Control the World by Adding and Multiplying

Andres Kwasinski

Department of Computer Engineering

Rochester Institute of Technology

What is this talk about?

- It is about Digital Signal Processing (DSP): the technology of performing operations on digital signals (signals represented as 1s and 0s).

- DSP is used everywhere

Something Interesting

- Many of the DSP algorithms are based on a very simple operation:

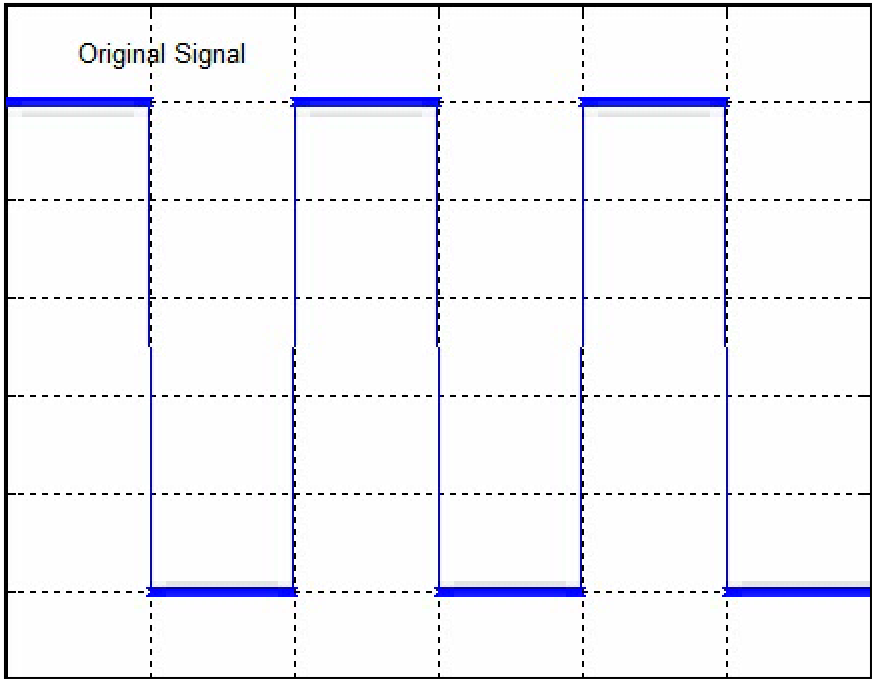
$$y[n] = \sum_{k=N}^{M} c_k x_n[k]$$

- Multiplications
- Accumulate the multiplications

- A simple algorithm with many uses…

Fourier Series

- **Periodic signals can be expressed as a sum of products between weighting coefficients and sinusoids**



$$x(t) = \frac{1}{2} + \sum_{n=0}^{20} \frac{2}{(2n+1)\pi} \cos((2n+1)\omega_o t - \frac{\pi}{2})$$

Fourier Series

$$x(t) = \frac{1}{2} + \sum_{n=0}^{20} \frac{2}{(2n+1)\pi} \cos((2n+1)\omega_o t - \frac{\pi}{2})$$

- **Instead of sinusoids, we use the more general expression of complex exponentials:** $e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$

- **We still use the same idea: periodic signals can be expressed as a sum of products between weighting coefficients and complex exponentials:**

$$x(t) = \sum_{k=-\infty}^{\infty} X[k] e^{jk\omega_o t}$$

- **<u>Fourier Series</u> (Representation) of *x(t)***

$$y[n] = \sum_{k=N}^{M} c_k x_n[k]$$ (our reference algorithm)

$$x(t) = \sum_{k=-\infty}^{\infty} X[k] e^{jk\omega_o t}$$

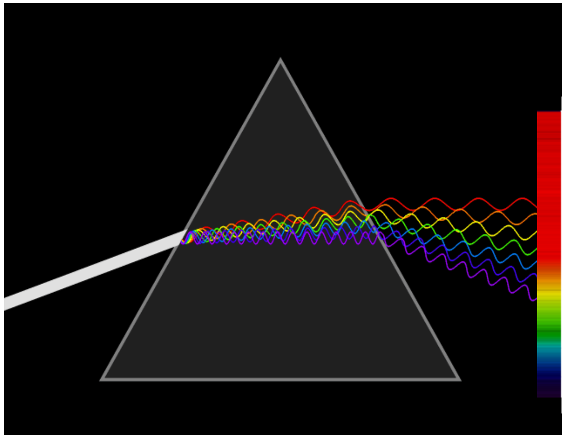
- **Fourier Series (Representation) of *x(t)***

Coefficients (weights) are complex numbers

- **Each term of the sum multiplies a coefficient with a complex exponential function:**
  - **Each complex exponentials function has a different frequency:** $k\omega_0$
  - **Two different complex exponential functions are orthogonal.**
- **This gives rise to the idea of "spectrum".**

Spectrum

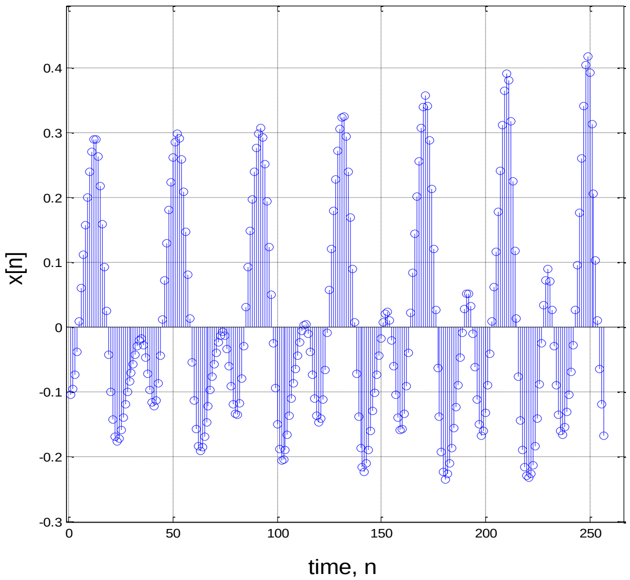
- Signals can be decomposed into separable (orthogonal, independent) components determined by complex exponentials (sinusoids) of different frequencies. The result is the “spectrum” of the signal.
- Signals are made from the composition of sinusoids.

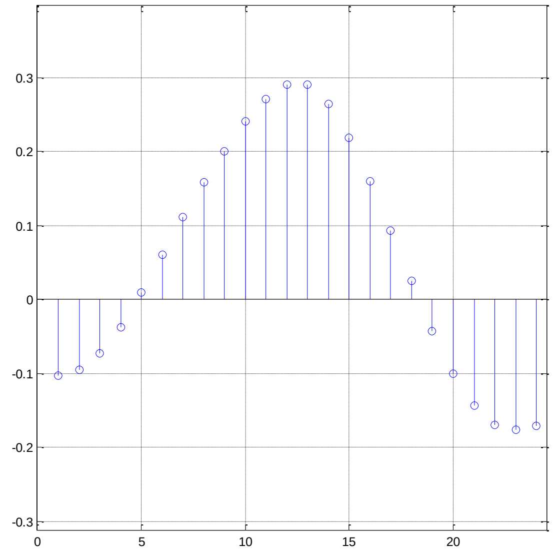


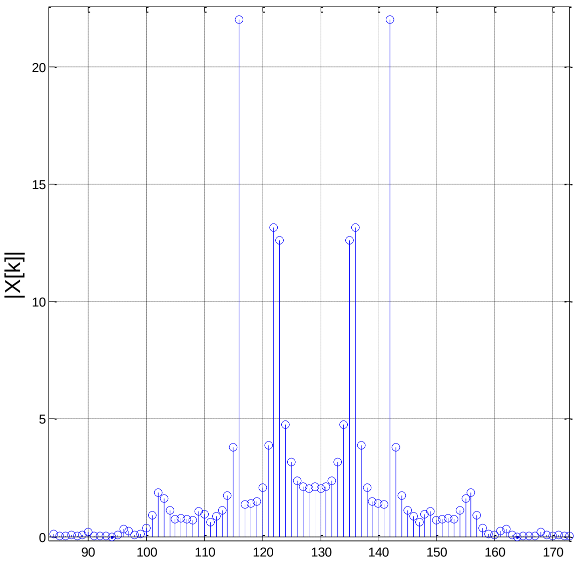
Spectrum: term originally used in the 17th century to describe the decomposition of white light into its colors (frequencies).

