)
   3. $A = 1, \phi = 0, \omega = \pi$ (high frequency)
   4. $A = 1, \phi = \pi/2, \omega = \pi$
3. $x[n] = \begin{cases} 1 & \text{at } n = 0 \\ 0 & \text{otherwise.} \end{cases}$
   - *This is the most important signal of all! It is called a delta function or a unit impulse, denoted by $\delta[n]$.*

#2: Difference Equations and Flow Graphs

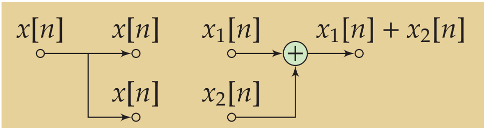
The flow graphs now operate on the *entire signal* (vector vs scalar)

1. Multiplication (gain):



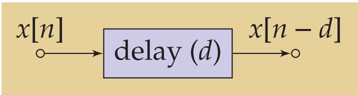
(*k* can be integer, fraction, negative number. . .)

2. Split/add (adder):



(A signal becomes two **identical** copies)
(Two signals added together)

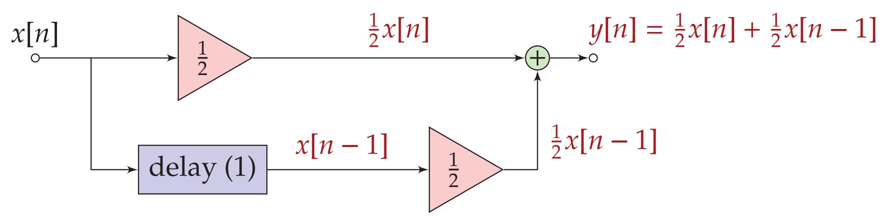
3. Delay:



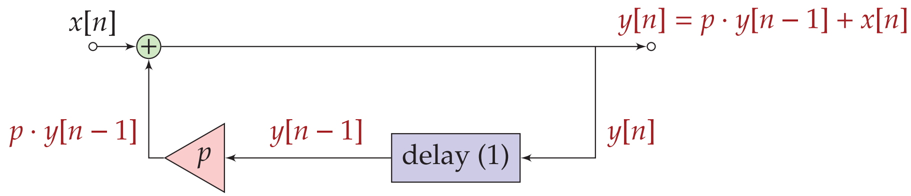
(A signal is delayed by *d* **integer** units)

#2: Difference Equations Example

Example (A):

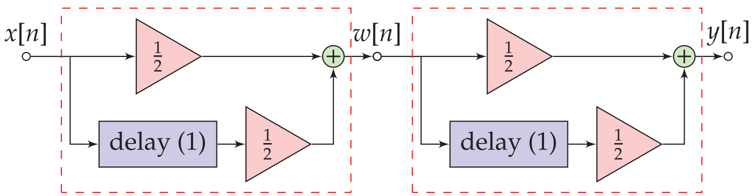


Example (B):



#2: Difference Equations Example

Example (C): The smoothing system in cascade



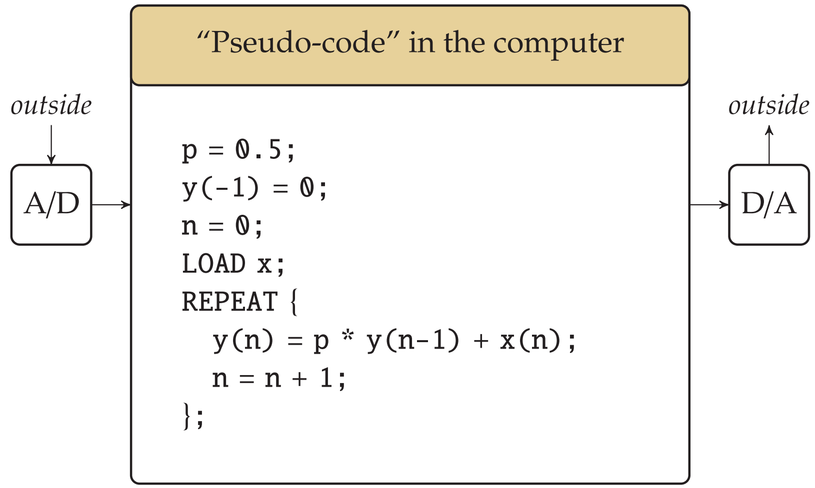
Tedious to go through the flow graph for *each point*. Make use of difference equations:

$$
\begin{aligned}
w[n] &= \frac{1}{2}x[n] + \frac{1}{2}x[n-1] \\
y[n] &= \frac{1}{2}w[n] + \frac{1}{2}w[n-1] \\
&= \frac{1}{4}x[n] + \frac{1}{4}x[n-1] + \frac{1}{4}x[n-1] + \frac{1}{4}x[n-2] \\
&= \frac{1}{4}\big(x[n] + 2x[n-1] + x[n-2]\big).
\end{aligned}
$$

Effect: further smoothing

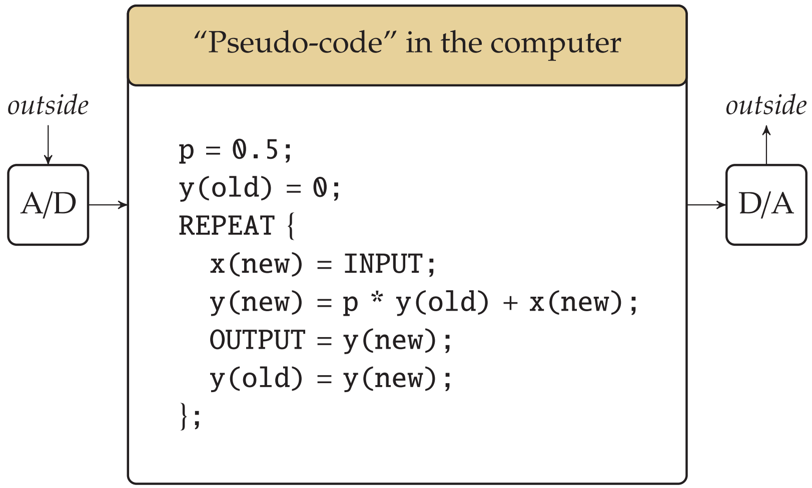
#2: Difference Equations Implementation

Software implementation: e.g. $y[n] = p \cdot y[n-1] + x[n]$



#2: Difference Equations Implementation

Real-time software implementation: e.g. $y[n] = p \cdot y[n-1] + x[n]$



#2: Difference Equations Mini-conclusions

Some conclusions about difference equations:

1. Flow graphs and difference equations are *equivalent*
   - Can go from flow graphs to difference equations
   - Can go from difference equations to flow graphs *(Do you know how?)*
2. They correspond to different ways of implementation
   - Flow graph is more "hardware"
   - Difference equation is more "software"
3. A general form of the difference equation:

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + \ldots + b_0 x[n] + b_1 x[n-1] + \ldots$$

Hence, the name "constant-coefficient"

#1–#2: Time-domain Conclusions

Both flow graphs and difference equations are "time-domain" methods

- We compute the results as time passes
- Generally speaking, these methods allow for easy computation of the output
- Not so good with understanding the system behavior

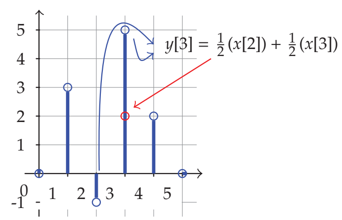
As a result, if we want to *analyze* (understand) and even *design* a discrete-time system, we need more advanced tools.

Next, we turn to "transform" methods.

