



CMX940 PLL Simulation Tool

Reference: AT0787 Issue: 1

Date: 8th October 2020

1 INTRODUCTION

This document describes the CMX940 PLL Simulation Tool which has been created to give CML Microcircuits and its customers the ability to simulate the noise profile generated by the phase-locked loop (PLL) of the CMX940 High Performance RF Synthesizer.

The simulation is created in Octave (<https://www.octave.org/download.html>) and can be accessed by running the file named CMX940_LNC_1.m (where 1 is the version number). This is contained in a folder named (CMX940_Loop_Noise_Calculator) which also contains the instructions required to run the program. (Copy all relevant files and folders.)

It should be noted that the simulation tool does not account for the effect of bleed current on the noise and spur levels presented to the user. It is assumed that the user will have to investigate and optimise the bleed current settings based on their usage case. Refer to the CMX940 Datasheet for more information.

Disclaimer: This simulation tool/calculator is a free tool provided by CML Microcircuits (UK) Ltd. It is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

1.1 History

Version	Author	Changes	Date
A	MW	Initial Draft	9/6/2020
B	JLF/JMG/MW	- Editorial changes - Updated functionality and screenshots through review - Updated following review.	29/9/2020
C	JMG/JLF/MW	Reformatting to add history table and standard document flow. Editorial changes. Updated plots based on measurement data	1/10/2020
1	JMG	Issue for release.	8/10/2020

1.2 Glossary

Fcomp	Comparison Frequency used in PLL
CPC	Charge pump current
MCLK	Master Reference clock input
PLL	Phase Locked Loop
PN	Phase Noise
VCO	Voltage Controlled Oscillator
CP	Charge Pump
PC	Phase Comparator
OLG	Open Loop Gain
CLG	Closed Loop Gain
UI	User Interface
RO	Reference Oscillator
PM	Phase Margin
Z	Set of all integers
F_x	Frequency of component X
G_x	Power/Voltage gain of component/path X
K_x	Gain constant of component X
N_x	Noise generated by component X

1.3 Contents

1	INTRODUCTION.....	2
1.1	HISTORY	2
1.2	GLOSSARY	2
1.3	CONTENTS.....	3
2	PRELIMINARY MATERIAL	4
3	PHASE-LOCKED LOOP THEORY	4
3.1	REFERENCE OSCILLATOR	5
3.2	DIVIDERS	6
3.3	PHASE COMPARATOR	7
3.4	LOOP FILTER	8
3.5	VOLTAGE CONTROLLED OSCILLATOR.....	9
4	NOISE IN A PHASE LOCKED LOOP.....	9
5	CMX940 SIMULATOR.....	12
5.1	1 HZ NORMALISED PHASE NOISE	12
5.2	VCO GAIN SELECTION.....	12
5.3	LOADING AND SAVING DATA IN THE PARAMETER SETTINGS TAB	13
5.4	ANALYSIS TAB	13
5.5	SPUR AVOIDANCE.....	15
6	BIBLIOGRAPHY	18
7	APPENDIX.....	19

2 PRELIMINARY MATERIAL

Referring to the block diagram of Figure 2, the input reference signal to a negative feedback control loop is supplied by a highly stable frequency source. In this way, this reference source (often derived from a quartz crystal oscillator) imparts its superior stability characteristics to a higher frequency tuneable oscillator. Frequency tuning of the high frequency oscillator is achieved by programming of the control loop dividers. Along with our wanted signal we find that there is some unwanted noise which can be observed in Figure 1. In order for the wanted signal to be useful we need the level of the signal to be much greater than the level of the noise, so we will be calculating how much noise this circuit generates. We measure the power of the signal and the noise in dB (decibels) for relative measurements or dBm (which is dB relative to 1 mW {dBW + 30}) for absolute power levels. We can also measure a dBc value which is the difference between the noise level (dBm) and the peak signal level (dBm). The noise level will be measured in a given bandwidth but this is usually referenced to nominal bandwidth of 1 Hz in which case the measurement is dBm/Hz.

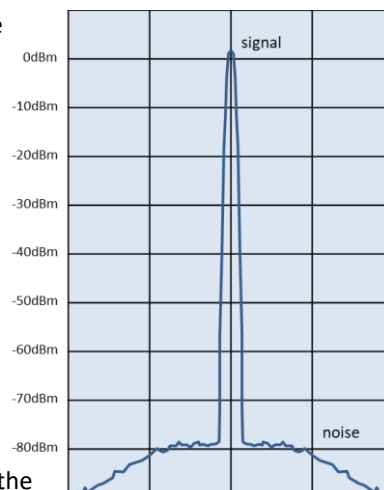


Figure 1: Example of a signal

Later on we shall see that we need to input various noise profiles. To do this we must specify the centre frequency, f_c that they are measured at. What we can then do is adjust the noise, N_0 , to any other centre frequency, f , and calculate the new noise profile, N :

$$N = N_0 * \frac{f}{f_c}$$

Please refer to the datasheet for a detailed description of the CMX940 operation.

3 PHASE-LOCKED LOOP THEORY

In order to understand the mathematics behind the simulation we must first take a look at the block diagram of the PLL we shall be considering in Figure 2.

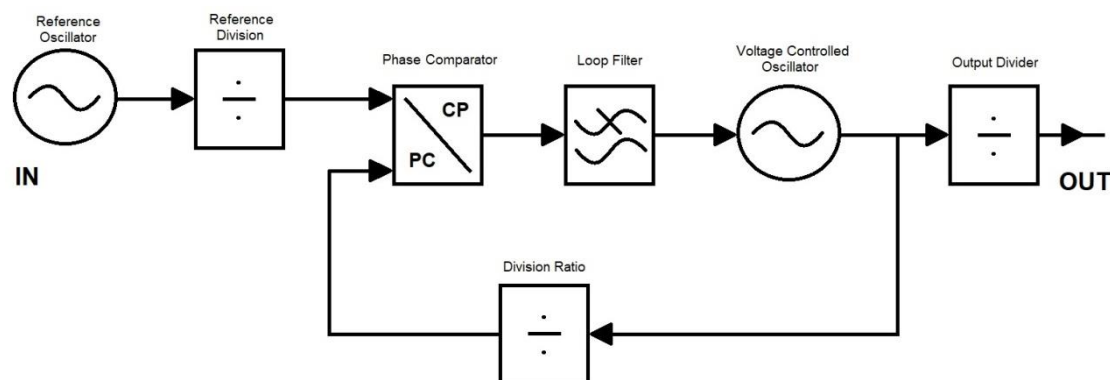


Figure 2: Block Diagram of the CMX940 PLL

We can see there are three sections to consider in the circuit: the input, loop and output. At the input we have a signal generated by the Reference Oscillator (or the output of the Reference Phase-Locked Loop – REF-PLL in CMX940). This signal passes through the reference division path to make sure it enters the loop at the required comparison frequency. The signal then passes through three components in the forward path of the loop: the phase comparator (PC) which has a charge pump, the transfer function (loop filter) which filters out unwanted frequencies and the voltage controlled

oscillator (VCO) at 2.7 - 4 GHz which converts the signal from a voltage to a frequency. Some of the signal is then fed back to the PC via a feedback path. This contains a divider which reduces the frequency so that it is equal to the comparison frequency at the PC from the input path. The VCO output signal is also fed to an optional output divider so that it is at the required output frequency.

