PHASE-LOCKED LOOP MACROMODELS

by

Emy Tan

Memorandum No. UCB/ERL M90/114

10 December 1990

cover page

PHASE-LOCKED LOOP MACROMODELS

by

Emy Tan

Memorandum No. UCB/ERL M90/114

10 December 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ABSTRACT

In this report, the monolithic 560B PLL has been simulated and simplified; the simulation times are excessive compared to those of a known PLL macromodel introduced by D. O. Pederson and K. Mayaram. Even faster PLL macromodel performance is possible by chosing faster PLL macromodel components. Thus, alternative phase comparators (PCs) and voltage-controlled oscillators (VCOs) are examined. Of the configurations proposed, the simulation times of the simple voltage-controlled voltage source (VCVS) PC is the fastest; the Integrating Sine VCO, which modifies the argument of the sine function, has the fastest simulation times of the examined VCOs. From these results, two PLL macromodels have been developed which provide proper PLL operation at significant simulation-time reductions compared to actual PLLs. The first PLL proposed, PLL(1), has a voltage-controlled voltage source (VCVS) implementing a sinusoidal phase-comparator (PC) characteristic. The second PLL proposed, PLL(2), has a VCVS with an amplified/limited output, providing a triangular phase-comparator characteristic. Both proposed PLLs use an Integrating Sine VCO.

ACKNOWLEDGEMENTS

I would like to thank Professor Donald O. Pederson, my research advisor, for everything he has done for me. His advice and guidance, both on technical issues and non-technical issues, have been invaluable. Equally invaluable is his confidence in me, which has been my motivation through many rough times.

I am thankful to Prof. Robert Meyer for taking the time to read this report, and also to both Prof. Meyer and Prof. Paul Gray for giving me a strong integrated-circuit background from the courses they have taught. Their work and insights have been quite helpful for this report.

I am indebted to Jean Hsu, Hoa Luong, David Gates, and Beom Johnson for their help in making this report presentable. In addition, I would also like to thank my husband, Charles Ron Musick, Jr., for his continuous advice and support.

I acknowledge with gratitude fellowship support from the Office of Naval Research and Tektronix Inc..

Table of Contents

CHAPTER 1: INTRODUCTION

Monolithic phase-locked loops (PLLs) are commonplace in today's circuit applications because of their versatility, low cost, and high performance capabilities (a summary of the basic operation of a PLL is included in Appendix A for reference). Due to the complexity of even the most basic PLL designs, however, simulating an actual PLL alone or in a system is too costly in terms of both real time and CPU time. The need arises for simple PLL macromodels that provide fast yet accurate simulation of the PLL. Each subcircuit function of the phase-locked loop is replaced by the simplest possible circuit simulator elements that accurately perform the correct function. In this report, PLL macromodels are investigated using SPICE3 as the circuit simulator. These macromodels are then viable SPICE3 simulation substitutes (subcircuits) for actual phase-locked loops, with applications in stand-alone or system simulation.

In Chapter 2, the 560B Monolithic PLL circuit is simulated to show the operation of an actual PLL. The 560B circuit is then simplified to show that comparable operation is obtained by using simulation elements in place of transistor circuits. The simulation times of these PLLs are noted for future comparison.

In Chapter 3, a PLL macromodel introduced by Pederson and Mayaram, PLL(0) [2], is examined. The PLL(0) macromodel has a voltage-controlled voltage source (VCVS) analog multiplier for the phase comparator (PC), a simple RC filter, and a Wien voltage-controlled oscillator (VCO) macromodel. Limitations of the PLL(0) macromodel are noted (r_o effects are derived in Appendix B), improvements made, and the resulting PLL(0) macromodel is simulated. The performance of the PLL(0) macromodel and its simulation times are noted.

In Chapter 4, alternative PCs are investigated for use in a faster PLL macromodel. Close attention is paid to the PC characteristic, as that significantly affects the lock and capture ranges of the PLL. The PLL(0) macromodel has a VCVS analog multiplier PC with a sinusoidal PC characteristic; alternative PCs that have triangular PC characteristics are introduced and simulated (Appendix E examines the triangular PC characteristic). Sawtooth PC characteristics are mentioned, but implementation in a macromodel is difficult. A problem of DC offsets in the input is presented, but the input to the PLL is assumed to be AC-coupled. The double-balanced analog multiplier (the PC of the 560B) is then examined and simulated, and a future possible PC macromodel is introduced. Simulation times of all presented PCs are compared; the VCVS analog multiplier PC of the PLL(0) macromodel (sinusoidal PC characteristic) is the fastest, while the fastest PC with a triangular PC characteristic is the VCVS analog multiplier with an amplified and limited output.

In Chapter 5, alternative VCOs are investigated for use in a faster PLL macromodel. The PLL(0) macromodel has a Wien VCO macromodel; however, that configuration has limitations as brought out in Chapter 3. Modifying the argument of a sine function is investigated; Appendix C illustrates the incorrect and correct way to generate such a VCO. An Integrating Sine VCO is presented as a result of modifying the argument of a sine function, and the configuration is simulated. Multivibrator circuits are also presented and simulated; the two-inverter multivibrator has inferior performance compared to the Wien VCO macromodel, and the emitter-coupled pair (ECP) multivibrator takes much longer to simulate. Simulation times of all presented VCOs are compared; the Integrating Sine VCO is superior.

In Chapter 6, two proposed PLL macromodels are introduced based on the results of

Chapters 4 and 5. These macromodels are simulated, and their performance noted. An asymmetric capture range is the result of the particular control-voltage scheme, explained in Appendix D. Overall, the behavior is much as expected.

In Chapter 7, the viable PLL macromodels are summarized. Two PLL macromodels are recommended for use based on performance and simulation time.

CHAPTER 2: SIMULATION OF A MONOLITHIC PLL

2.1 Circuit Simulation of the 560B PLL

The 560B PLL, whose schematic-circuit diagram is shown in Figure 2.1, is an early monolithic PLL representing the second generation of PLL design. It consists of a double-balanced analog multiplier, an ECP multivibrator VCO, external capacitors for filtering and setting the free-running frequency of the VCO, biasing/level-shifting circuitry, and an output buffer. The 560B's operation has been described and analyzed by several sources.[1,4] A SPICE3 file for the 560B is shown in Figure 2.2. All simulation runs in this report are performed using SPICE3. The free-running frequency is chosen to be 1MHz, and a low-pass filter of 20kHz is placed at the differential output of the PC. Figures 2.3 and 2.4 show the output and phase relationships of the 560B PLL for various input frequencies. The capture range is approximately ±0.2MHz, as shown in the cases of f_{in}=0.81MHz, 0.82MHz, and 1.21MHz. For the 560B, the control voltage is limited to between +14.58V/+9.75V by the output stage; this limits both the capture the lock ranges to be approximately ±0.2MHz. Note that the output voltage has a DC offset of 12.1V. This is the control voltage for the VCO to operate at its free-running frequency. For an input frequency of 1MHz, the PLL output settles to an approximate DC value of 12.1V, as expected. For input frequencies of f_{in}=0.6MHz and 1.4MHz, the PLL cannot capture the signal, and the VCO oscillates near its free-running frequency of 1MHz.

For this report, the concern is with the PLL operation and its simulation time. Table 2.1 shows the average simulation times of the 560B compared to the Simplified 560B PLL and the PLL(0) macromodel, introduced in Section 2.2 and Chapter 3, respectively. The simulations were performed on four different machines: the VAX 8650, SUN 4/370, DEC 3100, and IBM RS6000. The 560B PLL takes a tremendous amount of CPU time in simulation in all cases, which is the basis of the need for a PLL macromodel as described in Chapter 1. These simulation results show that there are tremendous CPU-time savings if a PLL macromodel is used in place of an actual PLL.

2.2 Simulation of the Simplified 560B PLL

For simulation purposes, the 560B PLL can be simplified to reduce the simulation time. Such a simplified 560B PLL is shown in Figure 2.5 and is called the Simplified 560B. The principle PLL sub-functions [PC, filter, VCO] are retained as is, while the remainder of the circuitry (bias, level shifting, buffering, etc.) is replaced by ideal voltage sources, current sources, and dependent-sources. The required voltage and current levels in the Simplified PLL are taken from the DC operating point of the 560B, given in Figure 2.6. The PLL output is now taken differentially from the filtered PC output (does not have a DC bias), is not limited in any way (does not restrict the capture and lock ranges), and controls the VCO through a voltage-dependent current source.

The SPICE3 input file for the Simplified 560B PLL is shown in Figure 2.7; the corresponding output graphs are shown in Figures 2.8 and 2.9. Though the voltage waveforms of the 560B and the Simplified 560B cannot be directly compared due to output voltage limiting and offset of the 560B, the Simplified 560B operates much as expected for a PLL of its type. A larger capture range (approximately ±0.4MHz) is shown for the Simplified 560B, which is correct because the output voltage is not limited. The loop gain constant K_L is approximately 2 times the free-running frequency of the PLL[1], and the calculated capture and lock ranges are ±0.25MHz and ±3.14MHz, respectively. The simulated capture range is about 1.6 times larger than the calculated capture range, which is expected (see Appendix A), and the simulated lock range (shown in Figures 2.10 and 2.11), is very close to the calculated one if the lower lock limit is ignored (lower lock range predictions are too small). Also, the Simplified 560B has smaller fluctuations around the DC control voltage than the actual 560B; this results because the 560B amplifies the PC output through a differential pair while the Simplified 560B takes the PC output directly (this effect is best illustrated in the cases where the input frequency is not captured). Figure 2.12 demonstrates the capture and lock phenomena more vividly. The Simplified 560B captures the input signal with frequency 1.2MHz. Lock occurs when the input frequency linearly changes from 1.2MHz to 2.5MHz. However, when the input frequency linearly changes from 2.5MHz to 3.5MHz, the Simplified 560B loses lock. The simulation time of the Simplified 560B compares favorably to the 560B (about 2.4 times faster, on average). However, the PLL(0) macromodel introduced in Chapter 3 is over an order of magnitude faster than either.

CHAPTER 3: A PHASE-LOCKED LOOP MACROMODEL

The PLL simplification can be taken a step further by simplifying the main PLL sub-functions: the result is a PLL macromodel. Using a PLL macromodel is in contrast to another popular method of reducing simulation time: namely, behavioral modeling of a PLL[5]. With a PLL macromodel, the modeled circuit is evaluated in real-time, requires no special computer coding or delay-time equations, and is not dependent on special computer subroutines and/or compilers (needed to link the behavioral model to the rest of the circuit simulation). As such, a PLL macromodel is easier to incorporate into the mainstream of analog design simulation.

3.1 Configuration of the PLL(0) Macromodel

An early PLL macromodel, PLL(0), has been introduced by Pederson and Mayaram.[2] Its circuit schematic is shown in Figure 3.1. In this macromodel, each of the necessary functions of a PLL is replaced by a very simple SPICE2 circuit or element that realizes the desired function. The PC is a VCVS analog multiplier, the filter is a simple one-pole RC filer, and the VCO is a Wien VCO macromodel[2]. The operation of each of these PLL functions has been examined in detail[2]. There are, however, a few bothersome problems concerning the Wien VCO macromodel. One problem is created by the output resistance associated with the amplifier. This output resistance causes the VCO frequency to vary nonlinearly (see Appendix B). The VCO frequency with r_o effects is:

$$\omega_{osc} = \frac{1}{\sqrt{1 + r_o(\frac{1}{R} + kV_c)}}(\frac{1}{RC} + \frac{k}{C}V_c) \qquad [3.1]$$

where k is a chosen constant.

Table 3.1 shows the degradation in VCO frequency for a few given control voltages. Thus, for the PLL(0) macromodel without modification, control voltages larger than those normally expected result due to the r_o effects. For example, for $f_{osc} = 1MHz$, the expectation is V_c=0V. However, because of the r_o effects, a non-zero V_c is needed. For a VCO frequency of 1MHz, a control voltage that produces a frequency f_{oscx} such that $f_{oscx} = \frac{1}{.995} 1\,MHz = 1.005MHz$ is needed. That is a V_c=50mV instead of 0V to achieve a VCO frequency of 1MHz. Since the maximum limit on possible control voltage values is what limits the capture range, the capture range will be smaller than expected. To overcome this r_o frequency deviation, a very small r_o can be used with the amplifier. As an alternative, an additional VCVS can be used to buffer the effects of r_o.

Another problem with the VCO macromodel is found in the VCO output amplitude. One purpose of the limiting diodes and voltage sources is to limit the VCO output to a set value, 10V in this case. This is very hard to control, however, since the voltage drop across the diode depends nonlinearly on the current through it, and the current is constantly changing. Any change in r_o also changes the value of the constant voltage sources needed to generate a fixed limit on the VCO output voltage. In the PLL(0) macromodel with $r_o = 0.001\Omega$, the needed voltage source is 8.8825V. This value is for a diode with $I_s = 1x\,10^{-16}A$.

Fixing the output amplitude of the VCO is troublesome because the limiting process tends to create a distorted sinusoid (squashed tips), which alters the frequency of the output a little. This distortion is actually necessary for the proper operation of the Wien oscillator[2]. Because of this, however, the Wien VCO does not provide a good, fixed-amplitude, reliable frequency sinusoid as desired for a PLL macromodel. To compensate for the altered VCO output frequency, the VCO frequency-setting capacitor can be adjusted to C=150.1pF (value found by trial and error).

Yet another problem is a simulation problem: the Wien VCO macromodel must be shock excited to start its oscillation. A zero-valued voltage source with a voltage spike at time=0s, V_{trig}, is provided for this purpose. However, the value of the initial voltage spike affects the capture of the input. For a given transient print size and V_{max} step, the voltage spike must be adjusted so that the initial VCO output has already reached a 10V oscillation frequency but isn't delayed at the 10V limit. Figure 3.2 illustrates the former case (capture is not achieved at first due to a small VCO voltage) where the print size and V_{max} step of the .tran statement should be larger. Figure 3.3 illustrates the latter case (capture should not have been achieved, but was achieved due to the VCO staying at the maximum value for a long period of time) where the print size and V_{max} step of the .tran statement should be smaller. Care must be taken when running simulations to ensure that a proper initial VCO characteristic is achieved for more accurate capture ranges. This involves adjusting the V_{trig} voltage spike and the print size & V_{max} steps of the .tran statement.

To compensate for the problems encountered with the VCO macromodel, PLL(0) is adjusted so $r_o = 0.001\Omega$, the limiting voltage supplies are 8.8825V, C=150.1pF, and a proper Vtrig/.tran statement is used for the transient analysis.

3.2 Simulation of the PLL(0) Macromodel

The PLL(0) macromodel of Figure 3.1 has been tested for various sinusoidal input frequencies (constant & changing) and under various different PLL parameter value changes (K_L, filter bandwidth, ω_o). The PLL parameters are chosen to be as follows: $K_p = 5\frac{V}{rad}$, $K_o = .1\omega_o \frac{rad}{volts-sec}$, filter bandwidth=20KHz, the VCO output is sinusoidal with a 10V amplitude, and the input is sinusoidal with a 1V amplitude.

The SPICE3 input file for the PLL(0) macromodel is shown in Figure 3.4 and is used to generate Figures 3.5 though 3.7. Figure 3.5 shows capture for an input that is at a higher (1.147MHz) and lower (0.842MHz) frequency from the free-running frequency (1MHz). Note the high-frequency ripples, which are enlarged for their respective cases. Figure 3.5 also shows the steady-state phase differences for the two cases. The unsuccessful capture of an input frequency both higher (1.148MHz) and lower (0.841MHz) than the VCO free-running frequency is shown in Figure 3.6. This illustrates the capture range of the PLL(0) macromodel. An attenuated difference-frequency is apparent, and there is no steady-state phase difference between the input and the VCO signals for the uncaptured cases. For these cases, the VCO should oscillate at its free-running frequency of 1MHz since it cannot capture the input signal. However, due to the difference-frequency and sum-frequency ripples, the VCO oscillates at a lower frequency. Figure 3.7 shows the demodulated output of the PLL(0) macromodel for a frequency-modulated input. The carrier frequency is 1MHz, the signal frequency is at 10KHz, and the modulation index is 10.

Figures 3.8, 3.9 and 3.10 show the captures ranges for various simulated center frequencies, filter bandwidths, and K_L. As the center frequency increases, the general trend is that the capture range decreases. This is due to the filter attenuations being more pronounced for higher frequencies. As the filter bandwidth increases, the capture range also increases (shown in Figure 3.9) since the attenuation factor is less. As the loop gain constant K_L increases, the maximum magnitude of the control voltage increases, thus the capture range also increases. Figure 3.10 shows this general trend. Figure 3.10 also shows a comparison of the capture ranges for various K_L for both simulated and calculated (see Appendix A, Equation [A.10]) cases. The actual capture range is larger than the calculated capture range, which is expected (see Appendix A, paragraph following Equation [A.10]). The simulated capture range is consistently around 1.5 times larger than calculated.

The SPICE3 input file for the PLL(0) is also used in conjunction with a variable frequency input signal (commented out in Figure 3.4) to show the lock process and range of the PLL(0) macromodel. The graphs generated from this input file are shown in Figure 3.11. Both a multi-step, linearly increasing (1MHz→1.2MHz→1.4MHz→1.55MHz) and decreasing (1MHz→0.8MHz→0.6MHz→0.45MHz) input frequency is illustrated. The behavior of the PLL(0) macromodel shows an effective lock range of ±0.55MHz. The value is very close to the theoretical prediction of ±0.5MHz.

Figure 3.12 shows the simulated and calculated lock ranges for various K_L. The simulated lock range is always slightly larger than calculated. As a point of comparison, both the capture range and lock range are plotted together for various K_L in Figure 3.13; as mentioned in Appendix A, the lock range is usually larger than the capture range.

Typical simulation times for the PLL(0) macromodel are shown in Table 2.1, in comparison to the 560B and Simplified 560B PLLs. Using a PLL macromodel is clearly much faster than using the 560B or Simplified 560B PLL in simulation. PLL(0) shows a simulation time improvement (on average) of approximately 13.5 to 1 compared to the Simplified 560B simulation time and 32 to 1 compared to the 560B simulation time.

CHAPTER 4: BETTER/FASTER PHASE COMPARATORS

4.1 PCs with Different PC Characteristics

For the VCVS PC of the PLL(0) macromodel, the PC output is dependent upon the amplitude and shape of the inputs. If a plot of PC output voltage vs phase difference is made (general PC characteristics[3] are shown in Figure 4.1), the PC transfer function shows a sinusoidal characteristic; thus, K_p, the slope of the transfer function at a given operating point, is not constant over the various phase differences. As a consequence, the loop gain changes for different phase differences, and the capture and lock ranges are smaller than for a PLL with a constant K_p. Better phase comparators and phase comparator characteristics can be achieved to obtain better PLL performance.

A desirable PC characteristic is the triangular one (Curve 2 of Figure 4.1), which has a constant K_p (slope) for every 180^o. This triangular PC characteristic can be achieved in general if the PC output is a rectangular wave of a fixed amplitude independent of the input (see Appendix E). One way to accomplish this is to amplify the inputs of the PC, and then to limit them to a desired value to achieve a specific product. Alternatively, instead of modifying the input, the output of the PC can be amplified and limited to a fixed value. Either way, the sign of the product of the input and the VCO is the desired information while the PC output amplitude is predetermined and fixed. This amplification and limiting can be achieved by using a VCVS amplifier and two limiting diodes & constant voltage sources, c.f. the Wien VCO macromodel. The amplifying/limiting circuit is shown in Figure 4.2. Limiting the inputs to the PC would involve two amplifying/limiting circuits (a total of four diodes); a better approach is to amplify and limit the output of the multiplier, which uses only one amplifying/limiting circuit (two diodes) and conserves simulation time. Also, the amplifying/limiting circuit is not an extremely accurate voltage limiter, c.f. Section 3.1. In fact, the SPICE runs using amplified/limited input signals (see Figures 4.5 and 4.6) show the product of the input signals as a pseudo-square wave (a square wave with somewhat rounded tops instead of flat tops); the flaws of the amplifying/limiting circuit are apparent. Figure 4.3 is the SPICE3 input file for a PLL macromodel with amplified/limited inputs to the VCVS PC . Its circuit schematic is shown in Figure 4.4, and its output runs (showing capture and no-capture) are presented in Figures 4.5 and 4.6. Figure 4.7 is a SPICE3 input file of a PLL with an amplifier/limiter after the analog multiplication. Figure 4.8 shows a circuit schematic of this PLL, and its corresponding output runs showing capture and no-capture are in Figures 4.9 and 4.10.

The K_p for these PCs (see Appendix E) is:

$$K_p = \frac{2|V_{PC-max}|}{\pi} = \frac{20V}{\pi} = 6.37\frac{V}{rad} \qquad [4.1]$$

The lock and capture ranges are $\frac{\pi}{2}$ times the normal expression (see Appendix E):

$$\omega_c = \frac{\pi}{2}K_pAK_o\,|F(j\omega_i - \omega_o)| = 2\pi x\,10^6\,|F(j\omega_i - \omega_o)|$$

$$\approx 2\pi x\,141.4x\,10^3\frac{rad}{sec} \qquad [4.2]$$

$$\omega_l = \frac{\pi}{2}K_pAK_o = 2\pi x\,10^6\frac{rad}{sec} \qquad [4.3]$$

Great care must be used when establishing the two limiting voltages, as the capture and lock ranges are very sensitive to the amplitude of the PC output. This translates to a very careful choice of the values for the diodes and limiting voltage sources.

The simulation times (on the VAX 8650) of the two amplified/limited PLL macromodel cases are compared to the PLL(0) macromodel, and the results are shown in Table 4.1. Amplifying/limiting the PC inputs is about 2.25 times slower in simulation time than with no amplifying and limiting; amplifying/limiting the PC output is about 2.1 times slower in simulation time than without the amplifying and limiting.

The gains of having a triangular PC characteristic are having a constant K_p and larger lock and capture ranges ($\frac{\pi}{2}$ times the lock and capture ranges of a sinusoidal PC as explained in Appendix E). The immediate cost to the PLL macromodel, however, is the added simulation time to evaluate the diodes in the limiting scheme. Also, there are additional harmonics and high-frequency signals (square wave) that need to be filtered out; thus, for the same performance, a better filter is needed. But PLL operation still suffers from higher-order frequency components.

To obtain a sawtooth type of PC characteristic, the PC must distinguish between each individual sign of the input and VCO, not just the sign of their product. This is done in actual digital PLLs via rising and falling edge-triggered flip-flops; however, it is not easily achieved for macromodel purposes.

4.2 DC-Offset Canceling PCs

One of the drawbacks of the VCVS multiplier is its inability to handle a DC offset at the input. If there is an unwanted DC bias at the input, an extra term is produced at the output of the multiplier. The term is a bias-dependent version of the VCO signal; thus, a noise source at the VCO frequency is added to the system, though filtering may reduce the noise. The problem exists only if an intentional low-frequency component (i.e., for amplitude modulation) is not purposely introduced; for unwanted DC components in the input (offsets), the undesired VCO frequency feed-through is the problem. The methods of canceling a DC offset include using a differential PC or AC-coupling the input signal to the PC. For the purposes of this report, the input is assumed to be AC-coupled already.

4.3 Double-Balanced Analog Multiplier as a PC

The double-balanced analog multiplier, shown in Figure 4.11 is often used as a PC (as in the 560B). The double-balanced PC operates under the premise that the double-balanced analog multiplier is highly overdriven; as such, it is not a linear multiplication, but more of a switching between two predetermined values inherent to the PC. This is comparable to the amplified/limited PCs of Section 4.1. When the analog multiplier is overdriven, the PC output is independent of input shape or amplitude, and the PC characteristic is triangular with a fixed K_p. The double-balanced PC is analyzed by P. Gray and R. Meyer[1]; the value of K_p is

$$K_p = \frac{2}{\pi} I_{EE} R_C \qquad [4.4]$$

If the input is small, the double-balanced analog multiplier is not overdriven and K_p is not constant. This case is also addressed by Gray and Meyer[1]; the PC output becomes

$$V_{PC} = -\frac{2}{\pi} g_m R_C V_{in} \cos\phi \qquad [4.5]$$

However, in this report, the input is assumed to be large enough to overdrive the double-balanced analog multiplier.

The differential nature of the analog multiplier serves to cancel out unwanted DC offsets in the input. In addition, some noise and harmonic-distortion cancellations are achieved due to the cross-coupled nature of the upper differential pairs. All these properties (especially the triangular PC characteristic) make the double-balanced analog multiplier a popular circuit for a phase comparator. However, this PC has six transistors, too many to warrant using in a PLL macromodel unless the benefit is great. The same general effect (minus the noise and harmonic distortion cancellation) is achievable with an AC-coupled VCVS PC with an amplified/limited output (which only has two diodes).

