The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 Solutions Version 3.0

Date: March 8, 2023 Course: EE 445S Evans

Name: **Place,** **The Good**
Last, First

- **Exam duration**. The exam is scheduled to last 75 minutes.
- **Materials allowed**. You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks**. Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers**. When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab**. No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test**. All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except the proctor for the test, and that you did not provide help, directly or indirectly, to another student taking this exam.

	Problem	*Point Value*	*Your score*	*Topic*
Jason	1	25		IIR Filter Analysis
Eleanor	2	24		Increasing the Sampling Rate
Chidi	3	27		Equalizer Design
Tahani	4	24		Potpourri
	Total	100		

Problem 1.1 *IIR Filter Analysis*. 25 points.

Consider the following causal linear time-invariant (LTI) discrete-time infinite impulse response (IIR) filter with input $x[n]$ and output $y[n]$ described by

$$y[n] - b^2\, y[n-2] = x[n]$$

for $n \geq 0$, where b is a real-valued positive coefficient less than one, i.e. $0 < b < 1$.

Please note that the coefficient in front of the $y[n-1]$ term is zero.

(a) What are the initial conditions and their values? Why? *6 points*.

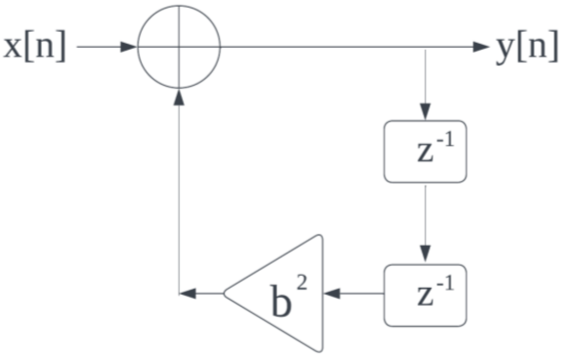
$$\textbf{Consider } \boldsymbol{n = 0}:\ \boldsymbol{y[0] = b^2\, \underbrace{y[-2]}_{\text{Initial Condition}} + x[0]}$$

$$\textbf{Consider } \boldsymbol{n = 1}:\ \boldsymbol{y[1] = b^2\, \underbrace{y[-1]}_{\text{Initial Condition}} + x[1]}$$

Our initial conditions must equal zero as necessary conditions for LTI to hold:

$$\boldsymbol{y[-2] = y[-1] = 0}$$

(b) Draw the block diagram of the filter relating input $x[n]$ and output $y[n]$. *6 points*.



(c) Derive a formula for the transfer function in the z-domain and the region of convergence. *4 points*.

$$\begin{aligned}
&\boldsymbol{Y(z) = b^2 z^{-2}\, Y(z) + X(z)} && \textbf{take z-Transform of } \boldsymbol{y[n] = b^2\, y[n-2] + x[n]} \\
&\boldsymbol{Y(z) - b^2 z^{-2}\, Y(z) = X(z)} && \\
&\boldsymbol{Y(z)(1 - b^2 z^{-2}) = X(z)} && \textbf{combine like terms} \\
&\boldsymbol{H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - b^2\, z^{-2}}} && \textbf{recall the transfer function is } \boldsymbol{H(z) = \frac{Y(z)}{X(z)}}
\end{aligned}$$

poles are at $\boldsymbol{z = b}$ and $\boldsymbol{z = -b}$.

$$\boldsymbol{H(z) = \frac{1}{(1-b\, z^{-1})(1+b\, z^{-1})}, for\ |z| > b} \qquad \textbf{final answer}$$

(d) Give a formula for the discrete-time frequency response of the filter. *3 points*.