We shall now look at the various components and how they influence system performance. In broad terms, the overall PLL noise at close offsets is determined by the reference path and flicker noise; at mid-offsets within the loop bandwidth this is determined by the loop section (N divider, phase comparator, charge pump and loop filter); at wide offsets the noise is determined by the wideband noise of the VCO.

3.1 Reference Oscillator

The reference oscillator generates a signal that acts as our input. We are only interested in the noise profile generated by the REF-PLL for our simulation, an example of which is given in Figure 3.



Figure 3: Example CMX940 REF-PLL Input Noise at 960 MHz

For the purpose of our simulation we allow up to seven frequency-noise input points and this will generate a profile as in Figure 4.

Note: The inputs are as in Figure 3.

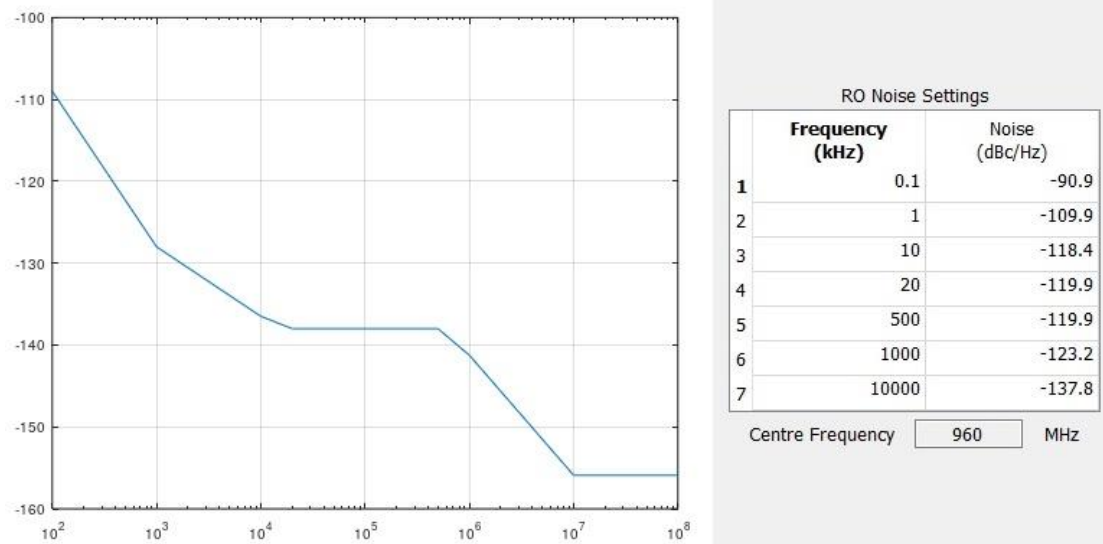


Figure 4: RO Noise Profile Generated by the Simulator

We can see that the simulator interpolates linearly between each data point. Outside of the specified frequency-noise values the simulator assumes the noise to be flat which can be seen at 10 MHz in Figure 4.

3.2 Dividers

There are three dividers within our circuit which divide the input frequency to a new output frequency. These dividers are: the reference divider R, the loop divider N and the output divider O. These can all be set as the user requires.

The reference divider is used to convert the frequency of the reference oscillator signal to the comparison frequency which is the signal frequency we want at the PC input. The loop divider takes the signal outputted by the VCO and converts it back to the comparison frequency. Finally, the output divider takes the signal outputted by the VCO and converts it to the required operating frequency.

We now want to know what effect the dividers have on our noise profile. Let's say that our noise has a power of N in the non-dB form and it passes through a divider of division value O. This simply reduces the noise so that it now has a power N/O. For example in the case of divide by 2 the noise will reduce in dB terms by ~6 dB (i.e. $20 \times \log(2) = 6.02$ dB).

To calculate the noise of a signal after it has passed through a divider with division value Div we need only to subtract $20 \times \log_{10} Div$ from its original noise power in dB form. As an example, see Figure 5, we will look at our RO noise profile again and send it through the reference divider with a division value of 8. This will give a comparison frequency of 120 MHz, the maximum specified in the CMX940.

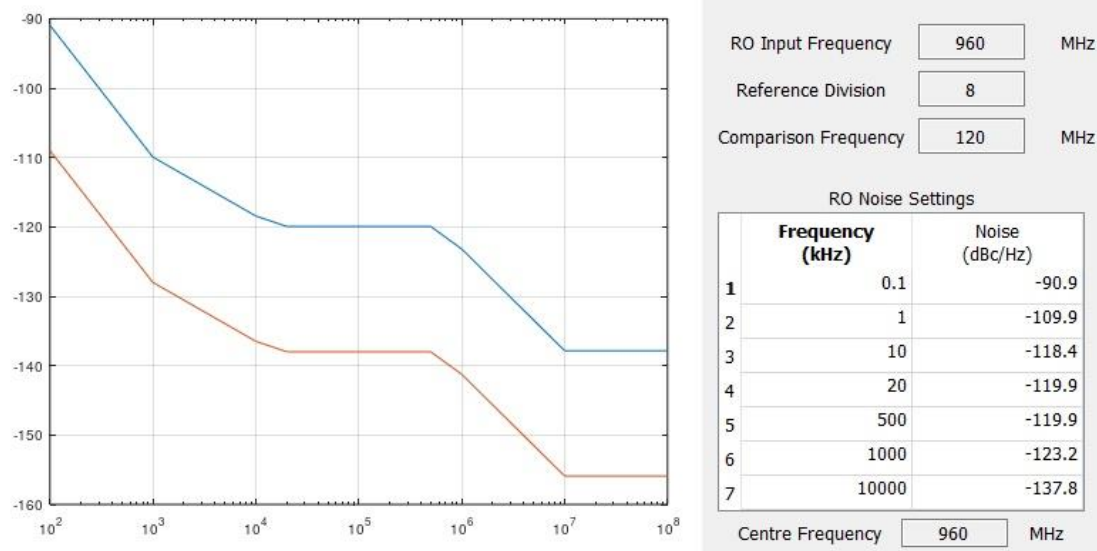


Figure 5: RO Noise Profile when Inputted (blue line) and after it has passed through the reference divider (red line).

The reference divider and output divider are both 'integer dividers' meaning they only allow integer inputs. The loop divider is a 'fractional divider' which allows a much wider range of output frequencies. Note: if only integer division is used the noise level is improved however this is not modelled in this simulator.

$$Noise_{Output} = Noise_{RO} - 20 * \log_{10} R + 20 * \log_{10} N - 20 * \log_{10} O$$

3.3 Phase Comparator

In the PC the input signal from the divided reference oscillator and the feedback signal from the VCO are compared and pulses proportional to the phase difference are applied to the loop filter via a charge pump. As both signals are at the same frequency (the comparison frequency) their noise profiles will add together. If we say N_1 is the noise coming from the reference oscillator and N_2 is the noise from the feedback path, both at some frequency f , then the combined noise is:

$$Combined\ Noise = 20 * \log_{10}(N_1 + N_2)$$

The noise then passes through a charge pump that operates at a certain current. The combination of the phase comparator and charge pump gives rise to a gain constant: $K_{PHI} = \frac{CPC}{2\pi}$. This will give a gain to the noise (N) passing through it:

$$Total\ Noise = 20 * \log_{10}(N * K_{PHI})$$

The phase comparator will also generate noise which is added to the system. This noise is difficult to measure directly so for our simulator we will use the 1 Hz normalised phase noise to calculate it. It is first necessary to evaluate the 1 Hz normalised phase noise, we use the following formula to do this:

$$PN_{1Hz} = PLL_{flat} - 20 * \log_{10} N - 10 * \log_{10} f_{comp} + 20 * \log_{10} O$$

where PN_{1Hz} is the 1 Hz normalised phase noise, PLL_{flat} is the noise level generated by the PLL at the flat point (in-band before the bandwidth peak), N is the division ratio and f_{comp} is the comparison frequency.

