SECTION 2.

Developing Constructs

2 Developing Constructs.

Before we are ready to leave to analog shore, before we can cast off the dock lines and sail the digital domain, we must first gather tools and become acquainted with the helm.

To do this we will start out with some basic constructs. These constructs will be used as building blocks later on, so that we can specify the shape our curve in the frequency domain and eliminate or at least minimize the need to do complex Fourier and z – transforms.

2.1 Basic constructs.

Four basic constructs will be introduced. These constructs can be used to derive almost any practical control loop. Derived constructs will be developed later on. Developing these will give us some practice using the basic constructs, and they may be of use implementing some practical system in our future experience.

2.1.1 Construct 1 – The Low pass filter – Flat gain transitioning to a pole at f_b. (LPF)

This was already developed in section 1, so there is no need to repeat it here. We'll simply re-iterate the results for easy reference.

$$H(z) = \frac{b_0}{1 - a_1 \cdot z^{-1}} \qquad \text{Where} \quad \alpha = 2\pi f_b$$

$$a_1 = e^{-\alpha \cdot T_m} \qquad b_0 = T_m \cdot \alpha$$

2.1.2 Construct 2 – Zero frequency Pole – (ZFP)

The second construct will be very similar to the first. It will be an integrator with a pole at zero frequency. Analog designers recognize that they use and inverting integrator as their first stage error amplifier. In the frequency domain, the non-inverting flavor will take the following form.

$$H(\omega) = \frac{\alpha_g}{j \cdot \omega}$$

where α_g is a constant that sets the gain.

This equation has a constant slope of -20db per decade, a constant phase shift of $\pi/2$ and a gain of 1 (0db) where $\omega = \alpha_g$.

Taking the liberty to look ahead to a target application and having determined a few values we will use for this example, I would like the 0db point for this stage to be 10 kHz, and I have selected the math frequency (F_m) to be 40 kHz. The value for α_g is can now determined.

$$F_{unity} = 10k \qquad \alpha_g = 2\pi F_{unity} \qquad \alpha_g = 6.283 \times 10^4$$

Following the steps of the example in Section 1, we will now take the inverse Fourier transform to get the time domain impulse response of the system, in this case it is too easy to do any other way other than by inspection (or lookup table).

$$h(t) = -\alpha_g \cdot u(t)$$

Simple substitution is applied to bring this into the discrete time domain.

$$h(n) = -T_m \cdot \alpha_g \cdot u(n \cdot T_m)$$

Now this can be converter into the z – domain in order to simplify the math by avoiding the need to apply convolution theorem. Once again, this example is easy enough to do by inspection.

$$H(z) = T_m \cdot \alpha_g \cdot \frac{1}{1 - z^{-1}}$$

You should note that the results here have the same form as Sequence 1 if we make the following substitutions.

$$a_1 = 1 \qquad b_0 = T_m \cdot \alpha_g \qquad b_0 = 1.571$$

In this case, the form of the answer will be identical to Sequence 1 with different coefficients for a_1 and b_0.

$$g_n = b_0 \cdot f_n - a_1 \cdot g_{n-1}$$

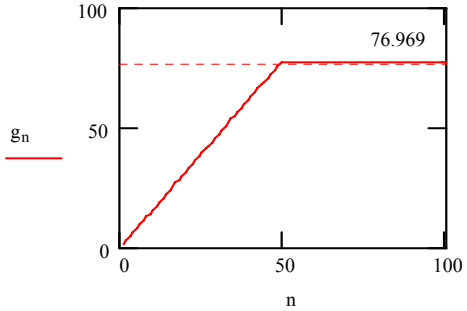
We should now plot the discrete domain response. My tool of choice for this is MathCad®.

$n = 1 .. 100$

$f_n = u(n) - u(n - 50) \qquad g_0 = 0$

$g_n = b_0 \cdot f_n + a_1 \cdot g_{n-1}$

The digital time domain response is plotted.