We substitute $z = e^{j\omega}$, because the unit circle is contained inside the Region of Convergence $|z| > b$ where $0 < b < 1$:

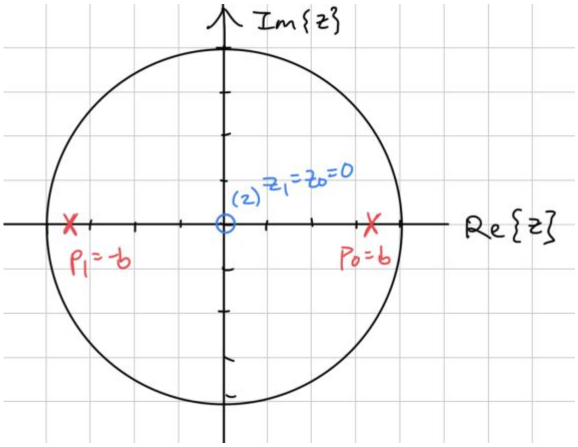
$$H(e^{jw}) = \frac{1}{(1 - be^{-j\omega})(1 + be^{-j\omega})} = \frac{1}{1 - b^2 e^{-j2\omega}}$$

(e) What frequency responses are possible among lowpass, highpass, bandpass, bandstop, allpass and notch? For each possibility, give a value of *b* that would give that response. *6 points*.

From our transfer function, $H(z) = \frac{1}{1-b^2z^{-2}} = \frac{z^2}{z^2-b^2}$

Thus, we know that…

- **2 zeros exist at $z = 0$**
- **2 poles exist at $p = \pm b$**



Consider as b→1

The 2 poles p_0→$1, p_1$→-1

(Note that the 2 poles are not equal to 1! Because then the poles would be on the unit circle, and we would encounter Bounded-Input Bounded-Ouput instability issues.)

With these pole locations, it means the poles are amplifying low and high frequencies.

Thus, we can see that this as a bandstop filter.

Consider as b→0

The 2 poles approach $p_0 = p_1 = 0$

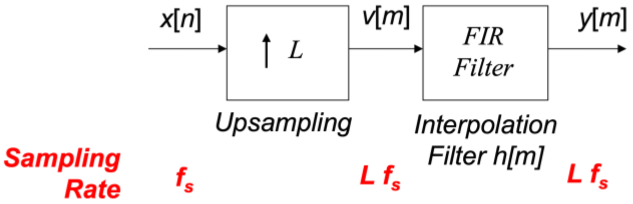
Now the 2 zeros and the 2 poles are on the same location, cancelling out.

Thus, we can see that this as an allpass filter.

Problem 1.2 *Increasing the Sampling Rate.* 24 points.

Upsampling by L can be used to increase the sampling rate of the input signal by a factor of L.

A lowpass finite impulse response (FIR) filter can then be applied to the output of the upsampler to attenuate the high frequencies introduced by upsampling.



On the right, discrete-time index n is associated with sampling rate f_s and discrete-time index m is associated with sampling rate $L\ f_s$.

(a) What is the maximum continuous-time frequency f_{max} that is present in $x[n]$? What discrete-time frequency does f_{max} correspond to? *6 points.*

$$\boldsymbol{x(t) = \cos(2\pi f_0 t)\,, \qquad for - \infty < t < \infty}$$

$$\boldsymbol{x(t) = \cos\left(\underbrace{2\pi f_0 \left(\frac{n}{f_s}\right)}_{\omega_0 = 2\pi \frac{f_0}{f_s}}\right)}$$

From Nyquist Theorem, we require that $f_s > 2f_{max}$
Thus, we have the continuous-time frequency is:

$$\boldsymbol{f_{max} < \frac{1}{2} f_s}$$

And the corresponding discrete-time frequency is:

$$\boldsymbol{\omega_{max} = 2\pi \frac{f_{max}}{f_s} = 2\pi \frac{(\frac{1}{2} f_s)}{f_s} = \pi}$$

Thus, from sampling, we can capture a frequency range of…

$$\boldsymbol{-\frac{1}{2} f_s < f < \frac{1}{2} f_s \ (continuous\ time\ frequencies)}$$