White light is made from the composition of monochrome light.

- **Same ideas can be extended to a digital signal** $x[n]$



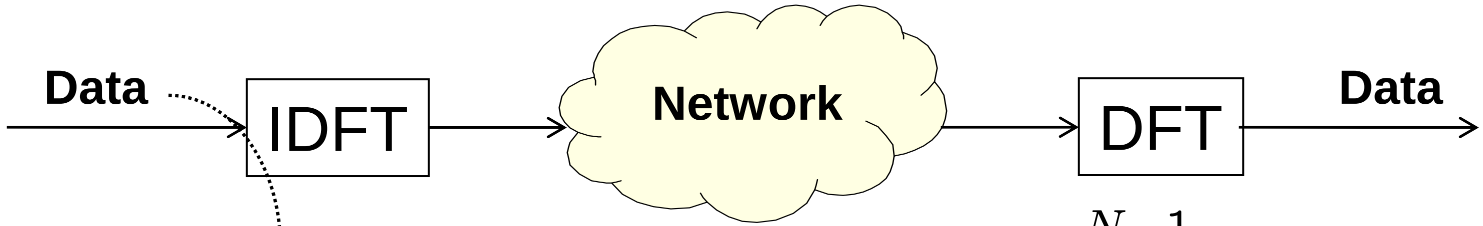




$$\hat{X}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-jk\quad o n}, k = 0, \ldots, N-1$$

$$y[n] = \sum_{k=N}^{M} c_k x_n[k]$$ (our reference algorithm)

DFT Application: OFDM



$$x[n] = \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi kn/N}$$

$$n = 0, \ldots, N-1$$

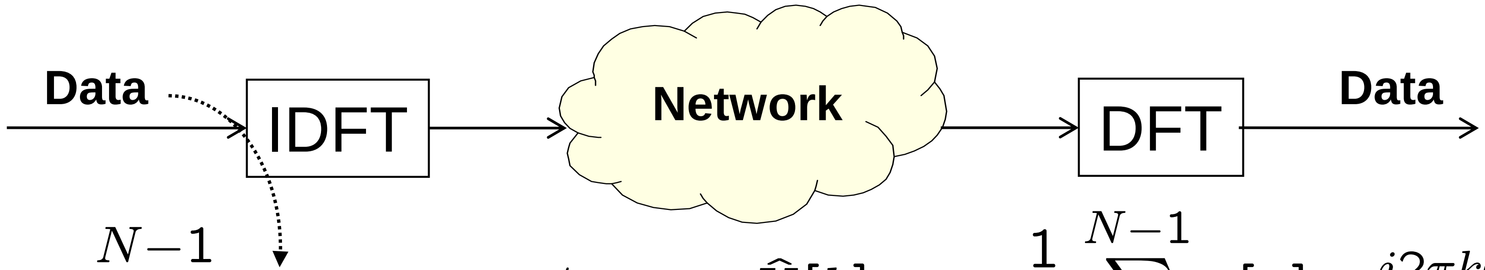
$$\hat{X}[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

$$k = 0, \ldots, N-1$$

$$y[n] = \sum_{k=N}^{M} c_k x_n[k]$$ (our reference algorithm)

- Each bit is sent through a separable (orthogonal) signal.
- Reverse operation at receiver extracts each bit by exploiting signal orthogonality.

DFT Application: OFDM



$$x[n] = \sum_{k=0}^{N-1} \hat{X}[k] e^{j2\pi kn/N} \qquad n = 0, \ldots, N-1$$

$$\hat{X}[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \qquad k = 0, \ldots, N-1$$

$$y[n] = \sum_{k=N}^{M} c_k x_n[k] \text{ (our reference algorithm)}$$

- Applications:

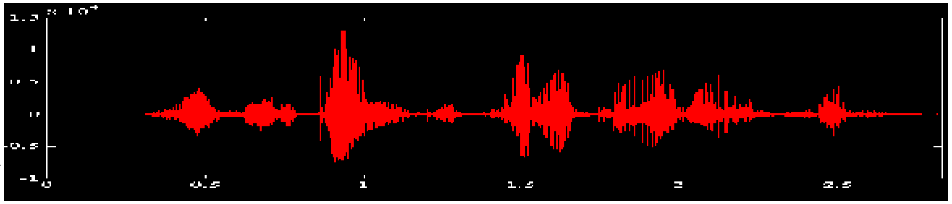
DSL

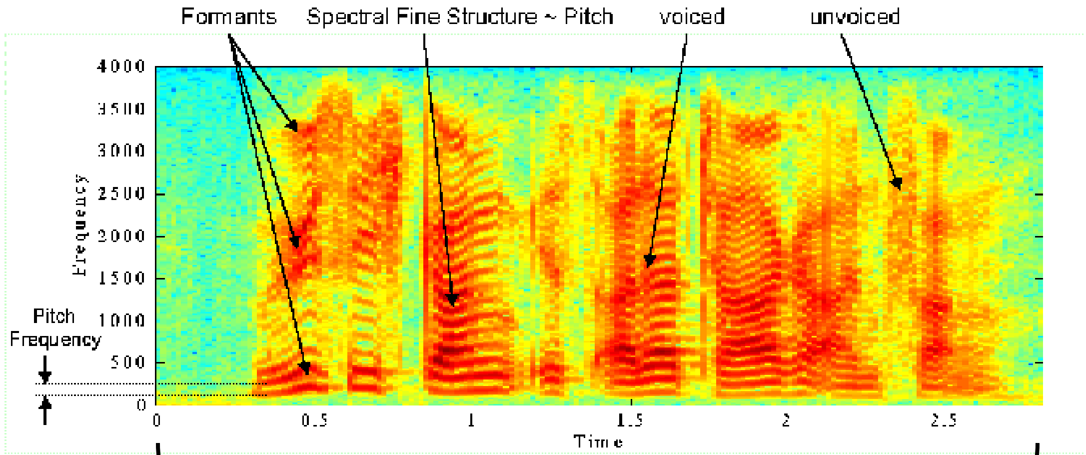
4G Phones

WiFi

DFT Application: Speech Recognition

Digital speech samples $x[n]$



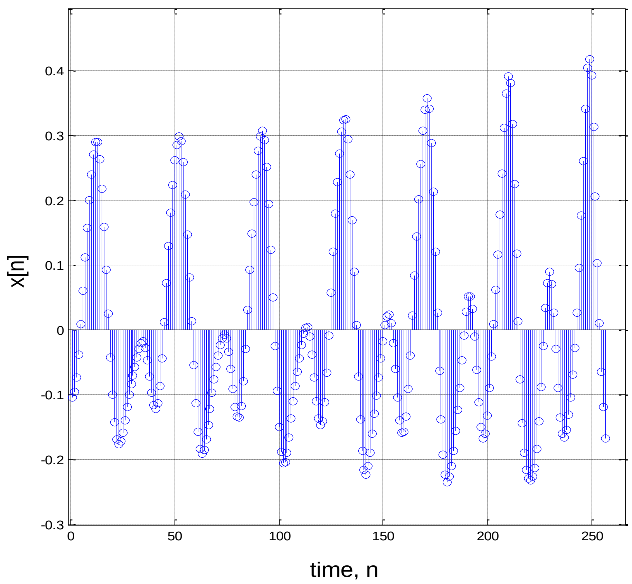


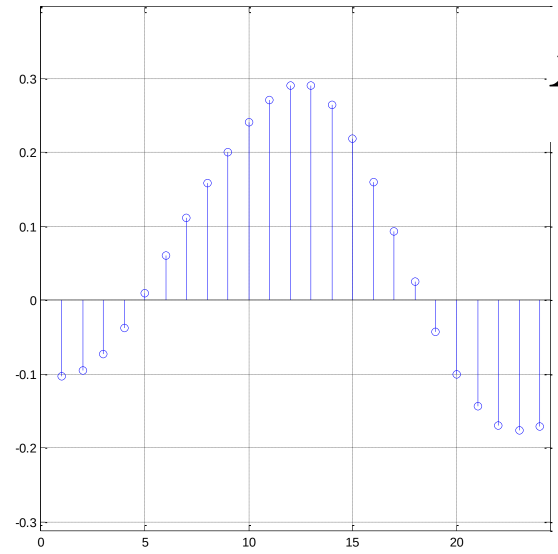
Every short period of time we take N samples and calculate the DFT:

$$\hat{X}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \qquad k = 0, \ldots, N-1$$