In Figure 4.12, a SPICE3 file for a PLL macromodel with a double-balanced analog multiplier PC is given. This PLL macromodel (schematic shown in Figure 4.13) includes a capacitor across the differential output of the PC for the low-pass filtering and a VCO that depends on a differential control voltage. Other possible configurations for the filtering are shown in Figure 4.14: two capacitors, or a separate, unilateral filter.

The simulated results for the PLL configuration with a double-balanced analog multiplier PC are shown in Figures 4.15 through 4.18. The corresponding capture and lock ranges are very close to those of a PLL with an amplified/limited VCVS PC. All output plots have a high-frequency ripple (shown in Figure 4.16), but the ripple is triangular instead of sinusoidal (as for the PLL(0) macromodel). Due to the feed-through of the square wave's additional high-frequency harmonics, there is more ripple in the control voltage to the VCO than for a sinusoidal PC characteristic.

The corresponding simulation times (on the VAX 8650) of PLL macromodels with the double-balanced analog multiplier PC and the AC-coupled VCVS PC with an amplified/limited output are shown in Table 4.2. The cost in simulation time for the double-balanced analog multiplier is about double that of the AC-coupled VCVS PC with output amplified.

4.4 A Future Possible PC Macromodel

It is desired to have a fixed amplitude rectangular wave at the output of the PC. One method is to use an amplified VCVS output with limiting diodes, but that does not produce accurate amplitudes. Another method is to use an overdriven, double-balanced multiplier for the PC; however, that configuration uses six transistors and adds to the simulation time. A simple method of achieving a rectangular-wave PC output would be possible if the sgn(x) function were available on SPICE3 (currently not available). A rectangular PC output would be a simple V=A sgn{Vin*Vvco}, where A is the desired amplitude of the rectangular wave, and the sgn(x) function provides the sign of the product. The amplitude would be constant, the PC characteristic would be triangular, K_p would be constant, and the evaluation would take minimal transient analysis time.

4.5 Phase Comparator Summary

A final comparison of the PCs considered in this chapter is shown in Table 4.3 (simulations done on the VAX 8650). Of all the possible PCs presented, the VCVS analog multiplier from the PLL(0) macromodel is the fastest in simulation time. To achieve a triangular PC characteristic, the amplified and limited PC output configuration is the best, a little over 2 times slower in simulation than the original VCVS multiplier. Clearly, the double-balanced analog multiplier PC is the most costly of all the PCs, at almost 4 times slower than the original VCVS PC.

CHAPTER 5: BETTER/FASTER VOLTAGE-CONTROLLED OSCILLATORS

5.1 Limitations of the Wien VCO Macromodel

The Wien VCO macromodel in the PLL(0) macromodel has several limitations, as presented in Chapter 3. In particular, the two most troublesome problems with the Wien VCO are that the VCO output does not have a well-controlled amplitude and the VCO output is a distorted sinusoid. These two problems cannot be fixed if diodes are used to limit the VCO output. Implementing a more complex configuration to produce the limiting would probably not be worth the increased simulation time. With this in mind, alternative VCOs are now explored.

5.2 Sinusoid with a Voltage-Controlled Argument

The simplest VCO possible is a sinusoid or a square pulse whose phase function changes linearly with a control voltage. In SPICE, the pulse function must have a fixed period, so it cannot be used as a VCO. However, the argument of a sine wave can be modified to produce a VCO function in SPICE3 and PSPICE, though not is SPICE2. The correct method for modifying the sine argument is to integrate the control voltage (see Appendix C for an explanation). Such an Integrating Sine VCO is shown in Figure 5.1, and can be implemented in SPICE3 as follows:

```
BVCO1  7  0  V=10*sin(2*pi*1e+6*V(9))
Ccap  9  0  1pF
Bcap  0 9 I=1e-12* (1+0.1V(3))                              [5.1]
```

where V(3) is the control voltage.

Figure 5.2 is a SPICE3 input file for this Integrating Sine VCO. Figure 5.3 is the output waveform for this VCO using a changing input (also shown in the same figure). The desired VCO output for a changing control voltage is seen for the integrating case. For comparison, the schematic for the Wien VCO macromodel is shown in Figure 5.4, and its corresponding SPICE3 input file is listed in Figure 5.5. The outputs of the Integrating Sine VCO and the Wièn VCO macromodel are shown together in Figure 5.6; the output waveforms are very similar. The simulation times (on the VAX 8650) of these two VCOs are shown in Table 5.1; the Integrating Sine VCO is about 2 times faster in simulation time than the Wien VCO macromodel for the same input and analysis situations.

5.3 Multivibrators as VCOs

5.3.1 Two-Inverter Multivibrators

A two-inverter multivibrator circuit is shown in Figure 5.7. This system of two inverters has two possible stable states. The charging/discharging action of the capacitor between the two inverters causes the inverters to switch between their known, quasi-stable states. This switching action is what constitutes a multivibrator. The oscillating output is a rectangular waveform or a triangular waveform, depending on where the output is taken. This oscillator configuration is presented in detail by Pederson and Mayaram[2]. Alternatively, a ring oscillator (n inverters connected in a ring) can be made into a VCO as in Figure 5.8 for n=2; however, multiple resistors and capacitors are needed, which would increase the simulation time with no gain over the two-inverter multivibrator configuration. Thus, ring oscillators are not considered.

Many potential VCOs for the PLL macromodel can be derived from the configuration of Figure 5.7, depending on the implementation of the inverters. The three most common inverters are the resistive-load inverter, the depletion-load inverter, and the CMOS inverter. For simulation purposes, MOS transistors are used instead of bipolar transistors because bipolar transistors have a base current that detracts from the ideal behavior of the oscillator. Figures 5.9 through 5.11 show the MOS circuit schematics of the Resistive-Load, Depletion-Load, and CMOS 2-Inverter VCOs, respectively. Their corresponding SPICE3 input files and output waveforms are shown in Figures 5.12 through 5.14, respectively. Values of R and C are picked to produce a free-running frequency of 1MHz. As in the Wien VCO macromodel, a voltage-controlled linear conductance is used to produce a variable oscillator frequency. The output waveforms are all produced by using a 0V control signal.

The output waveforms for the Resistive-Load, Depletion-Load, and CMOS 2-Inverter VCOs are very similar. The top waveform shows the inverter input switching at the MOS threshold levels; the middle waveform shows the charging/discharging voltage across the capacitor, and the bottom waveform is the square wave from an inverter output. All three output waveforms oscillate at the correct frequency and are possible oscillator outputs.

There are many disadvantages, however, that would result from using these VCOs in a PLL macromodel. One disadvantage is that the oscillating signals have a DC value; thus, the VCO signal needs to be AC-coupled or level-shifted prior to the PC. Another problem is that of an asymmetric duty cycle. The charging and discharging times of the capacitor are not equal, leading to a duty cycle that is not 50%. This introduces an unwanted average value or DC offset at the output of the PC. By meticulously picking component values and sizing transistors, however, a duty cycle of 50% can be achieved. Problems that are not solvable include the non-constant square-wave output amplitude and the very noticeable exponential curvatures of the "triangular" waveform. In addition, the transistors have inherent conductances across their terminals and various junction and bulk capacitances that cannot be eliminated for simulation purposes. These additional components affect the VCO frequency in an undesired manner. A final disadvantage is that the simulation times of the two-inverter multivibrators also compare unfavorably to the Wien VCO macromodel. The simulation times (on the VAX 8650) are shown in Table 5.2; the Resistive-Load, Depletion-Load, and CMOS 2-Inverter VCOs are 2.3, 3.2, and 3.3 times slower than the Wien VCO macromodel, respectively.

5.3.2 Emitter-Coupled Pair Multivibrator

One popular VCO in actual PLLs like the 560B is the emitter-coupled pair (ECP) multivibrator, shown in Figure 5.15 (a bipolar case is used to be consistent with the 560B). The same principles apply as in the 2-inverter multivibrator case; however, the frequency is dictated by controlling the current that charges/discharges the state-switching capacitor. A SPICE3 input file for the ECP multivibrator of Figure 5.15 is shown in Figure 5.16. The corresponding output signals (both square wave and triangular wave) are plotted in Figure 5.17. The output signal is differential and does not have DC-bias problems. The duty cycle of the output is well-controlled at 50% since the charging/discharging current is controlled; also, the amplitude is set by two limiting current cases (either all the current or a preset bleed current goes through the output transistor), so the amplitude is well maintained. The ECP multivibrator is examined by P. Gray and R. Meyer[1]; the free-running frequency of the VCO is:

$$f_o = \frac{1}{T} = \frac{I_{bias}}{4CV_{BE(on)}} \qquad [5.2]$$

If the current is linearly changed by a control voltage $I = I_{bias}(1+kV_c)$ where k is a chosen constant, the value of K_o is

$$K_o = k\omega_o \qquad [5.3]$$

where ω_o is the free-running frequency in $\frac{radians}{sec}$. For PLL macromodel purposes, however, the ECP multivibrator appears to have too many transistors for fast simulation. Table 5.3 shows the simulation times (on the VAX 8650) between the ECP multivibrator and the Wien VCO macromodel and verifies the drastic speed disadvantage (over 8 times slower in CPU time) of the ECP VCO.

5.4 Voltage-Controlled Oscillator Summary

When the simulation times (on VAX 8650) of all investigated VCOs are compared (see Table 5.4), the Integrating Sine VCO is superior, close to two times faster than the next fastest VCO (the Wien VCO macromodel). The Wien VCO is a larger circuit with more circuit equations (16) than the Integrating Sine VCO (9 circuit equations). This translates to a larger matrix to invert and solve, and thus slows down the overall Wien VCO simulation time compared to the Integrating Sine VCO (which has a simpler integrating-with-a-capacitor circuit). The remaining VCOs are much slower; clearly, the ECP multivibrator VCO is the most CPU-time costly, at over an order of magnitude slower in simulation time than the Integrating Sine VCO.

CHAPTER 6: TWO PROPOSED PLL MACROMODELS

6.1 Two Proposed Configurations

From the information on PLL components presented in Chapters 4 and 5, two configurations for PLL macromodels are proposed. The first proposed PLL macromodel, shown in Figure 6.1, uses a simple VCVS analog multiplier, a simple RC filter, and an Integrating Sine VCO. This macromodel is labeled PLL(1) and is a combination of the fastest (CPU time) elements for each PLL component. This PLL, however, has a PC with a sinusoidal characteristic; the effective K_p factor, and therefore K_L, depend on the amplitudes of the PC inputs.

The second PLL macromodel, PLL(2), is shown in Figure 6.2. It is the same as PLL(1), but has an additional amplifying/limiting function at the PC output and a separate, unilateral RC filter. This extra amplifying/limiting function causes the PC output to be independent of the input amplitudes; thus, the PC has a triangular characteristic, and K_p is independent of the PC input amplitudes. The PC output, however, is not a constant amplitude square wave as desired, but only a near-constant amplitude due to the voltage drop changes across the limiting diodes. To best approximate the desired square wave (10V zero-to-peak in this example), the maximum amplitude of the PC output is allowed to be slightly above 10V so that the area of the pseudo-square wave can be made approximately equal to that of a true square-wave.

6.2 Simulating the Two Proposed PLL Macromodels

The SPICE3 input files for PLL(1) and PLL(2) are listed in Figures 6.3 and 6.4, respectively. The two PLL macromodels are designed to have a PC output voltage with a maximum zero-to-peak value of 10V, a low-pass filter bandwidth of 20kHz, a K_o of $0.1\frac{rad}{sec-V}$, a VCO center frequency of 1MHz, a sinusoidal input amplitude of 1V, and a VCO output amplitude of 10V. With these values, K_p and K_L for the two proposed PLL macromodels are:

Macromodel	K_p	K_L	
PLL(1)	$5\frac{V}{rad}$	0.5MHz	[6.1]
PLL(2)	$6.37\frac{V}{rad}$	0.637MHz	[6.2]

Figures 6.5 through 6.8 show output plots for PLL(1). The capture range is +0.13MHz/-0.14MHz, as shown in Figures 6.5 and 6.6. This capture range is larger than the theoretical prediction (approximately 1.35 times larger) but is expected. The lock range is ±0.5MHz, as shown in Figures 6.7 and 6.8, and is precisely what theoretical results predict.

Figures 6.10 through 6.13 show the output plots for PLL(2). As mentioned above and shown in Figure 6.9, the outputs of the PLL(2) PC are allowed be a little above the maximum 10V so the areas of the pseudo-square wave are approximately that of a true square-wave. The capture range is +0.165MHz/-0.175MHz and is shown in Figures 6.10 and 6.11. The simulated capture range is approximately 1.5 times larger than theoretical predictions. The lock range, shown in Figures 6.12 and 6.13, is +0.97MHz/-0.85MHz. The lower lock range is smaller, most probably being affected by being too close to DC. The lock range is very close to the theoretical prediction, which predicts a lock range equal to $\frac{\pi}{2}$ times the loop gain at DC for a triangular PC characteristic, i.e.,

$f_l = \frac{\pi}{2} 0.637MHz = 1MHz$.

The capture ranges of both PLL cases slightly favor frequencies less than the VCO's center frequency. This is most probably due to initial conditions, which are very important to capture. The initial conditions for all cases are the same: both inputs to the PC start off positive, and the PLLs are non-inverting control voltage configurations (see Appendix D).

The average simulation times for PLL(1) and PLL(2) are presented in Table 6.1. Simulations were conducted on the VAX 8650, SUN 4/370, DEC 3100, and IBM RS6000. PLL(1) is the fastest in CPU time, which is expected. PLL(2) is the approximately 4 times slower in simulation time.

CHAPTER 7: CONCLUSIONS

7.1 Operating Characteristic

The simulated lock and capture ranges for various PLL macromodels are shown in Table 7.1 along with theoretical values calculated from the appropriate equations. The 560B and Simplified 560B are listed for completeness. The PLL macromodels all behave very much as predicted. All the simulated capture ranges are slightly larger (1.2 to 1.6 times) than the theoretical ranges, which is expected (see Appendix A). PLL(1) is most like the sinusoidal PLLs described in books, and it conforms well to the theoretical ranges derived for such a system. The PLL(0) macromodel has a VCO output that is a distorted sinusoid with an average half-cycle value somewhat larger than that of PLL(1). This results in a larger actual K_L and larger lock & capture ranges for PLL(0). PLL(2) has a much larger lock and capture ranges due to a triangular PC characteristic. For the same PC output amplitude, the effective K_p of a triangular PC characteristic is also larger than that of a sinusoidal PC characteristic ($\frac{10}{\frac{\pi}{2}} \frac{V}{rad}$ instead of $\frac{10}{2} \frac{V}{rad}$).

7.2 Simulation Time

The relative simulation times for various PLL macromodels are shown in Table 7.2. The 560B and Simplified 560B simulation times are listed to show the CPU-time savings when using a PLL macromodel. Relative simulation times are obtained per computer by normalizing the fastest time and lowest iteration count on that machine to one, while scaling every other PLL configuration's simulation times and iteration counts accordingly (the normalization factors are given at the bottom of the table). The simulations were performed on four machines: the VAX 8650, SUN 4/370, DEC 3100, and IBM RS6000. Though the number of iterations for each circuit is nearly equal for all computers tested, the actual simulation times for each machine are very different, depending on the speed of the respective machine.

From the data in Table 7.2, the slowest macromodel is over an order of magnitude faster in simulation time than the 560B; the fastest macromodel is over 55 times faster (on average) in simulation time. Clearly, the fastest PLL macromodel is PLL(1), approximately 1.8 times faster (on average) than PLL(0); both PLL(1) and PLL(0) are PLLs with a sinusoidal PC characteristic. The fastest PLL macromodel with a triangular PC characteristic is PLL(2), which is approximately 4 times slower than PLL(1).

7.3 Overall Proposed PLL Macromodels

The fastest, most accurate PLL macromodel investigated is PLL(1). Its operation is based on replacing the PC and VCO functions in a PLL with simple circuit equations that are quickly evaluated by the simulator. It has, however, a sinusoidal PC: the PLL's lock and capture ranges are affected by the input amplitude. If this is not acceptable, an alternative PLL macromodel, PLL(2), is recommended. PLL(2) modifies PLL(1) to be independent of input amplitude and shape. PLL(2) is less than 4 times slower than PLL(1), but it is still much faster than the 560B PLL.

DIRECTORY OF TABLES

PLL	VAX 8650	SUN 4/370	DEC 3100	IBM RS6000
Config.	Time(sec) : # Iter.	Time(sec) : # Iter.	Time(sec) : # Iter.	Time(sec) : # Iter.
560B	4949.7 : 138187	2554.4 : 138476	2662.7 : 137755	1687.9 : 138009
# msec/iter.	35.82	18.45	19.33	12.23
Sim. 560B	2111.7 : 145099	1531.7 : 144766	952.0 : 143238	662.8 : 142319
# msec/iter.	14.55	10.58	6.65	4.66
PLL(0)	136.8 : 33349	114.5 : 33349	70.4 : 33349	55.3 : 33349
# msec/iter.	4.10	3.43	2.11	1.66

TABLE 2.1: AVERAGE SIMULATION TIMES OF THE 560B, SIMPLIFIED 560B, AND PLL(0)

Ro	Vc=-5V	Vc=-2.5V	Vc=0V	Vc=2.5V	Vc=5V
10	0.9975093	0.9962710	0.9950372	0.9938080	0.9925833
1	0.9997501	0.9996252	0.9995004	0.9993756	0.9992508
0.1	0.9999750	0.9999625	0.9999500	0.9999375	0.9999250
0.01	0.9999975	0.9999963	0.9999950	0.9999938	0.9999925
0.001	0.9999997	0.9999996	0.9999995	0.9999993	0.9999992

Table 3.1: FREQUENCY REDUCTION FACTORS FOR R=1Kohm AND VARIOUS Ro & Vcontrol

Vin Freq.	No Amp/Lim	Amp/Lim @ PC Input	Amp/Lim @ PC Output
	Time(sec) : # Iter.	Time(sec) : # Iter.	Time(sec) : # Iter.
0.808meg	85.01 : 23037	178.05 : 31196	162.63 : 33069
0.809meg	79.69 : 22961	169.15 : 29470	164.38 : 33459
1.000meg	84.00 : 23157	199.64 : 35600	198.57 : 38671
1.189meg	83.53 : 23136	190.21 : 33284	171.31 : 34960
1.190meg	80.73 : 23114	188.37 : 32648	163.36 : 34225
Average:	82.59 : 23081	185.08 : 32440	172.05 : 34877
# msec/Iter.:	3.58	5.71	4.93

Table 4.1: SIMULATION TIMES OF NO AMPLIFYING/LIMITING OF THE PC AND AMPLIFYING/LIMITING AT THE PC INPUT AND OUTPUT (on VAX 8650)

Vin Freq.	Double-Balanced PC	AC-Coupled, Amp/Lim PC
	Time(sec) : # Iter.	Time (sec) : # Iter.
1.2meg	345.75 : 31915	176.42 : 36092
1.1meg	334.27 : 30905	178.63 : 34209
1.0meg	318.17 : 29567	180.78 : 38204
0.9meg	288.24 : 28101	162.57 : 31892
0.8meg	311.70 : 29745	166.00 : 33155
Average:	319.63 : 30047	172.88 : 34710
# msec/Iter.:	10.64	4.98

Table 4.2: SIMULATION TIMES (VAX 8650) FOR PLL MACROMODELs WITH A DOUBLE-BALANCED PC AND WITH AN AC-COUPLED, VCVS MULTIPLIER WITH AMPLIFIED/LIMITED PC OUTPUT

PC Implementation in PLL	Avg# sec : # Iter.	Avg# msec/# Iter.
VCVS Analog Multiplier PC:	84.27 : 23077.2	3.65
VCVS w/ Amplified/Limited Output:	172.05 : 34876.8	4.93
VCVS w/ Amplified/Limited Inputs:	185.08 : 32439.6	5.71
Double-Balanced PC:	319.63 : 30046.6	10.64

Table 4.3: SUMMARY OF SIMULATION TIMES FOR PLL MACROMODELs WITH VARIOUS PCs (VAX 8650)

VCO Freq.	Integrating Sine VCO Time(sec) : # Iter.	Wien VCO Time(sec) : # Iter.
0.5MHz	18.37 : 12016	30.07 : 9074
0.6MHz	19.53 : 12016	31.52 : 9761
0.7MHz	18.03 : 12017	33.75 : 10091
0.8MHz	18.63 : 12017	31.65 : 10439
0.9MHz	17.43 : 12017	36.35 : 11012
1.0MHz	19.30 : 12017	34.72 : 11558
1.1MHz	18.75 : 12017	35.87 : 12161
1.2MHz	18.27 : 12017	37.58 : 12519
1.3MHz	17.27 : 12017	38.53 : 12933
1.4MHz	16.93 : 12018	40.59 : 13841
1.5MHz	16.65 : 12018	45.06 : 14708
Average:	18.11 : 12017	35.97 : 11645
# msec/Iter.:	1.51	3.09

Table 5.1: SIMULATION TIMES (VAX 8650) FOR THE INTEGRATING SINE VCO AND THE WIEN VCO

VCO Freq.	Resistive Load Time(sec):Iter.	CMOS Inverters Time(sec):Iter.	Depletion Loa Time(sec):Iter.	Wien Macromodel Time(sec):Iter.
0.5MHz	57.10 : 17392	76.81 : 17510	74.62 : 17058	30.07 : 9074
0.6MHz	60.47 : 18791	87.08 : 19896	83.82 : 19007	31.52 : 9761
0.7MHz	63.83 : 20549	97.30 : 21873	88.56 : 20882	33.75 : 10091
0.8MHz	71.17 : 22328	99.83 : 23711	93.29 : 22448	31.65 : 10439
0.9MHz	79.19 : 24162	109.56 : 25564	107.39 : 24587	36.35 : 11012
1.0MHz	81.23 : 25677	112.41 : 27067	111.51 : 26906	34.72 : 11558
1.1MHz	84.29 : 27217	123.70 : 29264	123.60 : 28652	35.87 : 12161
1.2MHz	87.24 : 28582	135.93 : 31133	131.30 : 30368	37.58 : 12519
1.3MHz	96.78 : 30311	143.83 : 33036	142.84 : 32506	38.53 : 12933
1.4MHz	106.20 : 31876	154.19 : 34916	147.30 : 34196	40.59 : 13841
1.5MHz	110.33 : 33574	154.22 : 36658	163.55 : 35975	45.06 : 14708
Average:	81.62 : 25496	117.71 : 27330	115.25 : 26599	35.97 : 11645
# msec/Iter:	3.20	4.31	4.33	3.09

Table 5.2: SIMULATION TIMES (VAX 8650) OF THE 2-INVERTER MULTIVIBRATORs AND THE WIEN MACROMODEL

VCO Freq.	ECP Multivibr. DE Time(sec) : # Iter.	ECP Multivibr. SE Time(sec) : # Iter.	Wien Macromodel Time(sec) : # Iter.
0.5MHz	177.89 : 23083	170.76 : 23081	30.07 : 9074
0.6MHz	193.56 : 25692	198.40 : 25885	31.52 : 9761
0.7MHz	231.66 : 29242	220.08 : 29299	33.75 : 10091
0.8MHz	250.54 : 32333	246.39 : 32258	31.65 : 10439
0.9MHz	267.19 : 35384	276.76 : 35245	36.35 : 11012
1.0MHz	313.56 : 38398	309.98 : 38375	34.72 : 11558
1.1MHz	334.31 : 41411	322.56 : 41248	35.87 : 12161
1.2MHz	341.76 : 44284	346.62 : 44469	37.58 : 12519
1.3MHz	381.98 : 47505	367.88 : 47499	38.53 : 12933
1.4MHz	397.77 : 51295	402.99 : 51280	40.59 : 13841
1.5MHz	417.83 : 54458	401.17 : 54329	45.06 : 14708
Average:	300.73 : 38463	296.69 : 38452	35.97 : 11645
# msec/Iter.:	7.82	7.71	3.09

Table 5.3: SIMULATION TIMES (VAX 8650) FOR ECP MULTIVIBRATOR AND WIEN MACROMODEL (Differential and Single-Ended Control Voltage for ECP)

VCO MODEL	Avg# sec : # Iter.	Avg# msec/# Iter.
Integrating Sine VCO:	18.11 : 12017	1.51
Wien VCO Macromodel:	35.97 : 11645	3.09
Resistive-Load 2-Inv. Multivibrator:	81.62 : 25496	3.20
CMOS 2-Inv. Multivibrator:	117.71 : 27330	4.31
Depletion-Load 2-Inv. Multivibrator:	115.25 : 26599	4.33
ECP Multivibrator Single-Ended Vin:	296.69 : 38452	7.72
ECP Multivibrator Differential Vin:	300.73 : 38463	7.82

Table 5.4: SUMMARY OF SIMULATION TIMES FOR VARIOUS VCOs (VAX 8650)