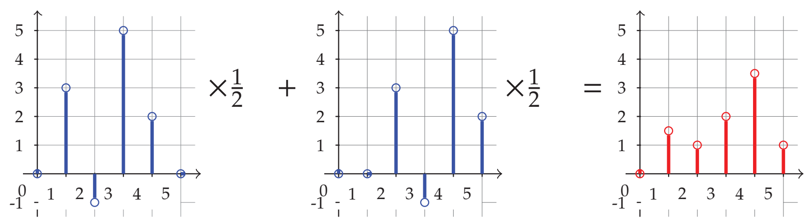
#3: Operator Notation and Arithmetics

Motivation: What does $y[n] = \frac{1}{2}x[n] + \frac{1}{2}x[n-1]$ mean?

1. Iterate for different n, e.g., for $n = 3$:



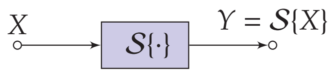
2. Scale and add the entire signal together, i.e.,



#3: Operator Notations

Assume we have the following:

1. X represents the *entire input signal* ($x[n]$ for all n).
2. Y represents the *entire output signal* ($y[n]$ for all n).
3. $Y = \mathcal{S}\{X\}$ represents an *operation*:



From X, operate on it (called $\mathcal{S}$), and output Y.

e.g. Can think about $\mathcal{S}$ in qualitative terms: take X, halving every value, and then add to a delayed version of X, also halving every value.

#3: Operator Notations

A very important operation is called a delay operation, written as $Y = \mathcal{D}\{X\}$. It means simultaneously the following:

$$y[0] = 0$$
$$y[1] = x[0]$$
$$y[2] = x[1]$$
$$\vdots$$

$$\boxed{y[n] = x[n-1]}$$

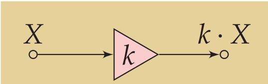
Often write this way instead of $y[n+1] = x[n]$.

In this specific case, we can omit the bracket and write $Y = \mathcal{D}X$. It *behaves* like ordinary multiplication: e.g. $\mathcal{D}(X_1 + X_2) = \mathcal{D}X_1 + \mathcal{D}X_2$.

#3: Operator Arithmetics

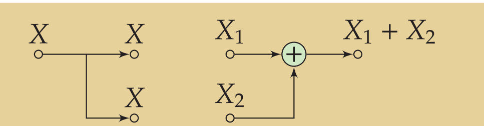
The flow graphs now acts as *operators*

1. Multiplication (gain):



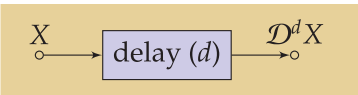
(k can be integer, fraction, negative number...)

2. Split/add (adder):



(Two **identical** copies / signals added together)

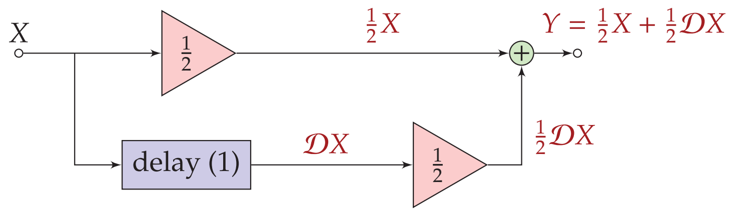
3. Delay:



(A signal is delayed by d **integer** units)
(We write $\mathcal{D}^d X$ to mean $\underbrace{\mathcal{D}\{\ldots\mathcal{D}}_{d \text{ of them}}\{\, X \,\}\ldots\})$

#3: Operator Example

Example (A):

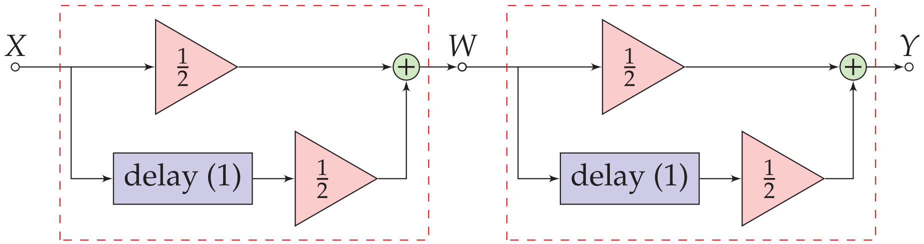


- We are allowed to write $Y = \frac{1}{2}(1 + \mathcal{D})X$. *How to interpret this?*
- There is a direct correspondence with $y[n] = \frac{1}{2}\big(x[n] + x[n-1]\big)$.

#3: Operator Example

We can tackle composite flow graphs using operator arithemtics.

Example (B): The smoothing system in cascade

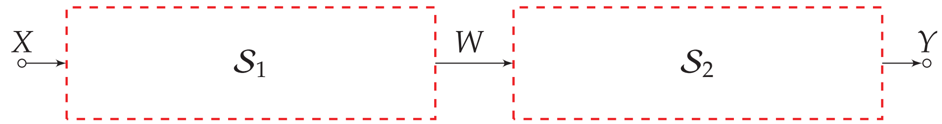


$$W = \tfrac{1}{2}(1+\mathcal{D})X$$
$$Y = \tfrac{1}{2}(1+\mathcal{D})W = \tfrac{1}{2}(1+\mathcal{D})\cdot\tfrac{1}{2}(1+\mathcal{D})X = \tfrac{1}{4}(1+2\mathcal{D}+\mathcal{D}^2)X$$

Each of the sub-system is $\mathcal{S} = \frac{1}{2}(1+\mathcal{D})$ and $Y = \mathcal{S}^2X$.

#3: Operator Example

We can generalize the cascade system to the following:



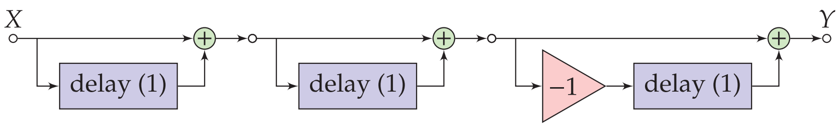
$$W = \mathcal{S}_1X$$
$$Y = \mathcal{S}_2\mathcal{S}_1X$$

This gives us a "hierarchical" way of analyzing a system: look at the "big picture", then more details of each sub-system, and further details of each sub-system, etc.

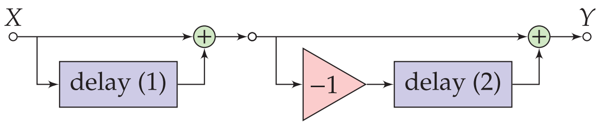
#3: Operator Example

Exercise: What are the respective operator equations?

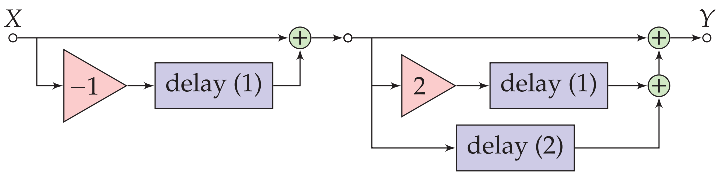
1



2



3

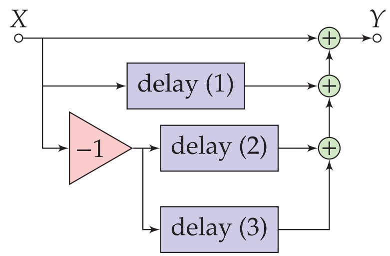


#3: Operator Example

These expressions are algebraically equivalent:

$$Y = (1+\mathcal{D})^2(1-\mathcal{D})X$$
$$Y = (1+\mathcal{D})(1-\mathcal{D}^2)X$$
$$Y = (1+2\mathcal{D}+\mathcal{D}^2)(1-\mathcal{D})X$$
$$Y = (1+\mathcal{D}-\mathcal{D}^2-\mathcal{D}^3)X$$

Hence the earlier three flow graphs are equivalent, as is the following:



#3: Operator: Output

Once we understand the system, we can compute the specific output when we are given a particular input.

Often, we are concerned with an impulse input: $x[n]$ is zero everywhere, except $x[0] = 1$.