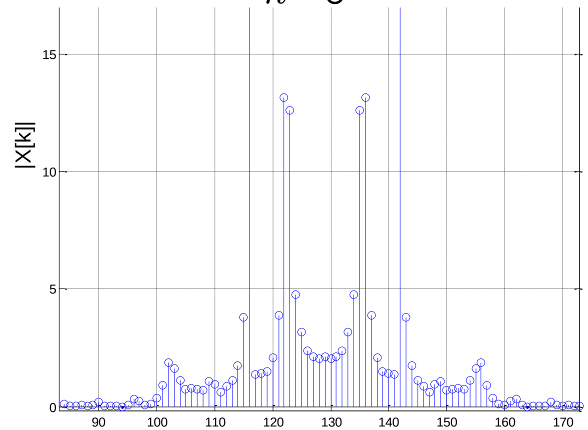
- As a result, we get the evolution of the spectrum over time.
- Computer use this information to recognize what is being said.

Understanding System Response





$$\hat{X}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$



- Inverse Discrete Fourier Transform $x[n] = \sum_{k=0}^{N-1} \hat{X}[k]e^{j2\pi kn/N}$
- The formula tells us that any signal can be thought of as being made from adding sinusoids (after multiplying each with an appropriate complex number).

Understanding System Response

- To get started, two important observations:
  - Any signal can be thought of as being made from adding sinusoids (after multiplying each with an appropriate complex number).
  - If we test a linear system with a sinusoidal signal, the system responds with a sinusoidal signal of the same frequency as the input but with different amplitude and phase.

Understanding System Response

- If we test a linear system with a sinusoidal signal, the system responds with a sinusoidal signal of the same frequency as the input but with different amplitude and phase.
- With math:
  - For input $x[n] = A\cos(\hat{\omega}_1 n + \phi)$
  - Output is $y[n] = A\left|H(e^{j\hat{\omega}_1})\right|\cos(\hat{\omega}_1 n + \phi + \angle H(e^{j\hat{\omega}_1}))$

Change in amplitude (↑ $\left|H(e^{j\hat{\omega}_1})\right|$)

Change in phase (↑ $\angle H(e^{j\hat{\omega}_1})$)

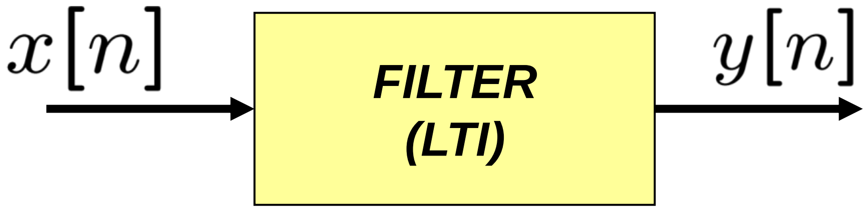
The effect of the system at frequency $\hat{\omega}_1$ is represented through the complex number

$$H(e^{j\hat{\omega}_1})$$

Understanding System Response

- ***Any signal can be thought of as being made from adding sinusoids (after multiplying each with an appropriate complex number).***
  - Decompose input signal into sinusoidal components; get its spectrum.
- ***If we test a linear system with a sinusoidal signal, the system responds with a sinusoidal signal of the same frequency as the input but with different amplitude and phase.***
  - By testing the system at different frequencies, find the function that maps the response of the system (amplitude and phase) as a function of frequency. This is called the system frequency response.
  - Multiply the input signal spectrum with the system frequency response. The result shows the components of the output signal at the sinusoidal components of different frequencies. This is the output signal spectrum.
  - Assemble the output signal from its spectrum (e.g. inverse Discrete Fourier Transform).

Understanding System Response



- Decompose input signal into sinusoidal components; get its spectrum.

$$\hat{X}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

- Multiply the input signal spectrum with the system frequency response. The result shows the components of the output signal at the sinusoidal components of different frequencies. This is the output signal spectrum.

$$\hat{Y}[k] = \hat{H}[k]\hat{X}[k]$$

- Assemble the output signal from its spectrum (e.g. inverse Discrete Fourier Transform).

$$y[n] = \sum_{k=0}^{N-1} \hat{Y}[k]e^{j2\pi kn/N}$$

Z-transform

$$X(z) = \sum_{k=-\infty}^{\infty} x[k] z^{-k}$$

- z is a complex variable.

Z-transform

$$X(z) = \sum_{k=-\infty}^{\infty} x[k] z^{-k}$$

- z is a complex variable.

(Think of it as a more general form of the DFT)

$$\hat{X}[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Z-transform

$$X(z) = \sum_{k=-\infty}^{\infty} x[k] z^{-k}$$

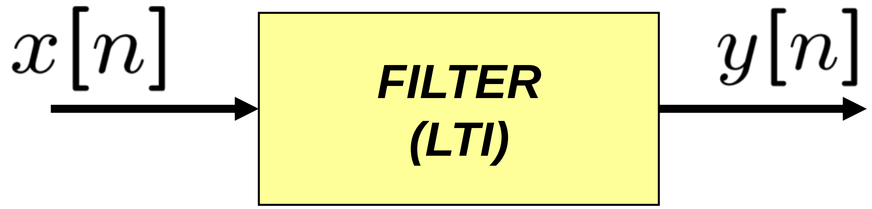
- z is a complex variable.

(Think of it as a more general form of the DFT)

$$\hat{X}[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

$$y[n] = \sum_{k=N}^{M} c_k x_n[k]$$ (our reference algorithm)

Understanding System Response



- Decompose input signal into sinusoidal components; get its spectrum.

$$\hat{X}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

$$X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k}$$

- Multiply the input signal spectrum with the system frequency response. The result shows the components of the output signal at the sinusoidal components of different frequencies. This is the output signal spectrum.

$$\hat{Y}[k] = \hat{H}[k]\hat{X}[k]$$

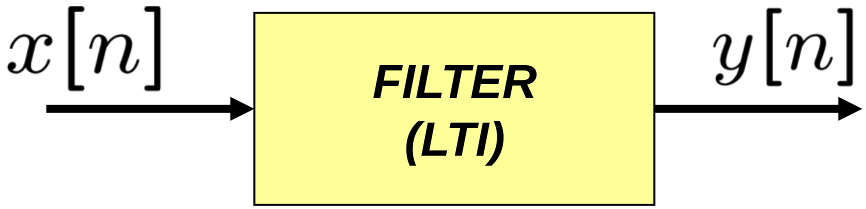
$$Y(z) = H(z)X(z)$$

- Assemble the output signal from its spectrum (e.g. inverse Discrete Fourier Transform).

$$y[n] = \sum_{k=0}^{N-1} \hat{Y}[k]e^{j2\pi kn/N}$$

$$y[n] = \mathcal{Z}^{-1}\{y[n]\}$$

Understanding System Response



- Decompose input signal into sinusoidal components; get its spectrum.

$$\hat{X}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \qquad X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k}$$

- Multiply the input signal spectrum with the system frequency response. The result shows the components of the output signal at the sinusoidal components of different frequencies. This is the output signal spectrum.