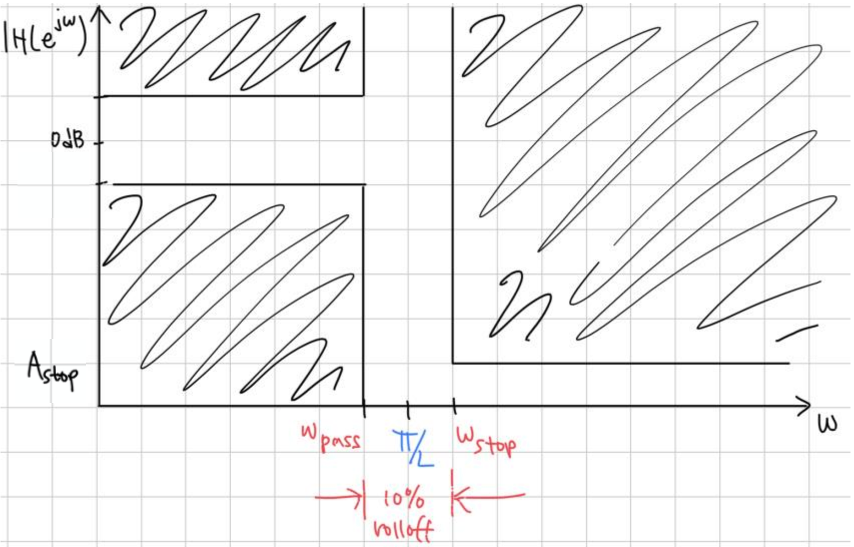
$$\boldsymbol{-\pi < \omega < \pi \ (discrete - time\ frequencies)}$$

(b) What discrete-time frequency in $v[m]$ corresponds to the maximum continuous-time frequency f_{max} that is present in $x[n]$? *6 points.*

From our solution in part (a), $f_{max} < \frac{1}{2} f_s$
And we are given that at $v[m]$, our new sampling rate is Lf_s
Using the same equation for discrete-time frequency, we find that our solution is:

$$\boldsymbol{\omega_{\max\ \_at\_v[m]} = 2\pi \frac{f_{max}}{f_{s\_at\_v[m]}} = 2\pi \frac{(\frac{1}{2} f_s)}{Lf_s} = \frac{\pi}{L}}$$

(c) Any discrete-time frequencies present in $v[m]$ higher than your answer in part (b) but less than π rad/sample correspond to frequencies introduced by upsampling. Give the discrete-time passband frequency ω_{pass} and stopband frequency ω_{stop} you would use for the lowpass filter design. *6 points*.



To determine our $\boldsymbol{\omega_{\text{pass}}}$ and $\boldsymbol{\omega_{\text{stop}}}$ values, we want to consider a reasonable value of 10% for the roll-off factor. Meaning, $\boldsymbol{\omega_{\text{stop}} = 1.1\omega_{\text{pass}}}$

Thus, our values result in…

$$\boldsymbol{\omega_{\text{pass}} = 0.95\frac{\pi}{L}}$$

$$\boldsymbol{\omega_{\text{stop}} \cong 1.05\frac{\pi}{L}}$$

(d) Give two ways to design a lowpass FIR filter with linear phase to meet the specifications of a discrete-time passband frequency ω_{pass} and stopband frequency ω_{stop}. A “way” to design the filter could be a formula, algorithm, etc. *6 points*.

Solution I:

Design an averaging filter with N coefficients

The first zero should occur at $\boldsymbol{\frac{2\pi}{N}}$

To determine N, we know that $\boldsymbol{\frac{2\pi}{N} = \frac{\pi}{L}}$

Thus, $\boldsymbol{N = 2L}$

Solution II:

Use filterDesigner from MATLAB
Use the design methods such as Parks-McClellan, Least-Squares, or Kaiser Window

Solution III:

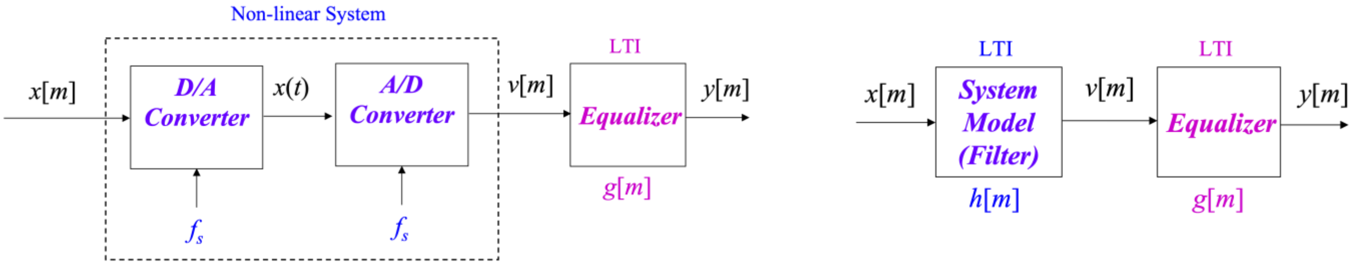
Place zeros on a pole-zero plot for a low-pass filter

Problem 1.3 *Equalizer Design.* 27 points.