Example: What's the output of the previous system with $x[n] = \delta[n]$?

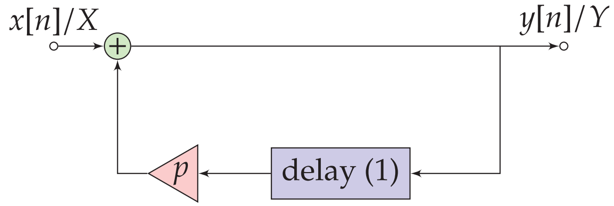
- $Y = X + \mathcal{D}X - \mathcal{D}^2X - \mathcal{D}^3X$;
- $\mathcal{D}^kX$ represents $\delta[n-k]$; therefore,
- Output is $y[n] = \delta[n] + \delta[n-1] - \delta[n-2] - \delta[n-3]$.

You can try tracing the flow graph or the difference equation to see if you can arrive at the same result.

#3: Operator Arithmetics for Feedback

Operation notation provides a powerful tool to analyze feedback, where output is "looped back" to the input.

- Flow graph:



- Difference equation: $y[n] = py[n-1] + x[n]$
- Operator equation: $Y = p\mathcal{D}Y + X$, or, $(1 - p\mathcal{D})Y = X$

#3: Operator Arithmetics for Feedback

If we can work on $\mathcal{D}$ algebraically, we can express Y in terms of X:

$$Y = \left(\frac{1}{1 - p\mathcal{D}}\right)X$$

What does it mean by "performing the delay operation in the denominator"?

Ans: Use the relationship

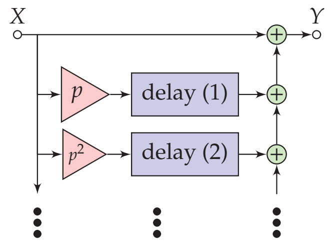
$$(1 - p\mathcal{D})(1 + p\mathcal{D} + p^2\mathcal{D}^2 + \ldots) = 1. \qquad \textit{(why?)}$$

Therefore,

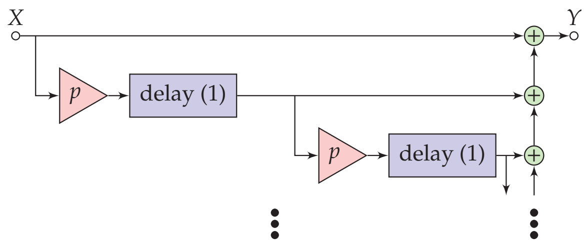
$$Y = (1 + p\mathcal{D} + p^2\mathcal{D}^2 + \ldots)X.$$

#3: Operator Arithmetics for Feedback

The corresponding flow graph:



Alternatively:



#3: Operator Arithmetics for Feedback

If $x[n] = \delta[n]$, what is $y[n]$?

Ans: $Y = X + p\mathcal{D}X + p^2\mathcal{D}^2X + \ldots$ *implies*

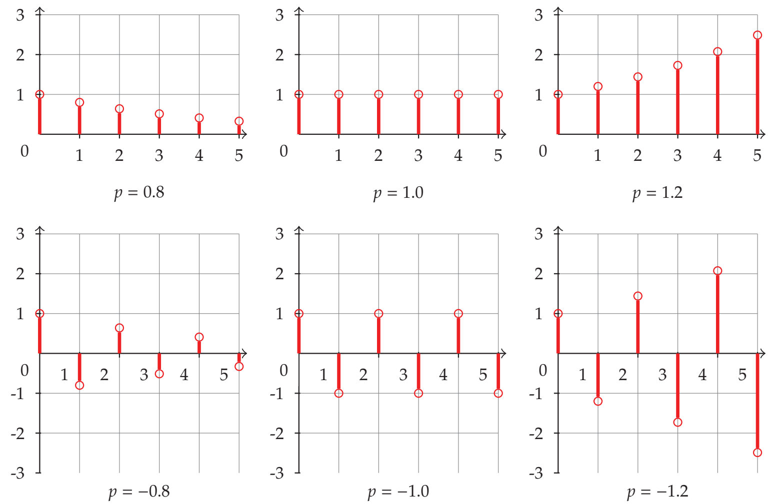
$$y[n] = \delta[n] + p\delta[n-1] + p^2\delta[n-2] + \ldots$$

We had the same result when we studied the flow graph before! For this feedback system, $\{1, 0, 0, \ldots\} \longrightarrow \{1, p, p^2, \ldots\}$.

Value of p determines whether the output in response to an impulse input is stable or not! In particular, three cases: $|p| < 1$, $|p| = 1$, $|p| > 1$.

#3: Operator Arithmetics for Feedback

"Stable" only if $|p| < 1$:



#3: Operator Mini-conclusions

We now have a good tool to analyze feedback systems.

- The key to understanding the behavior of a feedback system: think about *what happens to the signal when it goes through a loop or a cycle*
  - $|p| < 1$: signal weakens after the loop, so output decays
  - $|p| = 1$: signal magnitude remains the same after the loop, so output maintains
  - $|p| > 1$: signal is amplified after the loop, so output grows
- We are most concerned with an input that is a unit impulse: it's called the impulse response
  - Feedback gives rise to a *persistent response* with only a *transient input*
  - The system has a "similar" behavior as long as the input is of a finite duration. *(Why?)*

#4: z-transform and Transfer Function

Two (related) questions:

1. How to mathematically represent X that incorporates "the whole signal"?
2. How to mathematically represent the operations "multiplication", "addition", and "delay"?

A brilliant way: X is a polynomial where the coefficients are the various values of $x[n]$.

convention to use *negative* power

Example: $x[n] = \{3, 4, 1, -1\}$

$$X(z) = (3) + (4)z^{-1} + (1)z^{-2} + (-1)z^{-3}$$

X is now a polynomial in terms of z

#4: z-transform Notations

From $x[n]$ to $X(z)$, we call it the "z-transform":

$$X(z) = x[0] + x[1]z^{-1} + x[2]z^{-2} + x[3]z^{-3} + \dots$$

Using the summation symbol, we can write

$$\boxed{X(z) = \sum_{n=0}^{\infty} x[n]\, z^{-n}}$$

We "build" $X(z)$ by putting putting $x[n]$ as its coefficients; conversely, we can recover $x[n]$ by reading off the coefficients in $X(z)$.

#4: z-transform Notations

The z-transform notation makes it very convenient to represent *system operations*.

- A delay of one unit is equivalent to multiplication with z^{-1}, since

$$z^{-1}X(z) = \underbrace{x[0]z^{-1}}_{\text{value at time } n=1} + x[1]z^{-2} + x[2]z^{-3} + x[3]z^{-4} + \dots$$

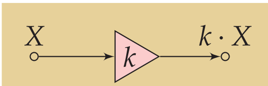
- A delay of d units is multiplication with z^{-d}.

This is very similar to the operator notation $\mathcal{D}$! You can think of z-transform as a "practical way" of realizing the operations.

#4: z-transform Arithmetics

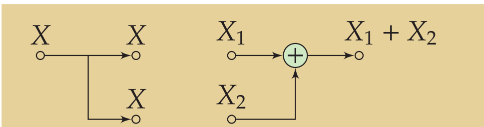
X, X_1, and X_2 are now z-transforms of the time-domain signals. We omit the "(z)" when there is no ambiguity.

1. Multiplication (gain):



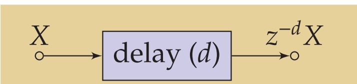
   (k can be integer, fraction, negative number...)