$$\hat{Y}[k] = \hat{H}[k]\hat{X}[k] \qquad Y(z) = H(z)X(z)$$

- Assemble the output signal from its spectrum (e.g. inverse Discrete Fourier Transform).

$$y[n] = \sum_{k=0}^{N-1} \hat{Y}[k]e^{j2\pi kn/N} \qquad y[n] = \mathcal{Z}^{-1}\{y[n]\}$$

Z-Transforms Properties – Convolution/Multiplication

$$W(z) = X(Z)Y(Z) = \left(\sum_{k=-\infty}^{\infty} x[k]z^{-k}\right)\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)$$

Z-Transforms Properties – Convolution/Multiplication

$$W(z) = X(Z)Y(Z) = \left(\sum_{k=-\infty}^{\infty} x[k] z^{-k} \right) \left(\sum_{m=-\infty}^{\infty} y[m] z^{-m} \right)$$

$$W(z) = \sum_{k=-\infty}^{\infty} x[k] \left(\sum_{m=-\infty}^{\infty} y[m] z^{-m} \right) z^{-k}$$

Z-Transforms Properties – Convolution/Multiplication

$$W(z) = X(Z)Y(Z) = \left(\sum_{k=-\infty}^{\infty} x[k]z^{-k}\right)\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)$$

$$W(z) = \sum_{k=-\infty}^{\infty} x[k]\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)z^{-k} = \sum_{k=-\infty}^{\infty} x[k]\sum_{n=-\infty}^{\infty} y[n-k]z^{-n}$$

$$n = m + k$$

Z-Transforms Properties – Convolution/Multiplication

$$W(z) = X(Z)Y(Z) = \left(\sum_{k=-\infty}^{\infty} x[k]z^{-k}\right)\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)$$

$$W(z) = \sum_{k=-\infty}^{\infty} x[k]\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)z^{-k} = \sum_{k=-\infty}^{\infty} x[k]\sum_{n=-\infty}^{\infty} y[n-k]z^{-n}$$

$$n = m + k$$

$$W(z) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]y[n-k]\right)z^{-n}$$

$$W(z) = X(Z)Y(Z) = \left(\sum_{k=-\infty}^{\infty} x[k]z^{-k}\right)\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)$$

$$W(z) = \sum_{k=-\infty}^{\infty} x[k]\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)z^{-k} = \sum_{k=-\infty}^{\infty} x[k]\sum_{n=-\infty}^{\infty} y[n-k]z^{-n}$$

$$n = m + k$$

$$W(z) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]y[n-k]\right)z^{-n}$$

$$x[n] * y[n] = \sum_{k=-\infty}^{\infty} x[k]y[n-k]$$

- Convolution sum

Z-Transforms Properties – Convolution/Multiplication

$$W(z) = X(Z)Y(Z) = \left(\sum_{k=-\infty}^{\infty} x[k]z^{-k}\right)\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)$$

$$W(z) = \sum_{k=-\infty}^{\infty} x[k]\left(\sum_{m=-\infty}^{\infty} y[m]z^{-m}\right)z^{-k} = \sum_{k=-\infty}^{\infty} x[k]\sum_{n=-\infty}^{\infty} y[n-k]z^{-n}$$

$$n = m + k$$

$$W(z) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]y[n-k]\right)z^{-n}$$

$$x[n] * y[n] = \sum_{k=-\infty}^{\infty} x[k]y[n-k]$$

- Convolution sum

$$W(Z) = \sum_{n=-\infty}^{\infty} (x[n] * y[n])z^{-n} = \sum_{n=-\infty}^{\infty} w[n]z^{-n}$$

Useful Z-Transforms Properties

- Convolution in time, Multiplication in z-domain:

$$\underbrace{y[n] = h[n] * x[n]}_{\text{Time domain}} \stackrel{\mathcal{Z}}{\longleftrightarrow} \underbrace{Y(Z) = H(z)X(z)}_{\text{Z domain}}$$

$$y[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

The <u>convolution sum</u> tells us, in time domain, how the filter (system) processes the input to generate the output

$$y[n] = \sum_{k=N}^{M} c_k x_n[k]$$ (our reference algorithm)

Z-Transforms Properties – Time Delay

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

$$\mathcal{Z}\{x[n-n_0]\} = \sum_{n=-\infty}^{\infty} x[n-n_0] z^{-n}$$

Z-Transforms Properties – Time Delay

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$m = n - n_0$$

$$\mathcal{Z}\{x[n-n_0]\} = \sum_{n=-\infty}^{\infty} x[n-n_0]z^{-n} = \sum_{m=-\infty}^{\infty} x[m]z^{-(m+n_0)}$$

Z-Transforms Properties – Time Delay

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$m = n - n_0$$

$$\mathcal{Z}\{x[n-n_0]\} = \sum_{n=-\infty}^{\infty} x[n-n_0]z^{-n} = \sum_{m=-\infty}^{\infty} x[m]z^{-(m+n_0)}$$

$$\mathcal{Z}\{x[n-n_0]\} = z^{-n_0} \sum_{m=-\infty}^{\infty} x[m]z^{-m}$$

Z-Transforms Properties – Time Delay

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

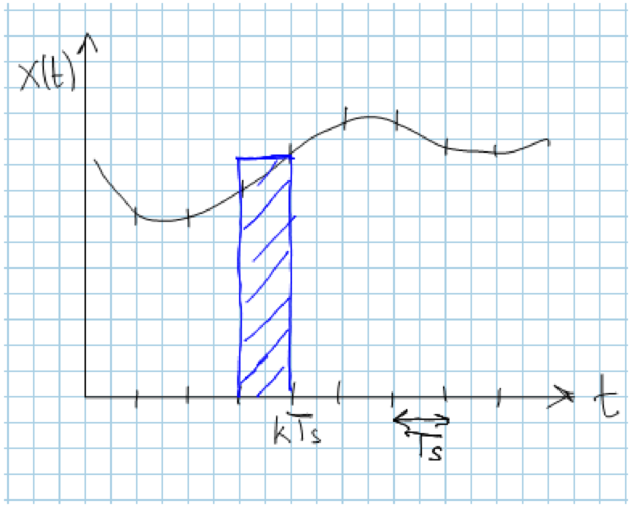
$$m = n - n_0$$

$$\mathcal{Z}\{x[n-n_0]\} = \sum_{n=-\infty}^{\infty} x[n-n_0] z^{-n} = \sum_{m=-\infty}^{\infty} x[m] z^{-(m+n_0)}$$

$$\mathcal{Z}\{x[n-n_0]\} = z^{-n_0} \sum_{m=-\infty}^{\infty} x[m] z^{-m}$$

$$\boxed{\mathcal{Z}\{x[n-n_0]\} = z^{-n_0} X(Z)}$$

- Rectangular rule (Euler's rule)



- Integration Increment: T_s

$$y(kT_s) \;=\; y(kT_s - T_s) + T_s x(kT_s)$$

- After digitalization, difference equation is:

$$y[k] \;=\; y[k-1] + T_s x[k]$$

A Digital Integrator

- Forward rectangular rule (Euler's rule)

$$y[k] = y[k-1] + T_s x[k]$$

- The **transfer function of** the corresponding **digital integrator** is (taking Z transforms)

$$\left(\mathcal{Z}\{y[k-k_0]\} = z^{-k_0} Y(Z)\right)$$

$$\mathcal{Z}\{y[k]\} = \mathcal{Z}\{y[k-1]\} + T_s \mathcal{Z}\{x[k]\}$$

$$Y(z) = z^{-1} Y(z) + T_s X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{T_s}{1 - z^{-1}}$$