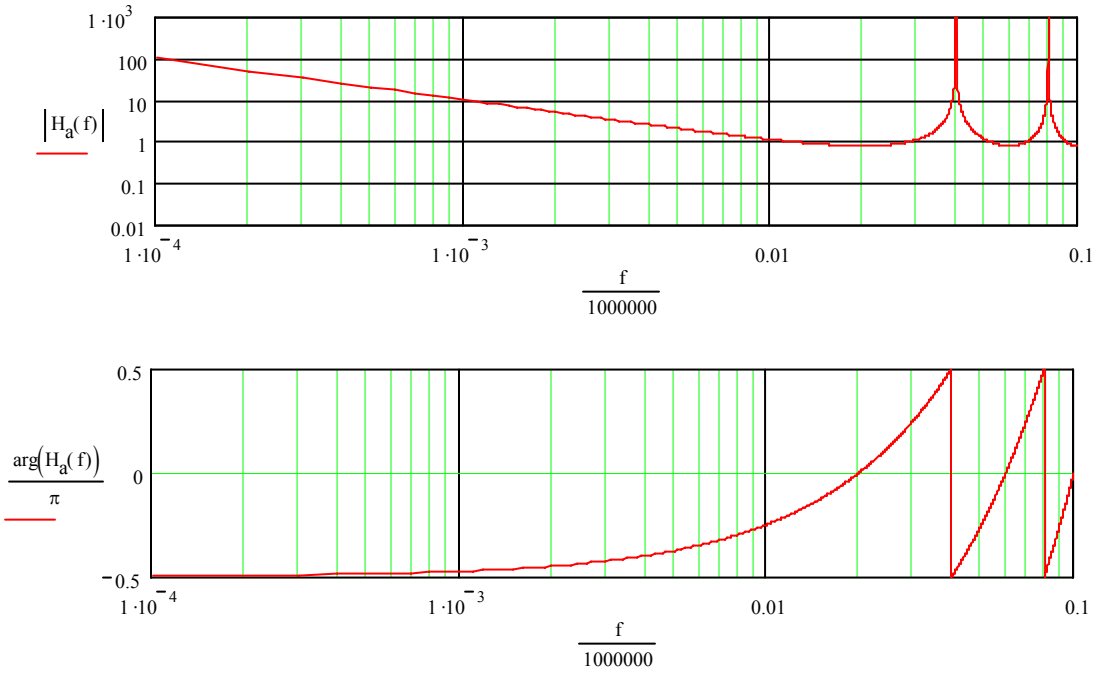


The value that the integrator converges to is 50 times b_0, this is the exact expected value for this integrator at n = 50.

The frequency domain response is plotted.

$$H_a(f) = H\left(e^{j\cdot 2\pi\cdot f\cdot T_m}\right)$$

$$f = \frac{f_b}{100}, \frac{f_b}{50} .. 10\cdot f_b$$



Here we see a fairly accurate representation of the desired result with the exception that the process breaks down in terms of gain at about ½ the math frequency.

2.1.3 Construct 3 – Zero frequency Zero (ZFZ)

A third useful construct will be a differentiator, otherwise known as a zero in the frequency domain. A zero taken by itself is rarely analyzed and practically of no use by itself, however as a construct it will be very useful.

The frequency domain representation of the single zero is…

$$H(\omega) = \frac{j \cdot \omega}{\alpha_g} \quad \text{Where} \quad \alpha_g = 2\pi f_b \quad \text{and} \quad f_b = 100$$

With ω in the numerator, a few steps are required to properly evaluate the term. Allowing myself to cheat a little, I will conclude that the result in the z – domain will boil down to the inverse of the integrator since they are inverse in the frequency domain.

$$H(z) = \frac{1 - z^{-1}}{T_m \cdot \alpha_g}$$

We can set the coefficients $b_0 = \frac{1}{T_m \cdot \alpha_g}$ and $b_1 = \frac{-1}{T_m \cdot \alpha_g}$

In the discrete time domain, we get …

$$g_n = b_0 \cdot f_n + b_1 \cdot f_{n-1}$$

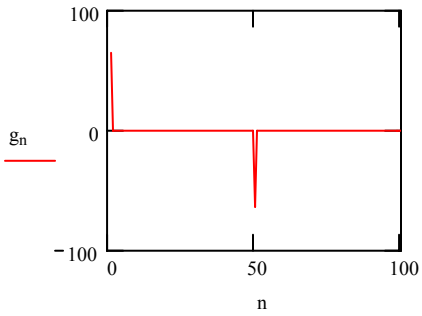
One should note that this represents a finite impulse response (FIR) filter, where as the previous examples were infinite impulse response (IIR) filters. The main difference is that the output of a FIR does not depend on previous states of the output.

The discrete time domain response is plotted.