2. Split/add (adder):



   (Two **identical** copies / signals added together)

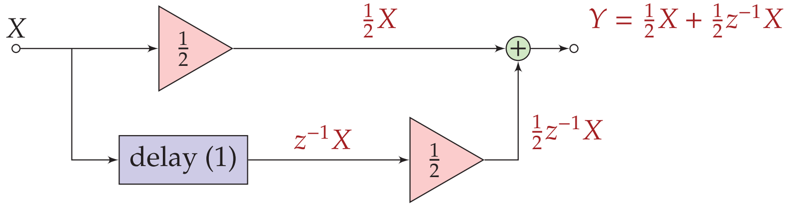
3. Delay:



   (A signal is delayed by d **integer** units)

#4: z-transform Example

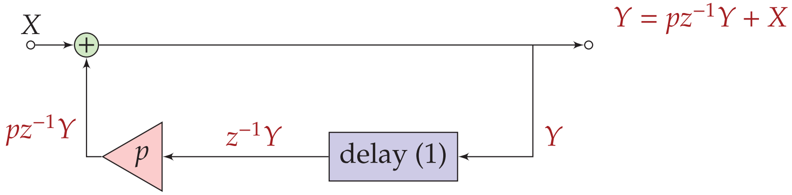
Example (A):



- We can write $Y = \frac{1}{2}\left(1 + z^{-1}\right)X$.
- There are direct correspondences with:
  - $Y = \frac{1}{2}(1 + \mathcal{D})\,X$
  - $y[n] = \frac{1}{2}\big(x[n] + x[n-1]\big)$

#4: z-transform Example

Example (B):

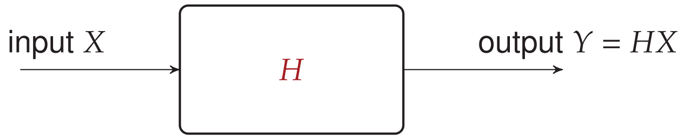


$$\left(1 - pz^{-1}\right) Y = X$$

$$Y = \frac{1}{1 - pz^{-1}} X$$

#4: Transfer Function

Define the transfer function $H(z) = \frac{Y(z)}{X(z)}$. (We'll also omit "(z)" later on)



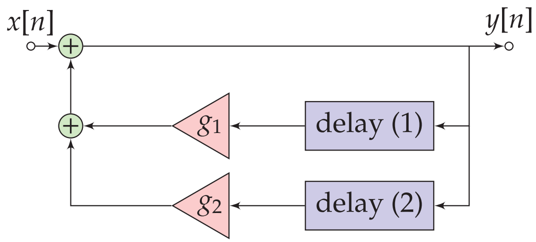
Example (A): $H = \frac{1}{2}\left(1 + z^{-1}\right)$.

Example (B): $H = \frac{1}{1 - pz^{-1}}$.

The transfer function "blows up" when $z = p$. Hence, we call p the "pole" of the system.

#4: z-transform Example

Example (C): A second-order feedback system, first with flow graph. Assume $g_1 = \frac{3}{4}$ and $g_2 = -\frac{1}{8}$:



$1 \to 1$

$0 \to 0 + \frac{3}{4}(1) = \frac{3}{4}$

$0 \to 0 + \frac{3}{4}(\frac{3}{4}) + (-\frac{1}{8})(1) = \frac{7}{16}$

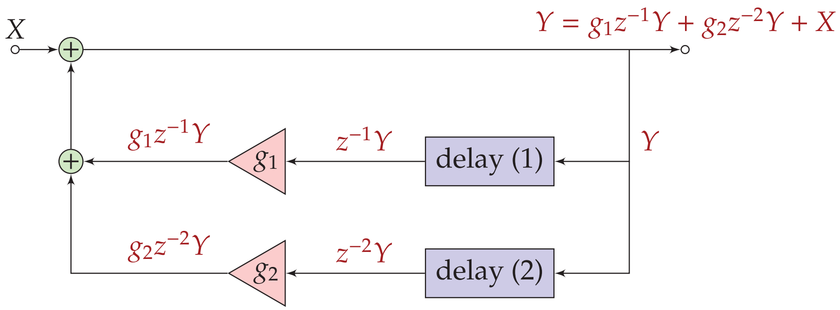
$0 \to 0 + \frac{3}{4}(\frac{7}{16}) + (-\frac{1}{8})(\frac{3}{4}) = \frac{15}{64}$

$0 \to 0 + \frac{3}{4}(\frac{15}{64}) + (-\frac{1}{8})(\frac{7}{16}) = \frac{31}{256}$

Do you see a pattern?

#4: z-transform Example

Using z-transform ($g_1 = \frac{3}{4}$, $g_2 = -\frac{1}{8}$):



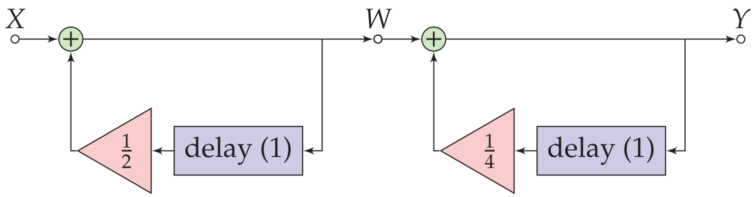
Algebra with polynomials provides a powerful tool for analysis!

$$\left(1 - \frac{3}{4}z^{-1} + \frac{1}{8}z^{-2}\right)Y = X$$

$$\left(1 - \frac{1}{2}z^{-1}\right)\left(1 - \frac{1}{4}z^{-1}\right)Y = X$$

#4: z-transform Example

We can convert the above second-order feedback system to the following equivalent form: cascade of two first-order feedback system

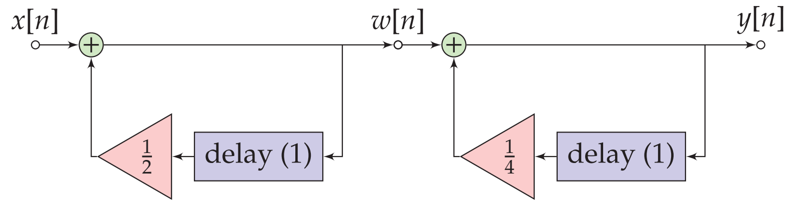


$$\left(1-\tfrac{1}{2}z^{-1}\right)W = X$$
$$\left(1-\tfrac{1}{4}z^{-1}\right)Y = W$$
$$\Longrightarrow \quad \left(1-\tfrac{1}{2}z^{-1}\right)\left(1-\tfrac{1}{4}z^{-1}\right)Y = X$$

A corollary: Can also interchange the order of the two systems

#4: z-transform Example

Verify with flow graph using $\{1, 0, 0, \ldots\}$ as input:



$$1 \rightarrow \quad 1 \quad \rightarrow 1$$
$$0 \rightarrow \quad 0+\tfrac{1}{2}(1)=\tfrac{1}{2} \quad \rightarrow \tfrac{1}{2}+\tfrac{1}{4}(1)=\tfrac{3}{4}$$
$$0 \rightarrow \quad 0+\tfrac{1}{2}(\tfrac{1}{2})=\tfrac{1}{4} \quad \rightarrow \tfrac{1}{4}+\tfrac{1}{4}(\tfrac{3}{4})=\tfrac{7}{16}$$
$$0 \rightarrow \quad 0+\tfrac{1}{2}(\tfrac{1}{4})=\tfrac{1}{8} \quad \rightarrow \tfrac{1}{8}+\tfrac{1}{4}(\tfrac{7}{16})=\tfrac{15}{64}$$
$$0 \rightarrow \quad 0+\tfrac{1}{2}(\tfrac{1}{8})=\tfrac{1}{16} \quad \rightarrow \tfrac{1}{16}+\tfrac{1}{4}(\tfrac{15}{64})=\tfrac{31}{256}$$

Do you see a pattern?

#4: z-transform Example

Further algebra reveals yet another form!