To calculate this we need to use a reference which will be measured at a given charge pump current (CPC_0) and output divider value (O_0), for example we have the following data set:

$PLL_{flat} = -125.3$ dBc/Hz measured at 1200 MHz, $N = 30$, $O = 3$ ($f_{VCO} = 3600$ MHz),
 $f_{comp} = 120$ MHz, $CPC = 2400$ mA.

From this data we can calculate the 1 Hz normalised phase noise:

$$\begin{aligned} PN_{1Hz} &= -125.3 - 20 * \log_{10} 30 - 10 * \log_{10} 120 * 10^6 + 20 * \log_{10} 3 \\ PN_{1Hz} &= -226.09 \\ PN_{1Hz} &\approx -226 \text{ dBc/Hz} \end{aligned}$$

We calculate this 1 Hz normalised phase noise because this has a given value from which we can calculate the PLL_{flat} value for any initial conditions. So in order to calculate the PC noise we take an adjusted 1 Hz normalised phase noise, say two dB lower, calculate the PC phase noise and in turn calculate the PLL_{flat} value, we can then adjust this until we obtain the correct PLL_{flat} value.

The noise of the phase comparator can now be calculated. We obtain the initial formula for this by consulting (Banerjee, 2006, p. 158) and taking the formulae for PN_{flat} :

$$N_{PC} = PN_{1Hz_Adj} + 10 * \log_{10} F_{PC}$$

Now we need to adjust this for various charge pump and output divider values. Given the PN_{1Hz_Adj} value was calculated at a charge pump current (CPC_0) and output divider value (D_0) we get the following general formula for phase comparator noise:

$$N_{PC} = PN_{1Hz_Adj} + 10 * \log_{10} F_{PC} - 10 * \log_{10} \frac{CPC}{CPC_0} - 20 * \log_{10} \frac{O}{O_0}$$

For a more detailed analysis of phase comparators see (Banerjee, 2006, pp. 14-21).

3.4 Loop Filter

The loop filter is a passive Resistor-Capacitor (RC) network that removes unwanted frequencies. It provides a gain to noise passing through it as well as generating some unwanted noise. Let's first look at how the filter is set up by considering Figure 6. Note that the input capacitance of the VCO is accounted for by the simulation tool and effectively adds a shunt capacitance to C3 of the loop filter shown in Figure 6.

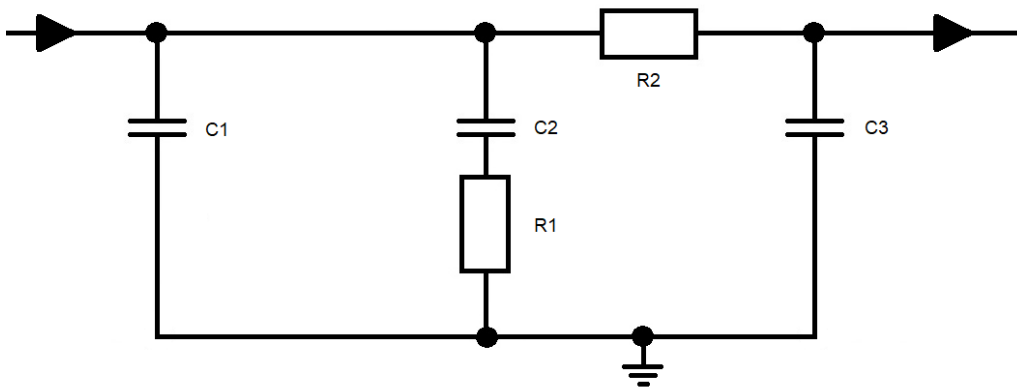


Figure 6: Loop Filter Design

In Section 7 (Appendix) we calculate the gain of the RC network and also the noise it produces. Here we shall just show the results:

$$Gain = \frac{1 + s * TC2}{s * (C3 * (1 + s * TC2) + C2 * (1 + s * TC3) + C1 * (1 + s * TC2) * (1 + s * TC3))}$$

Note: $TC2 = C2 * R1$ and $TC3 = C3 * R2$ are both time constants.

$$N_{Fil} = \frac{\sqrt{2}}{2} * \sqrt{4kT \left(R2 * \frac{Z_{R25}}{Z_{R21} + Z_{R25}} * \frac{Z_{R24}}{Z_{R23} + Z_{R24}} + R3 * \frac{Z_{R24}}{Z_{R31} + Z_{R24}} \right)}$$

3.5 Voltage Controlled Oscillator

The VCO converts the voltage output from the loop filter into a frequency and has a gain function denoted K_{VCO} . This is calculated from spot tuning voltages and frequencies and is measured in MHz/V; multiplying it by 2π gives radians/V. The gain obtained by any signal passing through the VCO is $\frac{K_{VCO}}{2\pi i f}$. The VCO has an internal noise which must be entered in the simulator by the user. This profile is entered into the simulator in a similar manner to the RO noise, but the output frequency is also required. This will allow the noise to be scaled to different frequencies (i.e. a noise profile from measurement at 1000 MHz can be entered, which can then be scaled to the relevant frequency, e.g. 3000 MHz). The VCO noise is modelled such that the profile is flat for the far out frequencies and the gradient is -20 dB/decade for near in frequencies steepening to -30 dB/decade for $f < 1$ kHz.