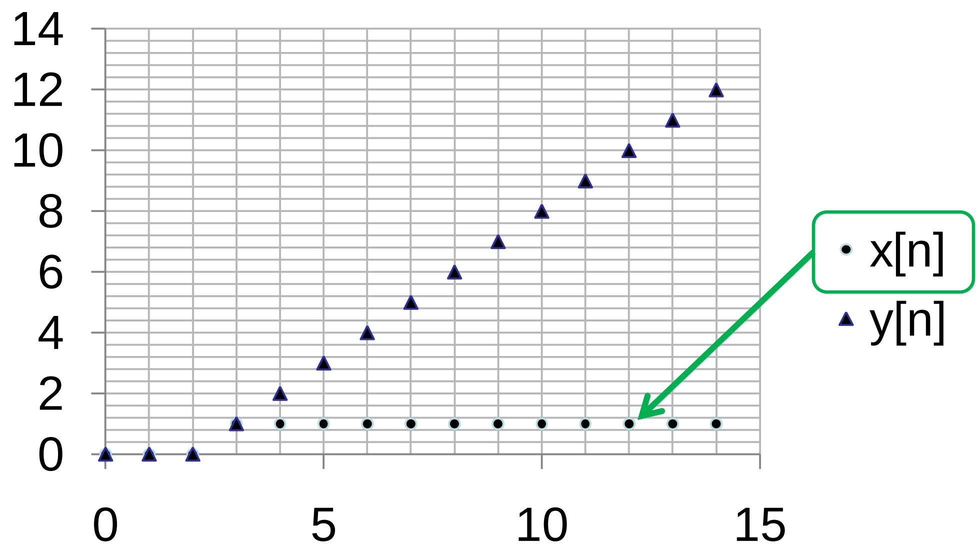
This is a filter whose output is the integral of its input

A Digital Integrator

$$y[k] = y[k-1] + T_s x[k]$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{T_s}{1 - z^{-1}}$$

- **The problem for this system is that is unstable for some input signals (not good if the system will be used for something that is not integration).**



- **When the input is just a constant (turned on to a value =1) the output grows for ever…**

Digital Systems

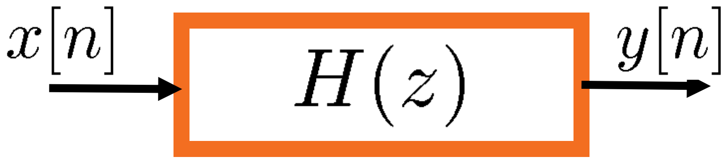
- How can we tell if and when a system will be unstable?

$$H(z) = \frac{Y(z)}{X(z)} = \frac{T_s}{1 - z^{-1}} = \frac{T_s z}{z - 1}$$

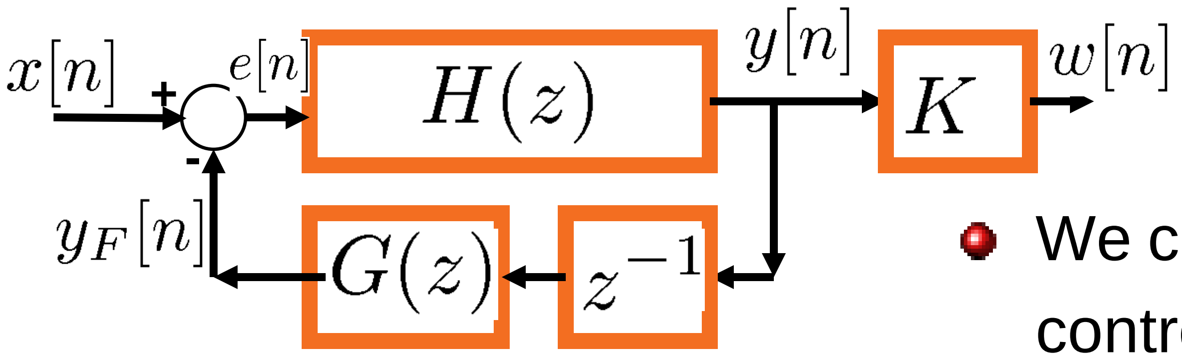
- We look at the roots of the denominator of the system's transfer function.

- If any root has magnitude larger of equal to 1, the system is unstable.

Making an Unstable System Stable



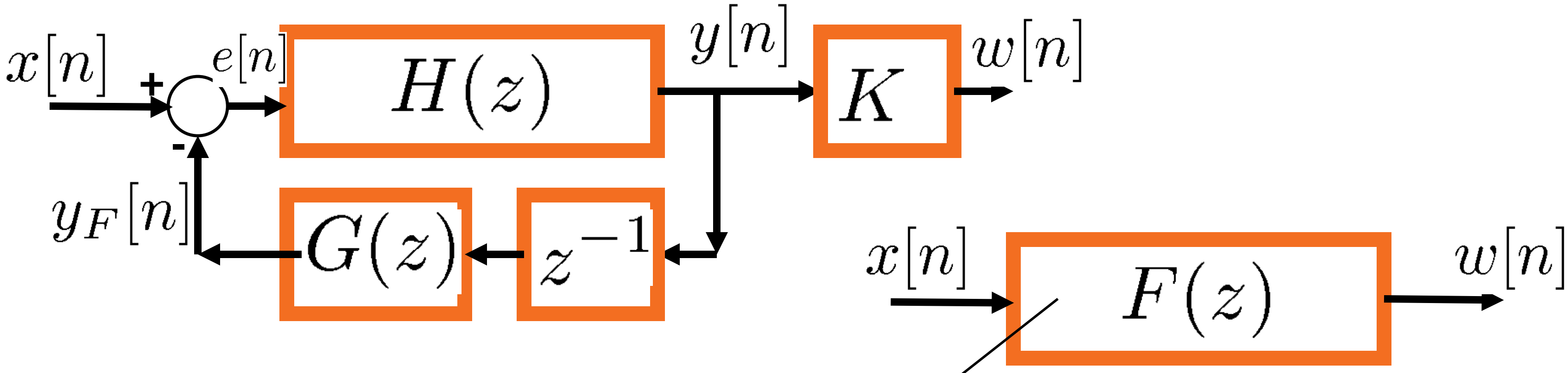
- Original unstable system



- We connect other system to control the original system

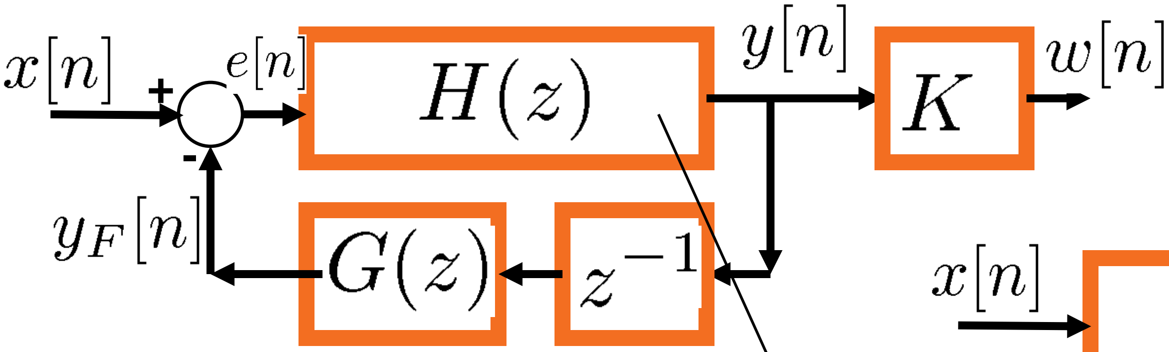
- The resulting overall stable system.