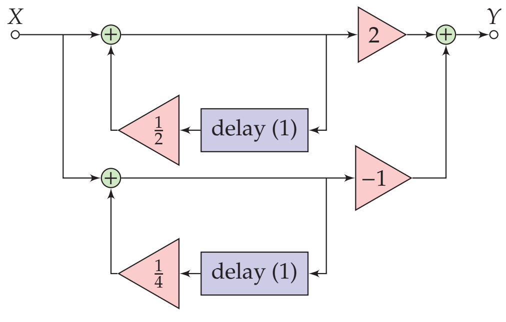
$$Y = \frac{1}{\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{1}{4}z^{-1}\right)}X$$
$$= \left(\frac{2}{1-\frac{1}{2}z^{-1}} + \frac{-1}{1-\frac{1}{4}z^{-1}}\right)X \qquad \textit{(why?)}$$
$$= \left(\frac{1}{1-\frac{1}{2}z^{-1}}\right)\cdot(2)\cdot X + \left(\frac{1}{1-\frac{1}{4}z^{-1}}\right)\cdot(-1)\cdot X$$

Two first-order feedback system in parallel. Poles are at $\frac{1}{2}$ and $\frac{1}{4}$.

The mathematics is called partial fraction.

#4: z-transform Example

Another equivalent form: two first-order feedback system in parallel



$$1 \rightarrow 2(1) - 1(1) = 1$$
$$0 \rightarrow 2[0+\tfrac{1}{2}(1)] - 1[0+\tfrac{1}{4}(1)] = \tfrac{3}{4}$$
$$0 \rightarrow 2[0+\tfrac{1}{2}(\tfrac{1}{2})] - 1[0+\tfrac{1}{4}(\tfrac{1}{4})] = \tfrac{7}{16}$$
$$0 \rightarrow 2[0+\tfrac{1}{2}(\tfrac{1}{4})] - 1[0+\tfrac{1}{4}(\tfrac{1}{16})] = \tfrac{15}{64}$$
$$0 \rightarrow 2[0+\tfrac{1}{2}(\tfrac{1}{8})] - 1[0+\tfrac{1}{4}(\tfrac{1}{64})] = \tfrac{31}{256}$$

Do you see a pattern?

#4: z-transform Example

Remember that for a first-order feedback system with gain p, an impulse input ($x[n] = \delta[n]$) gives an output

$$y[n] = p^n$$

Hence for our second-order feedback system, we can analytically represent the output as

$$y[n] = (2)(\tfrac{1}{2})^n + (-1)(\tfrac{1}{4})^n = (\tfrac{1}{2})^{n-1} - (\tfrac{1}{4})^n$$

This is not obvious at all from the original flow graph analysis!

#4: z-transform Example Mini-conclusions

z-transform allows us to use *polynomials* to analyze discrete-time systems, particularly feedback systems

- The poles are the roots of the polynomial in the denominator
- The poles (in particular, the one with the largest magnitude) determine the stability of the feedback system
- Partial fraction, as a technique in polynomial manipulation, is very useful to give insight into the system behavior
- (Almost all) high-order feedback systems can be turned to a sum of first-order feedback systems

#3 – #4: Transform Conclusions

Transform method is less straightforward compared with time-domain methods, but is more powerful, particularly in design.

- Control the rate of decay by designing the poles.
- Can break into sub-systems and treat the system hierarchically.
- For implementation, can always go back to time-domain.

Example: $H = \dfrac{1 - \frac{3}{2}z^{-1}}{1 - \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}}$. Thus,

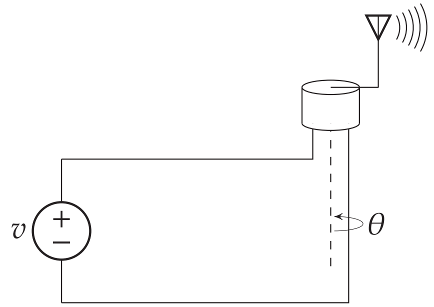
$$\left(1 - \tfrac{5}{6}z^{-1} + \tfrac{1}{6}z^{-2}\right) Y = \left(1 - \tfrac{3}{2}z^{-1}\right) X$$
$$Y - \tfrac{5}{6}z^{-1}Y + \tfrac{1}{6}z^{-2}Y = X - \tfrac{3}{2}z^{-1}X$$
$$y[n] - \tfrac{5}{6}y[n-1] + \tfrac{1}{6}y[n-2] = x[n] - \tfrac{3}{2}x[n-1]$$

Problems that Need Control

A design problem: **Fix the angle $\theta[n]$ of an antenna**. You have

1. A voltage source $v[n]$ (assumed discrete-time)
2. A motor that produces an angular movement proportional to the voltage input

A possible configuration:



Problems that Need Control

The corresponding discrete-time model:

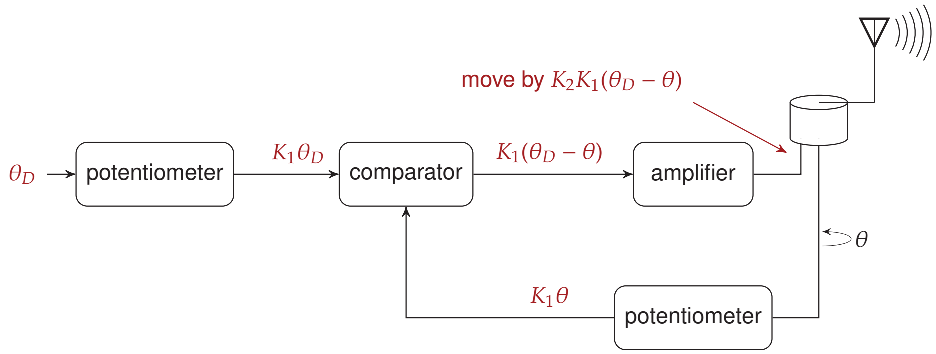
Problems:

- Need to know initial angle
- Need exact electrical and mechanical characteristics of the motor
- Need to give very precise instruction to the voltage source to first accelerate and then decelerate the motor

This is called an open-loop system, which is generally not desirable.

Problems that Need Control

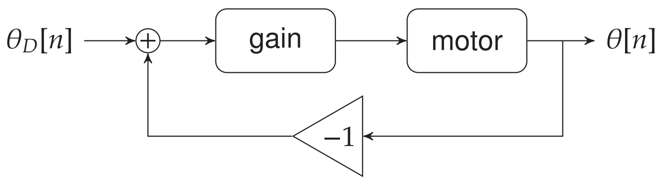
Modification by adding feedback control:



- A desired angle θ_D
- Potentiometer maps angle to voltage (K_1)
- Comparator takes the difference
- Amplifier magnifies the input signal (K_2)

Problems that Need Control

The corresponding discrete-time model:



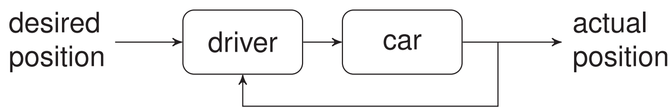
- gain = K_1K_2. This is our design parameter
- Can disturb the antenna position — the system will correct itself
- No need to know initial angle, model the exact electrical and mechanical characteristics of the motor, or give very precise instruction to the voltage source to first accelerate and then decelerate the motor

This is called a closed-loop system, which is generally desirable!

Problems that Need Control

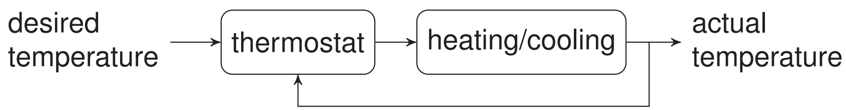
Many problems benefit from feedback control. *Examples:*

1. Proper driving: controlling the steering wheel to stay in lane



What about drunk driving?

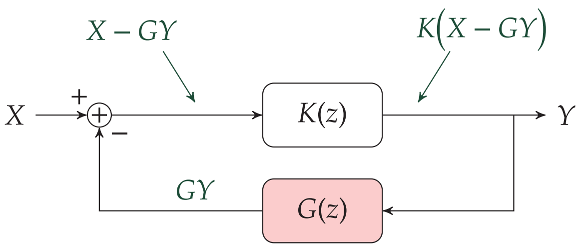
2. Air-conditioning: controlling the room temperature



What if you don't want to turn on and off the air-conditioning too frequently?