$n = 1..100$

$f_n = u(n) - u(n - 50)$

$g_n = b_0 \cdot f_n + b_1 \cdot f_{n-1}$

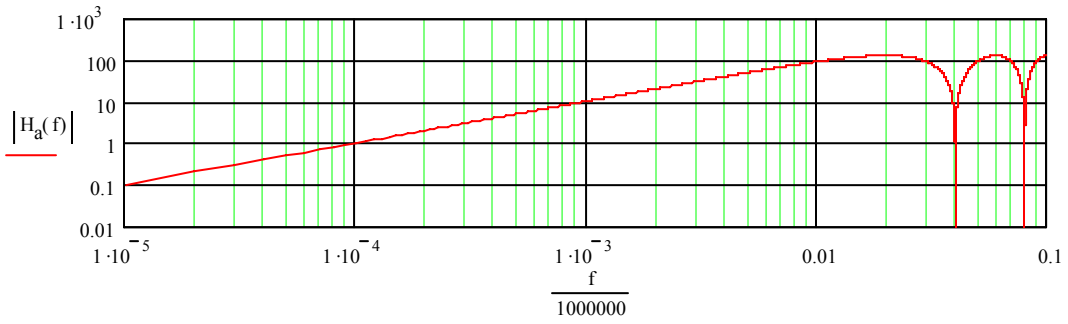


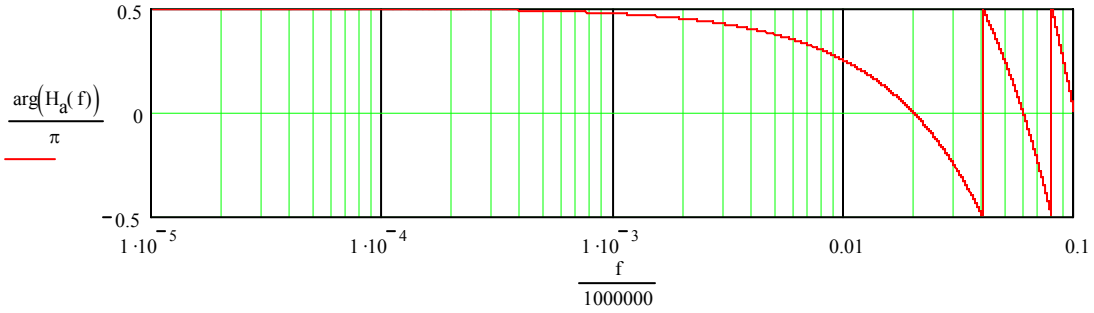
The output is an ideal differentiator in the discrete domain.

The frequency domain response of the differentiator is plotted.

$$H_a(f) = H\left(e^{j\cdot 2\pi\cdot f\cdot T_m}\right)$$

$$f = \frac{f_b}{10}, \frac{f_b}{5} .. 1000\cdot f_b$$





Here the gain is fairlyaccurate out to Tm / 2 , however there significant phase error even at
Tm / 4.

2.1.4 Construct 4 - Flat gain transitioning to a zero at f_b. (f_bZ)

This construct, along with the LPF, will become one of the major building blocks that can be used for any control loop. Anytime it is desired to introduce a zero at some break frequency, this construct can be used.

This construct will be created simply by inverting a LPF which is a flat gain transitioning to a pole.

$$H_{fbZ}(z) = \frac{1 - e^{-\alpha_g \cdot T_m} \cdot z^{-1}}{\alpha_g \cdot T_m} \qquad \text{where} \qquad \alpha_g = 2\pi f_b$$

We need to simplify the expression to get a leading "1" in the denominator.

$$H_{fbZ}(z) = \frac{\frac{1}{\alpha_g \cdot T_m} + \frac{-e^{-\alpha_g \cdot T_m}}{(\alpha_g \cdot T_m)} \cdot z^{-1}}{1}$$

Now we can just pick out the coefficients.

$$a_1 = 0 \qquad b_0 = \frac{1}{\alpha_g \cdot T_m} \qquad b_1 = \frac{-e^{-\alpha_g \cdot T_m}}{(\alpha_g \cdot T_m)}$$

The discrete time domain representation can now be written as…

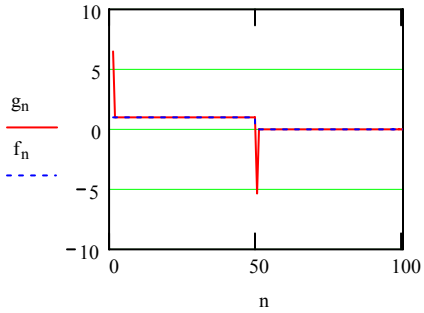
$$g_n = b_0 \cdot f_n + b_1 \cdot f_{n-1}$$

Values are plugged in to demonstrate the results. The break frequency value is set to 1 kHz and the math frequency value is set to 40 kHz.