Many applications use a digital-to-analog (D/A) converter and an analog-to-digital (A/D) converter.

- An audio system would use a D/A converter for playback over earbuds or speakers and an A/D converter in a microphone for recording and mixing.
- A digital communication system would use a D/A converter in the transmitter and an A/D converter in the receiver.

You'll design a linear time-invariant (LTI) bound-input bounded-output (BIBO) stable discrete-time equalizer to compensate the distortion in the nonlinear system (on the left) using an LTI model of the nonlinear system (on right):



(a) Give a formula for a test signal to use for *x*[*m*] that would be of finite duration. The test signal should have all discrete-time frequencies in it. The test signal would be used to estimate the impulse response *h*[*m*] for the LTI model of the cascade of the D/A and A/D converters. *12 points*.

Solution I: **Use a chirp signal:** $x(t) = \cos\underbrace{(2\pi f_0 t + \pi \mu t^2)}_{\theta(t)}$

The chirp will sweep continuous-time frequencies:

$$0\ Hz \rightarrow f_{max} = \frac{1}{2} f_s \text{ Hz}$$

This chirp signal would sweep discrete-time frequencies from $0\ rad/sample \rightarrow \pi\ rad/sample$

where the instantaneous frequency is:

$$\frac{d\theta(t)}{dt} = 2\pi f_0 + 2\pi\mu t = 2\pi\underbrace{(f_0 + \mu t)}\ rad/s$$

$$f_i(t) = f_0 + \mu t,\ f_0 = 0 \text{ Hz}$$

$$f_{max} = \frac{1}{2} f_s = \mu t_{max}$$

Solution II:

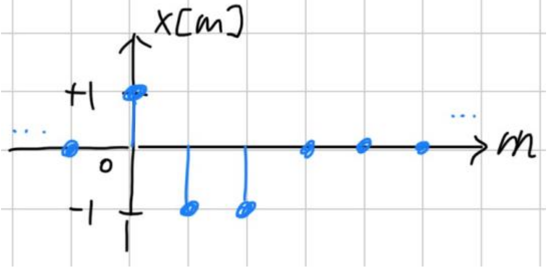
Use a PN (Pseudo-Random Noise) Sequence because it covers all frequencies

- **r^{th}-order connection polynomial**
- **$2^r - 1$ bits for fundamental period**
  We subtract 1 because the state of all zeros is a lockout state that does not change

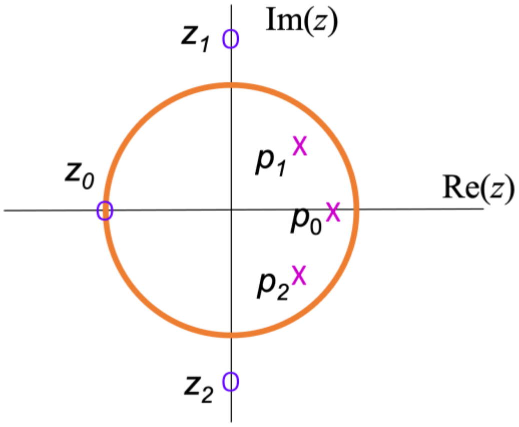
Solution III:

Use an impulse response: $x[m] = \delta[m]$
In theory this is a good/feasible idea.
In practice, it does not capturing variability over time in the unknown system.
Note: Impulsive events used in seismic modelling of acoustic layers (eg., earth layers)