PLL Config.	VAX 8650 Time(sec) : # Iter.	SUN 4/370 Time(sec) : # Iter.	DEC 3100 Time(sec) : # Iter.	IBM RS6000 Time(sec) : # Iter.
PLL(1)	73.743 : 29997	79.936 : 29997	34.055 : 29997	30.901 : 29997
# msec/iter.	2.46	2.66	1.14	1.03
PLL(2)	311.645 : 85300	404.552 : 85197	147.837 : 84995	121.792 : 85023
# msec/iter.	3.65	4.75	1.74	1.43

TABLE 6.1: AVERAGE SIMULATION TIMES OF THE PLL(1) AND PLL(2) MACROMODELS

PLL Macromodel Type	KL (MHz) Theor.	Capture Range (MHz) Theor.	 Simul.	Lock Range (MHz) Theor.	 Simul.
560B PLL:	~2	~+/-.25	volt. lim. +.21/-.19	~+/-3.14	volt. lim. unavail.
Simplified 560B PLL:	~2	~+/-.25	+.4/-.37	~+/-3.14	+3.2/-.8
Pederson/Mayaram PLL(0):	.5	+/-.1	+.15/-.16	+/-.5	+.55/-.55
Proposed PLL(1):	.5	+/-.1	+.13/-.14	+/-.5	+.5/-.5
Proposed PLL(2):	.64	+/-.14	+.17/-.18	+/-1.0	+.97/-.85

Table 7.1: SUMMARY OF PLL OPERATION FOR VARIOUS PLL CONFIGURATIONS

PLL Macromodel Type	Relative # sec : # Iterations VAX 8650	 SUN 4/370	 DEC 3100	 IBM RS6000
560B PLL:	67.12 : 4.61	31.96 : 4.62	66.44 : 4.59	54.62 : 4.60
Simplified 560B PLL:	28.64 : 4.84	19.16 : 4.83	27.95 : 4.77	21.45 : 4.74
Pederson/Mayaram PLL(0):	1.85 : 1.11	1.43 : 1.11	2.07 : 1.11	1.79 : 1.11
Proposed PLL(1):	1 : 1	1 : 1	1 : 1	1 : 1
Proposed PLL(2):	4.23 : 2.84	5.06 : 2.84	4.34 : 2.83	3.94 : 2.83
PLL(1) avg. actual data:	73.74 : 29997	79.94 : 29997	34.06 : 29997	30.90 : 29997

Table 7.2: SUMMARY OF RELATIVE SIMULATION TIMES FOR VARIOUS PLL CONFIGURATIONS

FIGURE CAPTIONS

Figure E.2a: Average PC Output vs Phase Difference for Non-Inverting Multiplier

Figure E.2b: Average PC Output vs Phase Difference for Inverting Multiplier

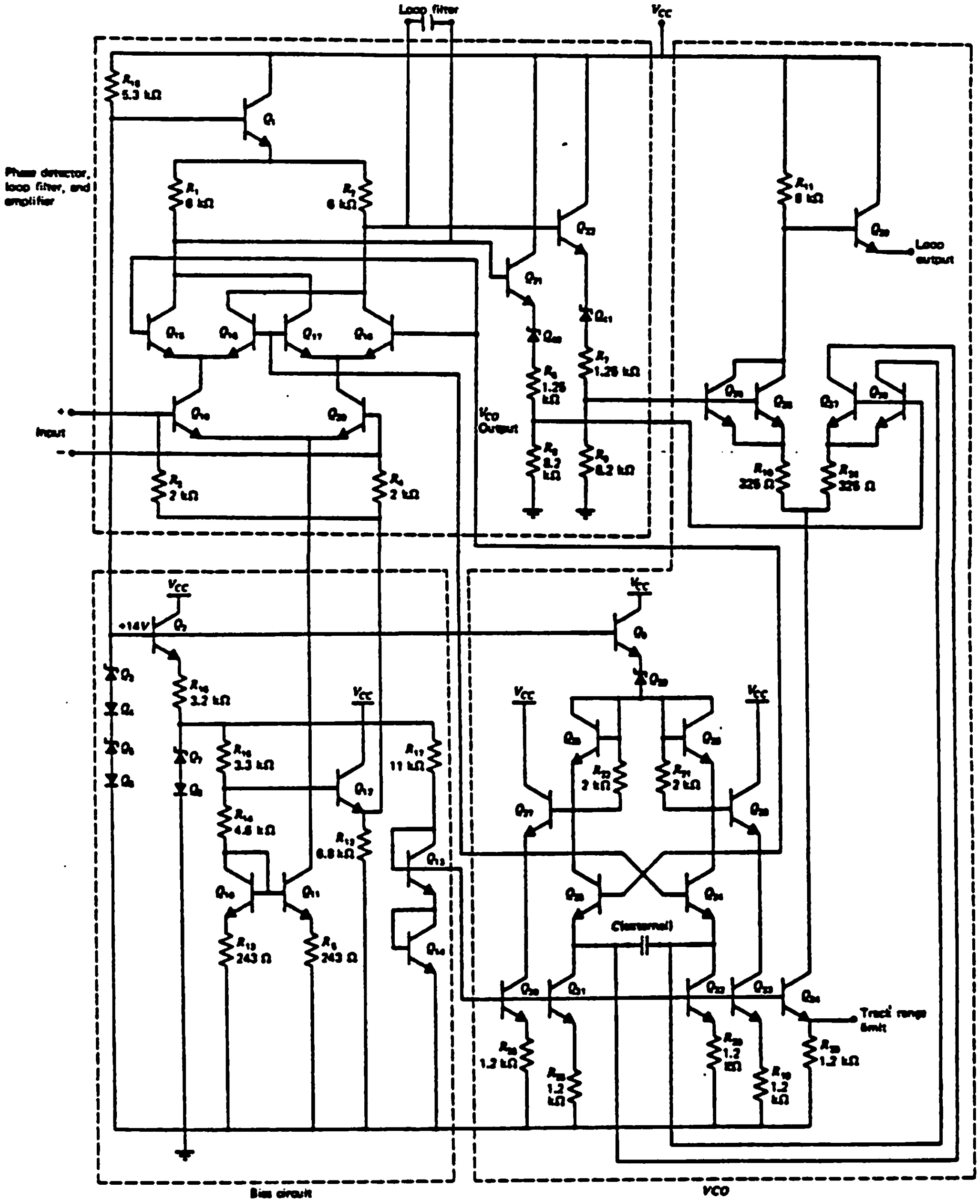
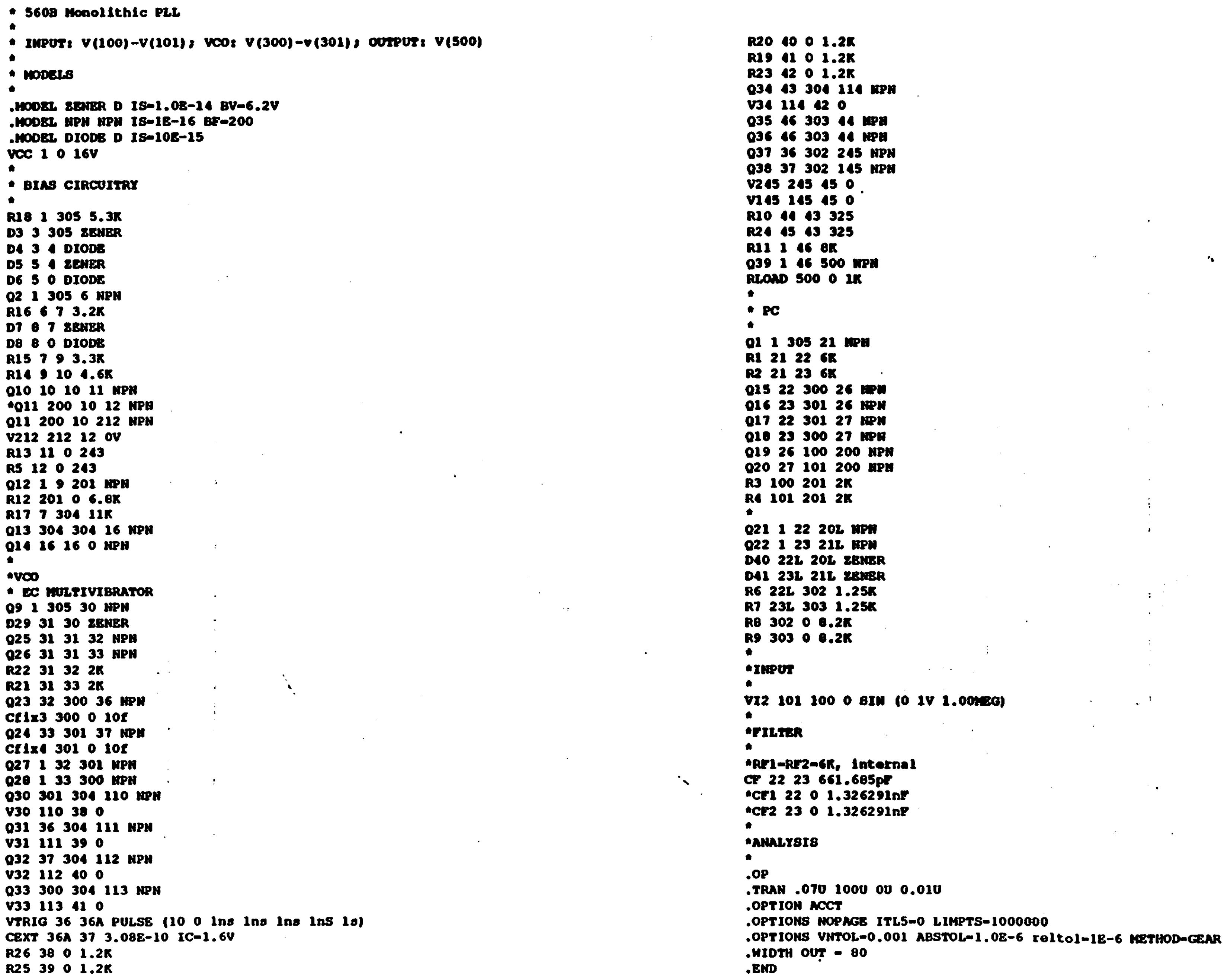


Figure 2.1: The 560B Monolithic PLL Circuit



```
* 560B Monolithic PLL
*
* INPUT: V(100)-V(101); VCO: V(300)-v(301); OUTPUT: V(500)
*
* MODELS
*
.MODEL ZENER D IS=1.0E-14 BV=6.2V
.MODEL NPN NPN IS=1E-16 BF=200
.MODEL DIODE D IS=10E-15
VCC 1 0 16V
*
* BIAS CIRCUITRY
*
R18 1 305 5.3K
D3 3 305 ZENER
D4 3 4 DIODE
D5 5 4 ZENER
D6 5 0 DIODE
Q2 1 305 6 NPN
R16 6 7 3.2K
D7 8 7 ZENER
D8 8 0 DIODE
R15 7 9 3.3K
R14 9 10 4.6K
Q10 10 10 11 NPN
*Q11 200 10 12 NPN
Q11 200 10 212 NPN
V212 212 12 0V
R13 11 0 243
R5 12 0 243
Q12 1 9 201 NPN
R12 201 0 6.8K
R17 7 304 11K
Q13 304 304 16 NPN
Q14 16 16 0 NPN
*
*VCO
* EC MULTIVIBRATOR
Q9 1 305 30 NPN
D29 31 30 ZENER
Q25 31 31 32 NPN
Q26 31 31 33 NPN
R22 31 32 2K
R21 31 33 2K
Q23 32 300 36 NPN
Cfix3 300 0 10f
Q24 33 301 37 NPN
Cfix4 301 0 10f
Q27 1 32 301 NPN
Q28 1 33 300 NPN
Q30 301 304 110 NPN
V30 110 38 0
Q31 36 304 111 NPN
V31 111 39 0
Q32 37 304 112 NPN
V32 112 40 0
Q33 300 304 113 NPN
V33 113 41 0
VTRIG 36 36A PULSE (10 0 1ns 1ns 1ns 1nS 1s)
CEXT 36A 37 3.08E-10 IC=1.6V
R26 38 0 1.2K
R25 39 0 1.2K
R20 40 0 1.2K
R19 41 0 1.2K
R23 42 0 1.2K
Q34 43 304 114 NPN
V34 114 42 0
Q35 46 303 44 NPN
Q36 46 303 44 NPN
Q37 36 302 245 NPN
Q38 37 302 145 NPN
V245 245 45 0
V145 145 45 0
R10 44 43 325
R24 45 43 325
R11 1 46 8K
Q39 1 46 500 NPN
RLOAD 500 0 1K
*
* PC
*
Q1 1 305 21 NPN
R1 21 22 6K
R2 21 23 6K
Q15 22 300 26 NPN
Q16 23 301 26 NPN
Q17 22 301 27 NPN
Q18 23 300 27 NPN
Q19 26 100 200 NPN
Q20 27 101 200 NPN
R3 100 201 2K
R4 101 201 2K
*
Q21 1 22 20L NPN
Q22 1 23 21L NPN
D40 22L 20L ZENER
D41 23L 21L ZENER
R6 22L 302 1.25K
R7 23L 303 1.25K
R8 302 0 8.2K
R9 303 0 8.2K
*
*INPUT
*
VI2 101 100 0 SIN (0 1V 1.00MEG)
*
*FILTER
*
*RF1=RF2=6K, internal
CF 22 23 661.685pF
*CF1 22 0 1.326291nF
*CF2 23 0 1.326291nF
*
*ANALYSIS
*
.OP
.TRAN .07U 100U 0U 0.01U
.OPTION ACCT
.OPTIONS NOPAGE ITL5=0 LIMPTS=1000000
.OPTIONS VNTOL=0.001 ABSTOL=1.0E-6 reltol=1E-6 METHOD=GEAR
.WIDTH OUT = 80
.END
```

Figure 2.2: A SPICE3 Input File for the 560B PLL

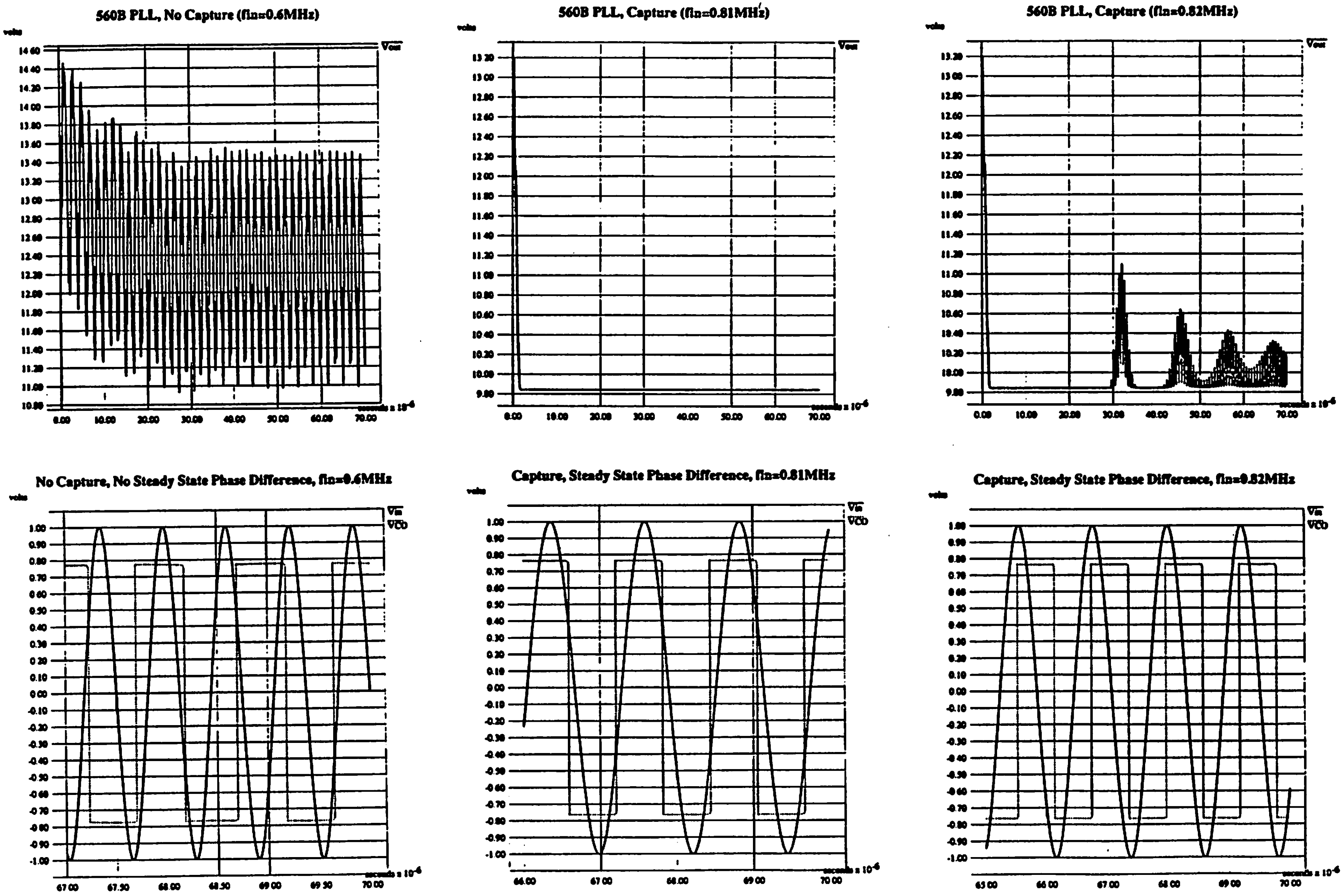


Figure 2.3: 560B Capture Behavior (0.6MHz,0.81MHz,0.82MHz)

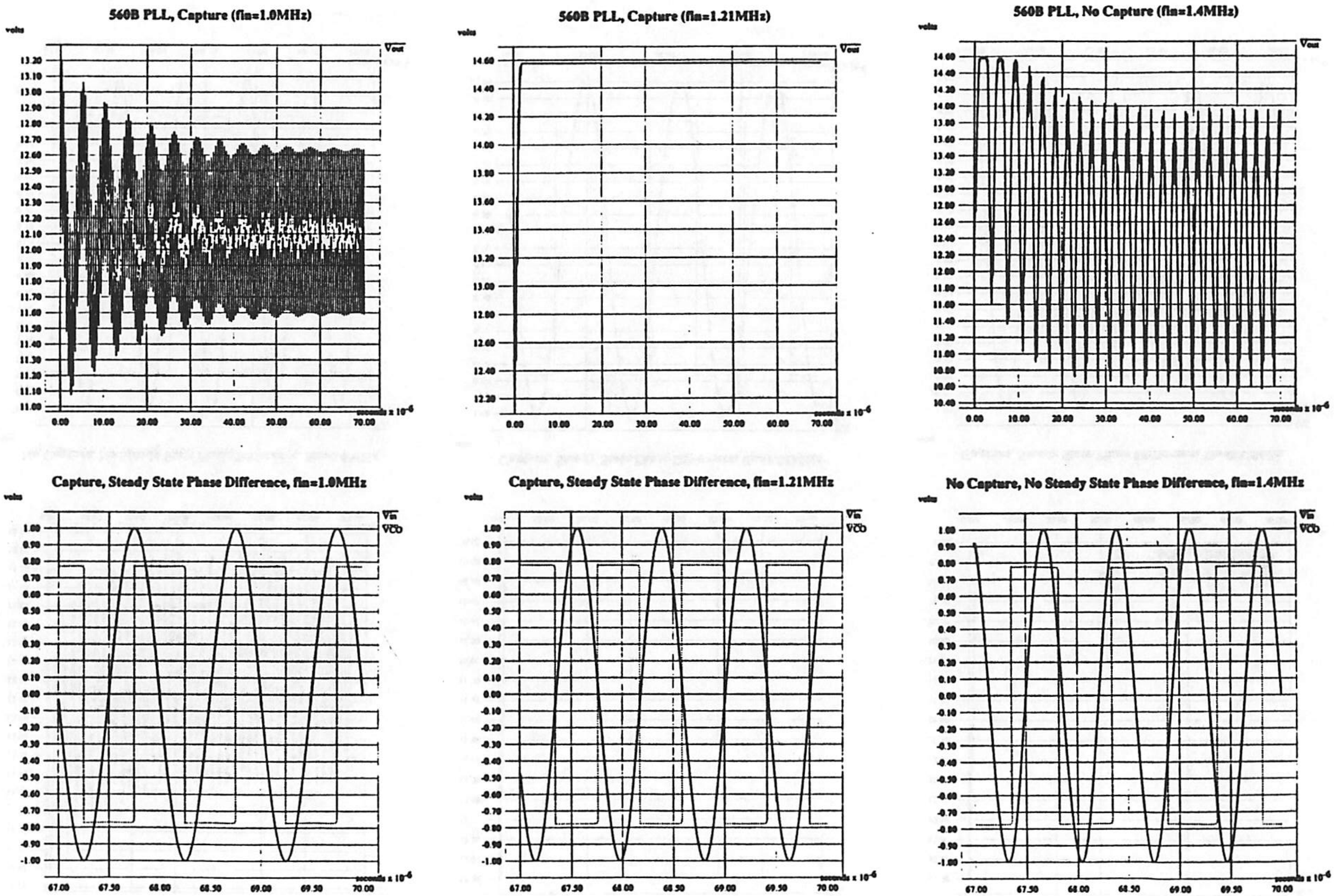
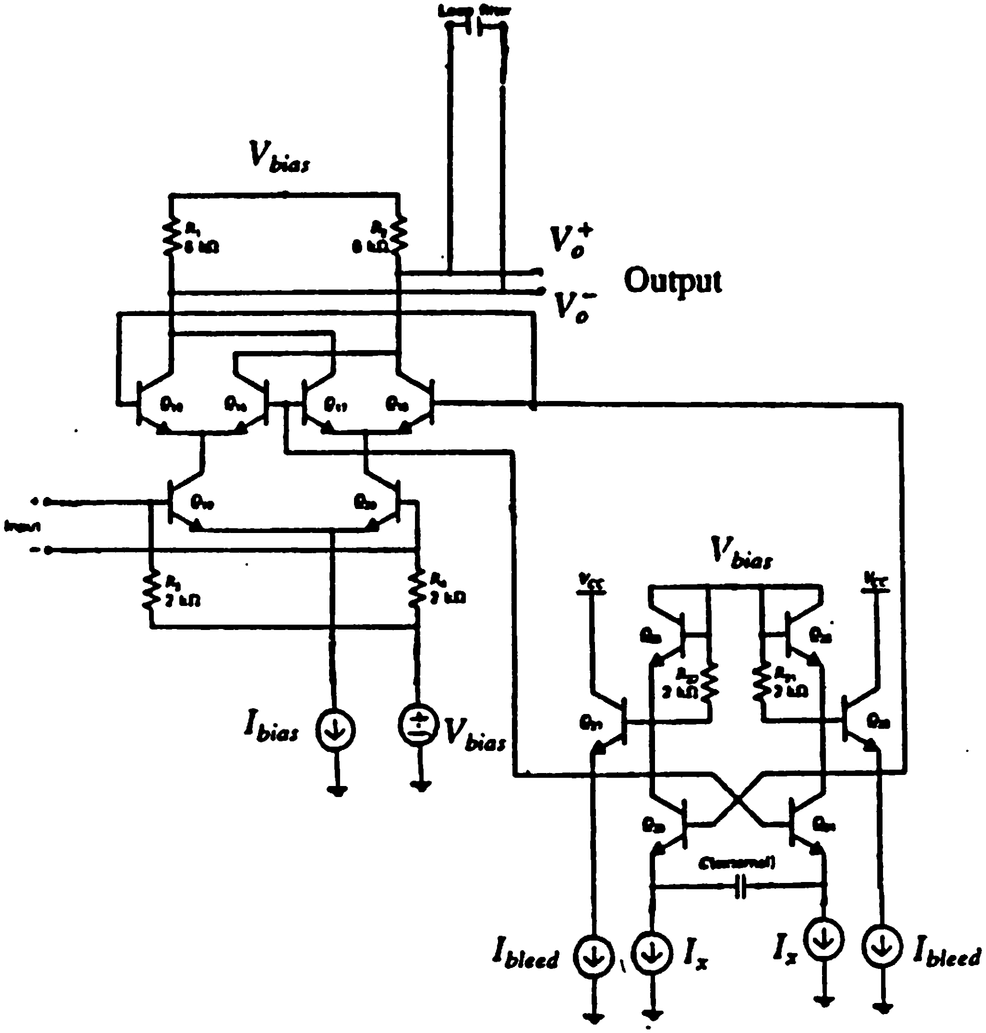