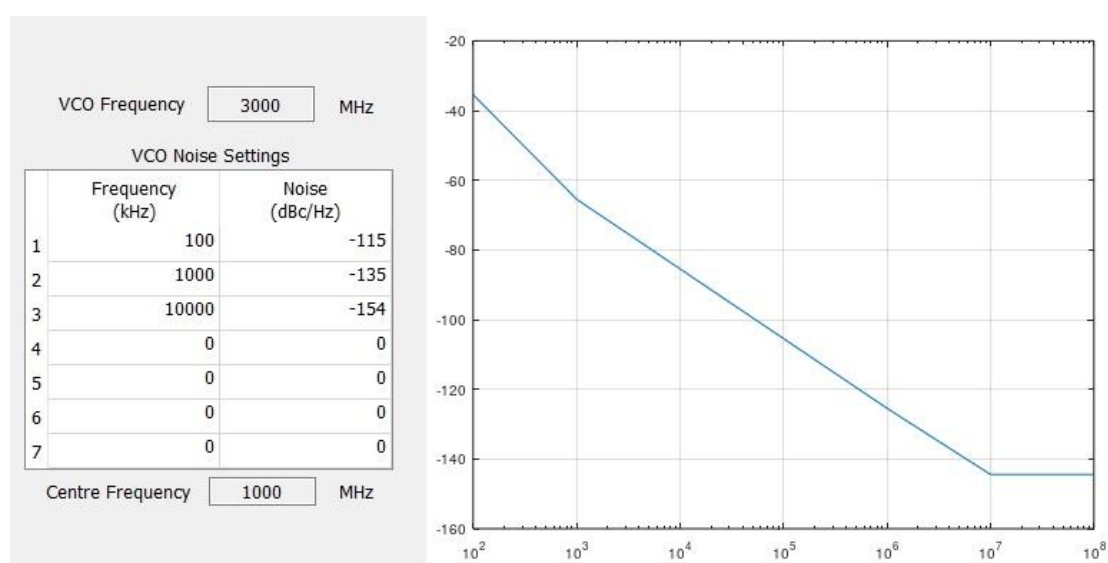


Figure 7: VCO Noise Input

The general noise profile of a VCO is well known (Kundert, 2012, p. 14). It has three main parts: an initial gradient of -30 dB/decade, which then reduces at the 'corner frequency', f_c , to a gradient of -20 dB/decade and eventually either it will flatten off or it will become limited by the noise floor of the measurement system or the output amplifier. We can easily find the VCO noise far out as it dominates the phase noise plot outside of the loop bandwidth. From this data we can infer the profile of the VCO. In our simulation we ignore the -30dB/decade noise as it is unlikely to have a significant effect on our total noise plot due to the low frequency in which it appears. It would also be challenging to measure. Therefore we model our VCO noise in such a way that for frequencies less than our input value the profile has a -20dB/decade gradient and for frequencies greater than our input values the noise is flat. This means when we measure our VCO noise profile we should aim to get points which start on the -20 dB/decade slope and end on the flat sections.

4 NOISE IN A PHASE LOCKED LOOP

Figure 8 below shows a block diagram that we can use to model the noise in a Phase Locked Loop.

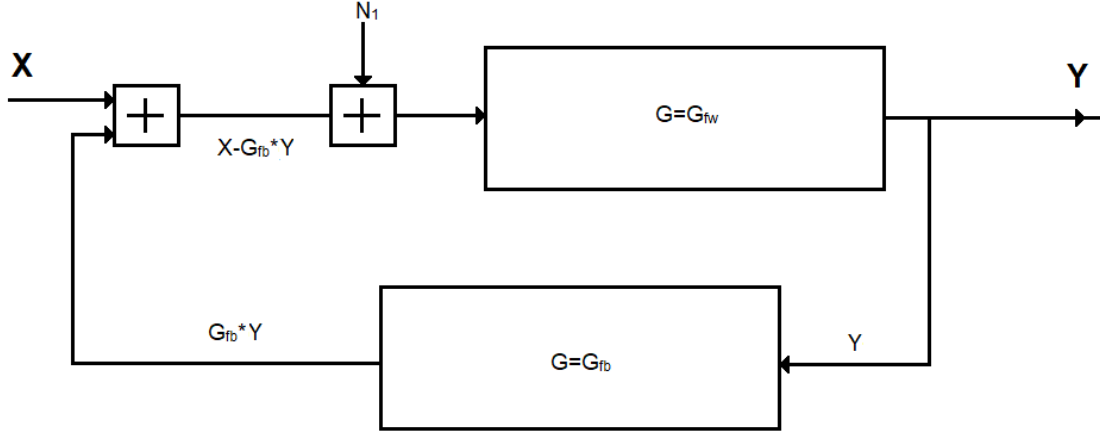


Figure 8: Block Diagram for the Noise in CMX940

By analysing the system we can obtain an expression for the output Y:

$$Y = (X - Y * G_{fb} + N_1) * G_{fw}$$

Rearranging this for Y we get:

$$Y = \frac{X * G_{fw}}{1 + G_{fw} * G_{fb}} + \frac{N_1 * G_{fw} * G_{fb}}{1 + G_{fw} * G_{fb}}$$

The first term gives the closed loop gain of the system. The second term gives the noise power. We can therefore see that the gain of the system is:

$$\text{System.Gain} = \frac{G_{fw}}{1 + G_{fw} * G_{fb}}$$

And the noise contribution of the loop is given by:

$$\text{Loop.Noise} = G_{fw} * \frac{N_1 * G_{fb}}{1 + G_{fw} * G_{fb}}$$

Let's now consider some properties of the loop, firstly we define the open loop gain to be:

$$OLG = G_{fw} * G_{fb}$$

This is clearly the gain we get when we go around the loop once. The closed loop gain is defined to be:

$$CLG = \frac{G_{fw}}{1 + G_{fw} * G_{fb}}$$

We can see this is the same as the aforementioned system gain and it describes the gain an input signal X gets, in comparison to output signal Y, when it is fed through the loop (ignoring additional noise generated by loop components).

Another interesting factor we can measure is the phase margin.

$$PM = -180 - \arg(OLG) * \frac{180}{\pi} \quad \text{for } \arg(OLG) \in [-\pi, \pi)$$

The block diagram can be broken down further to consider how the noise from each of the individual component is calculated. We shall note that the gains are represented as follows:

- $G_{PC} = K_{PHI} = \frac{CPC}{2*\pi}$
- $G_{Fil} = \frac{1 + s*TC2}{s*(C3*(1 + s*TC2) + C2*(1 + s*TC3) + C1*(1 + s*TC2)*(1 + s*TC3))}$
- $G_{VCO} = \frac{K_{VCO}}{2*\pi*f} = \frac{TC*2*\pi}{2*\pi*f} = \frac{TC}{f}$
- $G_{fb} = \frac{1}{N}$

Note: CPC = charge pump current; $s = 2 * \pi * i * f$; $TC2 = C2 * R2$; $TC3 = C3 * R3$; TC = tuning constant or VCO gain (MHz/V); N = division ratio.

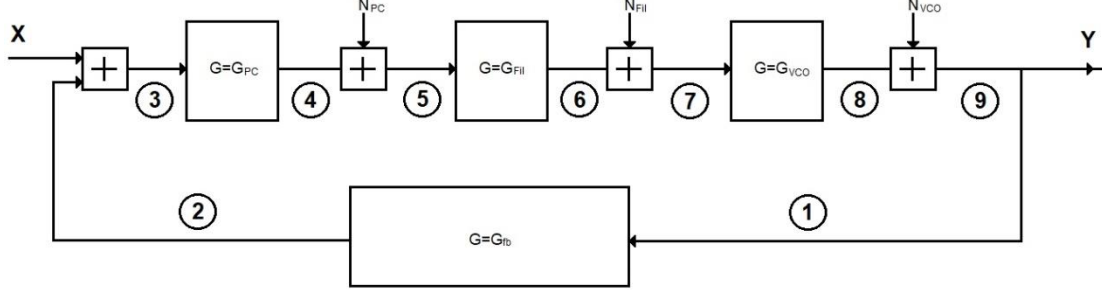


Figure 9: Detailed Block Diagram of the Noise in CMX940

The numbers 1 to 9 correspond to a new noise at each point, as described below:

1. Y
2. $Y * G_{fb}$
3. $X - Y * G_{fb}$
4. $(X - Y * G_{fb}) * G_{PC}$
5. $(X - Y * G_{fb}) * G_{PC} + N_{PC}$
6. $((X - Y * G_{fb}) * G_{PC} + N_{PC}) * G_{Fil}$
7. $((X - Y * G_{fb}) * G_{PC} + N_{PC}) * G_{Fil} + N_{Fil}$
8. $((X - Y * G_{fb}) * G_{PC} + N_{PC}) * G_{Fil} + N_{Fil}) * G_{VCO}$
9. $((X - Y * G_{fb}) * G_{PC} + N_{PC}) * G_{Fil} + N_{Fil}) * G_{VCO} + N_{VCO}$