$n = 1..100$

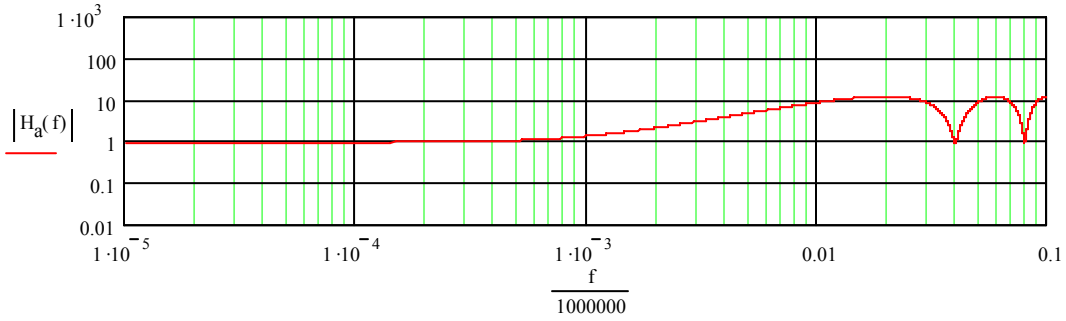
$f_n = u(n) - u(n - 50)$

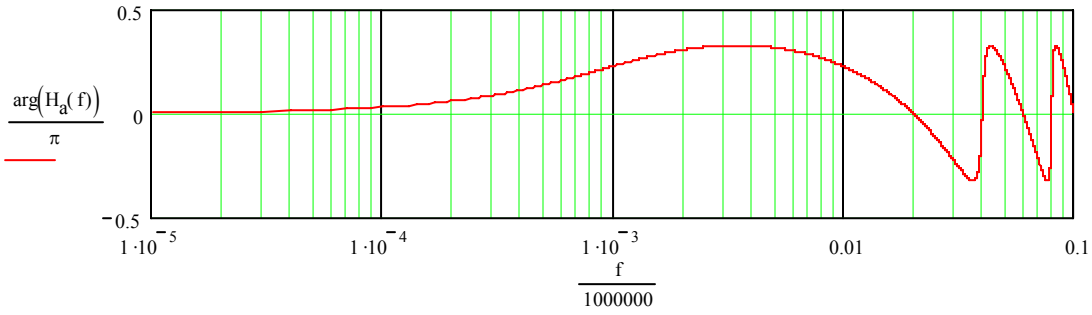
$g_n = b_0 \cdot f_n + b_1 \cdot f_{n-1}$



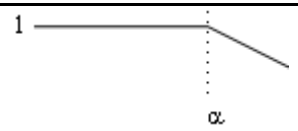
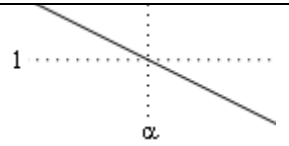
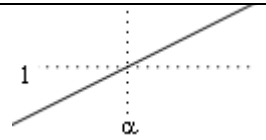
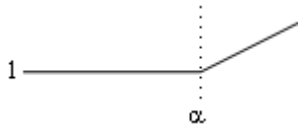
$H_a(f) = H_{fbZ}\left(e^{j \cdot 2\pi \cdot f \cdot T_m}\right)$

$f = \frac{f_b}{100}, \frac{f_b}{50} .. 100 \cdot f_b$





2.2 Summary of Basic Constructs

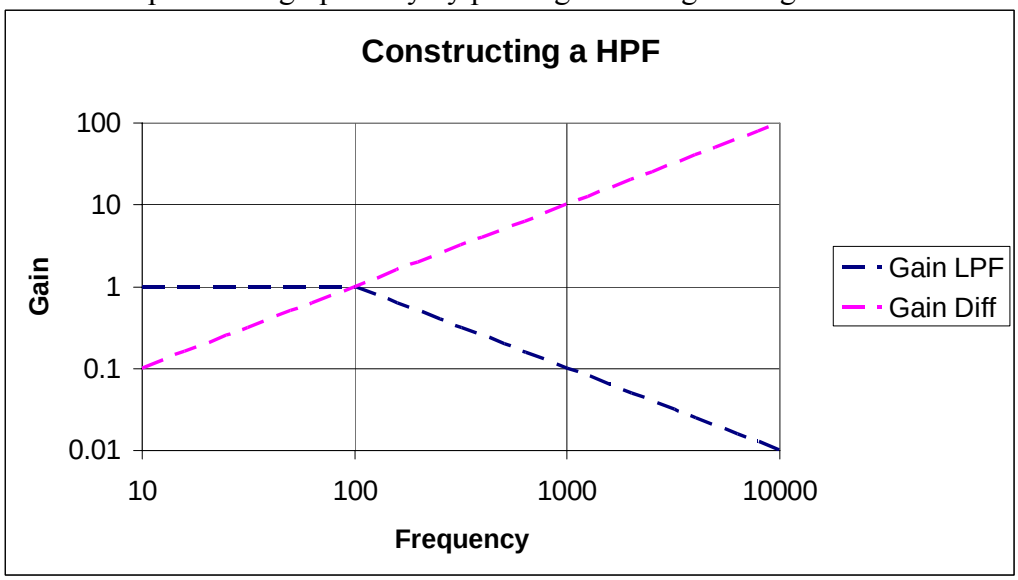
Construct	Description	z - domain function	Coefficients			Frequency domain graphic
			a1	b0	b1	
1	LPF Low Pass Filter	$H(z) = \frac{b_0}{1 - a_1 \cdot z^{-1}}$	$a_1 = e^{-\alpha \cdot T_m}$	$b_0 = \alpha \cdot T_m$	-	
2	ZFP Zero frequency pole	$H(z) = \frac{b_0}{1 - a_1 \cdot z^{-1}}$	$a_1 = 1$	$b_0 = \alpha \cdot T_m$	-	
3	ZFZ Zero frequency zero	$H(z) = b_0 + b_1 \cdot z^{-1}$	-	$b_0 = \frac{1}{\alpha \cdot T_m}$	$b_1 = \frac{-1}{\alpha \cdot T_m}$	
4	f_bZ Frequency f_b zero	$H(z) = b_0 + b_1 \cdot z^{-1}$	-	$b_0 = \frac{1}{\alpha \cdot T_m}$	$b_1 = \frac{-e^{-\alpha \cdot T_m}}{\alpha \cdot T_m}$	