Figure 2.4: 560B Capture Behavior (1MHz,1.21MHz,1.4MHz)



$$I_x = I_{bias}(1 + k\,(V_o^- - V_o^+))$$

Figure 2.5: The Simplified 560B PLL Circuit

```
* 560B PLL Operating Point
*
v(1) = 1.600000e+01          v(46) = 1.304840e+01
v(3) = 7.444340e+00          v(100) = 3.606923e+00
v(4) = 6.811194e+00          v(101) = 3.606923e+00
v(5) = 6.331466e-01          v(110) = 7.471097e-01
v(6) = 1.283205e+01          v(111) = 7.471097e-01
v(7) = 6.832362e+00          v(112) = 7.471097e-01
v(8) = 6.437306e-01          v(113) = 7.471097e-01
v(9) = 4.368471e+00          v(114) = 7.471097e-01
v(10) = 9.461923e-01         v(145) = 2.509491e+00
v(11) = 1.798906e-01         v(200) = 2.858678e+00
v(12) = 1.798906e-01         v(201) = 3.610587e+00
v(16) = 7.544665e-01         v(212) = 1.798906e-01
v(201) = 9.906494e+00        v(245) = 2.509491e+00
v(21) = 1.285633e+01         v(300) = 5.113617e+00
v(211) = 9.906494e+00        v(301) = 5.113617e+00
v(22) = 1.065662e+01         v(302) = 3.235330e+00
v(221) = 3.730447e+00        v(303) = 3.235330e+00
v(23) = 1.065662e+01         v(304) = 1.508933e+00
v(231) = 3.730447e+00        v(305) = 1.362239e+01
v(26) = 4.383429e+00         v(500) = 1.220960e+01
v(27) = 4.383429e+00         v30#branch = 6.225914e-04
v(30) = 1.283704e+01         v31#branch = 6.225914e-04
v(31) = 6.625781e+00         v32#branch = 6.225914e-04
v(32) = 5.875547e+00         v33#branch = 6.225914e-04
v(33) = 5.875547e+00         v34#branch = 6.225914e-04
v(36) = 4.346177e+00         v145#branch = 1.548735e-04
v(36a) = -5.65382e+00        v212#branch = 7.402906e-04
v(37) = 4.346177e+00         v245#branch = 1.548735e-04
v(38) = 7.471097e-01         vcc#branch = -1.96639e-02
v(39) = 7.471097e-01         v12#branch = 5.170282e-20
v(40) = 7.471097e-01         vtrig#branch = 1.761829e-18
v(41) = 7.471097e-01
v(42) = 7.471097e-01
v(43) = 2.408824e+00
v(44) = 2.509491e+00
v(45) = 2.509491e+00
```

Figure 2.6: The DC Operating Point of the 560B PLL

```
* SIMPLIFIED 560B PLL
*
* INPUT: V(100)-V(101); VCO: V(300)-v(301); Vcontrol: V(22)-v(23)

* INPUT *
*********
*-----------------------------------------------------------------------
* Variable Frequency Input  For LOCK
*************************************
*
** Input *
*VIN 99 0 PWL 0 3 30u 3 60u 32 200u 32
*.ic v(99)=0
*Xinput 99 101 100 VarFreqIn
*******
*.subckt VarFreqIn 3 84 85
**Creates an input with a variable frequency, depending on the VIN
**control voltage; the VCO with VIN as the control voltage
*
** Ko=.1meg
*rb 4 10 1K
**gvar2 4 10 poly(2) 4 10 3 0    0 0 0 0 1e-4
*bvar2 4 10 i=1e-4 * v(3) * (v(4)-v(10))
*vtrig 10 0 pulse(0 .5 0 0 0 1ps 1s)
*c2 4 0 159.15pF
*ra 4 6 1K
**gvar1 4 6 poly(2) 4 6 3 0      0 0 0 0 1e-4
*bvar1 4 6 i=1e-4 * v(3) * (v(4)-v(6))
*c1 6 7 159.15pF
**egain 5 0 4 0 3.05
*bgain 5 0 v=3.05*v(4)
*ro 5 7 10
*d1 7 8 md
*v1 8 0 0.191v
*d2 0 9 md
*v2 9 7 0.191v
*bvco1 84 0 v=0.5*v(7)
*rvco1 84 0 1k
*bvco2 85 0 v=-0.5*v(7)
*rvco2 85 0 1k
*.model md d is=1e-16
*.ends VarFreqIn
*******

* Fixed Input Frequency FOR CAPTURE
*************************************
VIN 101 100 0 SIN (0 1V 1.000MEG)
*-----------------------------------------------------------------------
*
* MODELS *
**********
.MODEL ZENER D IS=1.0E-14 BV=6.2V
.MODEL NPN NPN IS=1E-16 BF=200
.MODEL DIODE D IS=10E-15
VCC 1 0 16V
*
*VCO EC MULTIVIBRATOR
**********************************************
*Ko is .93 the center freq. like in the G&M book
*The center freq. is 1MHz
**********************************************
V31 31 0 6.625781V
Q25 31 31 32 NPN
Q26 31 31 33 NPN
R22 31 32 2K
R21 31 33 2K
Q23 32 300 36 NPN
Cfix3 300 0 10f
Q24 33 301 37 NPN
Cfix4 301 0 10f
Q27 1 32 301 NPN
I301 301 0 622.5914uA
Q28 1 33 300 NPN
I300 300 0 622.5914uA
*
* Control Voltage Dictating VCO Frequency
**************************************************
*BI36 36 0 I=778.23925e-6*(1+0.93*v(73))
*BI37 37 0 I=778.23925e-6*(1+0.93*v(73))
BI36 36 0 I=778.23925e-6*(1+0.93*(v(22)-v(23)))
BI37 37 0 I=778.23925e-6*(1+0.93*(v(22)-v(23)))
**************************************************
**I36 36 0 622.5914uA
**I37 37 0 622.5914uA
VTRIG 36 36A PULSE (10 0 1ns 1ns 1ns 1nS 1s)
CEXT 36A 37 3.12E-10 IC=1.6V
*
* PC *
**************************************************
*Kc is around 2.55V/rad (maybe a little larger)
**************************************************
*
V21 21 0 12.85633
R1 21 22 6K
R2 21 23 6K
Q15 22 300 26 NPN
Q16 23 301 26 NPN
Q17 22 301 27 NPN
Q18 23 300 27 NPN
Q19 26 100 200 NPN
Q20 27 101 200 NPN
I200 200 0 740.2906uA
R3 100 201 2K
R4 101 201 2K
V201 201 0 3.610587V
*
*FILTER *
*********
*
*RF1=RF2=6K, internal
CF 22 23 663.146pF
*CF1 22 0 1.326291nF
*CF2 23 0 1.326291nF
*Bfilt 72 0 v=v(22)-v(23)
*Rfilt 72 73 1K
*Cfilt 73 0 7.95775n

*ANALYSIS
*
.TRAN .07U 100U 0U 0.01U
.OPTION ACCT
.OPTIONS NOPAGE ITL5=0 LIMPTS=1000000
.OPTIONS VNTOL=0.001 ABSTOL=1.0E-6 reltol=1E-6 METHOD=GEAR
.WIDTH OUT = 80
.END
```

Figure 2.7: A SPICE Input File for the Simplified 560B PLL

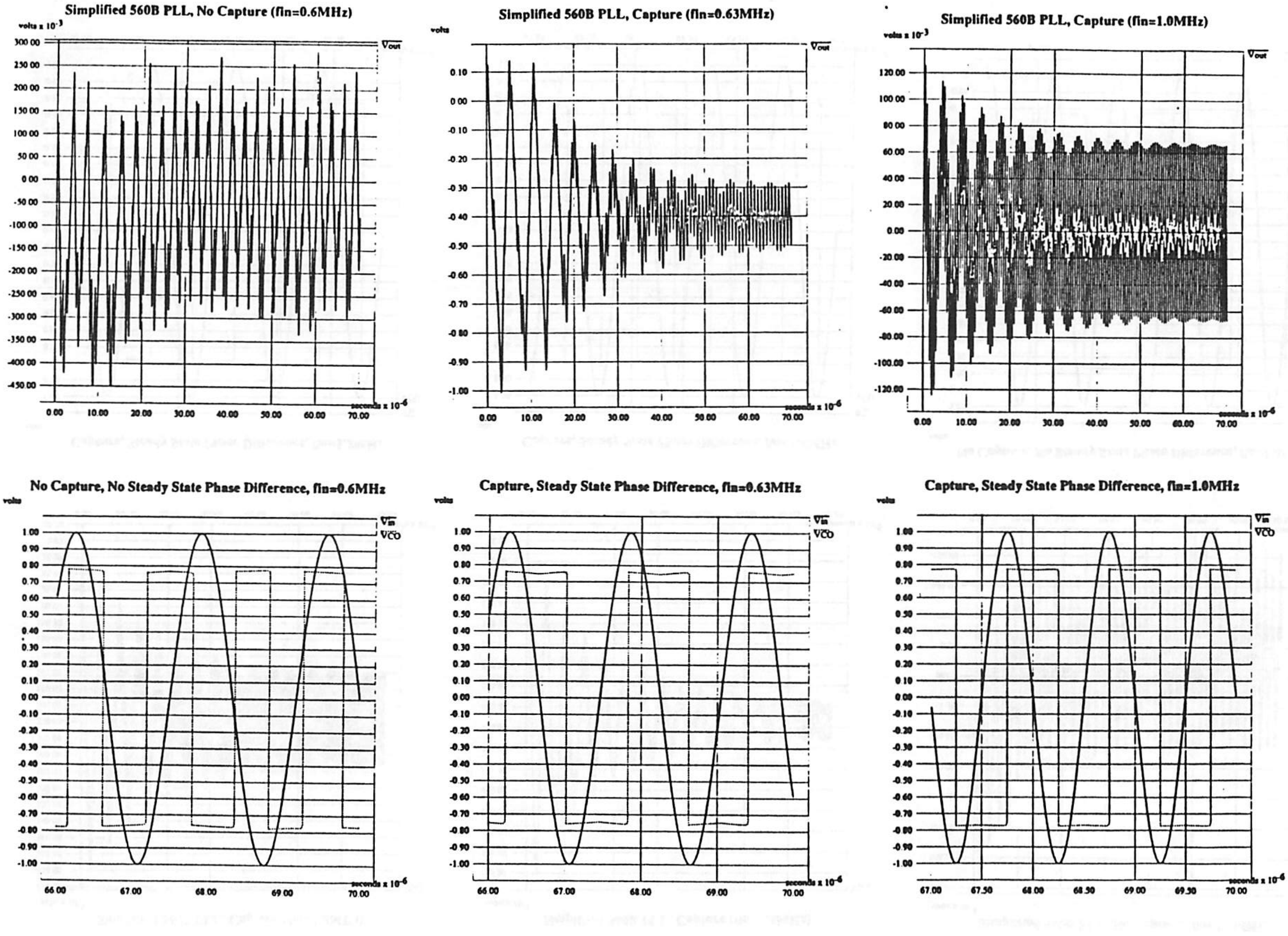
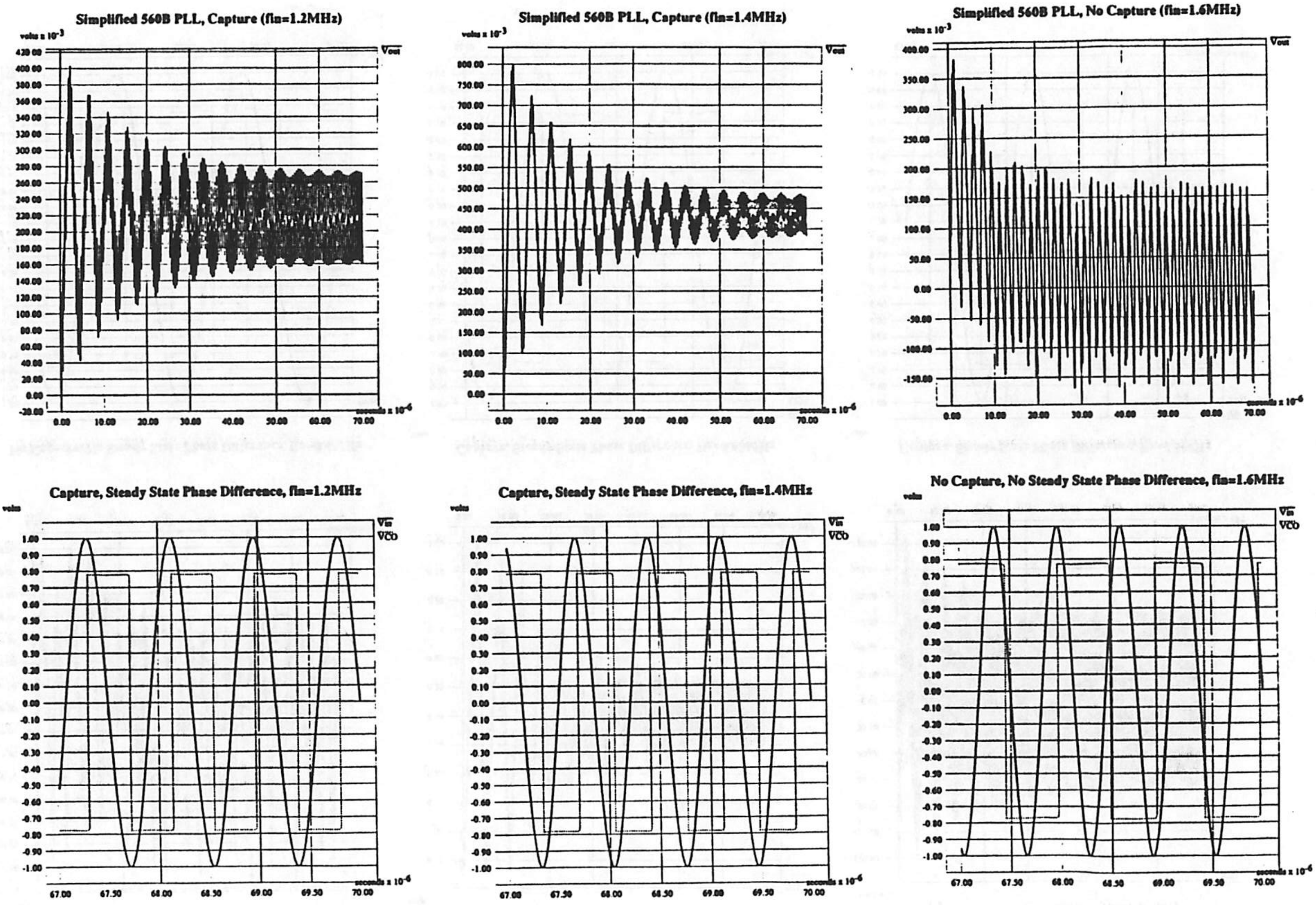


Figure 2.8: Simplified 560B Capture Behavior (0.6MHz,0.63MHz,1.0MHz)



Simplified 560B PLL, Capture (fin=1.2MHz)

Simplified 560B PLL, Capture (fin=1.4MHz)

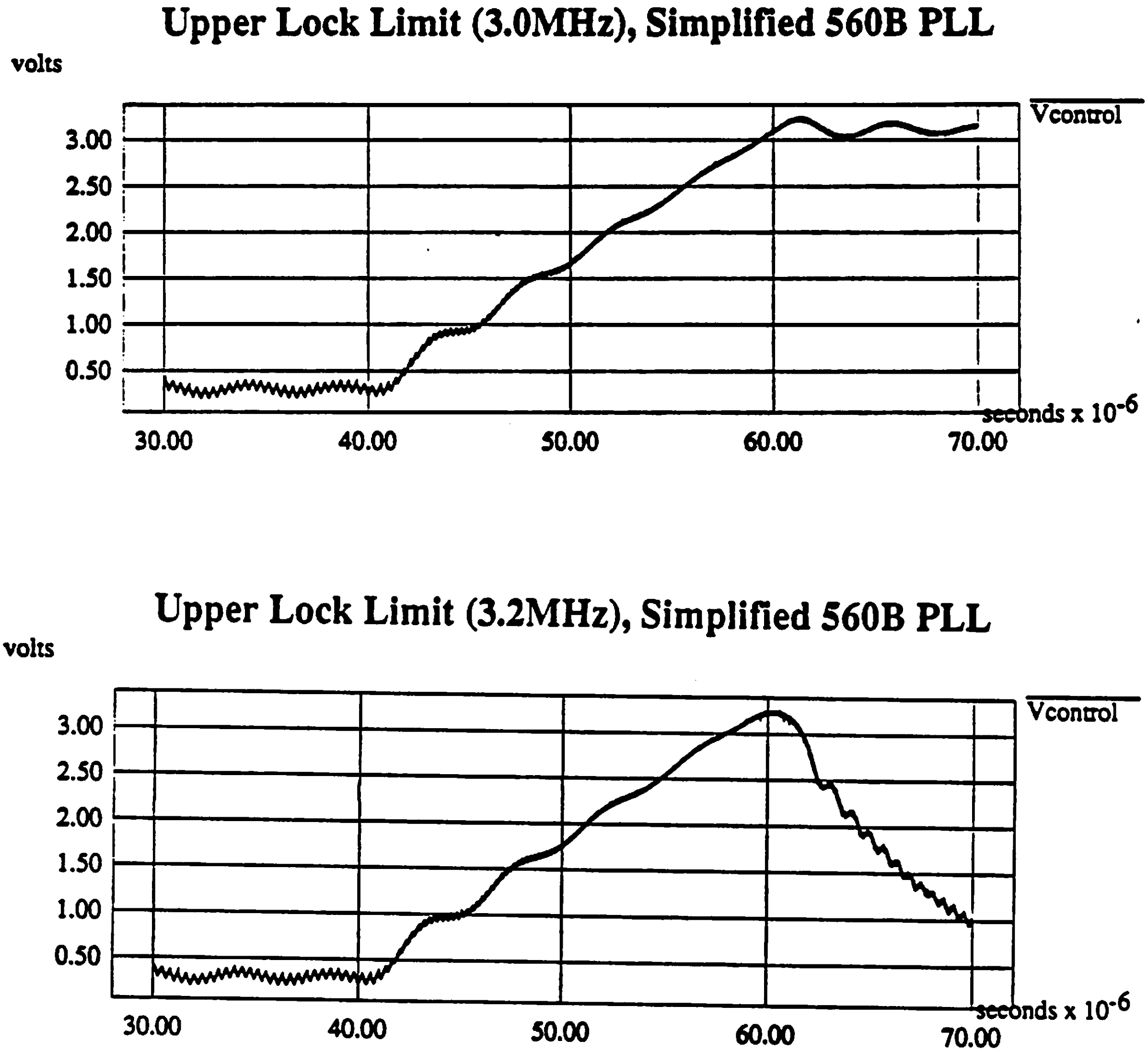
Simplified 560B PLL, No Capture (fin=1.6MHz)

Capture, Steady State Phase Difference, fin=1.2MHz

Capture, Steady State Phase Difference, fin=1.4MHz

No Capture, No Steady State Phase Difference, fin=1.6MHz

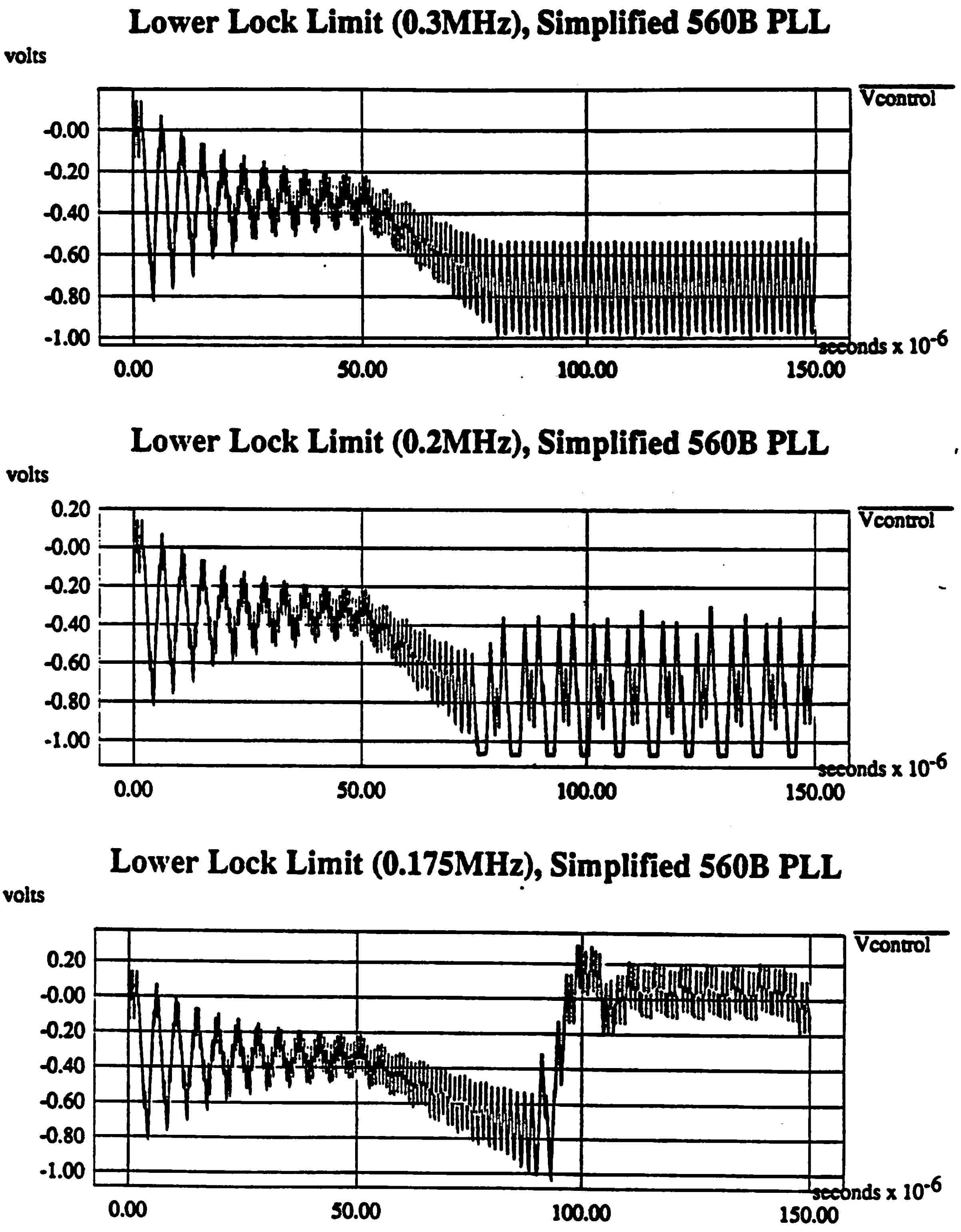
Figure 2.9: Simplified 560B Capture Behavior (1.2MHz,1.4MHz,1.6MHz)



Upper Lock Limit (3.0MHz), Simplified 560B PLL

Upper Lock Limit (3.2MHz), Simplified 560B PLL

Figure 2.10: Simplified 560B Lock Behavior (3.0MHz,3.2MHz)



Lower Lock Limit (0.3MHz), Simplified 560B PLL

Lower Lock Limit (0.2MHz), Simplified 560B PLL

Lower Lock Limit (0.175MHz), Simplified 560B PLL

Figure 2.11: Simplified 560B Lock Behavior (0.3MHz,0.2MHz,0.175MHz)