So we know that $Y = N^o 9$. Rearranging the resulting equation for Y we find that we can express it in the same form as before:

$$Y = \frac{X * G_{fw}}{1 + G_{fw} * G_{fb}} + \frac{N_1 * G_{fw} * G_{fb}}{1 + G_{fw} * G_{fb}}$$

$$= \frac{X * G_{fw}}{1 + OLG} + \frac{N_1 * OLG}{1 + OLG}$$

But in this case we have the following expression for $N_1 * \frac{OLG}{1 + OLG}$:

$$\frac{N_{PC} * G_{Fil} * G_{VCO}}{1 + OLG} + \frac{N_{Fil} * G_{VCO}}{1 + OLG} + \frac{N_{VCO}}{1 + OLG}$$

This formula shows the overall contribution from each noise source within the loop. Let's break it down and remind ourselves of the noise contributions from each source:

$$20 * \log_{10} N_{PC} = PN_{Hz_Adj} + 10 * \log_{10} F_{PD} - 10 * \log_{10} \frac{CPC}{CPC_0}$$

$$N_{Fil} = \frac{\sqrt{2}}{2} * \sqrt{4kT * \left(R2 * \frac{Z_{R25}}{Z_{R21} + Z_{R25}} * \frac{Z_{R24}}{Z_{R23} + Z_{R24}} + R3 * \frac{Z_{R24}}{Z_{R31} + Z_{R24}} \right)}$$

N_{VCO} is a profile given by our inputs. Our input at X is our reference oscillator profile which has noise N_{RO} but this passes through the reference divider, which has division value R, so the input at X is:

$$X = \frac{N_{RO}}{R}$$

Once our noise is outputted at Y it will pass through the output divider, which has division value O, so the final noise, N, will have a value as follows:

$$N = \frac{Y}{O}$$

Using these calculations we can simulate the noise generated by the CMX940.

5 CMX940 SIMULATOR

5.1 1 Hz Normalised Phase Noise

Recalling the definition of 1 Hz normalised phase noise from Section 3.3, this value can be used to calculate the noise generated by the charge pump as this is inherently a very difficult value to measure. The simulator is programmed so that it adjusts the noise generated by charge pump until the 1 Hz normalised phase noise is at the specified value. If the program cannot find a charge pump noise that meets the required 1 Hz normalised phase noise it will return the message 'Error: Unable to find correct Charge Pump Noise.'

Note: It is sometimes assumed that the 1 Hz normalised phase noise is a constant for a particular PLL design or integrated circuit. This is not true; often datasheets quote the best figure for the device and leave the user to discover that with different charge pump settings or different divider setting the value may vary. For this reason the simulator allows the user to adjust the value. The default values are taken from evaluation of the CMX940 (S-PLL) and should give a reasonable estimate of performance in typical scenarios.

5.2 VCO Gain Selection

The VCO gain, or tuning constant, will vary with VCO operating frequency and therefore the simulator has a built-in function to allow the user to select their own value or to allow the simulator to choose the tuning constant. To do this the user must click on the switch in the 'Tuning Constant Settings' box on the 'Parameter Settings' tab. A picture of the box in manual and automatic modes is shown in Figure 10.



Figure 10: Tuning Constant Settings Selection Box on CMX940 Simulator UI

When the manual setting is selected the simulator just takes the tuning constant value entered by the user in the 'General Values' list. When automatic is selected, the program finds the tuning constant values by considering the VCO frequency and using the tuning constant profiles which are stored in the program. The user then has two options, firstly whether they want the low VCO or the high VCO. This is because to cover the desired VCO frequency range the CMX940 needs two VCOs and these have an overlap and different tuning constant profiles. The second selection is whether the user wants 'Automatic Switching' or not. When this setting is turned off the simulator will run with whatever settings are selected regardless of whether they are realistic. If 'Automatic Switching' is

turned on the simulator will change between high and low VCO if the wrong one is selected, give a warning if a VCO frequency value is the crossover range is selected and revert to manual mode if a VCO frequency outside the calculated frequency range is selected.

5.3 Loading and Saving Data in the Parameter Settings Tab

A helpful setting built into the CMX940 simulator is the ability to load and save the input data in the 'Parameter Settings' tab.

Firstly let's look at the save function. Click on the 'Save Input Data' button at the bottom of the tab and a pop out box will appear giving the user the option of saving the input data to a numbered save slot or a custom save slot. The user can then select any of the 20 numbered save slots or type their own name in the custom save slot. Then the relevant save button must be pressed and the data will be saved into that save slot. Note that any data already in that save slot will be overwritten.

The load function essentially works in the same way. The user should click on the 'Load Input Data' button and this will produce a similar pop out box. The user should then choose where they would like to load data from; either a numbered save slot or a custom save slot and then click the corresponding load button. This will load the data to the simulator's UI and also update the values in the reset components box.

Every set of saved data is saved in a text file named 'Data_History.txt' which is in the 'DataStore' file. Every time a set of data is saved it saves the date, time, location and data set, see Figure 11 for an example, so that the user can find overwritten data sets and copy and paste them into one of the save files if they wish.

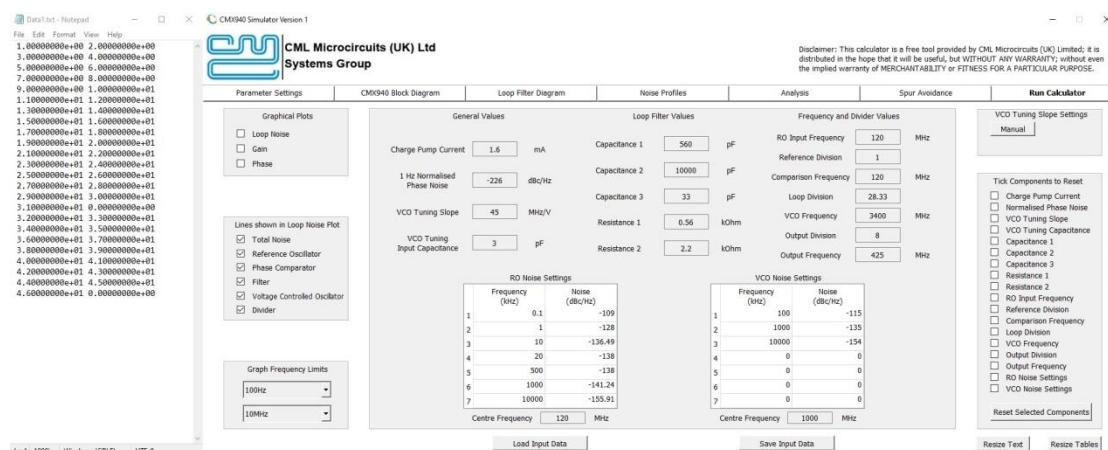


Figure 11: Example of How Data from the Parameter Settings tab in Saved in Save Slot Number 1

5.4 Analysis Tab

Within the simulator there is a tab labelled 'Analysis' which the user can use to calculate possible capacitance, resistance, frequency and divider input values. This can be seen in Figure 12 below.