Making an Unstable System Stable



$$W(z) = F(z)X(z)$$

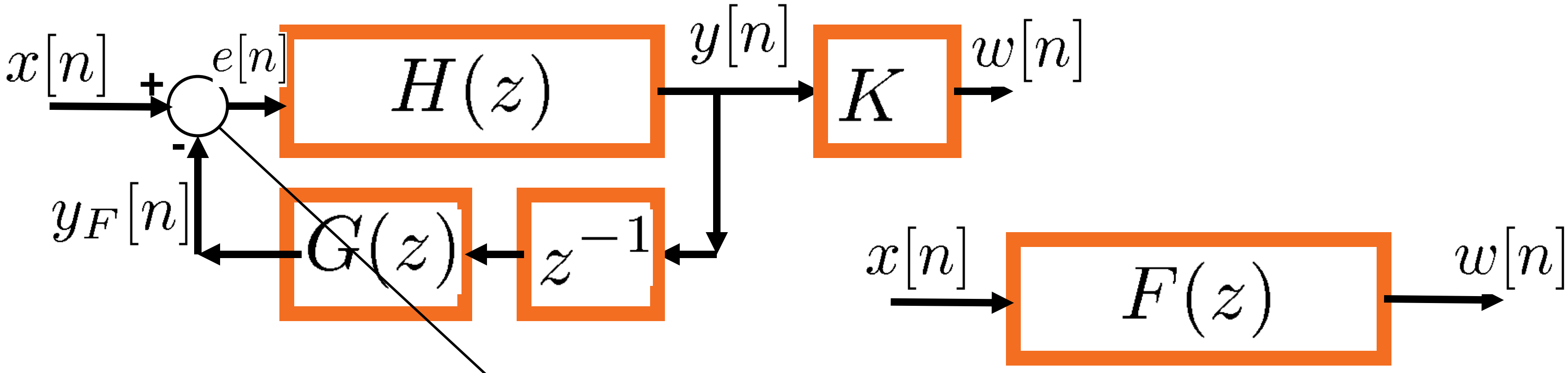
Making an Unstable System Stable



$$W(z) = F(z)X(z)$$

$$Y(z) = H(z)E(z)$$

Making an Unstable System Stable

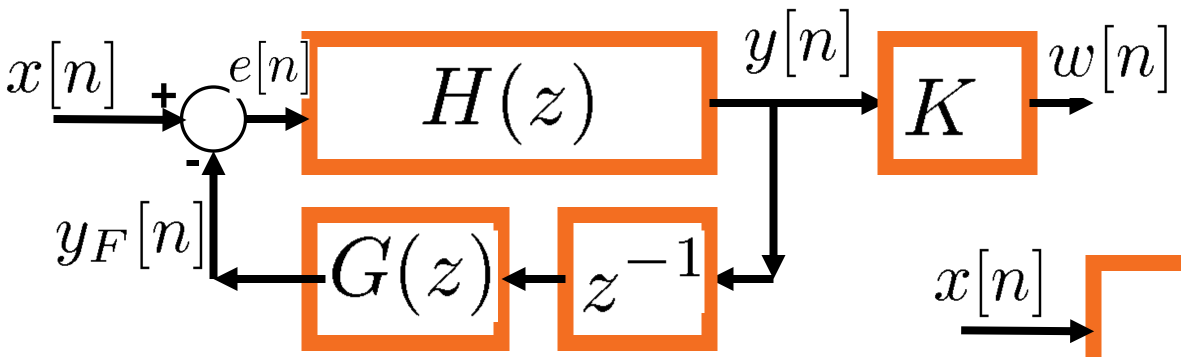


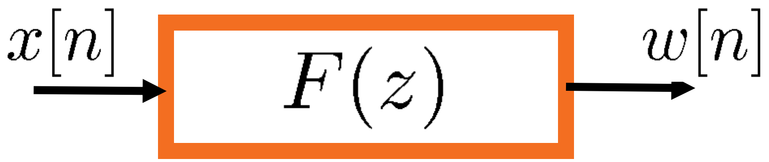
$$W(z) = F(z)X(z)$$

$$Y(z) = H(z)E(z)$$

$$E(z) = X(z) - Y_F(z)$$

Making an Unstable System Stable





$$W(z) = F(z)X(z)$$

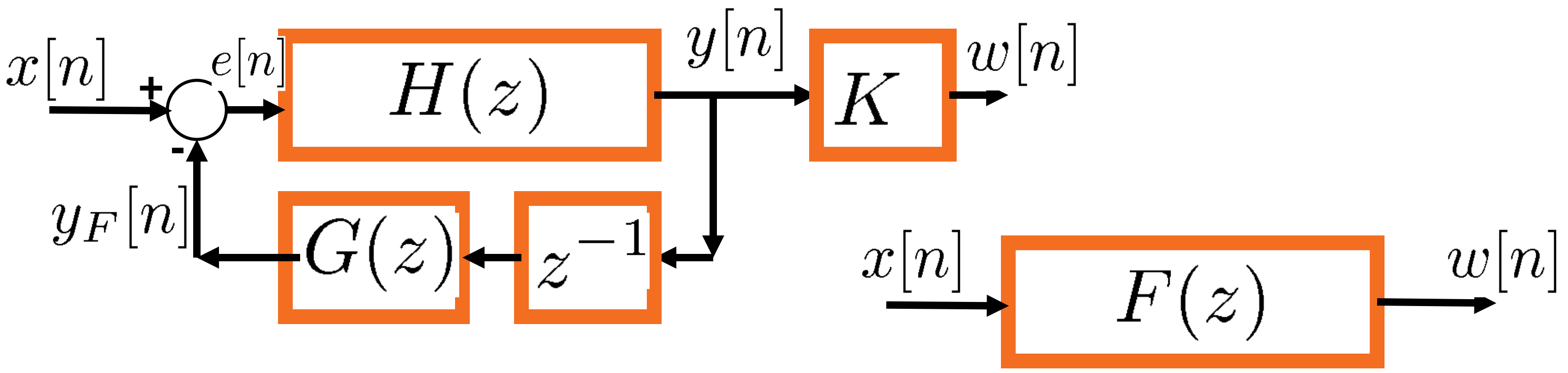
$$Y(z) = H(z)E(z)$$

$$E(z) = X(z) - Y_F(z)$$

$$\begin{aligned} Y(z) &= KH(z)\left(X(z) - Y_F(z)\right) \\ &= KH(z)\left(X(z) - z^{-1}G(z)Y(z)\right) \end{aligned}$$