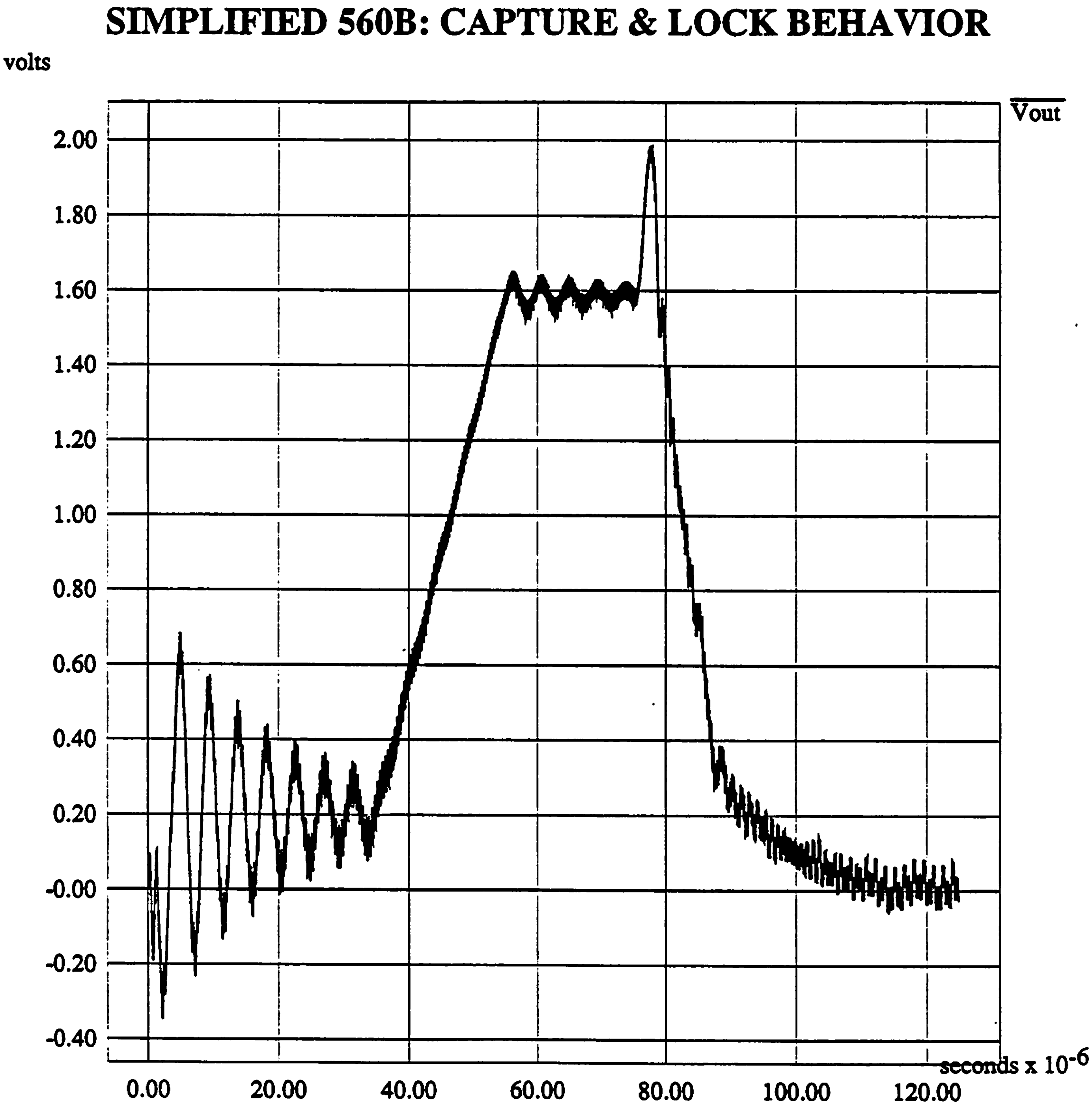


Figure 2.12: Simplified 560B Capture and Lock Behavior (1.2MHz,2.5MHz,4.5MHz)

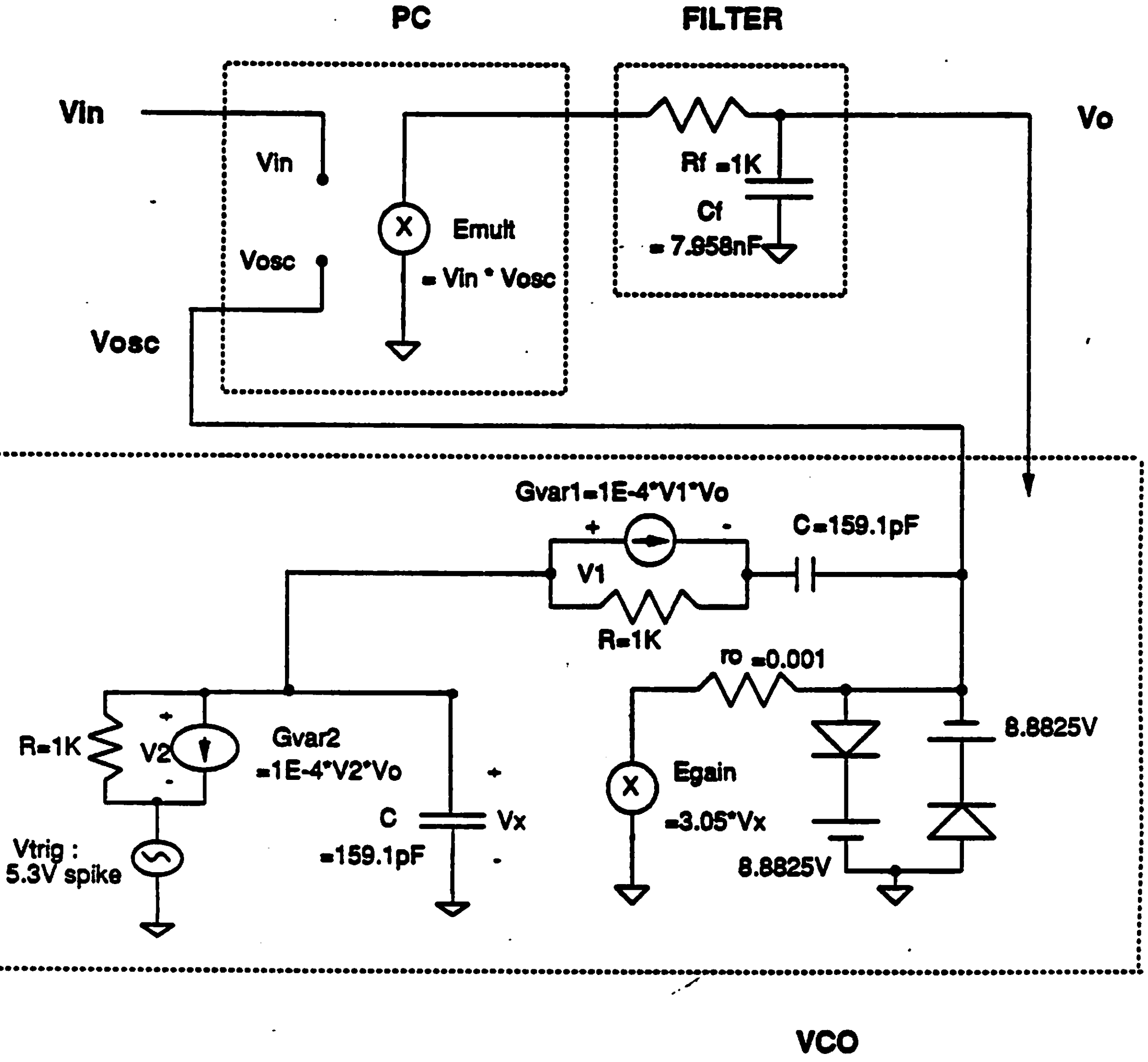


Figure 3.1: The PLL(0) Macromodel

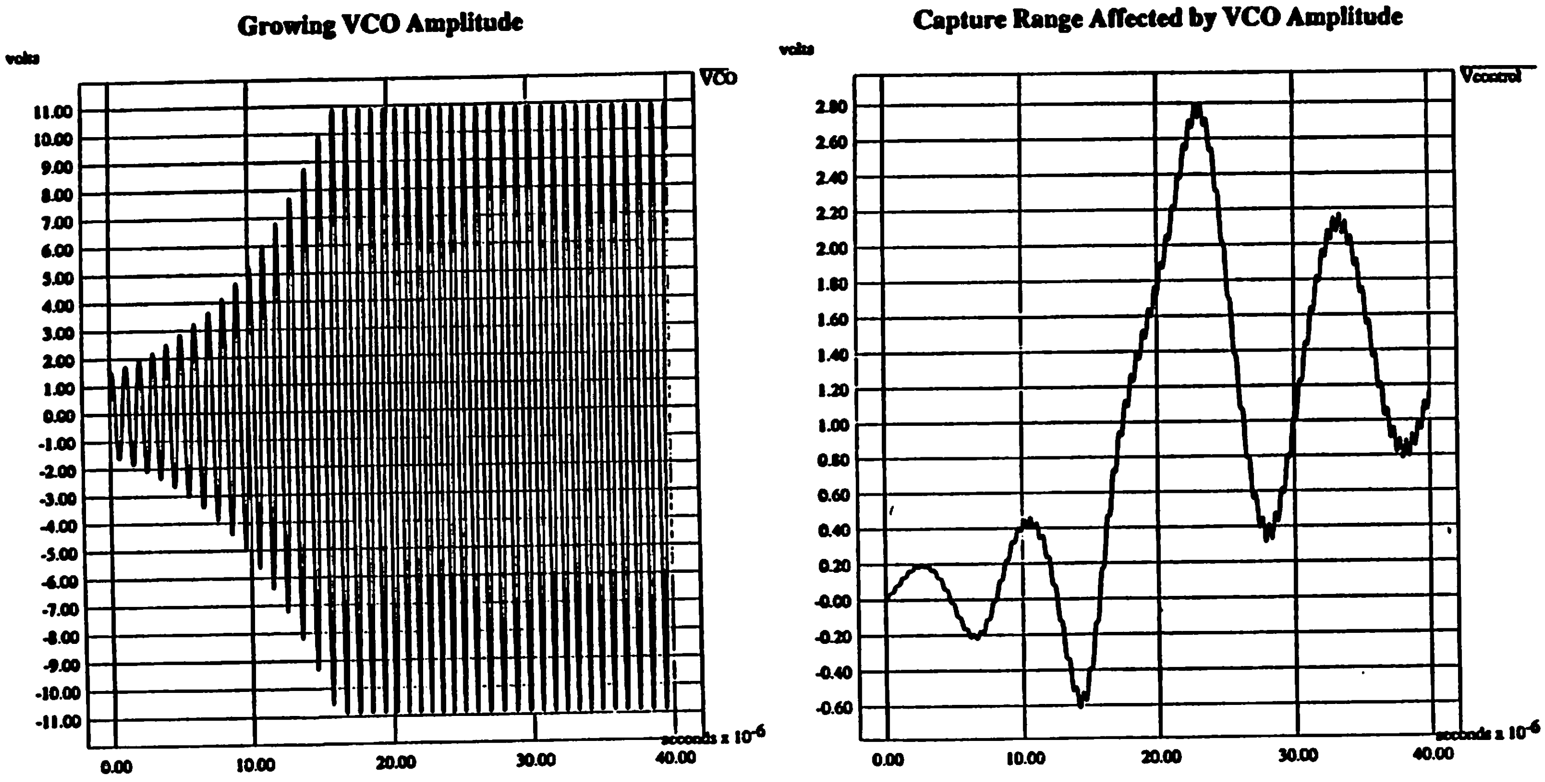
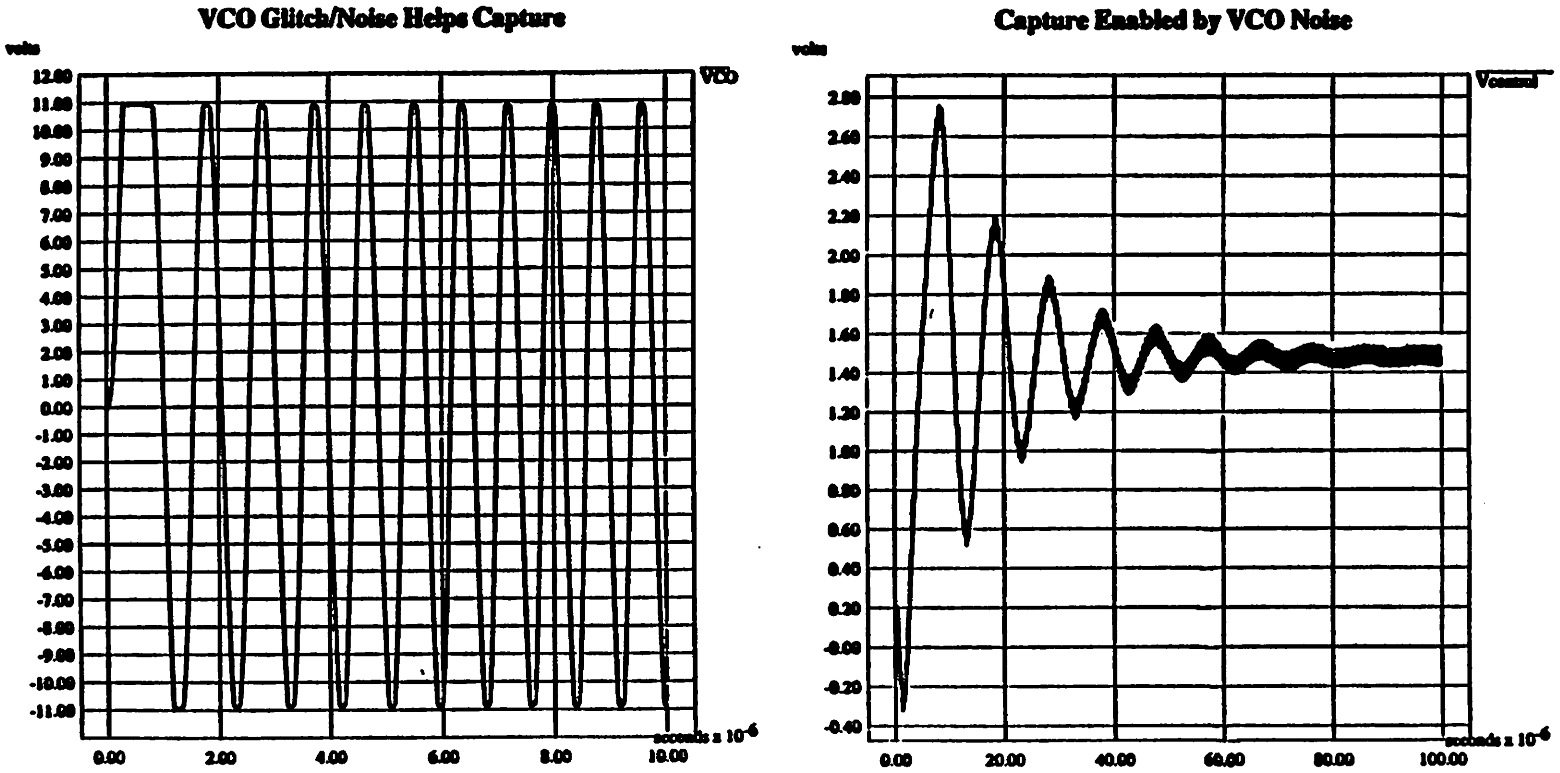


Figure 3.2: Capture Range Affected by VCO Amplitude (.tran/Vtrig settings)


VCO Glitch/Noise Helps Capture

Capture Enabled by VCO Noise

Figure 3.3: Capture Range Affected by VCO Glitch (.tran/Vtrig settings)

```
PLL(0) [Pederson/Mayaram] Macromodel
*
* Vin: V(1); VCO: V(7); Vcontrol: V(3)

************************************
* Input: Fixed Input Frequency *
*vin 1 0 sffm(0 1 1meg 10 10k)
vin 1 0 sin(0 1 0.841meg)
*vin 1 0 sin(0 1 0.842meg)
*vin 1 0 sin(0 1 1.148meg)
*vin 1 0 sin(0 1 1.147meg)
rin 1 0 1k

************************************
** Input: Variable Input Frequency *
*.ic v(99)=0
*Xinput 99 1 VarFreqIn

** Input
*VIN 99 0 PWL 0 0 70u 0 80u -2 130u -2 140u -4 190u -4 230u -5.3 250u -5.3 290u -5.4 315
u -5.4 385u -5.5 450u -5.5
**VIN 99 0 PWL 0 0 70u 0 80u +2 130u +2 140u +4 190u +4 230u +5.3 250u +5.3 290u + 5.4 3
15u +5.4 385u +5.5 450u +5.5

*************
*.subckt VarFreqIn 3 7
**Creates an input with a variable frequency, depending on the VIN
**control voltage; the VCO with VIN as the control voltage
**
*rb 4 10 1K
**gvar2 4 10 poly(2) 4 10 3 0     0 0 0 0 1e-4
*bvar2 4 10 i=1e-4 * v(3) * (v(4)-v(10))
*vtrig 10 0 pulse(0 5.3 0 0 0 1ns)
*c2 4 0 159.1pF
**c2 4 0 159.15pF
*ra 4 6 1K
**gvar1 4 6 poly(2) 4 6 3 0       0 0 0 0 1e-4
*bvar1 4 6 i=1e-4 * v(3) * (v(4)-v(6))
*c1 6 7 159.1pF
**c1 6 7 159.15pF
**egain 5 0 4 0 3.05
*bgain 5 0 v=3.05*v(4)
**ro 5 7 .001
*ro 5 7 10
*d1 7 8 md
*v1 8 0 0.191v
**v1 8 0 1v
*d2 0 9 md
*v2 9 7 0.191v
**v2 9 7 1v
*.model md d is=1e-16
*.ends VarFreqIn
*************
```

```
* Phase comparator multiplier *
* Kc = Km Vi Vosc / 2 = 5 V/rad
* emult 2 0 poly(2) 1 0 7 0   0 0 0 0 1
bmult 2 0 v=1.00*v(1)*v(7)

* Low-pass Filter *
* BW=20kHz
rf 2 3 1k
cf 3 0 7.958nF

* Wien feedback circuit *
* Ko=.1meg
rb 4 10 1K
*gvar2 4 10 poly(2) 4 10 3 0    0 0 0 0 1e-4
bvar2 4 10 i=1e-4 * v(3) * (v(4)-v(10))
vtrig 10 0 pulse(0 5.3 0 0 0 1ns)
*c2 4 0 159.1549pF
c2 4 0 159.1pF
ra 4 6 1K
*gvar1 4 6 poly(2) 4 6 3 0      0 0 0 0 1e-4
bvar1 4 6 i=1e-4 * v(3) * (v(4)-v(6))
*c1 6 7 159.1549pF
c1 6 7 159.1pF

* positive-gain block
*egain 5 0 4 0 3.05
bgain 5 0 v=3.05*v(4)
*ro 5 7 10
ro 5 7 .001
d1 7 8 md
*v1 8 0 10v
*v1 8 0 9.125v
v1 8 0 8.8825v
d2 0 9 md
*v2 9 7 10v
*v2 9 7 9.125v
v2 9 7 8.8825v
.model md d is=1e-16

* control cards *
*.tran 0.07u 300u 0 0.01u
.tran 0.07u 100u 0 0.01u
.option acct
.options nopage itl5=0 limpts=1000000
.options vntol=0.001 abstol=1.0E-6 reltol=1E-6 method=GEAR
.width out=80
.end
```

Figure 3.4: The SPICE3 Input File for the PLL(0) Macromodel

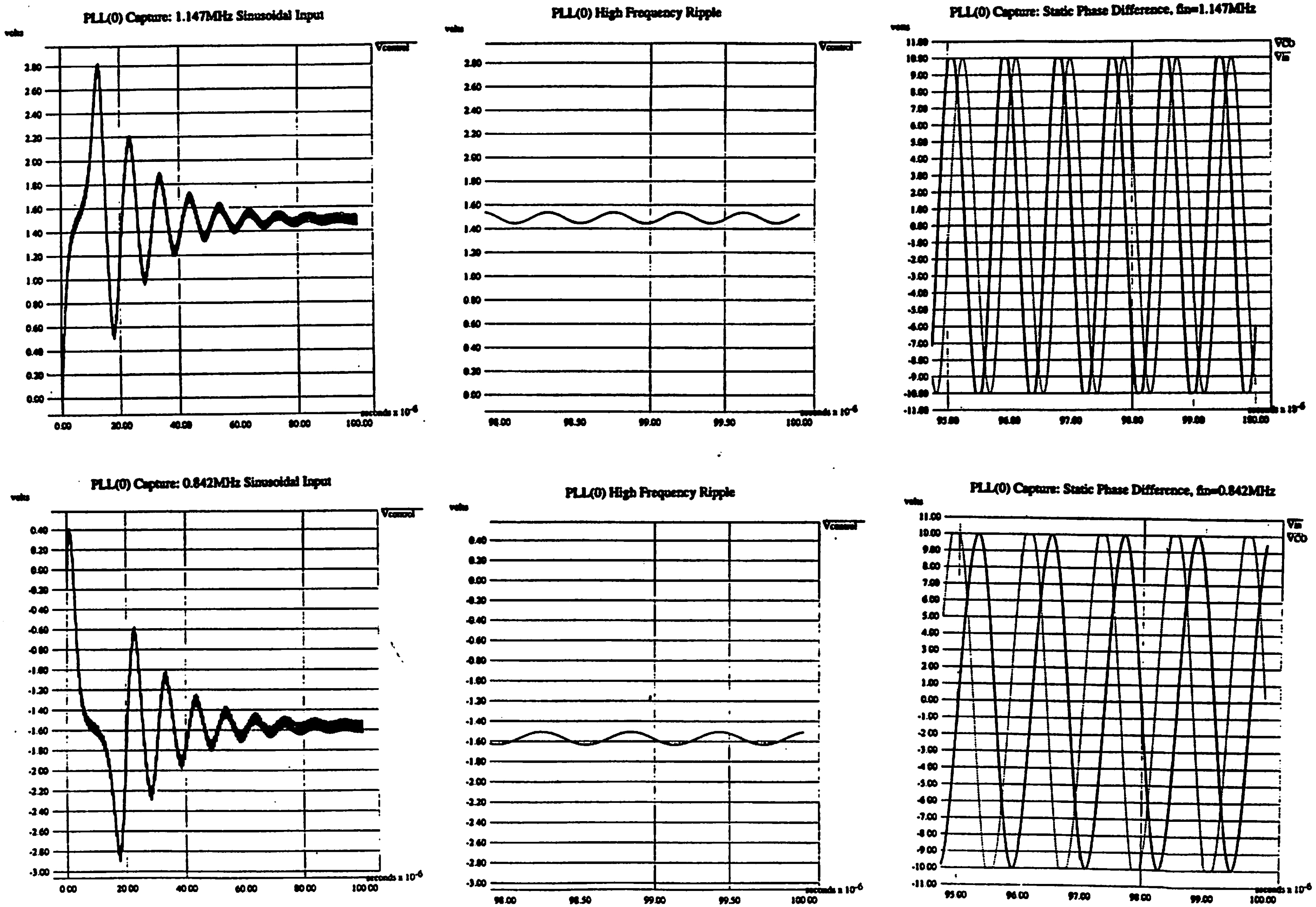


Figure 3.5: PLL(0) Macromodel Capture Behavior (1.147MHz & 0.842MHz)

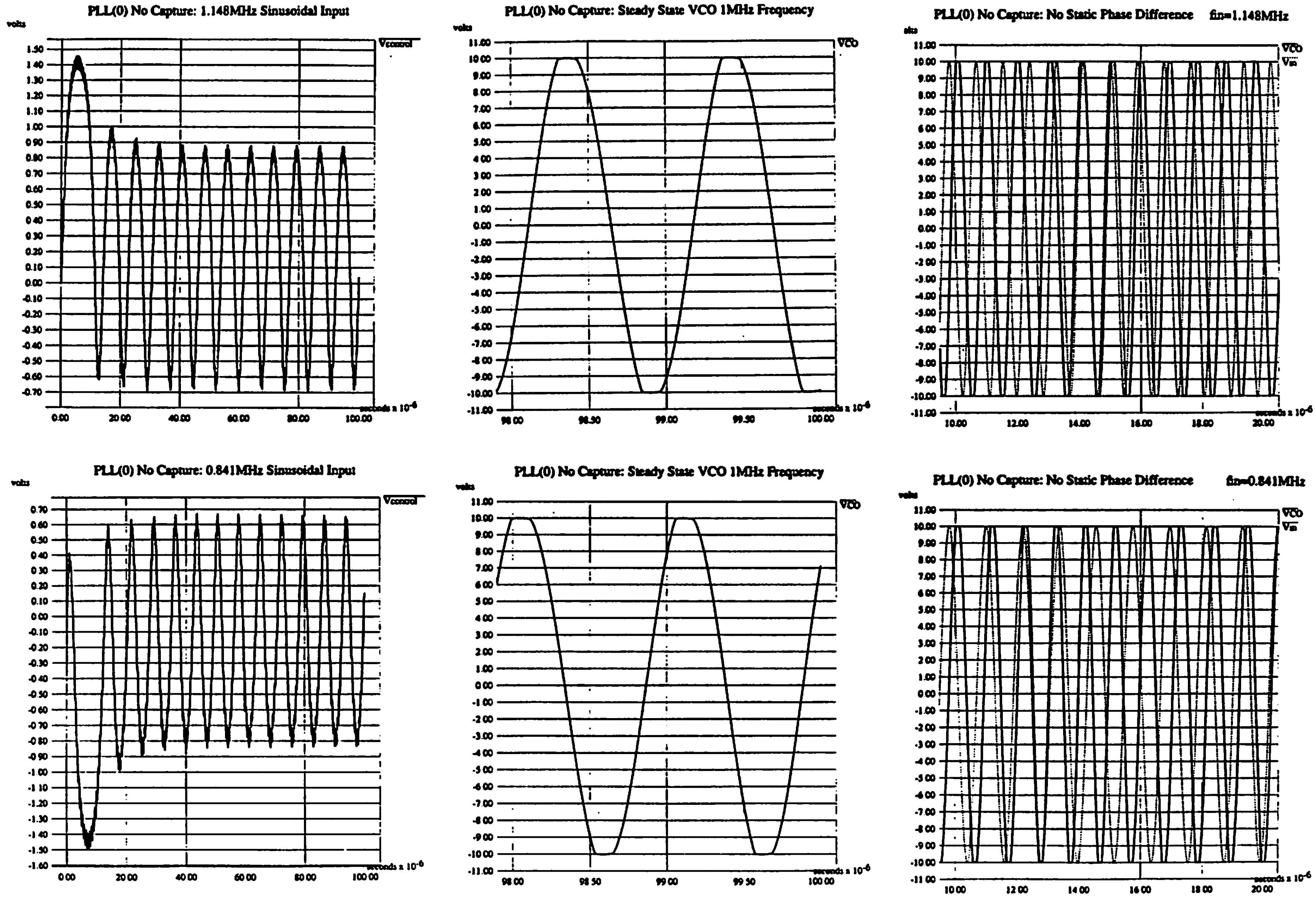


Figure 3.6: PLL(0) Macromodel No-Capture Behavior (1.148MHz & 0.841MHz)