Mathematics of Feedback Control

A general feedback pattern:



- Negative gain is embedded in the "+" and "−" symbols
- $Y = K(X - GY)$

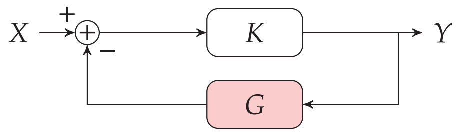
$$H = \frac{Y}{X} = \frac{K}{1 + KG}$$

Feedback Control Example

Example (A): Reliable amplification with a constant K

K is huge ($K \gg 1$) but unreliable. How to build a *reliable gain* system?

Ans: Assume we can build an attenuator G reliably.



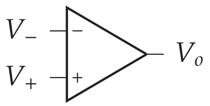
If $KG \gg 1$: *independent of the actual value of K,*

$$\frac{Y}{X} = \frac{K}{1 + KG} \approx \frac{1}{G}$$

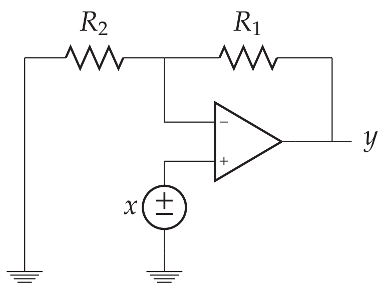
Feedback Control Example

Example (A): (cont.) Reliable amplification using circuits

Use an operational amplifier (op amp) where $V_o = K(V_+ - V_-)$



Build this circuit (V_- is the feedback):

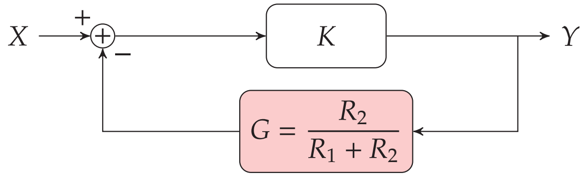


$$V_- = \left(\frac{R_2}{R_1 + R_2}\right) Y$$

$$Y = K(X - V_-)$$

Feedback Control Example

Example (A): (cont.) Reliable amplification using circuits



Assume $KG \gg 1$*: (precise value of K does not matter)*

$$H = \frac{K}{1 + KG} \approx \frac{1}{G} = \frac{R_1 + R_2}{R_1}$$

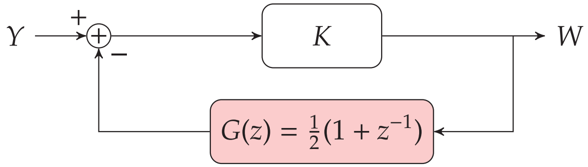
- Resistor values are much more accurate than op amp gain → this new circuit has much more reliable gain
- Tradeoff: the gain is much smaller, since we need $K \gg 1/G$
- Called non-inverting amplifier

Feedback Control Example

Example (B): Inverse system design

$y[n] = \frac{1}{2}(x[n] + x[n-1])$ has a smoothing effect; can we *undo* it?

Ans: Let $G = \frac{1}{2}(1 + z^{-1})$, so $Y = GX$. Pass $y[n]$ through:



Assume $KG(z) \gg 1$: (we'll ignore what that really means for a polynomial in z for now)

$$H = \frac{W}{Y} = \frac{K}{1+KG} \approx \frac{1}{G}$$

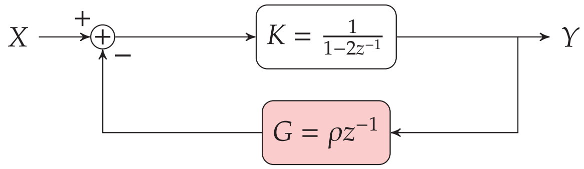
Hence, $W = HY \approx \left(\frac{1}{G}\right)(G)X = X$.

Feedback Control Example

Example (C): Stabilization of unstable systems (e.g., microphone echo)

$y[n] = 2y[n-1] + x[n]$ is an unstable system; can we *stabilize* it?

Ans: Let $K = \frac{1}{1-2z^{-1}}$ and $G = \rho z^{-1}$.

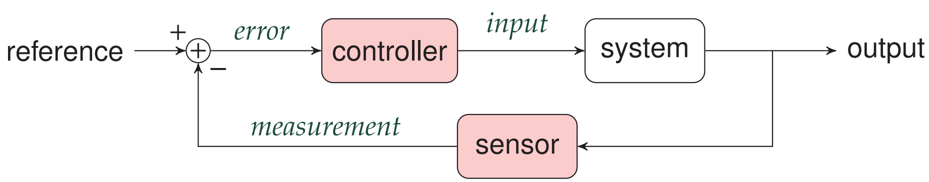


$$\frac{K}{1+KG} = \frac{\frac{1}{1-2z^{-1}}}{1+\left(\frac{1}{1-2z^{-1}}\right)(\rho z^{-1})} = \frac{1}{(1-2z^{-1})+(\rho z^{-1})} = \frac{1}{1-(2-\rho)z^{-1}}$$

ρ is our design parameter. Pick ρ so that $|2-\rho| < 1$.

Feedback Design

A more generic setup of feedback loop

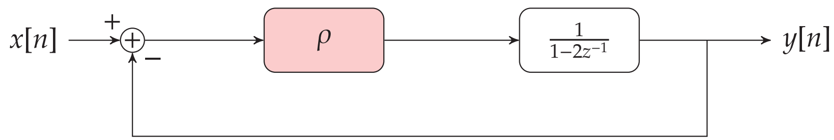


- controller = $C(z)$, system/plant = $P(z)$, sensor = $G(z)$
- $P(z)$ may be unstable
- Design C and G such that the closed-loop system

$$H = \frac{CP}{1+CPG} \quad \text{is stable}$$

Feedback Control Design Example

Example (A):

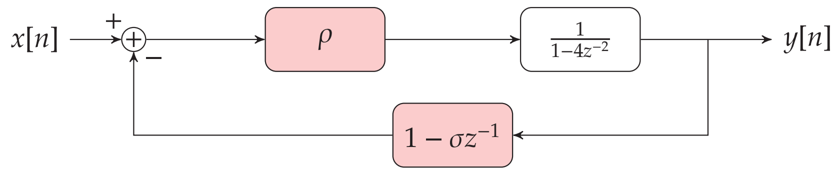


$$\frac{Y}{X} = \frac{\frac{\rho}{1-2z^{-1}}}{1+\frac{\rho}{1-2z^{-1}}} = \frac{\rho}{(1+\rho)-2z^{-1}} = \frac{\frac{\rho}{1+\rho}}{1-\left(\frac{2}{1+\rho}\right)z^{-1}}$$

- Bigger ρ means a smaller pole
- e.g. $x[n] = \delta[n]$, then a smaller pole means $y[n] \to 0$ faster

Feedback Control Design Example

Example (B):



- Original system is unstable because

$$\frac{1}{1-4z^{-2}} = \frac{1}{(1+2z^{-1})(1-2z^{-1})} = \frac{\frac{1}{2}}{1-2z^{-1}} + \frac{\frac{1}{2}}{1+2z^{-1}}$$

- Overall feedback system:

$$\frac{\frac{\rho}{1-4z^{-2}}}{1+\frac{\rho}{1-4z^{-2}}(1-\sigma z^{-1})} = \frac{\rho}{(1-4z^{-2})+\rho(1-\sigma z^{-1})} = \frac{\rho}{(1+\rho)-(\rho\sigma)z^{-1}-4z^{-2}}$$

Feedback Control Design Example

Example (B): (cont.) $H = \dfrac{\rho}{(1+\rho)-(\rho\sigma)z^{-1}-4z^{-2}}$

We can make different choices of ρ and σ.