2.3 Derived Constructs.

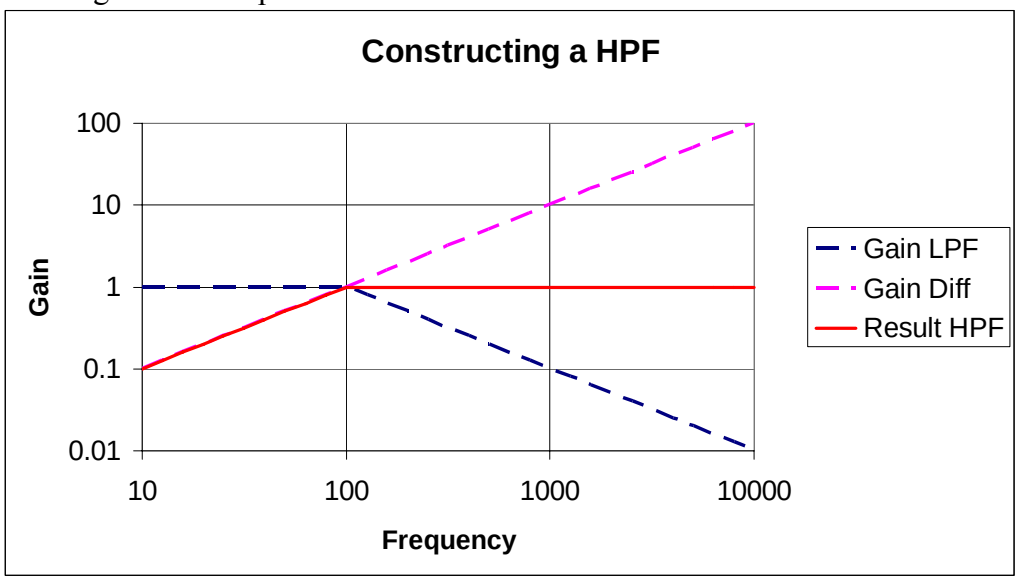
2.3.1 Building a High Pass Filter (HPF)

The first derived construct will be a single zero high pass filter (HPF). As a derived construct, it will be made of two of the previous basic constructs. We will multiply a differentiator and a LPF to achieve the results.

The idea is presented graphically by plotting the two gain stages.



The stages are multiplied to find the result.



Applying this to the z – domain, we need the two constructs.

Zero frequency zero (ZFZ) Construct

$$H_{zero}(z) = \frac{1 - z^{-1}}{T_m \cdot \alpha_g}$$

Low Pass Filter (LPF) Construct

$$H_{LPF}(z) = T_m \cdot \alpha_g \cdot \frac{1}{1 - e^{-\alpha_g \cdot T_m} \cdot z^{-1}}$$

Where $\alpha_g = 2\pi f_b$ and $T_m = \frac{1}{F_m}$

In this case we'll let the break frequency, f_b, be 100 Hz and the math frequency, F_m, be 40 kHz.

Using the construct method we multiply the basic constructs to give us the z – domain result.

$$H_{HPF}(z) = H_{zero}(z) \cdot H_{LPF}(z)$$

Some simple algebra provides our familiar form.

$$H_{HPF}(z) = \frac{1 - z^{-1}}{1 - e^{-\alpha_g \cdot T_m} \cdot z^{-1}}$$

The coefficients fall out $a_1 = e^{-\alpha_g \cdot T_m}$ $b_0 = 1$ $b_1 = -1$

We can write the discrete time domain representation.