$$\implies Y(z)\left(1 + z^{-1}G(z)Y(z)\right) = KH(z)X(z)$$

Making an Unstable System Stable



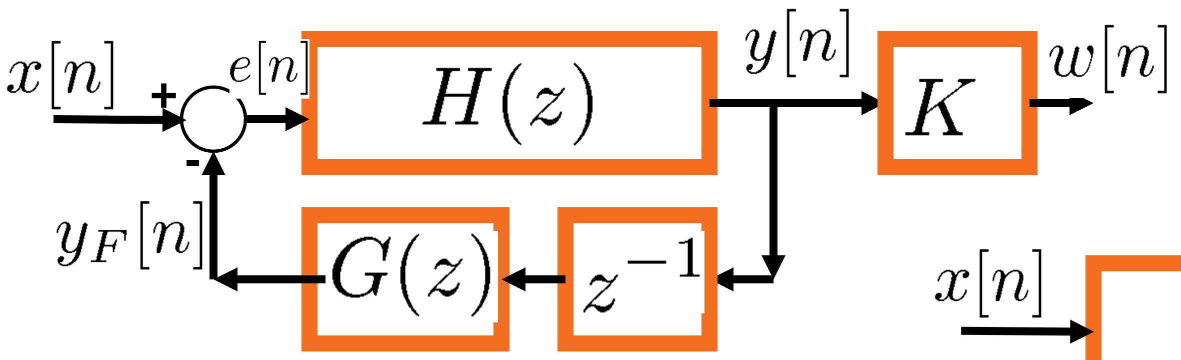
$$Y(z) = F(z)X(z)$$

$$Y(z)\left(1 + z^{-1}G(z)Y(z)\right) = KH(z)X(z)$$

$$\frac{Y(z)}{X(z)} = F(z) = \frac{KH(z)}{1 + z^{-1}G(z)H(z)}$$

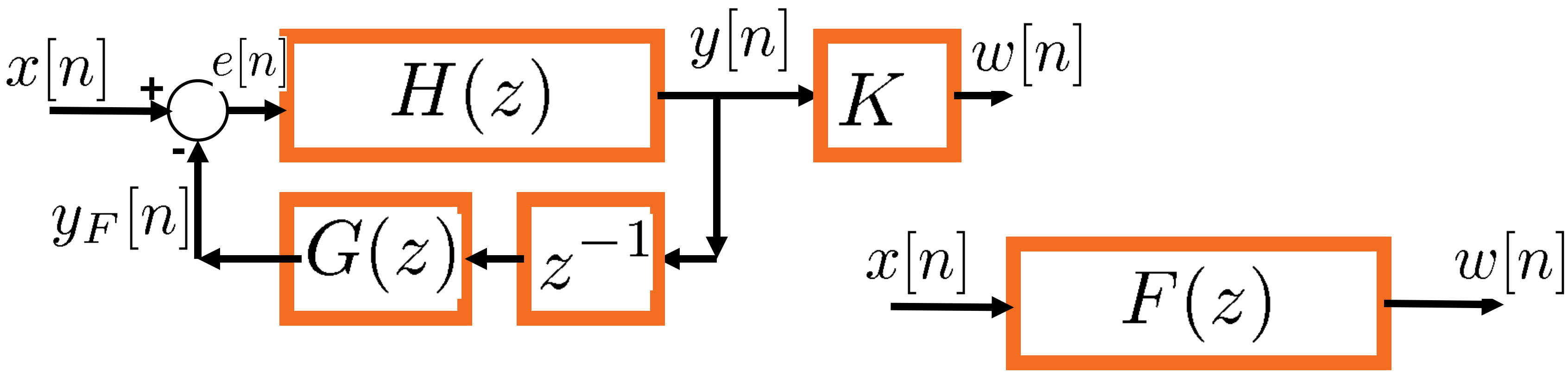
$$F(z) = \frac{K}{1 - z^{-1} + \frac{z^{-1}G(z)(1-z^{-1})}{1-z^{-1}}} = \frac{K}{1 - z^{-1} + z^{-1}G(z)}$$

Making an Unstable System Stable



$$F(z) = \frac{K}{1 - z^{-1} + z^{-1}G(z)}$$

Making an Unstable System Stable

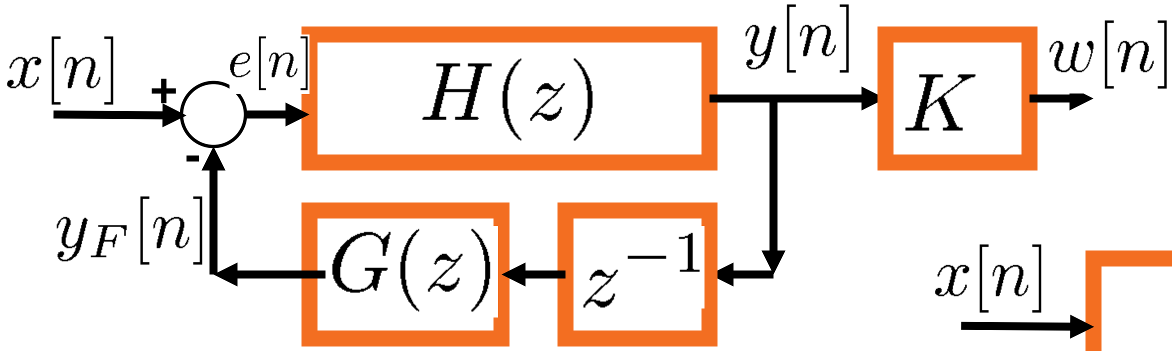


$$F(z) = \frac{K}{1 - z^{-1} + z^{-1}G(z)}$$

- By design, we choose: $G(z) = g_1 + g_2 z^{-1}$
  (this is called an FIR filter)
- Control system design means to find the constants g_1, g_2

$$F(z) = \frac{K}{1 - z^{-1} + z^{-1}\left(g_1 + g_2 z^{-1}\right)} = \frac{K}{1 + (g_1 - 1)z^{-1} + g_2 z^{-2}}$$

Making an Unstable System Stable



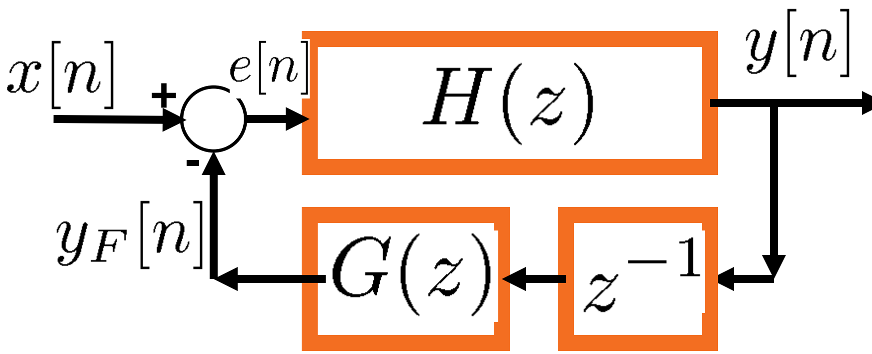
- Control system design means to find the constants g_1, g_2

$$F(z) = \frac{K}{1 - z^{-1} + z^{-1}\left(g_1 + g_2 z^{-1}\right)} = \frac{K}{1 + (g_1 - 1)z^{-1} + g_2 z^{-2}}$$

- Stability will be determined by the location (value) of the roots of the denominator

$$F(z) = \frac{K}{(1 - p_F z^{-1})(1 - p_F^* z^{-1})} = \frac{K}{1 - 2r_F z^{-1} + r_F^2 + q_F^2}$$

Making an Unstable System Stable



$$F(z) = \frac{K}{1 - z^{-1} + z^{-1}\left(g_1 + g_2 z^{-1}\right)} = \frac{K}{1 + (g_1 - 1)z^{-1} + g_2 z^{-2}}$$

$$F(z) = \frac{K}{(1 - p_F z^{-1})(1 - p_F^* z^{-1})} = \frac{K}{1 - 2r_F z^{-1} + r_F^2 + q_F^2}$$

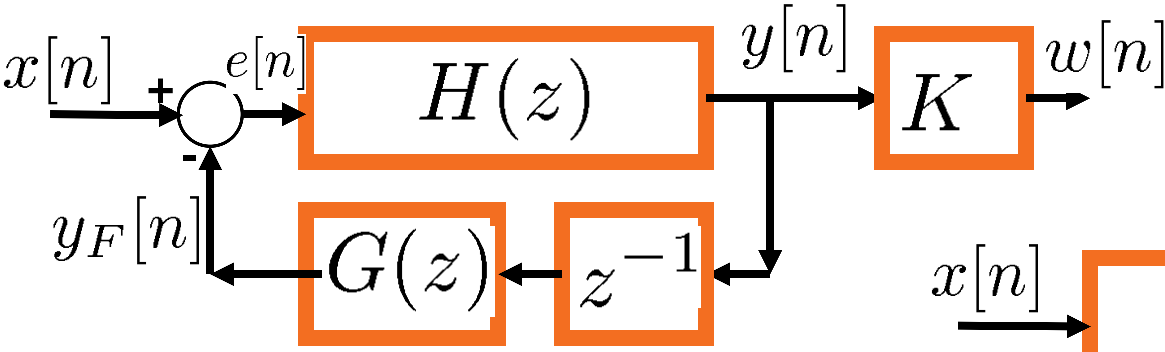
$$p_F = r_F \pm j q_F$$

$$g_1 = 1 - 2r_F$$

$$g_2 = r_F^2 + q_F^2$$

R·I·T

Making an Unstable System Stable



$$F(z) = \frac{K}{1 - z^{-1} + z^{-1}\left(g_1 + g_2 z^{-1}\right)} = \frac{K}{1 + (g_1 - 1)z^{-1} + g_2 z^{-2}}$$