1. $\rho = \frac{55}{9}$, $\sigma = 0$:

$$H = \frac{\frac{55}{9}}{\frac{64}{9}-4z^{-2}} = \frac{\frac{55}{64}}{1-\frac{9}{16}z^{-2}} = \frac{\frac{55}{128}}{1-\frac{3}{4}z^{-1}} + \frac{\frac{55}{128}}{1+\frac{3}{4}z^{-1}}$$

Therefore, if $x[n] = \delta[n]$, then

$$y[n] = \frac{55}{128}\left[\left(\frac{3}{4}\right)^n + \left(-\frac{3}{4}\right)^n\right] = \begin{cases} \frac{55}{64}\left(\frac{3}{4}\right)^n & \text{if } n \text{ is even.} \\ 0 & \text{if } n \text{ is odd.} \end{cases}$$

Feedback Control Design Example

Example (B): (cont.) $H = \dfrac{\rho}{(1+\rho)-(\rho\sigma)z^{-1}-4z^{-2}}$

2. $\rho = 31$, $\sigma = \frac{8}{31}$:

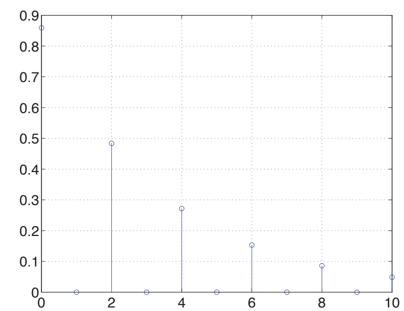
$$H = \frac{31}{32-8z^{-1}-4z^{-2}} = \frac{\frac{31}{32}}{1-\frac{1}{4}z^{-1}-\frac{1}{8}z^{-2}} = \frac{\frac{31}{32}\cdot\frac{2}{3}}{1-\frac{1}{2}z^{-1}} + \frac{\frac{31}{32}\cdot\frac{1}{3}}{1+\frac{1}{4}z^{-1}}$$

Therefore, if $x[n] = \delta[n]$, then

$$y[n] = \frac{31}{96}\left[2\left(\frac{1}{2}\right)^n + \left(-\frac{1}{4}\right)^n\right]$$

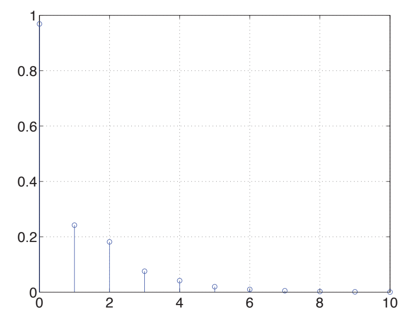
Feedback Control Design Example

Example (B): (cont.)



$\rho = \frac{55}{9}$, $\sigma = 0$

poles at $\frac{3}{4}$, $-\frac{3}{4}$



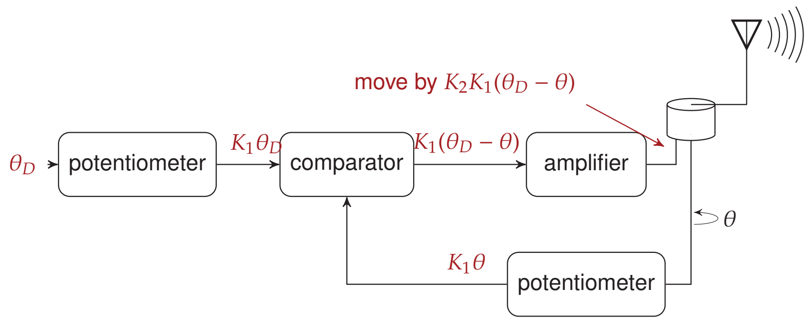
$\rho = 31$, $\sigma = \frac{8}{31}$

poles at $\frac{1}{2}$, $-\frac{1}{4}$

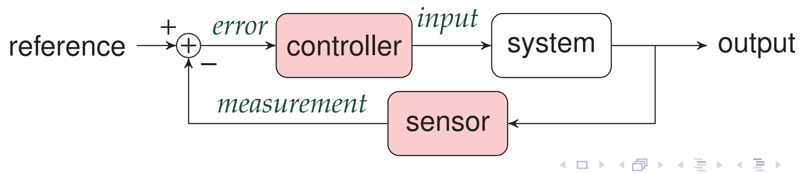
Decay is determined by the larger of the two poles (magnitude < 1)

Feedback Design: The Antenna Orientation Problem

Set the input angle, and expect the output to follow



Connect the "specific problem" to the "general model"



Feedback Design: The Antenna Orientation Problem

- Input is $\theta_D[n]$. Output is $\theta[n]$
- Sensor senses current angle to feed back to the input

$$\theta_s[n] = \theta[n]$$

- Controller produces an *angular velocity* proportional to the difference between desired angle and sensed data

$$v[n] = K(\theta_D[n] - \theta_s[n])$$

- System turns the angle of the antenna from the "previous angle" to the "current angle" (based on the previous angular velocity!)

$$\theta[n] = \theta[n-1] + Tv[n-1]$$

where T is the *time between the discrete samples (fixed beforehand)*

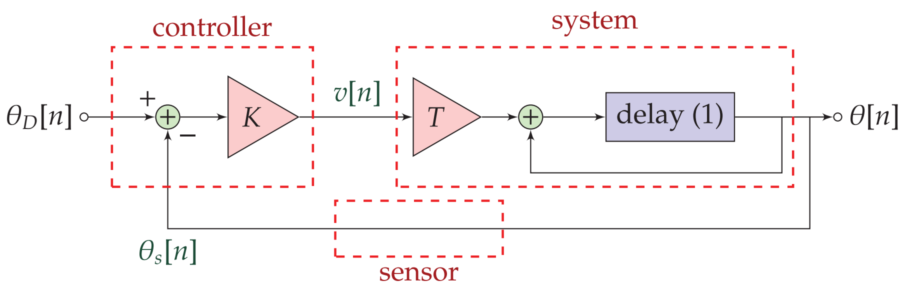
Design question: How to choose K?

Feedback Design: The Antenna Orientation Problem

Equation:

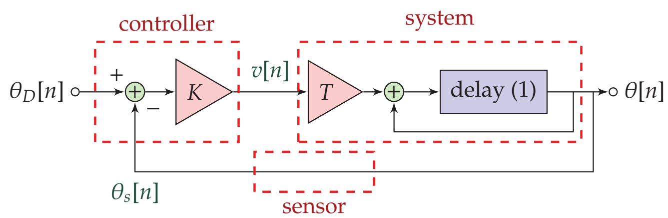
$$\theta[n] = \theta[n-1] + KT(\theta_D[n-1] - \theta[n-1])$$

Flow graph:



Make use of the mathematics of feedback control to find the transfer function of the entire process

Feedback Design: The Antenna Orientation Problem



- Within the *system* block, we have a feedback with positive loop! Hence, transfer function within this block has minus in the denominator:

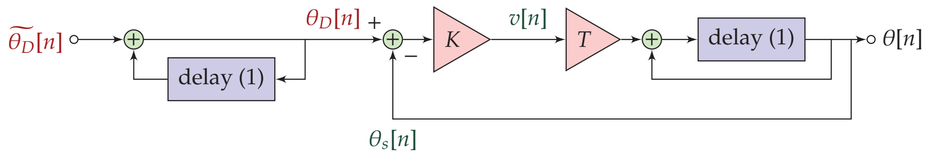
$$T \cdot \frac{z^{-1}}{1-z^{-1}}$$

- Substituting this into the system block, we can apply the feedback equation again to get

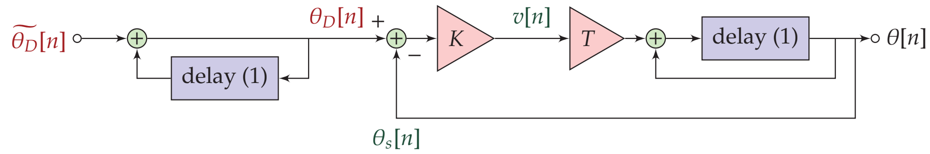
$$H = \frac{\Theta}{\Theta_D} = \frac{K\left(T \cdot \frac{z^{-1}}{1-z^{-1}}\right)}{1 + K\left(T \cdot \frac{z^{-1}}{1-z^{-1}}\right)} = \frac{(KT)z^{-1}}{1-(1-KT)z^{-1}}$$

Feedback Design: The Antenna Orientation Problem

- We can use this transfer function to understand the behavior of the output with an inpulse input.
- But what we want is a bit different: we want to understand the behavior of the output with an input staying at a certain angle.
- Change to the following system!