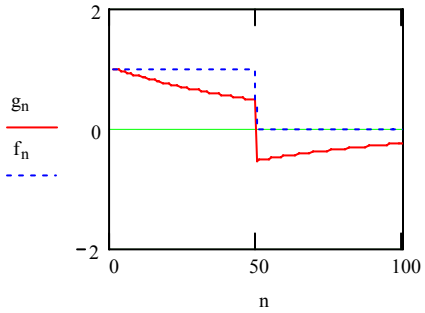
$$g_n = f_n - f_{n-1} + a_1 \cdot g_{n-1}$$

Once again the discrete time domain response is plotted.

$n = 1..100$

$f_n = u(n) - u(n - 50)$

$g_n = f_n - f_{n-1} + a_1 \cdot g_{n-1}$

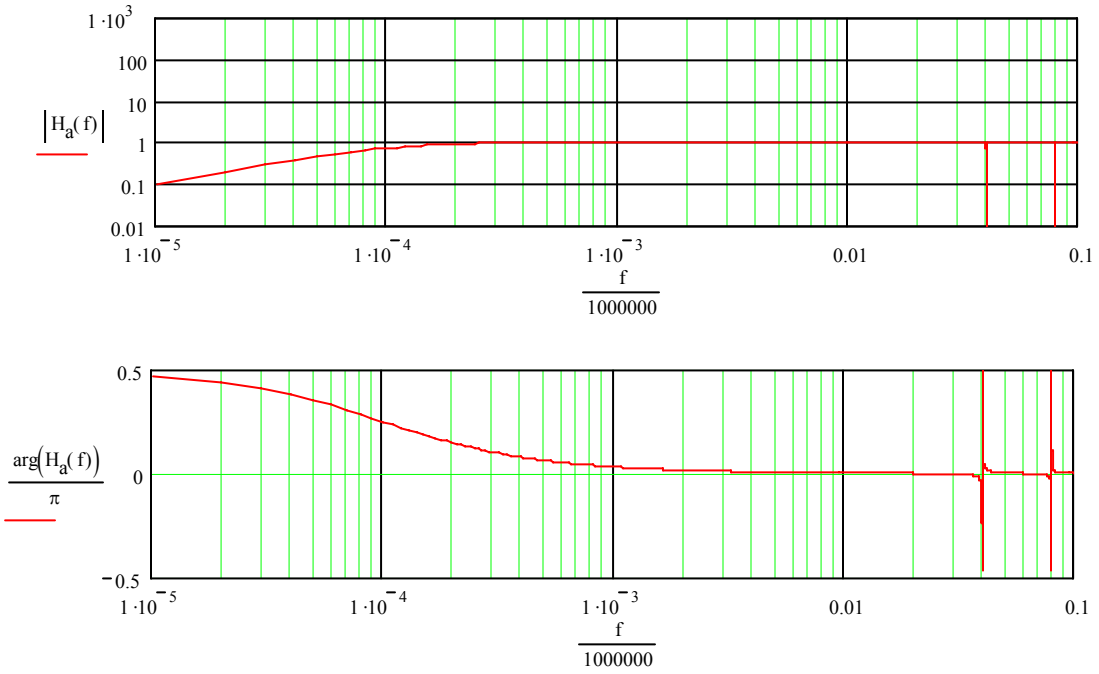


This result matches the output expected for a high pass filter.

The high pass filter is plotted in the frequency domain.…

$$H_a(f) = H_{HPF}\left(e^{j\cdot 2\pi\cdot f\cdot T_m}\right)$$

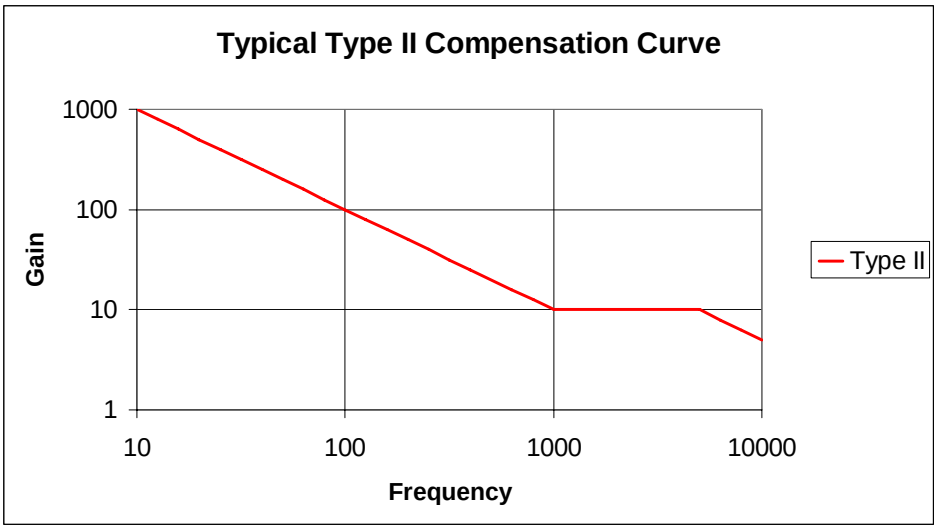
$$f = \frac{f_b}{10}, \frac{f_b}{5} .. 1000\cdot f_b$$