Solution IV: **Use a special case of a short PN sequence**



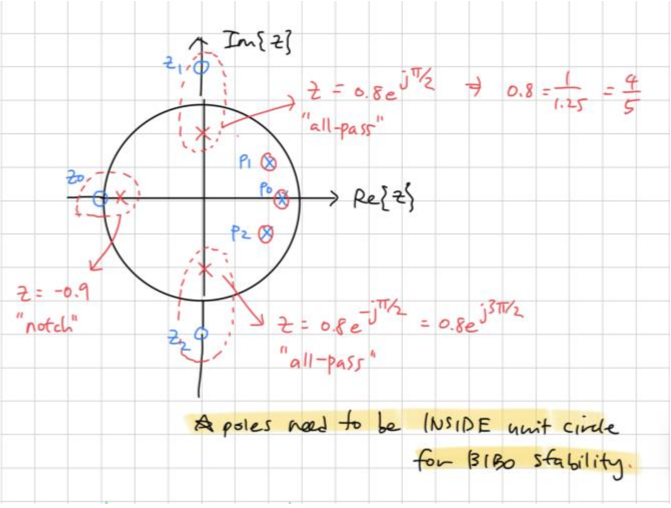
(b) Given the poles and zeros of $H(z)$ below, give the values for the poles and zeros of $G(z)$ and draw them on the pole-zero diagram on the right for an LTI equalizer $G(z)$ that would make the cascade of the LTI system model $H(z)$ and the LTI equalizer $G(z)$ be allpass. *15 points*.



$$p_0 = 0.9$$
$$p_1 = 0.9\, e^{j\frac{\pi}{4}}$$
$$p_2 = 0.9\, e^{-j\frac{\pi}{4}}$$

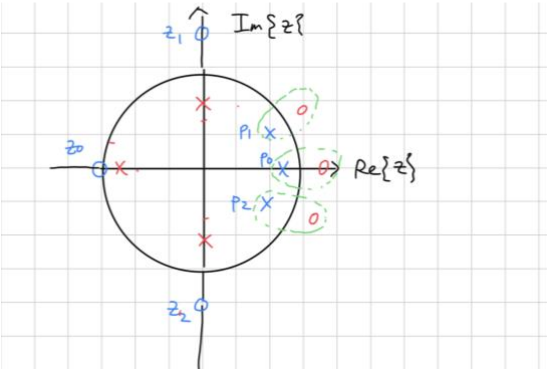
$$z_0 = -1$$
$$z_1 = 1.25\, e^{j\frac{\pi}{2}}$$
$$z_2 = 1.25\, e^{-j\frac{\pi}{2}}$$

Solution I


Im{z}
Re{z}
z_1
z_0
z_2
p_1
p_0
p_2
$z = 0.8e^{j\pi/2} \Rightarrow 0.8 = \frac{1}{1.25} = \frac{4}{5}$
"all-pass"
$z = -0.9$
"notch"
$z = 0.8e^{-j\pi/2} = 0.8e^{j3\pi/2}$
"all-pass"

⚠ poles need to be INSIDE unit circle for BIBO stability.

Solution II



Problem 1.4. *Potpourri*. 24 points.

(a) An integrator is a common building block in systems. The discrete-time version of the integrator is the running summation, which is defined for input $x[n]$ and output $y[n]$ for $n \geq 0$ as follows:

$$y[n] = \sum_{m=0}^{n} x[m]$$

The summation requires unbounded memory as $n \to \infty$. A more efficient implementation is

$$y[n] = y[n-1] + x[n] \ \text{ for } \ n \geq 0$$

I. Give the initial condition(s) for the more efficient implementation to be linear and time-invariant (LTI). *4 points*.

Rather than using the summation equation,

$$\boldsymbol{y[n] = \sum_{m=0}^{n} x[m]}$$

It is simpler to solve for the initial conditions using the equivalent recursive difference equation instead, $\boldsymbol{y[n] = y[n-1] + x[n] \ \text{ for } \ n \geq 0}$

Consider $\boldsymbol{n = 0}$: $\boldsymbol{y[0] = \underbrace{y[-1]}_{\text{Initial Condition}} + x[0]}$