Figure 12: 'Analysis' Tab of the CMX940 Simulator

There are two different strings of calculation on this tab, one on the left (LHS) and one on the right (RHS). Let's start with the LHS. We use this to calculate the correct filter values that we require. Firstly we have our input values: VCO frequency (f_{VCO}), comparison frequency (f_{comp}), natural frequency (f_N), damping factor (ξ), charge pump current (CPC) and VCO gain (G_{VCO}). These can be recalled from the 'Parameter Settings' tab if necessary. Once we have these we can use the following equations to calculate the filter values and bandwidth.

$$C2 = \frac{CPC * G_{VCO}}{N * (2\pi f_N)^2}$$

$$R1 = 2\xi * \sqrt{\frac{N}{CPC * G_{VCO} * C2}}$$

$$C1 = \frac{C2}{10}$$

$$R2 = 6 * R1$$

$$C3 = \frac{R1 * C2}{30 * R2}$$

$$BW = \pi f_N * \left(\xi + \frac{1}{4\xi} \right)$$

Note: $N = \frac{f_{VCO}}{f_{comp}}$

Once we have these values we need to think about what values we can actually input. To do this we have the option to convert our values into E series numbers. These are a set of numbers that can be used and we pick the closest number in the series as our choice. The E24 series is common for resistors and the E6 series more commonly available for capacitors. A drop down box will allow users to select one of the E3, E6, E12 or E24 series to specify their capacitor/resistor values. For more information on E series, refer to (E Series of Preferred Numbers, 2020). The mentioned series values are listed below:

$$E3 = \{1.0 \ 2.2 \ 4.7\}$$

$$E6 = E3 + \{1.5 \ 3.3 \ 6.8\}$$

$$E12 = E6 + \{1.2 \ 1.8 \ 2.7 \ 3.9 \ 5.6 \ 8.2\}$$

$$E24 = E12 + \{1.1 \ 1.3 \ 1.6 \ 2.0 \ 2.4 \ 3.0 \ 3.6 \ 4.3 \ 5.1 \ 6.2 \ 7.5 \ 9.1\}$$

Note: These equations are based on those in (Fujitsu Microelectronics).

Once calculated, the loop filter values can be entered into our 'Parameter Settings' tab. Our simulator will only input non-zero values so if the user only wants to input, say, C1 then they should manually set the other values to 0 or leave them blank.

Now we can consider the RHS. This section of the 'Analysis' tab allows us to calculate our input frequency and division values. The relation between the frequency and division values is already known but we shall restate three here from which all the relations can be found.

$$F_{comp} = \frac{F_{RO}}{R}$$

$$F_{comp} = \frac{F_{VCO}}{N}$$

$$F_{out} = \frac{F_{VCO}}{O}$$

Note: R = Reference Division, N = Division Ratio, O = Output Division.

What the user can now do is input known values or the desired values but leave all other values blank or as 0 then press the 'Calculate Frequency and Division Values' button resulting in three scenarios:

1. The input values do not match and therefore the calculator will compute multiple values for a component. This will produce an error message telling the user which component/s have multiple values. No values will be outputted and the user will have to adjust the inputs.
2. There may be insufficient input arguments to calculate values for every component. This will produce an error message telling the user which components have no value (the calculated values will still be outputted into the bottom table).
3. The simulator is happy that there are enough inputs and they all match, in this case the calculated values will all appear in the bottom table.

Once we have produced our frequency and division values we can send them to the 'Parameter Settings' tab – again ignoring any zero values or blanks.

5.5 Spur Avoidance

The 'Spur Avoidance' tab allows the user to test whether the output is going to be affected by integer, half, third, quarter or fifth order boundary spurs. Please note the simulation tool does not account for other spur types and more detailed information on these can be found in the CMX940 Datasheet (section 7.4) including guidance on how to avoid certain spur types.

In order for the simulation tool to calculate boundary spurs the following inputs are required:

- Reference frequency (MCLK) which is inputted to the CMX940
- S-PLL Bandwidth – the bandwidth of the loop filter for the CMX940
- Spur Finding Factor – determines over how many multiples of the loop bandwidth we are looking for spurs
- REF-PLL MDIV (M), SYN_SDIV (S) and SPLL_RDIV (R) Values; The overall reference division being SDIV * RDIV
- Output Divider value in the S-PLL

- Output frequency range to test: Starting Frequency, Stopping Frequency and the frequency step size required

From these inputs we produce a table where for every VCO frequency that has the potential to cause a boundary spur, that frequency is outputted along with S-PLL N divider value (decimal), the offset frequency of the spur, the actual frequency of the spur and the type of spur. Note that for all orders of boundary spurs, output division will not change the frequency offset of the spurs but will reduce their level by 6 dB for every halving of frequency.

In order to calculate the spurs we must first note that spurs occur at integer, half, third, quarter and fifth multiples of the loop comparison frequency. So, let's compare the SVCO frequency with the S-PLL comparison frequency:

$$Freq_{SPLL,Comp} = Freq_{REFPLL,ref} * M / (S * R)$$

$$Freq_{SVCO} = Freq_{Out} * Div_{Out}$$

$$Div_{SPLL,N} = Freq_{SVCO} / Freq_{SPLL,Comp}$$

Now that we have a divider value we want to consider whether this is 'close' to any of the following groups: Z , $Z + \frac{1}{2}$, $Z + \frac{1}{3}$, $Z + \frac{2}{3}$, $Z + \frac{1}{4}$, $Z + \frac{3}{4}$, $Z + \frac{1}{5}$, $Z + \frac{2}{5}$, $Z + \frac{3}{5}$, $Z + \frac{4}{5}$. We judge this by considering whether these problem numbers fall within a 'margin' of the PLL loop bandwidth.

$$Margin = Bandwidth_{SPLL} * Factor_{Spur} / Freq_{SPLL,Comp}$$

By checking whether the divider value is within the margin of any of the aforementioned groups we can determine whether a spur can potentially appear in the noise spectrum outputted by the CMX940. We can then calculate the offset frequency of the spur and the actual frequency the spur appears at. In the 'Spur Avoidance' tab there are two outputs, one a table listing all VCO frequencies that encountered a spur and information regarding what type of spur that was found and where it's located as well as a graph showing where all spurs are on the VCO frequency spectrum with their height determining the type of spur they represent. A further helpful feature is that if different settings are chosen and the calculator re-run, new spurs will also be plotted on the original graph. These will appear in a different colour and the plot will be added to the legend on the right hand side of the figure.

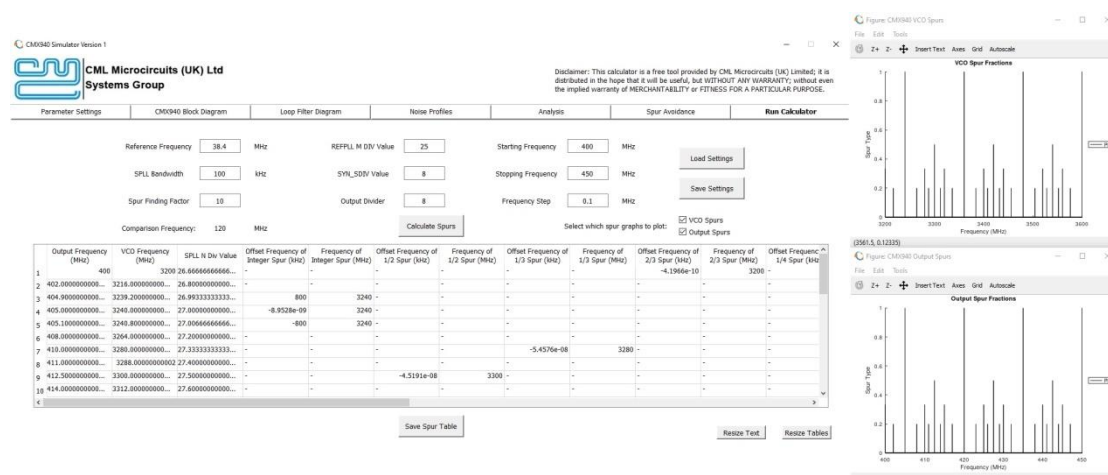


Figure 13: Spur Avoidance Tab Outputs

We can see in Figure 13 that there are 'Load' and 'Save Settings' buttons which work on the spur avoidance inputs in the same way as described in section 5.3. At the bottom of the tab we can also see a button labelled 'Save Spur Table'.