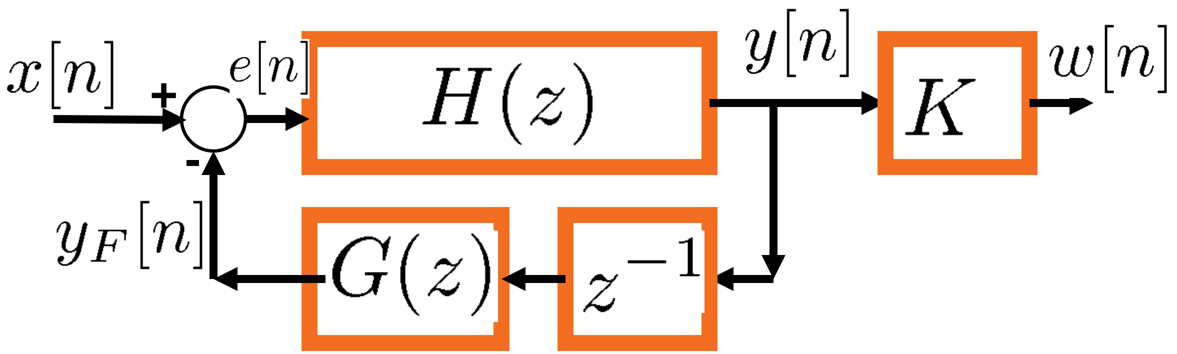
$$F(z) = \frac{K}{(1 - p_F z^{-1})(1 - p_F^* z^{-1})} = \frac{K}{1 - 2r_F z^{-1} + r_F^2 + q_F^2}$$

$$p_F = r_F \pm j q_F$$

$$g_1 = 1 - 2r_F$$
$$g_2 = r_F^2 + q_F^2$$

- The particular value of p_F is chosen based on the specific behavior we want of the system to some signals $p_F = 0.1768 + j0.1768$

Making an Unstable System Stable



- Making the filter run in a computer:

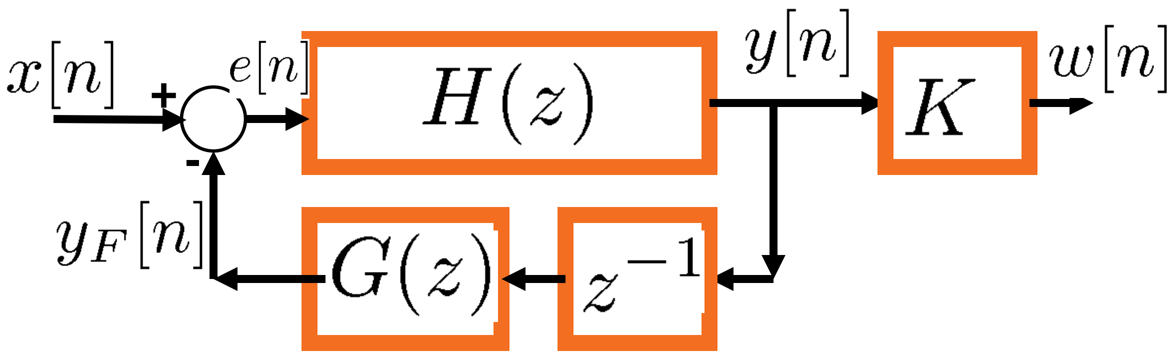
$$e[n] = x[n] - y_F[n]$$

$$Y_F(z) = z^{-1}G(z)Y(z) = (g_1 z^{-1} + g_2 z^{-2})Y(z)$$

$$y_F[n] = g_1 y[n-1] + g_2 y[n-2]$$

$$w[n] = Ky[n]$$

Making an Unstable System Stable



- Making the filter run in a computer:

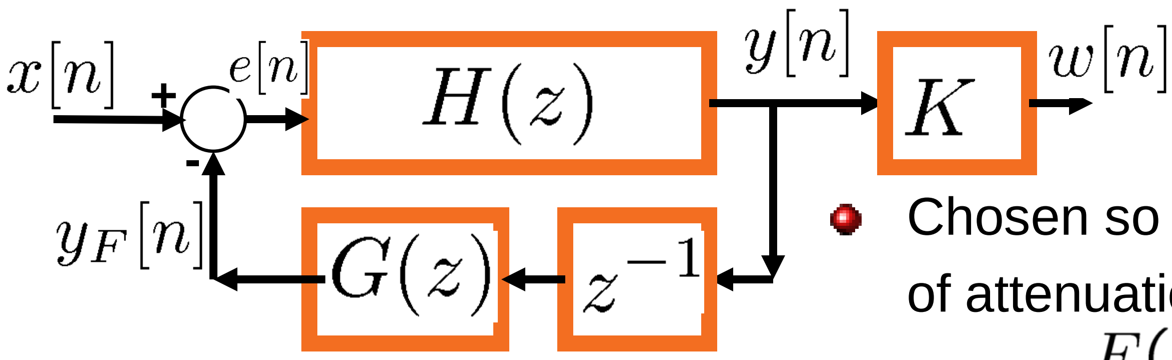
$$e[n] = x[n] - y_F[n]$$

$$Y_F(z) = z^{-1}G(z)Y(z) = (g_1 z^{-1} + g_2 z^{-2})Y(z)$$

$$y_F[n] = g_1 y[n-1] + g_2 y[n-2] \rightarrow \begin{cases} g_1 = 1 - 2r_F = 0.64 \\ g_2 = r_F^2 + q_F^2 = 1/16 \end{cases}$$

$$w[n] = Ky[n]$$

Making an Unstable System Stable



- Chosen so that there is no amplification of attenuation when input is a constant $F(z) = 1$ for $z = 1$.

- Making the filter run in a computer:

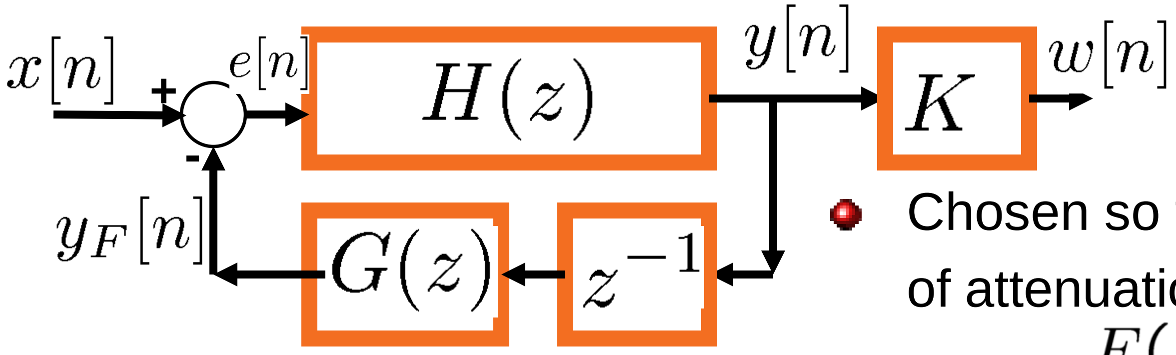
$$e[n] = x[n] - y_F[n]$$

$$Y_F(z) = z^{-1}G(z)Y(z) = (g_1 z^{-1} + g_2 z^{-2})Y(z)$$

$$y_F[n] = g_1 y[n-1] + g_2 y[n-2] \rightarrow \begin{cases} g_1 = 1 - 2r_F = 0.64 \\ g_2 = r_F^2 + q_F^2 = 1/16 \end{cases}$$

$$w[n] = Ky[n]$$

Making an Unstable System Stable



- Chosen so that there is no amplification of attenuation when input is a constant $F(z) = 1$ for $z = 1$.

- Making the filter run in a computer:

$$e[n] = x[n] - y_F[n]$$

$$Y_F(z) = z^{-1}G(z)Y(z) = (g_1 z^{-1} + g_2 z^{-2})Y(z)$$

$$y_F[n] = g_1 y[n-1] + g_2 y[n-2] \longrightarrow \begin{cases} g_1 = 1 - 2r_F = 0.64 \\ g_2 = r_F^2 + q_F^2 = 1/16 \end{cases}$$

$$w[n] = Ky[n] = 0.7089y[n]$$

Making an Unstable System Stable