In order for LTI to hold, our initial conditions must equal zero:

$$\boldsymbol{y[-1] = 0}$$

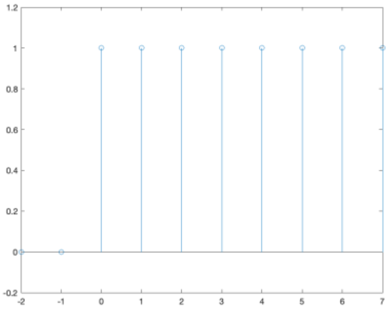
II. Give a formula for and plot the impulse response $h[n]$ of an LTI running summation. *4 points*

This time, let's use the summation form,

$$\boldsymbol{y[n] = \sum_{m=0}^{n} x[m]}$$

Let $\boldsymbol{x[m] = \delta[m]}$ **and then** $\boldsymbol{h[n] = y[n]}$**:**

$$\boldsymbol{h[n] = \sum_{m=0}^{n} \delta[m] = \begin{cases} 1, if\ n \geq 0 \\ 0, \quad else \end{cases}}$$



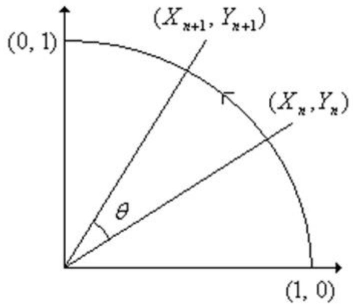
III. The running summation only gives an unbounded output for a bounded input when the input has a non-zero constant component. What filter would you apply before the running summation to prevent the running summation being bounded-input bounded-output unstable? *4 points*

Use a DC notch filter to remove the non-zero constant at 0 radians/sample

(b) There are several algorithms [1] to generate a cosine signal $x[n] = \cos(\omega_0\, n)$ and a sine signal $y[n] = \sin(\omega_0\, n)$ at the same time using rotation. For each signal, the argument is $0, \omega_0, 2\omega_0, \ldots,$ for $n = 0, 1, 2, \ldots$

Using the visual representation on the right, the next cosine value X_{n+1} and next sine value Y_{n+1} are computed from the current cosine value X_n and current sine value Y_n using a rotation operation:

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix}$$



Matrix Rotation [1]

Using rotation keeps the values of the cosine and sine for the same angle on the unit circle.

With $\theta = \omega_0$, we use the rotation approach starting at $n = 0$ and pre-compute $\cos(\theta)$ and $\sin(\theta)$.

Note: This is called the Coordinate Rotation Digital Computer (CORDIC) Algorithm. "The algorithm was used in the navigational system of the Apollo program's Lunar Roving Vehicle to compute bearing and range, or distance from the Lunar module."

I. What are the values of X_0 and Y_0? *4 points*.

Consider at $n = 0$: $\cos(\omega_0 n) = \cos(0) = 1$

$\text{Sin}(\omega_0 n) = \sin(0) = 0$

Thus, $X_0 = 1, Y_0 = 0$

II. How many multiplications per output sample are needed to compute the cosine and sine signals? *4 points.*

For the matrix operation, we have a 2x2 matrix multiplied by a 2x1 vector.

To compute X_{n+1} (cosine component), we need 2 multiplications

$$X_{n+1} = \cos(\theta)\, X_n + (-\sin(\theta))Y_n$$

To compute Y_{n+1} (sine component), we need 2 multplications

$$Y_{n+1} = \sin(\theta)\, X_n + \cos\,(\theta)Y_n$$

So, we require a total of 4 multiplications (and 4 additions) per output sample.

III. Compare your answer in part II to using second-order difference equations to compute cosine and sine signals separately. *4 points.*

Consider the difference equation for the cosine signal:

$$y[n] = (2\cos(\omega_0))\, y[n-1] - y[n-2] + x[n] - (\cos(\omega_0))\, x[n-1]$$

Consider the difference equation for the sine signal:

$$y[n] = (2\cos(\omega_0))\, y[n-1] - y[n-2] + (\sin(\omega_0))\, x[n-1]$$

We see that with this format, both cosine and sine require around 2 multiplications and 4 additions, for a total of 4 multiplications and 8 additions.

In comparison, the CORDIC algorithm only requires 4 multiplications and 4 additions, and so gives us computational savings.