Feedback Design: The Antenna Orientation Problem



- Let $p = 1 - KT$. Overall transfer function:

$$\widetilde{H} = \left(\frac{1}{1 - z^{-1}}\right)\left(\frac{(1-p)z^{-1}}{1 - pz^{-1}}\right) = \frac{1}{1 - z^{-1}} - \frac{1}{1 - pz^{-1}}$$

- Thus, if $\widetilde{\theta_D}[n]$ is an impulse input, output is *a difference of two geometrically decaying sequences*, i.e.

$$\theta[n] = (1)^n - (p)^n = 1 - p^n$$

Feedback Design: The Antenna Orientation Problem

$$\boxed{\theta[n] = 1 - p^n}$$

Interpretations:

- Let's say a value of 1 means 30°. We give an impulse input, or equivalently keep $\theta_D[n] = 1$, leads to $\theta[n] = 1 - p^n$, which approaches 1 when $|p| < 1$ and $n \to \infty$.
- Best value of p: $p = 0$, i.e. $KT = 1$.
- Note that $\lim_{p \to 0} p^0 = 1$. So, output is 1 only when $n \geq 1$.
- Let's say we get a discrete sample every one second, i.e. $T = 1$s. If we want to turn 30° (a value of 1 in our discrete model), then we should set $K = 30°$ per second (to multiply the input).

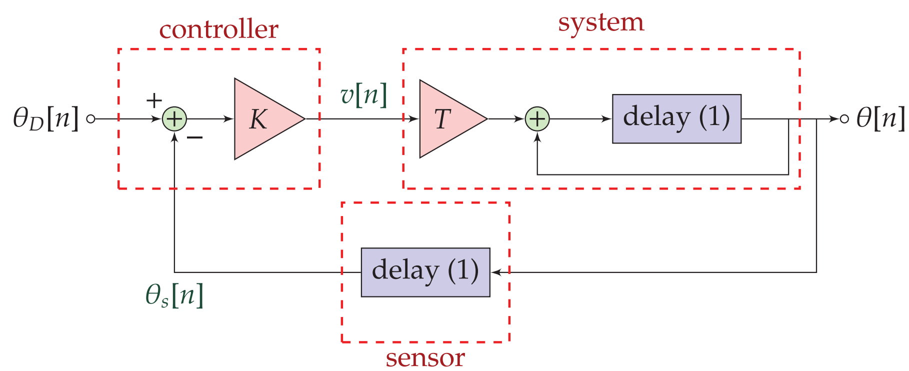
This makes **perfect sense**! We will be done in **one step**.

Feedback Design: The Antenna Orientation Problem

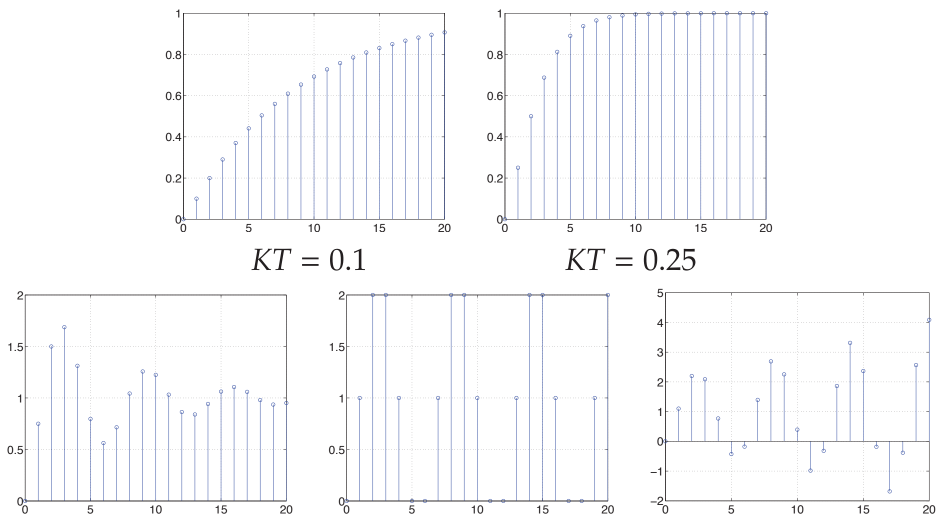
What if sensor feeds back to the input with a delay, i.e.

$$\theta_s[n] = \theta[n-1]$$

- Adding delay tends to destabilize the system
- Difficult to have an *intuitive way* to set K



Feedback Design: The Antenna Orientation Problem



$KT = 0.1$ $KT = 0.25$

$KT = 0.75$ $KT = 1.0$ $KT = 1.1$

Feedback Design: The Antenna Orientation Problem

Develop the transfer function

- Systems block unchanged: $\left(T \cdot \frac{z^{-1}}{1 - z^{-1}}\right)$
- Transfer function includes the delay in sensor

$$H = \frac{\Theta}{\Theta_D} = \frac{K\left(T \cdot \frac{z^{-1}}{1-z^{-1}}\right)}{1 + K\left(T \cdot \frac{z^{-1}}{1-z^{-1}}\right)(z^{-1})} = \frac{(KT)z^{-1}}{1 - z^{-1} + (KT)z^{-2}}$$

- Full transfer function including the "step" input:

$$\widetilde{H} = \frac{\Theta}{\widetilde{\Theta_D}} = \left(\frac{1}{1 - z^{-1}}\right)\frac{(KT)z^{-1}}{1 - z^{-1} + (KT)z^{-2}} = \frac{z}{1 - z^{-1}} - \frac{z}{1 - z^{-1} + (KT)z^{-2}}$$

Feedback Design: The Antenna Orientation Problem

- Poles are the values where the denominator is zero, i.e.

$$1 - z^{-1} + (KT)z^{-2} = 0$$
$$z^2 - z + KT = 0$$

So poles are at $\frac{1 \pm \sqrt{(-1)^2 - 4(1)(KT)}}{2(1)} = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - 4(KT)}$.

- For $0 < (KT) < \frac{1}{4}$: the *larger* of the pole is $\frac{1}{2} + \frac{1}{2}\sqrt{1 - 4(KT)}$. It is smallest when $KT = \frac{1}{4}$.
- Smaller values of KT also works, but not as good (due to a larger pole).
- Sensor delay results in slower response.

Feedback Design: The Antenna Orientation Problem

Some conclusions about feedback control:

1. Feedback allows us to change the system characteristics, e.g. from unstable to stable
2. Feedback allows design parameters
3. The mathematics of z-transform and pole locations are powerful tools to analyze discrete-time systems and feedback
4. *We now can analyze and design a system that would be impossible to do otherwise!*

Feedback Design: The Antenna Orientation Problem

A postscript: We had some observations earlier that

- For $\frac{1}{4} < (KT) < 1$, the system oscillates but still converges
- For $(KT) = 1$, the system oscillates *(Hint: what is the magnitude of the poles?)*
- For $(KT) \geq 1$, the system is unstable!

Why? You'll find out if you continue with EEE!

The Way Forward