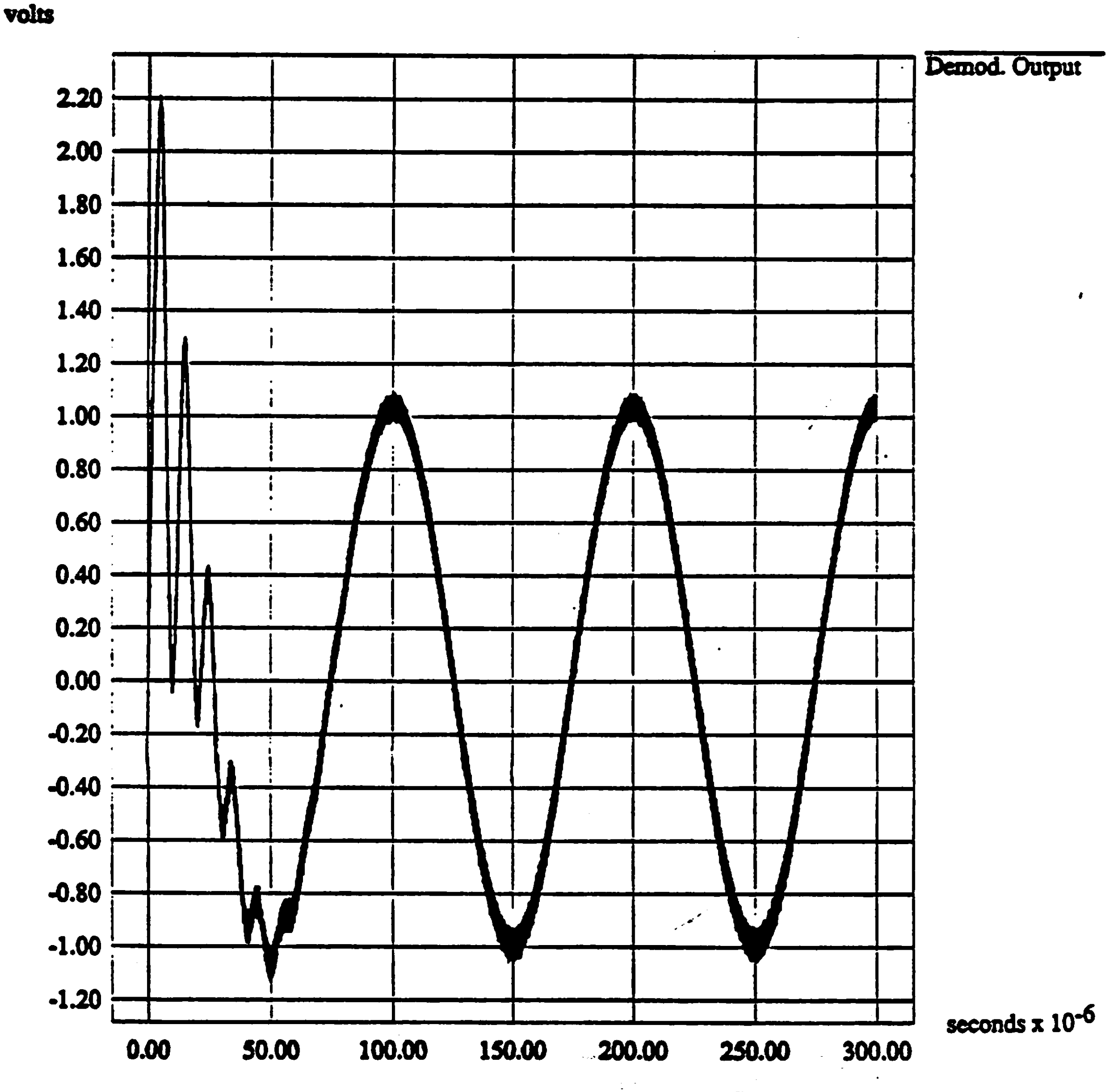


Figure 3.7: PLL(0) Macromodel Behavior for a Modulated Input (1MHz Carrier, 10KHz Signal)

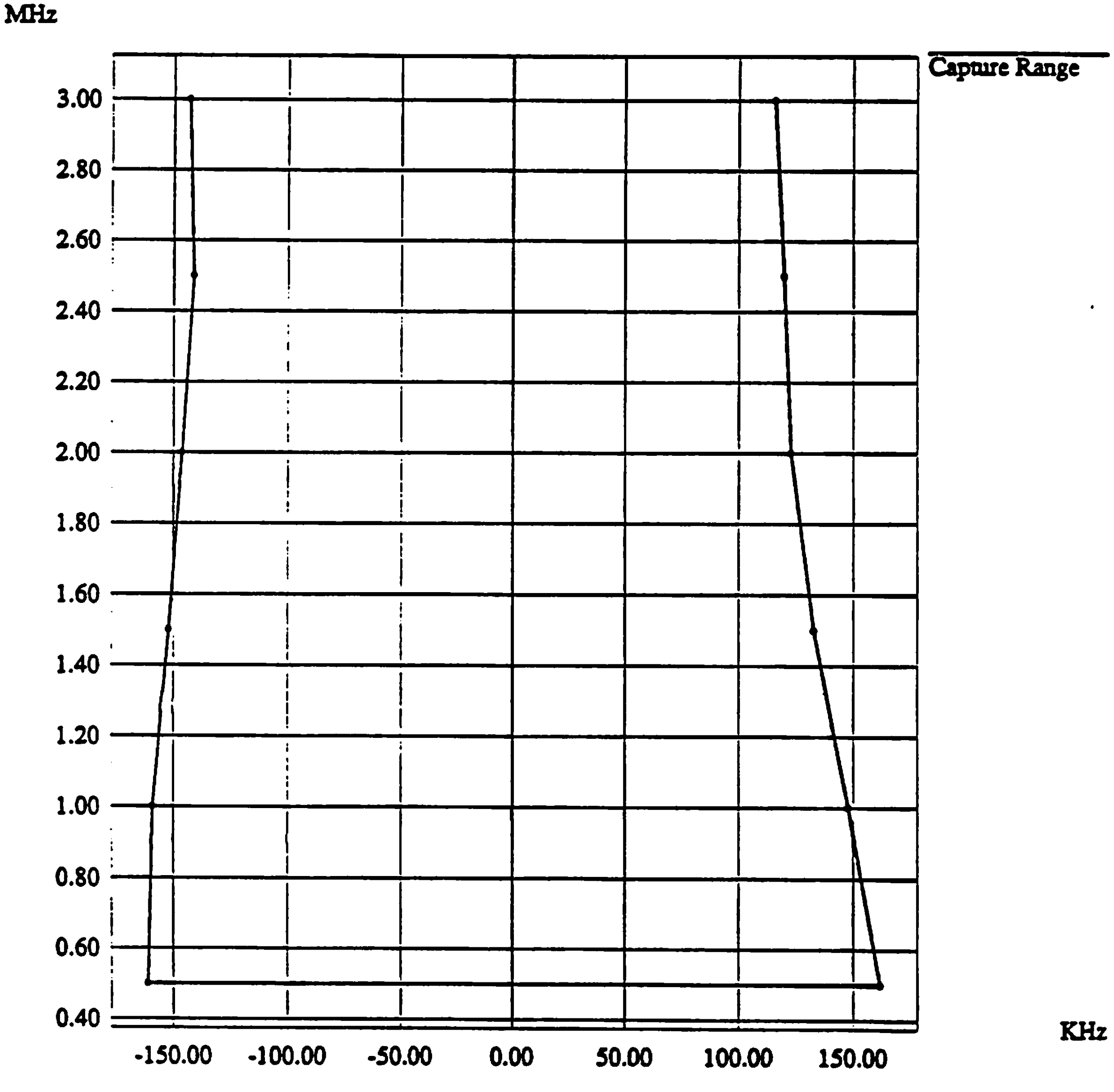


Figure 3.8: PLL(0) Macromodel, Center Frequency vs Capture Range

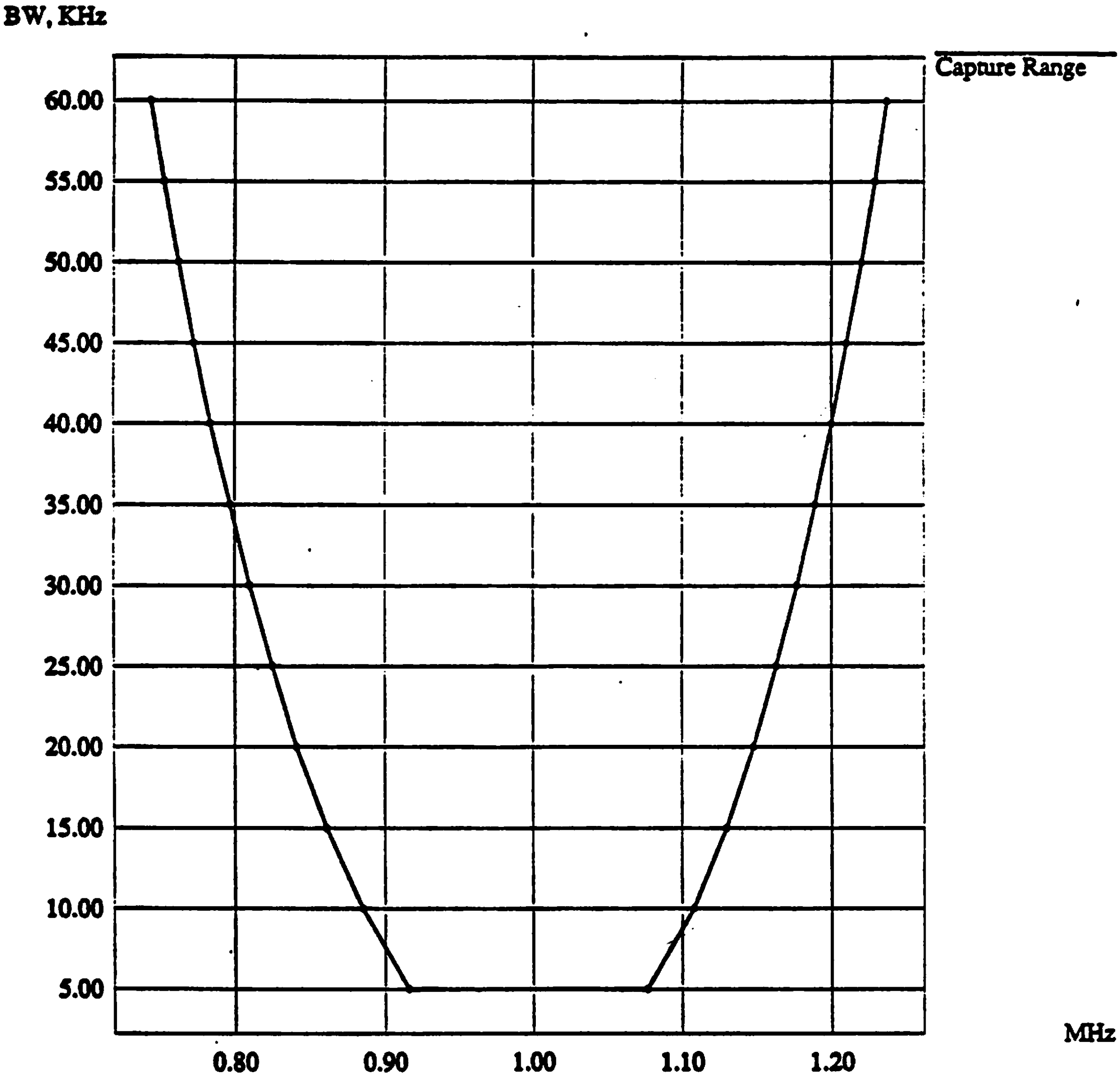


Figure 3.9: PLL(0) Macromodel, Filter Bandwidth (BW) vs Capture Range

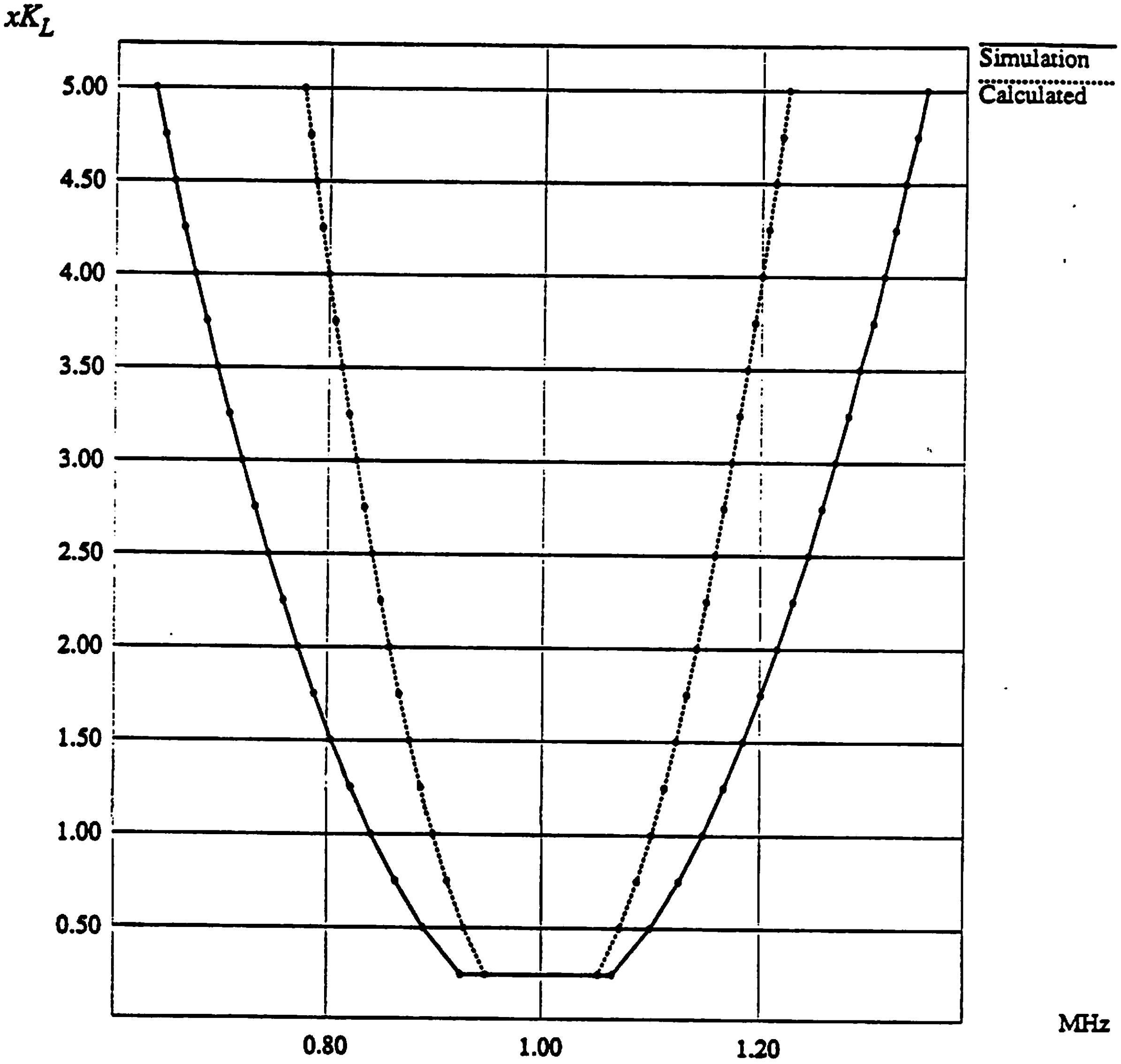


Figure 3.10: PLL(0) Macromodel, Loop Gain Constant (K_L) vs Capture Range

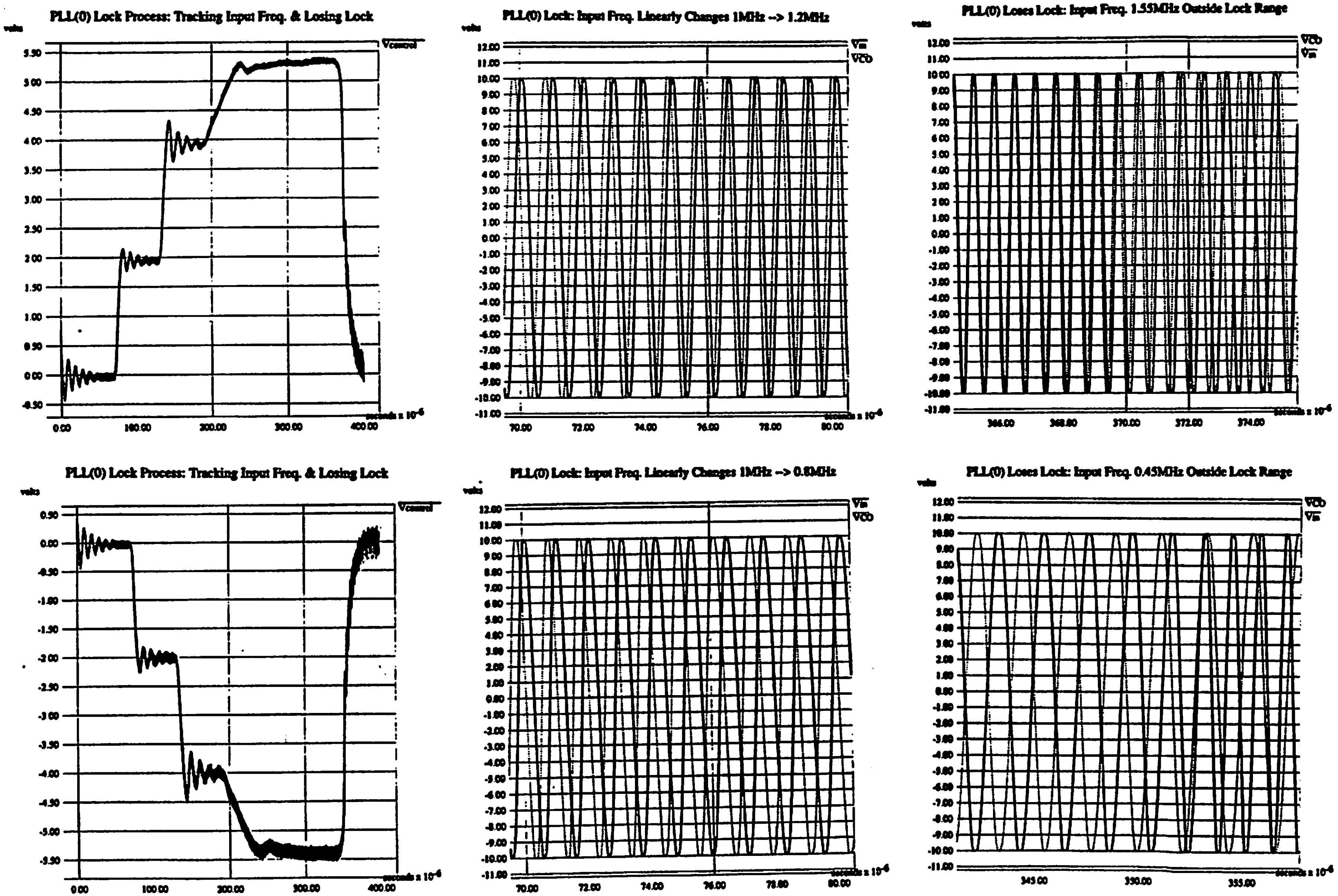


Figure 3.11: PLL(0) Macromodel Lock Behavior (±0.55MHz Lock Range)

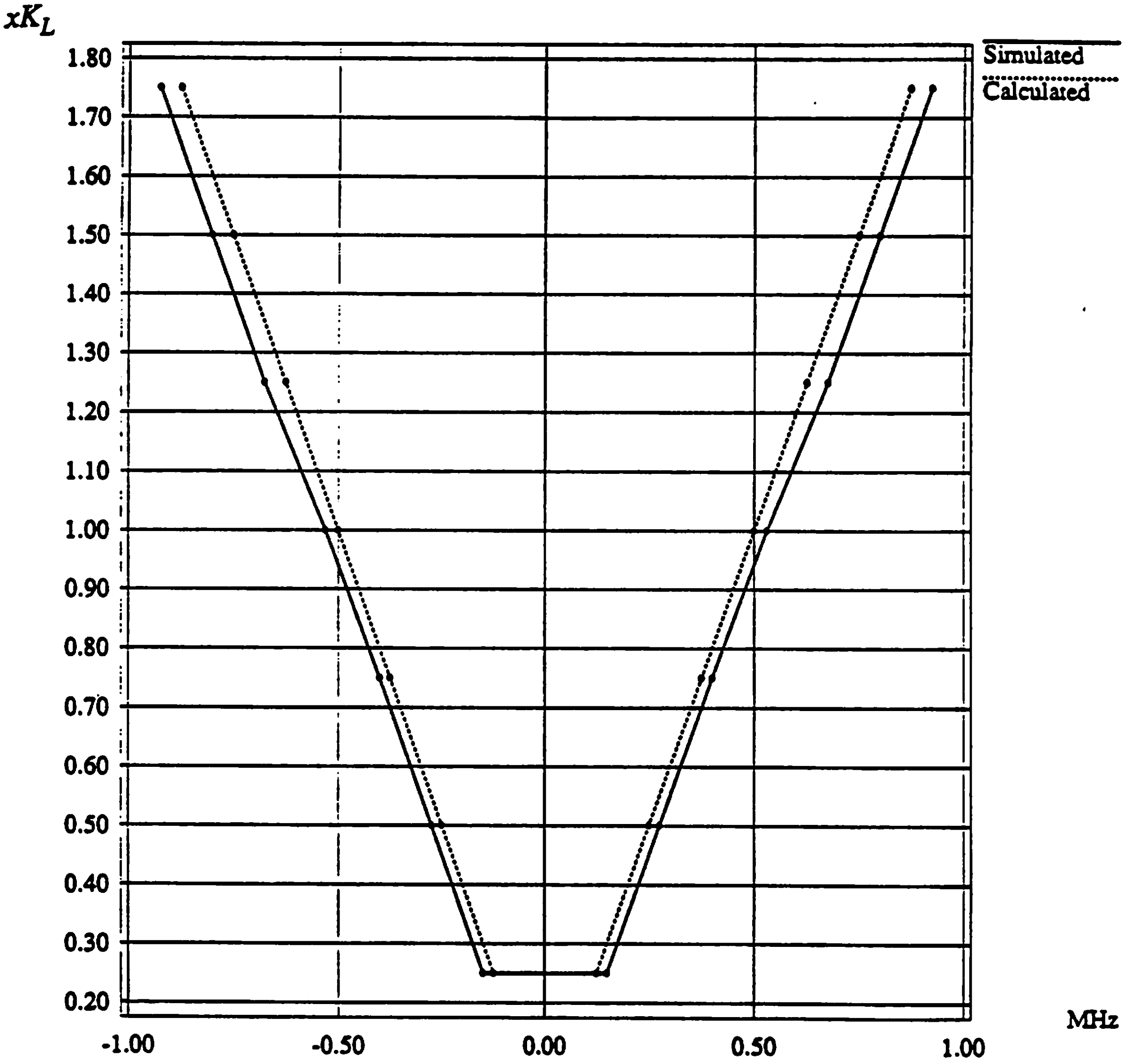


Figure 3.12: PLL(0) Macromodel, Loop Gain Constant (K_L) vs Lock Range

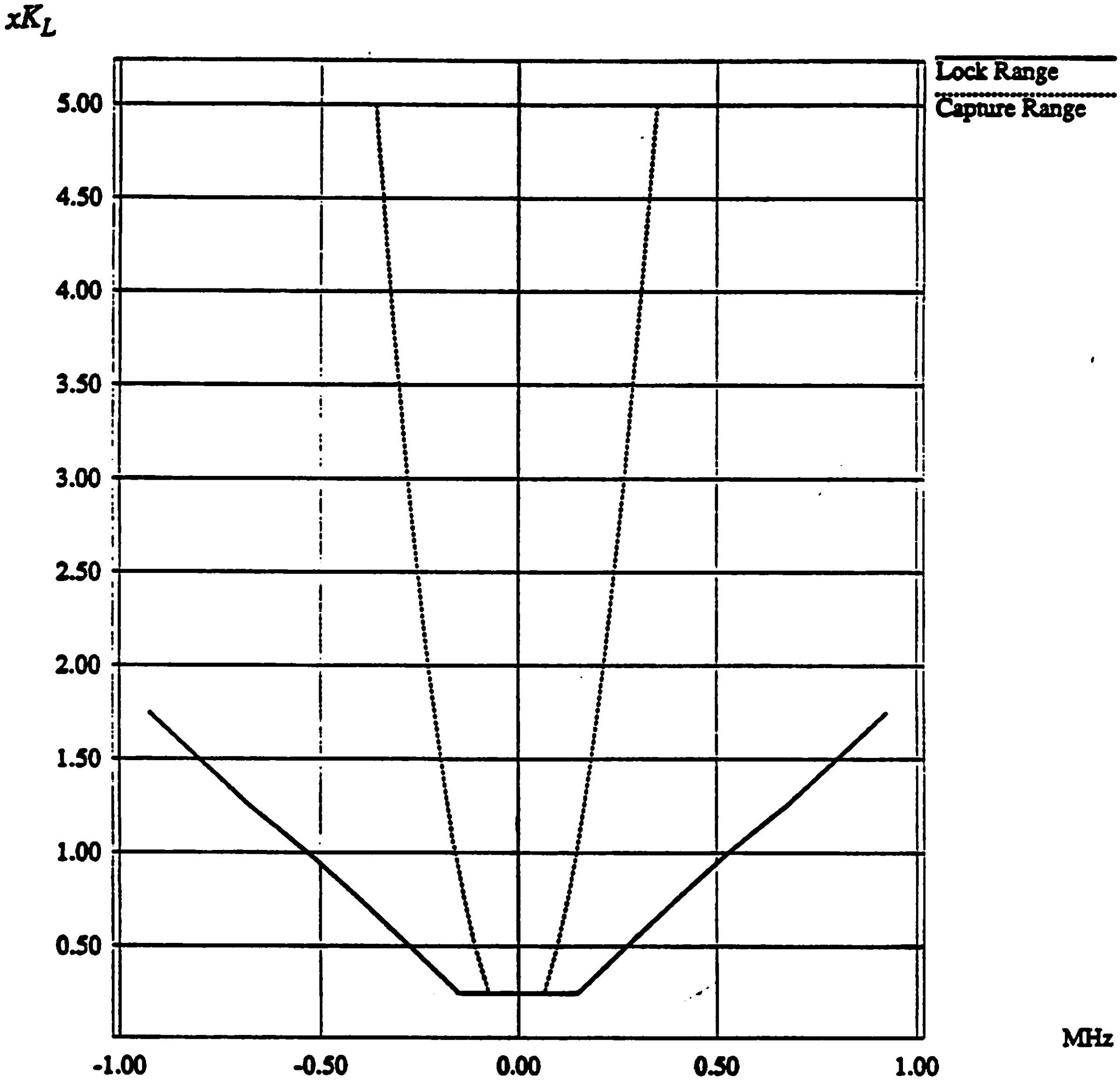


Figure 3.13: PLL(0) Macromodel, Simulated Lock and Capture Ranges for Various K_L