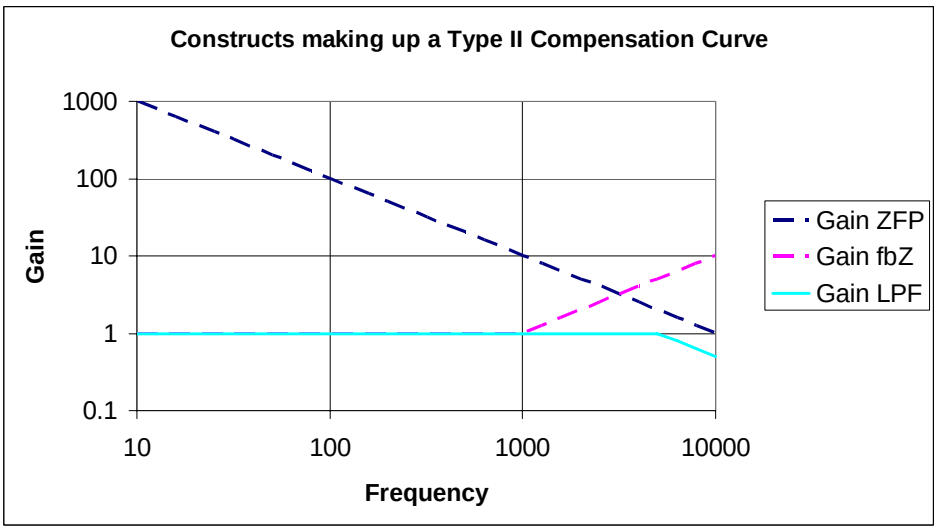
Here we see a single zero HPF with a breakpoint at the target frequency of 100 Hz. There is very little error in gain and phase out to 30 kHz.

2.4 Applying Construct Method to Derive a Type II Compensation Network.

A Type II compensation network is a common control scheme implemented in control systems. This next example will demonstrate a practical application of the construct method to derive a Type II network. The network "type" refers to the number of poles and zeros and the order in which they occur. In the frequency domain a type II network is a zero frequency pole followed by a zero then a pole, all multiplied by -1 to give 180 degree phase shift. The following figure represents the frequency domain response of a typical type II network.



This curve can be made of the following basic constructs.



2.4.1 Using the z – domain constructs to build the Type II network.

Let's go to work right in the z – domain.

We need to write down the z – domain transfer functions of the construct we are using. In this case they are as follows.

$$H_{ZFP}(z) = \frac{\alpha_1 \cdot T_m}{1 - z^{-1}} \qquad H_{fbZ}(z) = \frac{1}{\alpha_2 \cdot T_m} + \frac{-e^{-\alpha_2 \cdot T_m}}{(\alpha_2 \cdot T_m)} \cdot z^{-1} \qquad H_{LPF}(z) = \frac{\alpha_3 \cdot T_m}{1 - e^{-\alpha_3 \cdot T_m} \cdot z^{-1}}$$

The break frequencies or α's need to be defined. In this case, we can pick them off of the previous graph.

$$\alpha_1 = 2\pi 10k \qquad \alpha_2 = 2\pi 1k \qquad \alpha_3 = 2\pi 5k$$

Now we just need to choose our desired math frequency. We'll stay with 40 kHz for now.

$$T_m = \frac{1}{40k}$$

To come up with our new transfer function, we simply need to multiply these three stages and invert.

$$H_{T2}(z) = -H_{ZFP}(z) \cdot H_{fbZ}(z) \cdot H_{LPF}(z)$$

At this point, quite a bit of algebra would be required. We wish to make things easy, so we will use the "simplify" symbolic function in MathCad®.

$$H_{T2}(z) \text{ simplify } \rightarrow \frac{-5}{2} \cdot \pi \cdot z \cdot \frac{\left(z - \exp\left(\frac{-1}{20} \cdot \pi\right)\right)}{\left[(z-1) \cdot \left(z - \exp\left(\frac{-1}{4} \cdot \pi\right)\right)\right]}$$

With some minimal effort, this can be manipulated to our normal form.

$$H_{T2}(z) = \frac{\left(\frac{-5}{2}\cdot\pi\right) + \left(\frac{5}{2}\cdot\pi e^{\frac{-\pi}{20}}\right)\cdot z^{-1}}{1 - \left(e^{\frac{-\pi}{4}} + 1\right)\cdot z^{-1} - \left(-e^{\frac{-\pi}{4}}\right)\cdot z^{-2}}$$