This allows the user to save the information displayed in the table to a text document. It is saved in both a csv file and a text file. The csv file is in the same form as the table in Figure 13, whereas the text file is in a five-column format, an example of which is shown in Table 1 below. The .csv file can be imported into Excel for analysis. Note that the import into Excel does not include the column headers; these are the same as those shown in the “Spur Avoidance Tab” shown in Figure 13.

VCO Frequency (MHz)	SPLL N Div Value	Spur Type	Offset Frequency of Spur (kHz)	Frequency of Spur (MHz)
3200.32	26.669333	2/3	-960	3200
3210.24	26.752000	3/4	-960	3210
3216.16	26.801333	4/5	-800	3216
3240.96	27.008000	1	-960	3240
3264.16	27.201333	1/5	-800	3264

Table 1: Spur Data

6 BIBLIOGRAPHY

Banerjee, D. (2006). PLL Performance, Simulation, and Design.

Capacitive Reactance. (n.d.). Retrieved April 23, 2020, from Electronics Tutorials:
https://www.electornics-tutorials.ws/filter/filter_1.html

E Series of Preferred Numbers. (2020, February 12). Retrieved April 21, 2020, from Wikipedia:
https://en.wikipedia.org/wiki/E_series_of_numbers

Fujitsu Microelectronics, I. (n.d.). PLL Basics - Loop Filter Design.

Kundert, K. S. (2012, March). *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers*. Retrieved from The Designer's Guide Community: www.designers-guide.org

Lascari, L. (2002, March 13). *MathCad PLL worksheets*. Retrieved April 21, 2020, from RFDude:
tools.rfdude.com/PLL/pll.html

Resistors in Parallel. (n.d.). Retrieved April 23, 2020, from Electronics Tutorial:
https://www.electronics-tutorials.ws/resistor/res_4.html

Resistors in Series. (n.d.). Retrieved April 23, 2020, from Electronics Tutorials:
https://www.electronics-tutorials.ws/resistor/res_3.html

7 APPENDIX

Here we shall look at the effect resistors and capacitors have on a network.

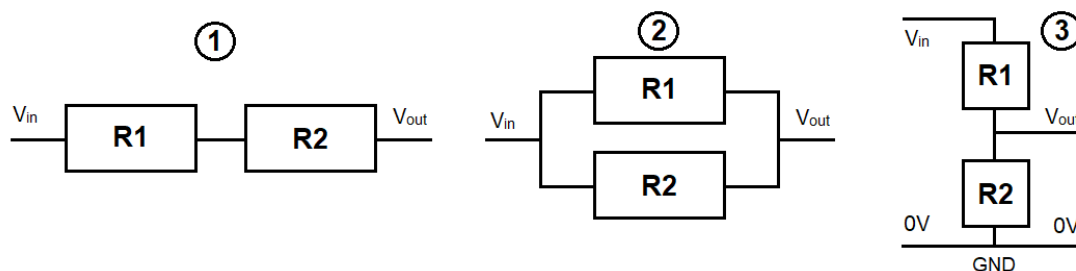


Figure 14: Resistor Diagrams

Consider Figure 14 part 1. We have two resistors in series. Their combined resistance is just the sum of their resistance values: $R1 + R2$. This is the gain factor of part 1. Let's now consider part 2 which has two resistors in parallel. To compute their combined resistance value we take the sum of their reciprocals and this is equal to the reciprocal of their combined resistance which is as follows: $\frac{1}{R} = \frac{1}{R1} + \frac{1}{R2} \Rightarrow R = \frac{R1 \cdot R2}{R1 + R2}$. This is the gain factor of part 2. Finally we come to part 3 (often known as a potential divider). Here we have two resistors in series but we want to take the output voltage across R2. In this case our gain factor and combined resistance is: $R = \frac{R2}{R1 + R2}$. Finally we want to know the effect of capacitors. We can consider capacitors in the same way as resistors if we use their capacitive reactance to replace resistance. The formula for this is as follows: $X_C = \frac{1}{2 \cdot \pi \cdot f \cdot C}$. All of these formulae are taken from the following sources: (Resistors in Series) (Resistors in Parallel) (Capacitive Reactance).

Let's now calculate the gain and noise generated by the loop filter. We must first consider this to be a network of resistors which are both in parallel and in series. We can do this by first considering the capacitive reactance (measured in Ohms) of the capacitors which allows us to treat them as resistors. For a capacitor of capacitance C, it's capacitive reactance is $X_C = \frac{1}{2 \cdot \pi \cdot f \cdot C}$. We must now convert Figure 6 into a simplified graph. Let's now consider the diagram which is in the first section of our filter.

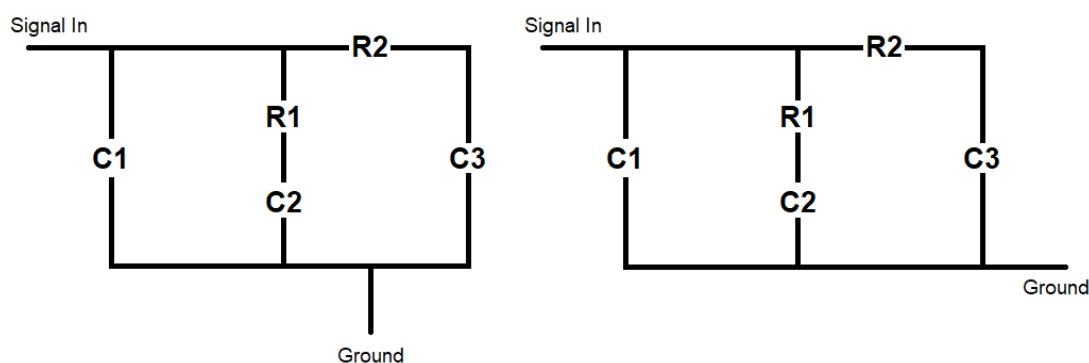


Figure 15: Loop Filter Nodal Graph

We are considering the signal input and the ground because we want to find the transfer function (for simplicity we will call this gain) of the network. We can see that some of the components in Figure 15 are in series, namely R1, C2 and R2, C3. Therefore let's simplify the network as in Figure 16.