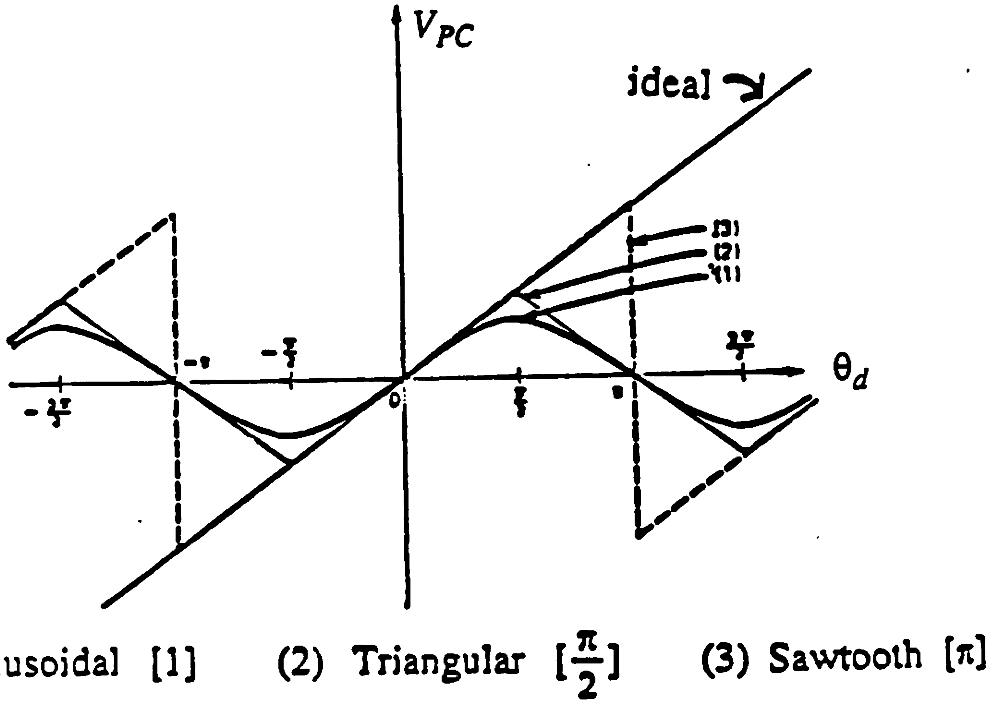
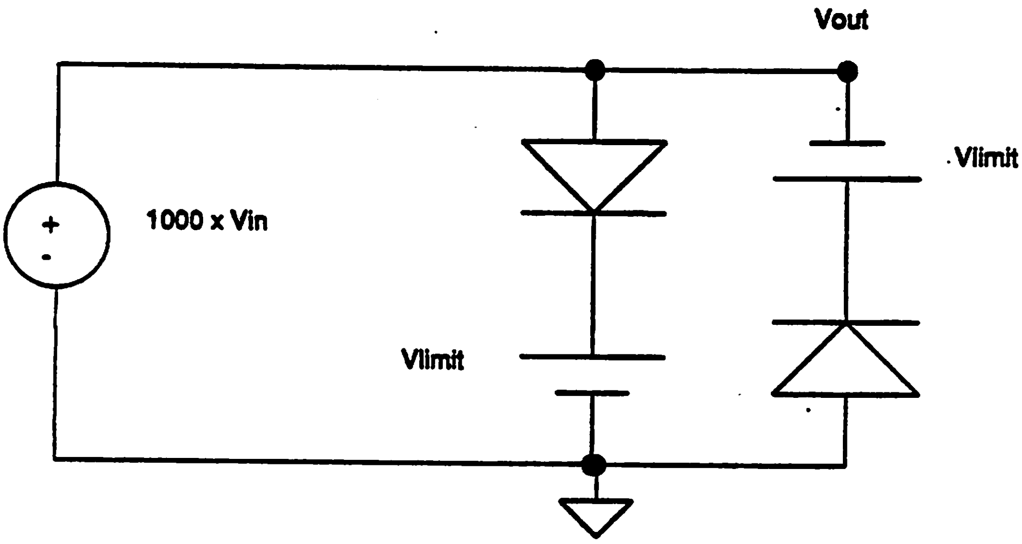


Figure 4.1: PC Output Voltage vs Phase Difference (PC Characteristic)



Vin: Signal to be amplified and limited

Vout: Amplified and limited signal

Figure 4.2: Circuit to Produce Amplifying and Limiting

```
PLL(0) Macromodel with PC inputs amplified & limited
*
* Vin: V(1); VCO: V(7); Vcontrol: V(3)
* Vin Amplified/Limited: V(31); VCO Amplified/Limited: V(27)
* PC Output: V(2)

* Input *
*vin 1 0 sin(0 1 1.190meg)
*vin 1 0 sin(0 1 1.189meg)
*vin 1 0 sin(0 1 0.809meg)
vin 1 0 sin(0 1 0.808meg)
rin 1 0 1k

* Phase comparator multiplier -- PC input limiting *
* Kc = Km Vi Vosc / 2 = 5 V/rad
*
* amplifying/limiting VCO
bpd1 23 0 v=1000*v(7)
ropd1 23 27 0.001
d1pd1 27 28 md2
v1pd1 28 0 8.6325v
d2pd1 0 29 md2
v2pd1 29 27 8.6325v
.model md2 d is=1e-16
*
* amplifying/limiting Vin
bpd2 33 0 v=1000*v(1)
ropd2 33 31 10
d1pd2 31 38 md1
v1pd2 38 0 .171v
d2pd2 0 39 md1
v2pd2 39 31 .171v
.model md1 d is=1e-12
*
* PC output
bpdout 2 0 v=v(27)*v(31)
*
```

```
* Low-pass Filter *
* BW=20kHz
rf 2 3 1k
cf 3 0 7.958nF

* Wien feedback circuit *
* Ko=.1meg
rb 4 10 1K
*gvar2 4 10 poly(2) 4 10 3 0    0 0 0 0 1e-4
bvar2 4 10 i=1e-4 * v(3) * (v(4)-v(10))
vtrig 10 0 pulse(0 5.3 0 0 0 1ns)
c2 4 0 159.1pF
ra 4 6 1K
*gvar1 4 6 poly(2) 4 6 3 0      0 0 0 0 1e-4
bvar1 4 6 i=1e-4 * v(3) * (v(4)-v(6))
c1 6 7 159.1pF

* positive-gain block
*egain 5 0 4 0 3.05
bgain 5 0 v=3.05*v(4)
ro 5 7 .001
d1 7 8 md
v1 8 0 8.8825v
d2 0 9 md
v2 9 7 8.8825v
.model md d is=1e-16

* control cards *
.tran 0.07u 70u 0 0.01u
.options acct
.options nopage itl5=0 limpts=2000
.options reltol=1e-4
.width out=80
.end
```

Figure 4.3: SPICE3 Input File for a PLL Macromodel with Amplified/Limited Inputs

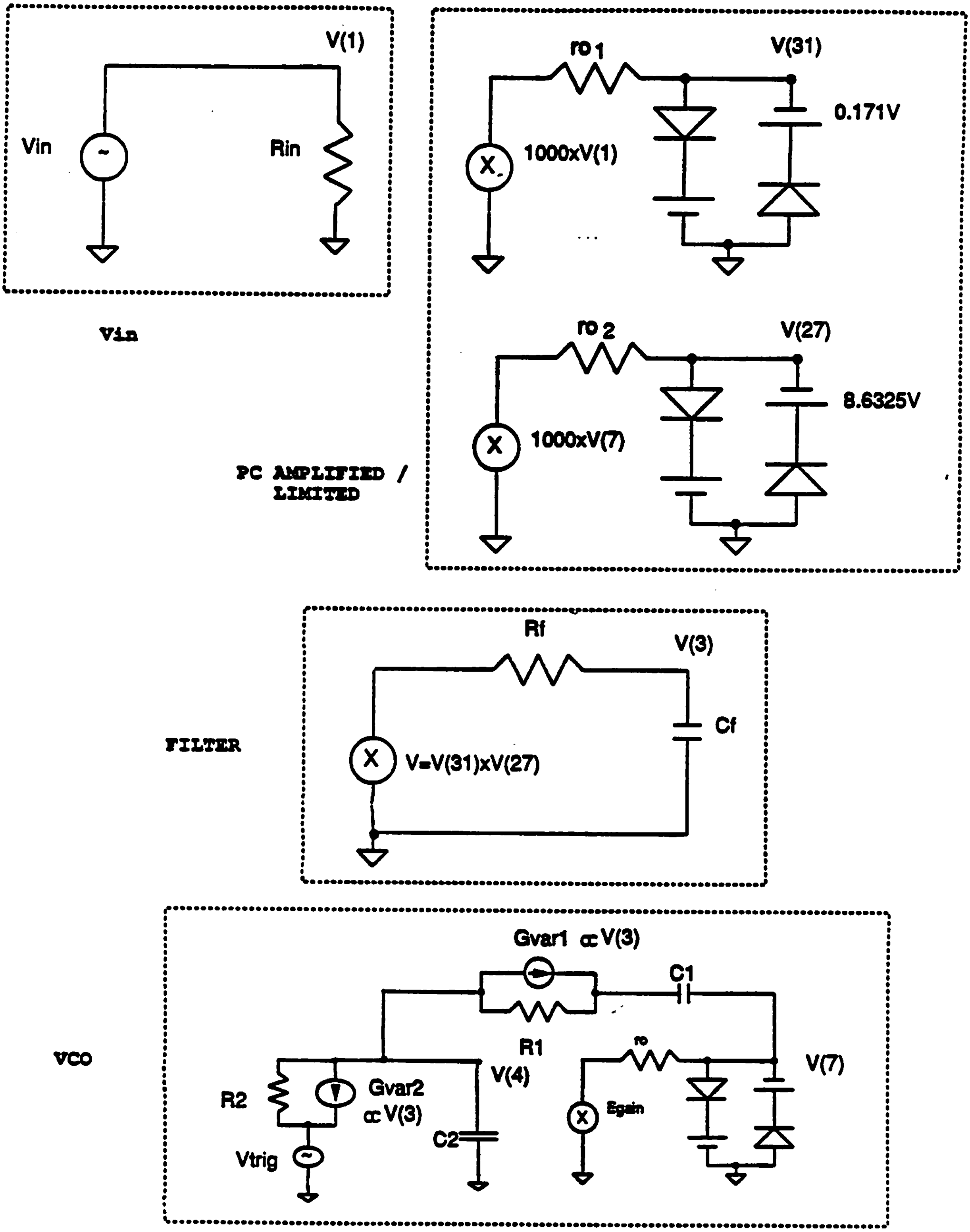


Figure 4.4: Schematic of a PLL Macromodel with Amplified / Limited Inputs

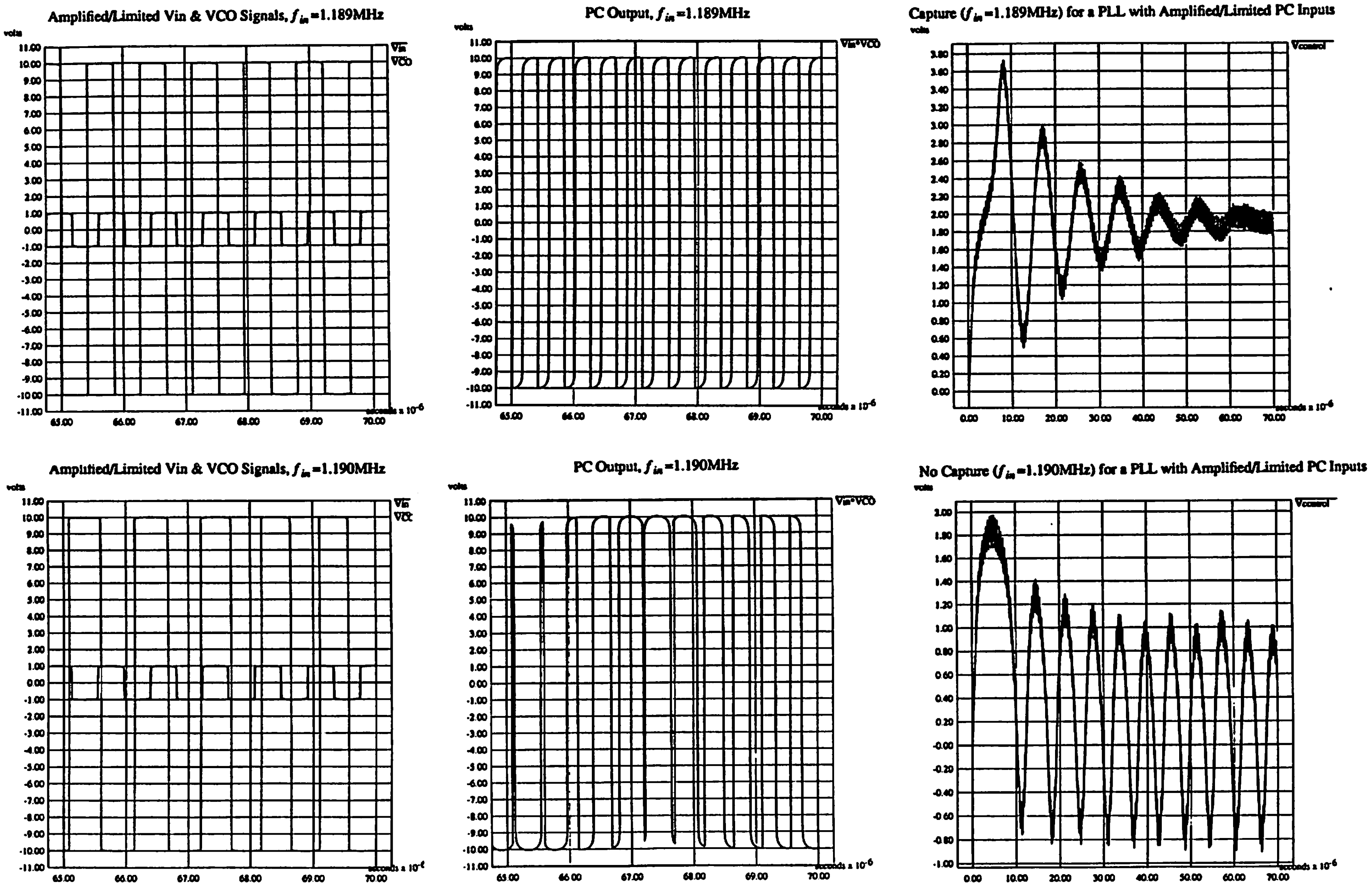
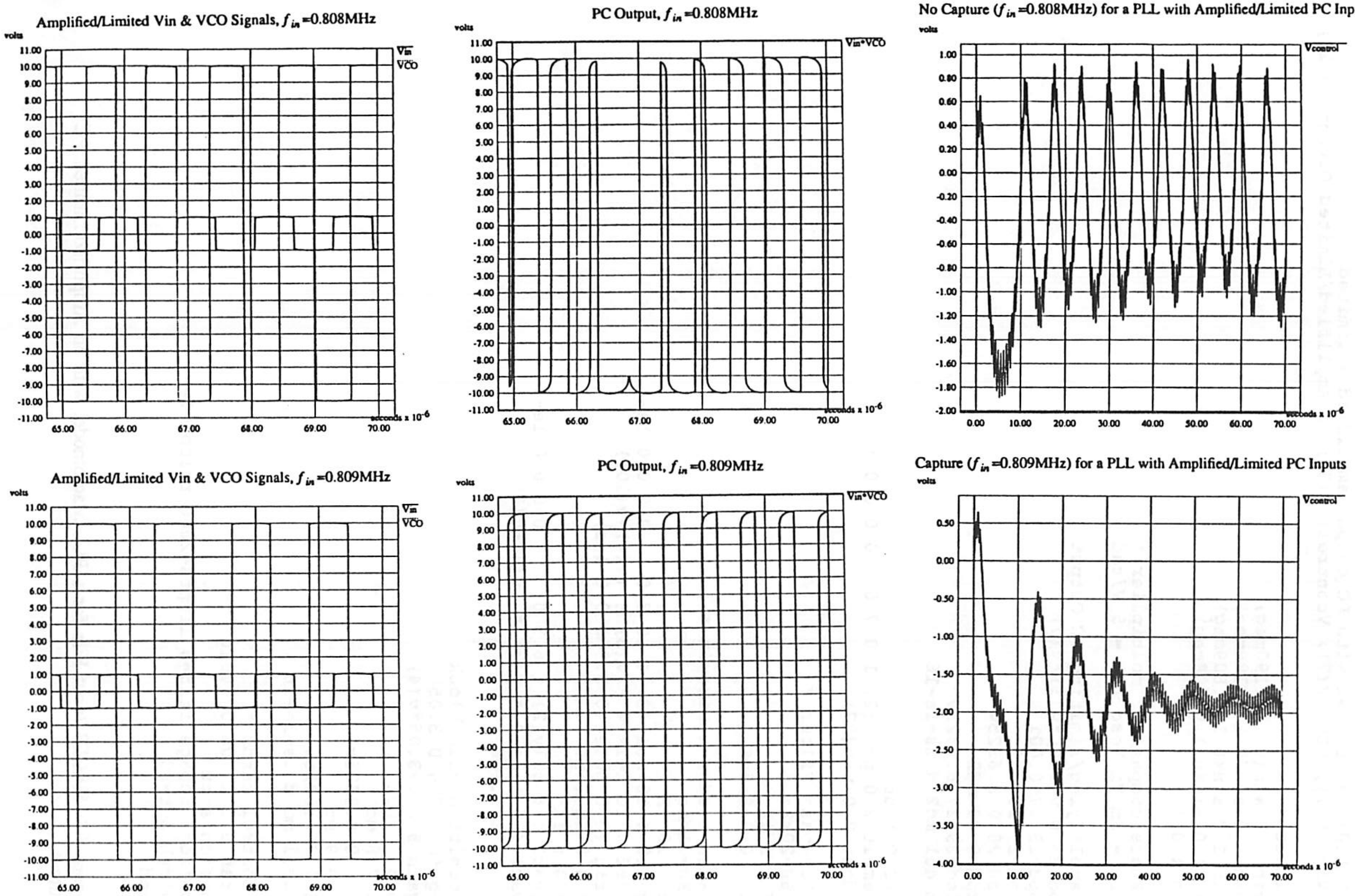


Figure 4.5: Upper Capture Range (1.189MHz,1.190MHz) for a PLL Macromodel having a VCVS PC with Amplified/Limited Inputs



Amplified/Limited Vin & VCO Signals, f_{in}=0.808MHz

PC Output, f_{in}=0.808MHz

No Capture (f_{in}=0.808MHz) for a PLL with Amplified/Limited PC Inputs

Amplified/Limited Vin & VCO Signals, f_{in}=0.809MHz

PC Output, f_{in}=0.809MHz

Capture (f_{in}=0.809MHz) for a PLL with Amplified/Limited PC Inputs

Figure 4.6: Lower Capture Range (0.808MHz,0.809MHz) for a PLL Macromodel having a VCVS PC with Amplified/Limited Inputs

```
* PLL(0) Macromodel with PC output amplified & limited
* Vin: V(1); VCO: V(7); Vcontrol: V(3); PC Amplified/Limited Output: V(27)

* Input *
*vin 1 0 sin(0 1 1.190meg)
*vin 1 0 sin(0 1 1.189meg)
*vin 1 0 sin(0 1 0.809meg)
vin 1 0 sin(0 1 0.808meg)
rin 1 0 1k

* Phase comparator multiplier *
* Kc = Km Vi Vosc / 2 = 5 V/rad
*
* amplifying/limiting PC Output
bpd 23 0 v=1000*v(1)*v(7)
ropd 23 27 0.001
d1pd 27 28 md2
v1pd 28 0 8.6325v
d2pd 0 29 md2
v2pd 29 27 8.6325v
.model md2 d is=1e-16
*
* PC Output
* emult 2 0 poly(2) 1 0 7 0   0 0 0 0 1
bpdout 2 0 v=v(27)

* Low-pass Filter *
* BW=20kHz
rf 2 3 1k
cf 3 0 7.958nF

* Wien feedback circuit *
* Ko=.1meg
rb 4 10 1K
*gvar2 4 10 poly(2) 4 10 3 0    0 0 0 0 1e-4
bvar2 4 10 i=1e-4 * v(3) * (v(4)-v(10))
vtrig 10 0 pulse(0 5.3 0 0 0 1ns)
c2 4 0 159.1pF
ra 4 6 1K
*gvar1 4 6 poly(2) 4 6 3 0      0 0 0 0 1e-4
bvar1 4 6 i=1e-4 * v(3) * (v(4)-v(6))
c1 6 7 159.1pF

* positive-gain block
*egain 5 0 4 0 3.05
bgain 5 0 v=3.05*v(4)
ro 5 7 .001
d1 7 8 md
v1 8 0 8.8825v
d2 0 9 md
v2 9 7 8.8825v
.model md d is=1e-16

* control cards *
.tran 0.07u 70u 0 0.01u
.option acct
.options nopage itl5=0 limpts=2000 reltol=1e-4
.width out=80
.end
```

Figure 4.7: SPICE3 Input File for a PLL Macromodel with an Amplified/Limited PC Output