From this form, the coefficients are easily identified.

$$b_0 = \frac{-5}{2}\cdot\pi \qquad b_1 = \frac{5}{2}\cdot\pi\cdot e^{\frac{-\pi}{20}}$$

$$a_1 = e^{\frac{-\pi}{4}} + 1 \qquad a_2 = -e^{\frac{-\pi}{4}}$$

Their values are approximately…

$b_0 = -7.853982$ $b_1 = 6.712295$

$a_1 = 1.455938$ $a_2 = -0.455938$

The final function is written as follows in the discrete time domain.

$$g_n = b_0\cdot f_n + b_1\cdot f_{n-1} + a_1\cdot g_{n-1} + a_2\cdot g_{n-2}$$

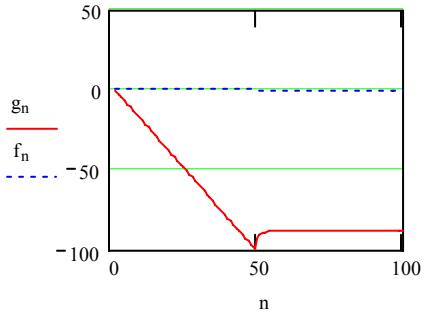
This function is now plotted in the discrete time domain.

$n = 2..100$

$f_n = u(n) - u(n - 50)$

$f_0 = 1 \quad f_1 = 1 \quad g_0 = 0 \quad g_1 = 0$

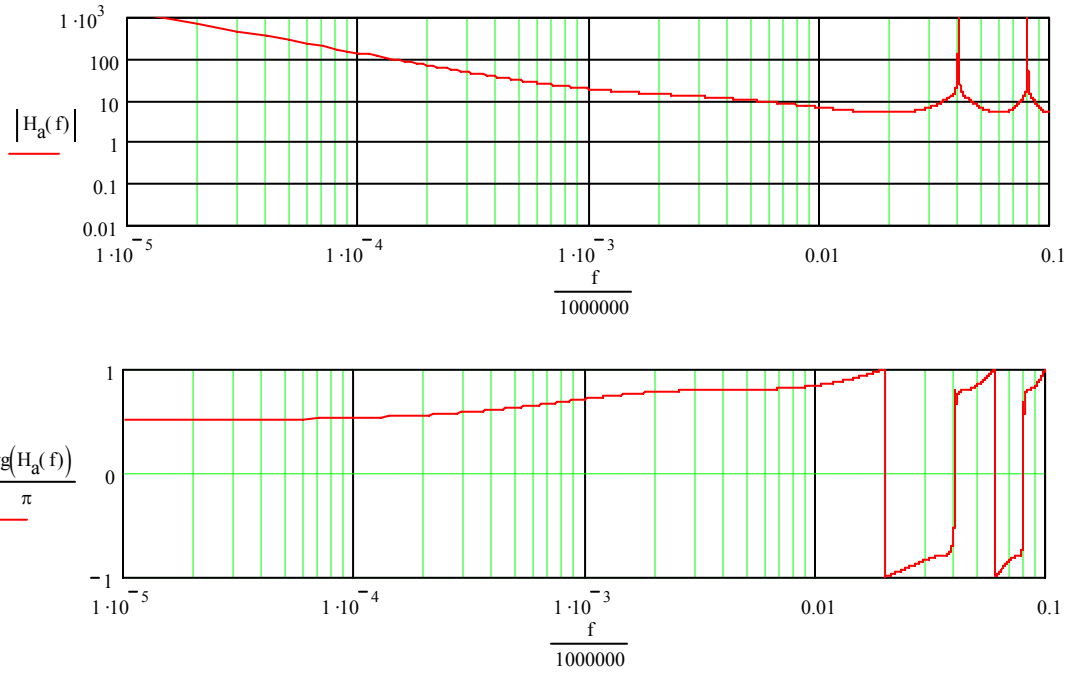
$g_n = b_0 \cdot f_n + b_1 \cdot f_{n-1} + a_1 \cdot g_{n-1} + a_2 \cdot g_{n-2}$



The frequency domain is also plotted.

$$H_a(f) = H_{T2}\left(e^{j\cdot 2\pi\cdot f\cdot T_m}\right)$$

$$f = \frac{f_b}{100}, \frac{f_b}{50} .. 100\cdot f_b$$



Here we see a fairly good representation of the original goal out to about 20 kHz. If we needed to have performance in that region, then we would need to process math at a higher rate, otherwise, the demonstration was successful.

2.5 Next Steps.

A companion MathCad® Sheet has been developed to follow this presentation. These tools should be reviewed and understood.

The next sections will focus on topics required for hardware implementation. These topics will be geared towards a Microchip® platform. An overview of data types, filter requirements, and DSP implementation will be covered.