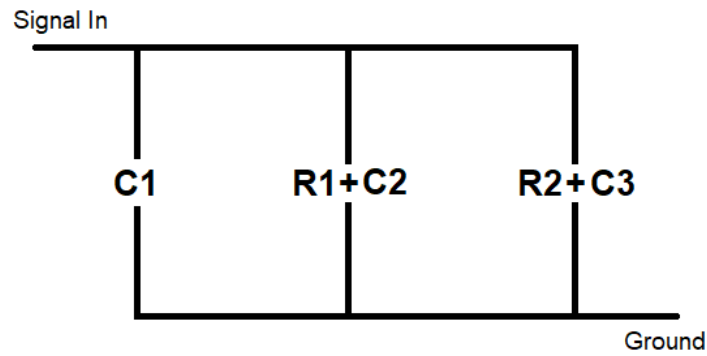


Figure 16: Loop Filter Simplified Nodal Graph

We can now see from Figure 16 that we have three parallel components so we can express the gain as follows:

$$\begin{aligned}
 \frac{1}{Gain} &= \frac{1}{X_{C1}} + \frac{1}{X_{C2} + R1} + \frac{1}{X_{C3} + R2} \\
 &= \frac{(X_{C2} + R1) * (X_{C3} + R2)}{X_{C1} * (X_{C2} + R1) * (X_{C3} + R2)} + \frac{X_{C1} * (X_{C3} + R2)}{X_{C1} * (X_{C2} + R1) * (X_{C3} + R2)} + \frac{X_{C1} * (X_{C2} + R1)}{X_{C1} * (X_{C2} + R1) * (X_{C3} + R2)} \\
 &= \frac{(X_{C1} + X_{C2} + R1) * (X_{C3} + R2) + X_{C1} * (X_{C2} + R1)}{X_{C1} * (X_{C2} + R1) * (X_{C3} + R2)} \\
 \Rightarrow Gain &= \frac{X_{C1} * (X_{C2} + R1) * (X_{C3} + R2)}{X_{C1} * (X_{C2} + R1) + (X_{C3} + R2) * (X_{C1} + X_{C2} + R1)}
 \end{aligned}$$

Now that we have calculated the gain of the network we want to consider where we are taking our output. We see in Figure 17 that we take our output across C3 and the path between input and output only has R2 on it. So to find the voltage gain at the output we must multiply our total gain of the network by $\frac{X_{C3}}{R2 + X_{C3}}$ (as per our third resistor formula).

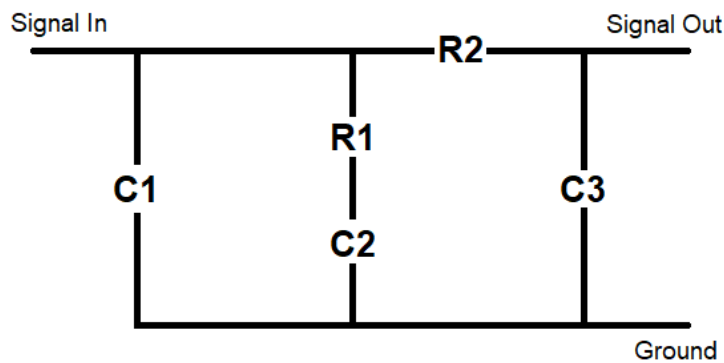


Figure 17: Loop Filter Nodal Graph

So we now find that the gain is as follows:

$$\begin{aligned}
 Gain &= \frac{X_{C1} * (X_{C2} + R1) * (X_{C3} + R2)}{X_{C1} * (X_{C2} + R1) + (X_{C3} + R2) * (X_{C1} + X_{C2} + R1)} * \frac{X_{C3}}{R2 + X_{C3}} \\
 &= \frac{X_{C1} * X_{C3} * (X_{C2} + R1) * (R2 + X_{C3})}{(R2 + X_{C3}) * (X_{C1} * (X_{C2} + R1) + (R2 + X_{C3}) * (X_{C1} + X_{C2} + R1))}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{X_{C1} * X_{C3} * (X_{C2} + R1)}{X_{C1} * (X_{C2} + R1) + (R2 + X_{C3}) * (X_{C1} + X_{C2} + R1)} \\
&\quad * \left(\frac{s^3 * C1 * C2 * C3}{s^3 * C1 * C2 * C3} \right) \\
&= \frac{1 + s * C2 * R1}{s * C3 * (1 + R1 * s * C2) + (s * C3 * R2 + 1) * (s * C2 + s * C1 + s^2 * C1 * C2 * R1)} \\
&= \frac{1 + s * TC2}{s * (C3 + s * TC2 * C3 + C2 + C1 + s * TC2 * C1 + s * TC3 * C2 + s * TC3 * C1 + s^2 * TC3 * TC2 * C1)} \\
&= \frac{1 + s * TC2}{s * (C3 * (1 + s * TC2) + C2 * (1 + s * TC3) + C1 * (1 + s * TC2 + s * TC3 * C1 + s^2 * TC3 * TC2))} \\
&= \frac{1 + s * TC2}{s * (C3 * (1 + s * TC2) + C2 * (1 + s * TC3) + C1 * (1 + s * TC2) * (1 + s * TC3))}
\end{aligned}$$

Note: $TC2 = C2 * R1$ and $TC3 = C3 * R2$ are both time constants.

We have now derived the transfer function for our filter which effects any noise passing through it. Remember that this is a complex function, to find the linear gain we must take the absolute value of the function.

If we consult (Banerjee, 2006, pp. 43-47) we observe another way of calculating the loop filter gain by considering a 4th order filter.

Now we would like to find the noise generated by the filter. To do this we must know that capacitors are noiseless and that resistors create a thermal noise. So to calculate the noise generated by the filter we must calculate the noise generated by each resistor and consider how the RC network around it controls that noise.

Firstly lets recall that the thermal noise generated by a resistor in V / $\sqrt{\text{Hz}}$ is $\sqrt{4 * k * T * R}$.

Where k = Boltzmann's constant in joules per kelvin,

T = the resistor's absolute temperature in kelvin

R = the resistor value in ohms (Ω)

We shall now define the following impedance equations:

$$\begin{aligned}
Z_{R2_1} &= R2 + \frac{1}{2 * \pi * f * C2} && \text{R2 in series with C2} \\
Z_{R2_2} &= \frac{1}{2 * \pi * f * C1} && \text{C1 impedance} \\
Z_{R2_3} &= R3 && \text{R3 impedance} \\
Z_{R2_4} &= \frac{1}{2 * \pi * f * C3} && \text{C3 impedance} \\
Z_{R2_5} &= Z_{R2_2} * \frac{Z_{R2_3} + Z_{R2_4}}{Z_{R2_2} + Z_{R2_3} + Z_{R2_4}} && \text{Z2 in parallel with Z3 and Z4} \\
Z_{R3_1} &= Z_{R2_3} + Z_{R2_1} * \frac{Z_{R2_2}}{Z_{R2_1} + Z_{R2_2}} && \text{Z1 and Z2 in parallel with Z3 in series}
\end{aligned}$$

We can therefore calculate the noise generated by R2 and R3 to be:

$$N_{R2} = \sqrt{4 * k * T * R2} * \left(\frac{Z_{R2_5}}{Z_{R2_1} + Z_{R2_5}} \right) * \left(\frac{Z_{R2_4}}{Z_{R2_3} + Z_{R2_4}} \right)$$

$$N_{R3} = \sqrt{4 * k * T * R3} * \left(\frac{Z_{R24}}{Z_{R31} + Z_{R24}} \right)$$

Combining these two equations together we find the total noise generated by the filter to be:

$$N_{Fil} = \frac{\sqrt{2}}{2} * \sqrt{4kT \left(R2 * \frac{Z_{R25}}{Z_{R21} + Z_{R25}} * \frac{Z_{R24}}{Z_{R23} + Z_{R24}} + R3 * \frac{Z_{R24}}{Z_{R31} + Z_{R24}} \right)}$$

More detailed calculation can be found at (Lascari, 2002, pp. 10-17).