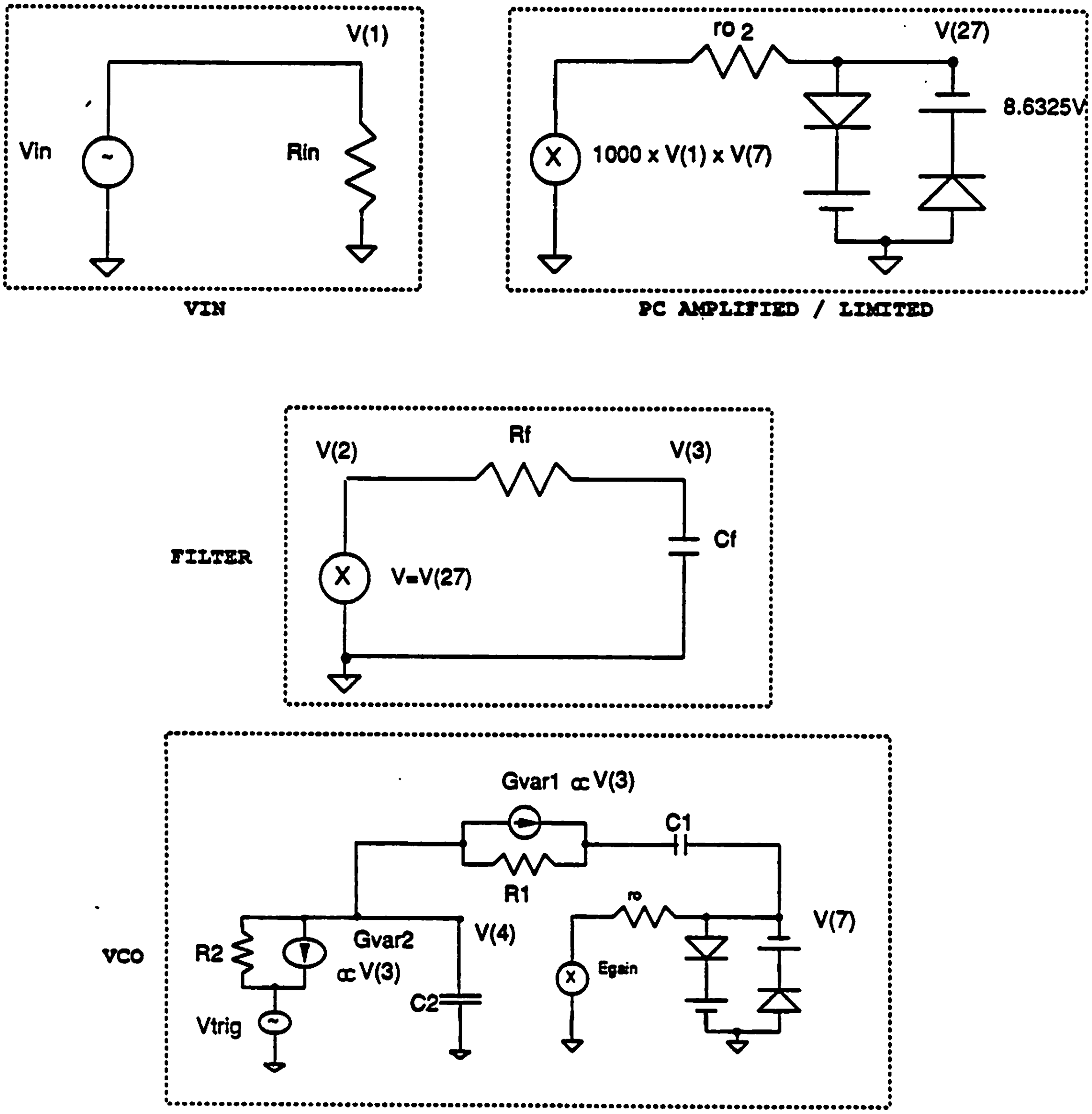


Figure 4.8: Schematic of a PLL Macromodel with an Amplified / Limited PC Output

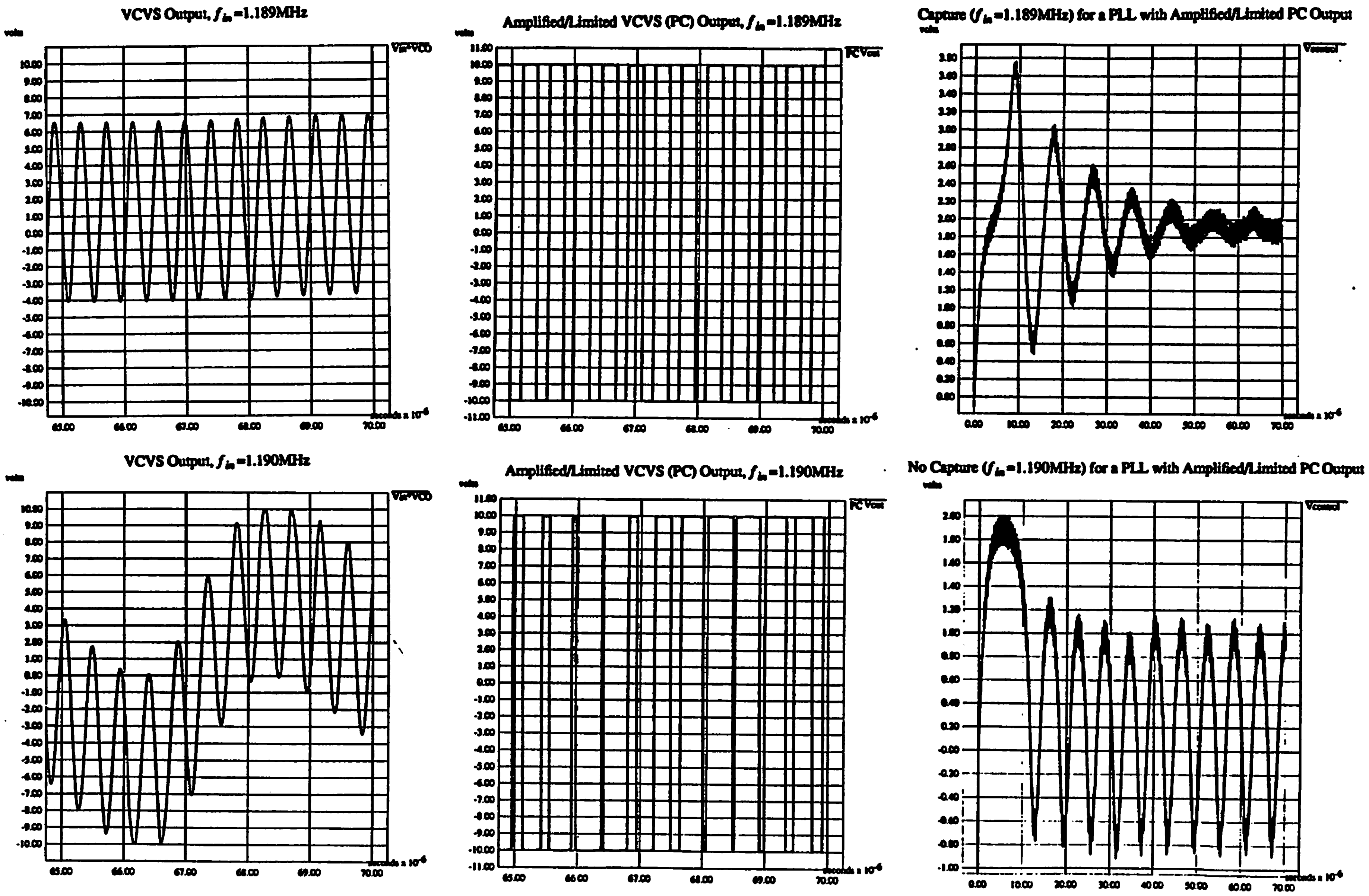
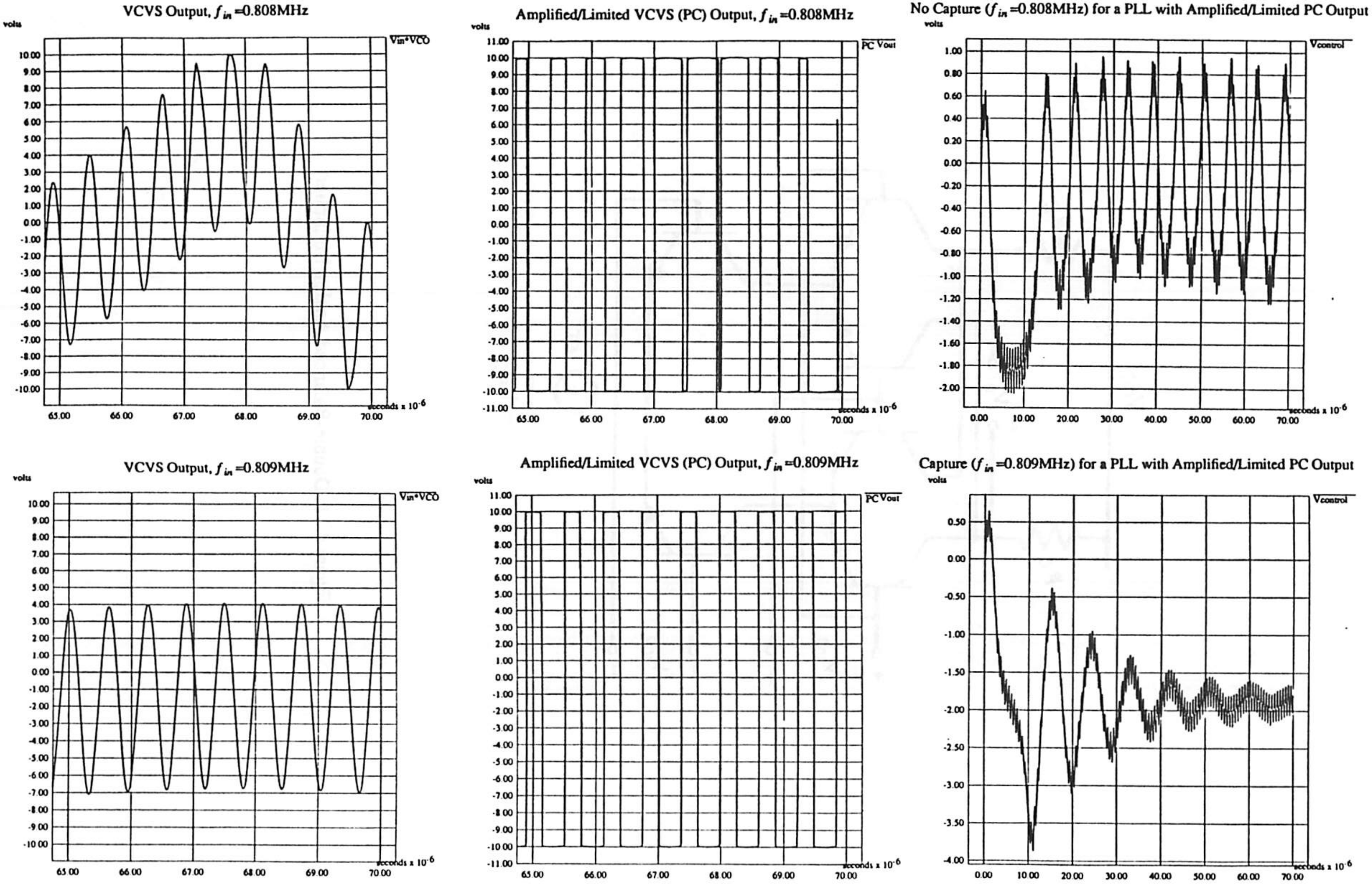


Figure 4.9: Upper Capture Range (1.189MHz,1.190MHz) for a PLL Macromodel having a VCVS PC with an Amplified/Limited PC Output



VCVS Output, f_{in}=0.808MHz

Amplified/Limited VCVS (PC) Output, f_{in}=0.808MHz

No Capture (f_{in}=0.808MHz) for a PLL with Amplified/Limited PC Output

VCVS Output, f_{in}=0.809MHz

Amplified/Limited VCVS (PC) Output, f_{in}=0.809MHz

Capture (f_{in}=0.809MHz) for a PLL with Amplified/Limited PC Output

Figure 4.10: Lower Capture Range (0.808MHz,0.809MHz) for a PLL Macromodel having a VCVS PC with an Amplified/Limited PC Output

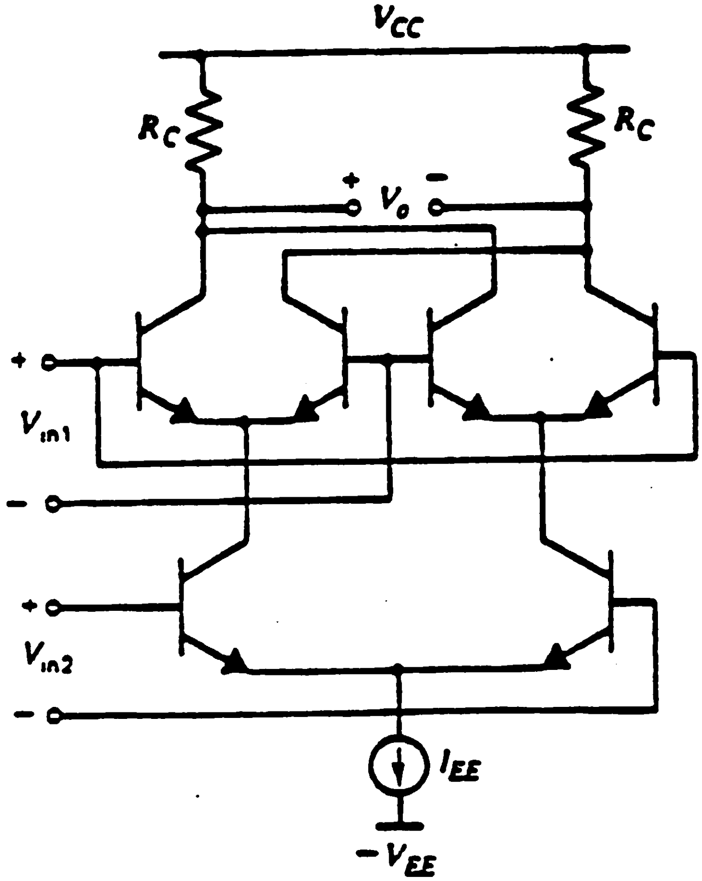
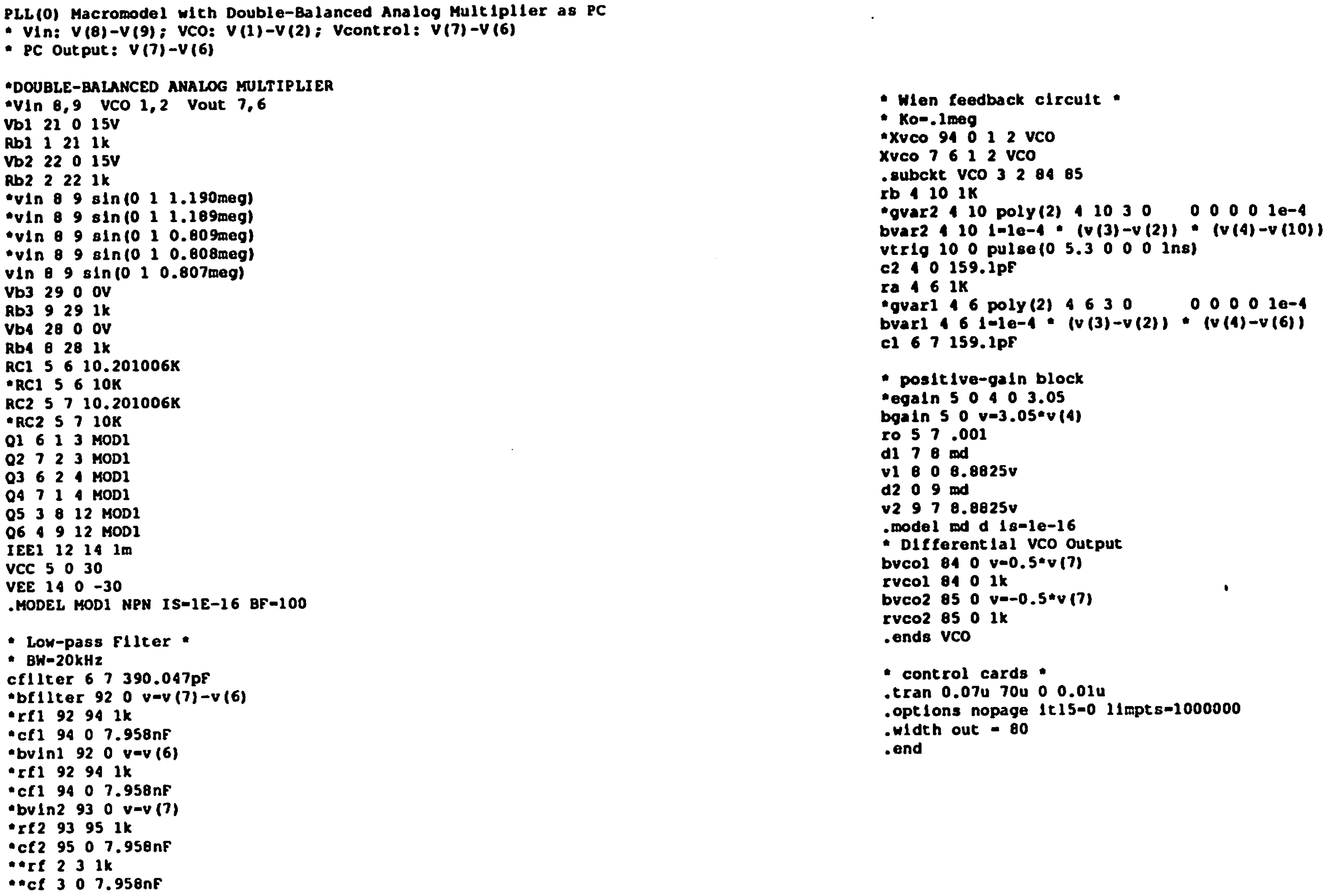


Figure 4.11: Double-Balanced Analog Multiplier



```
PLL(0) Macromodel with Double-Balanced Analog Multiplier as PC
* Vin: V(8)-V(9); VCO: V(1)-V(2); Vcontrol: V(7)-V(6)
* PC Output: V(7)-V(6)

*DOUBLE-BALANCED ANALOG MULTIPLIER
*Vin 8,9  VCO 1,2  Vout 7,6
Vb1 21 0 15V
Rb1 1 21 1k
Vb2 22 0 15V
Rb2 2 22 1k
*vin 8 9 sin(0 1 1.190meg)
*vin 8 9 sin(0 1 1.189meg)
*vin 8 9 sin(0 1 0.809meg)
*vin 8 9 sin(0 1 0.808meg)
vin 8 9 sin(0 1 0.807meg)
Vb3 29 0 0V
Rb3 9 29 1k
Vb4 28 0 0V
Rb4 8 28 1k
RC1 5 6 10.201006K
*RC1 5 6 10K
RC2 5 7 10.201006K
*RC2 5 7 10K
Q1 6 1 3 MOD1
Q2 7 2 3 MOD1
Q3 6 2 4 MOD1
Q4 7 1 4 MOD1
Q5 3 8 12 MOD1
Q6 4 9 12 MOD1
IEE1 12 14 1m
VCC 5 0 30
VEE 14 0 -30
.MODEL MOD1 NPN IS=1E-16 BF=100

* Low-pass Filter *
* BW=20kHz
cfilter 6 7 390.047pF
*bfilter 92 0 v=v(7)-v(6)
*rf1 92 94 1k
*cf1 94 0 7.958nF
*bvin1 92 0 v=v(6)
*rf1 92 94 1k
*cf1 94 0 7.958nF
*bvin2 93 0 v=v(7)
*rf2 93 95 1k
*cf2 95 0 7.958nF
**rf 2 3 1k
**cf 3 0 7.958nF
```

```
* Wien feedback circuit *
* Ko=.1meg
*Xvco 94 0 1 2 VCO
Xvco 7 6 1 2 VCO
.subckt VCO 3 2 84 85
rb 4 10 1K
*gvar2 4 10 poly(2) 4 10 3 0    0 0 0 0 1e-4
bvar2 4 10 i=1e-4 * (v(3)-v(2)) * (v(4)-v(10))
vtrig 10 0 pulse(0 5.3 0 0 0 1ns)
c2 4 0 159.1pF
ra 4 6 1K
*gvar1 4 6 poly(2) 4 6 3 0      0 0 0 0 1e-4
bvar1 4 6 i=1e-4 * (v(3)-v(2)) * (v(4)-v(6))
c1 6 7 159.1pF

* positive-gain block
*egain 5 0 4 0 3.05
bgain 5 0 v=3.05*v(4)
ro 5 7 .001
d1 7 8 md
v1 8 0 8.8825v
d2 0 9 md
v2 9 7 8.8825v
.model md d is=1e-16
* Differential VCO Output
bvco1 84 0 v=0.5*v(7)
rvco1 84 0 1k
bvco2 85 0 v=-0.5*v(7)
rvco2 85 0 1k
.ends VCO

* control cards *
.tran 0.07u 70u 0 0.01u
.options nopage itl5=0 limpts=1000000
.width out = 80
.end
```

Figure 4.12: SPICE3 Input File for a PLL Macromodel with a Double-Balanced Analog Multiplier as the PC

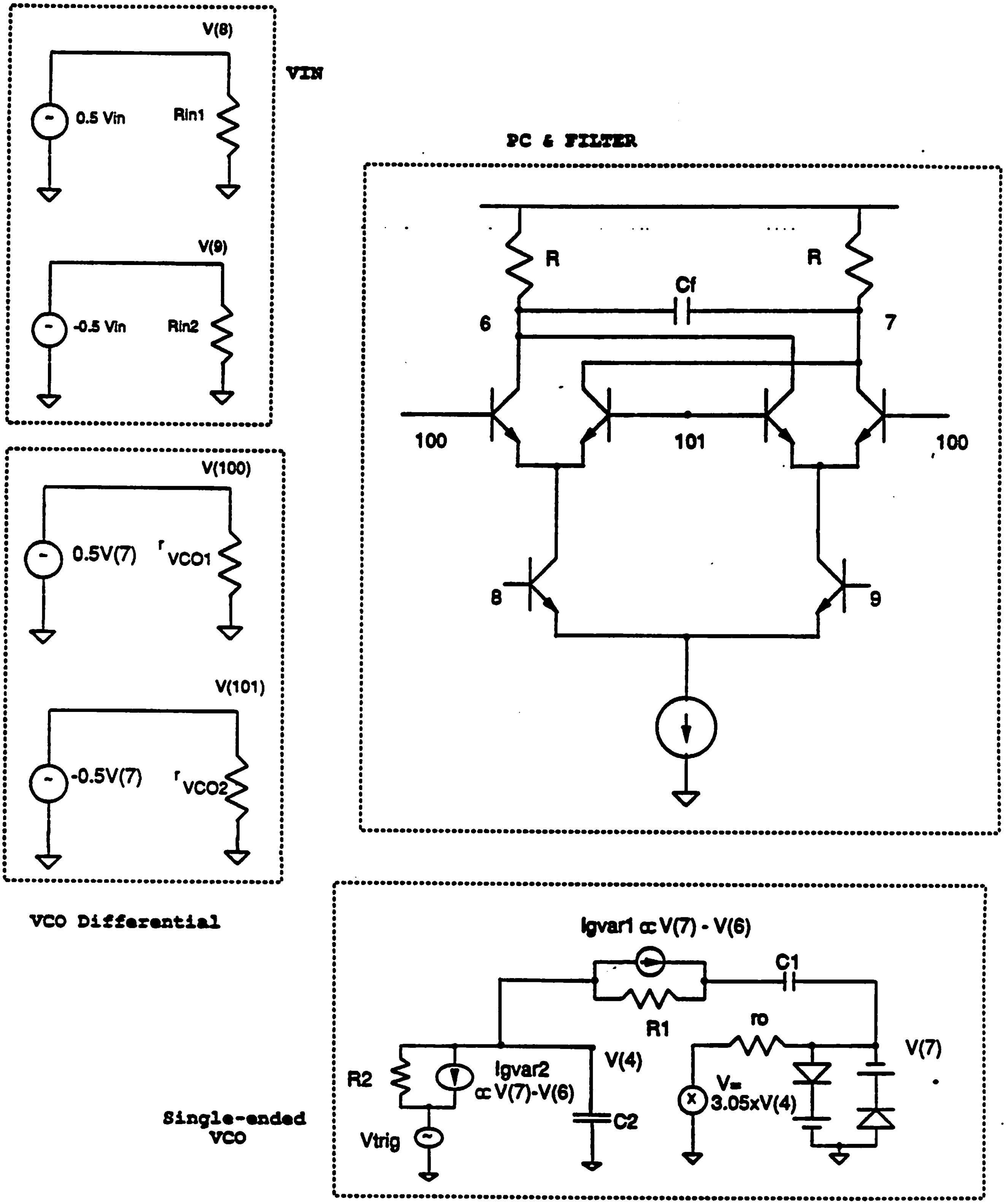


Figure 4.13: Schematic for a PLL Macromodel with a Double-Balanced Analog Multiplier as the PC

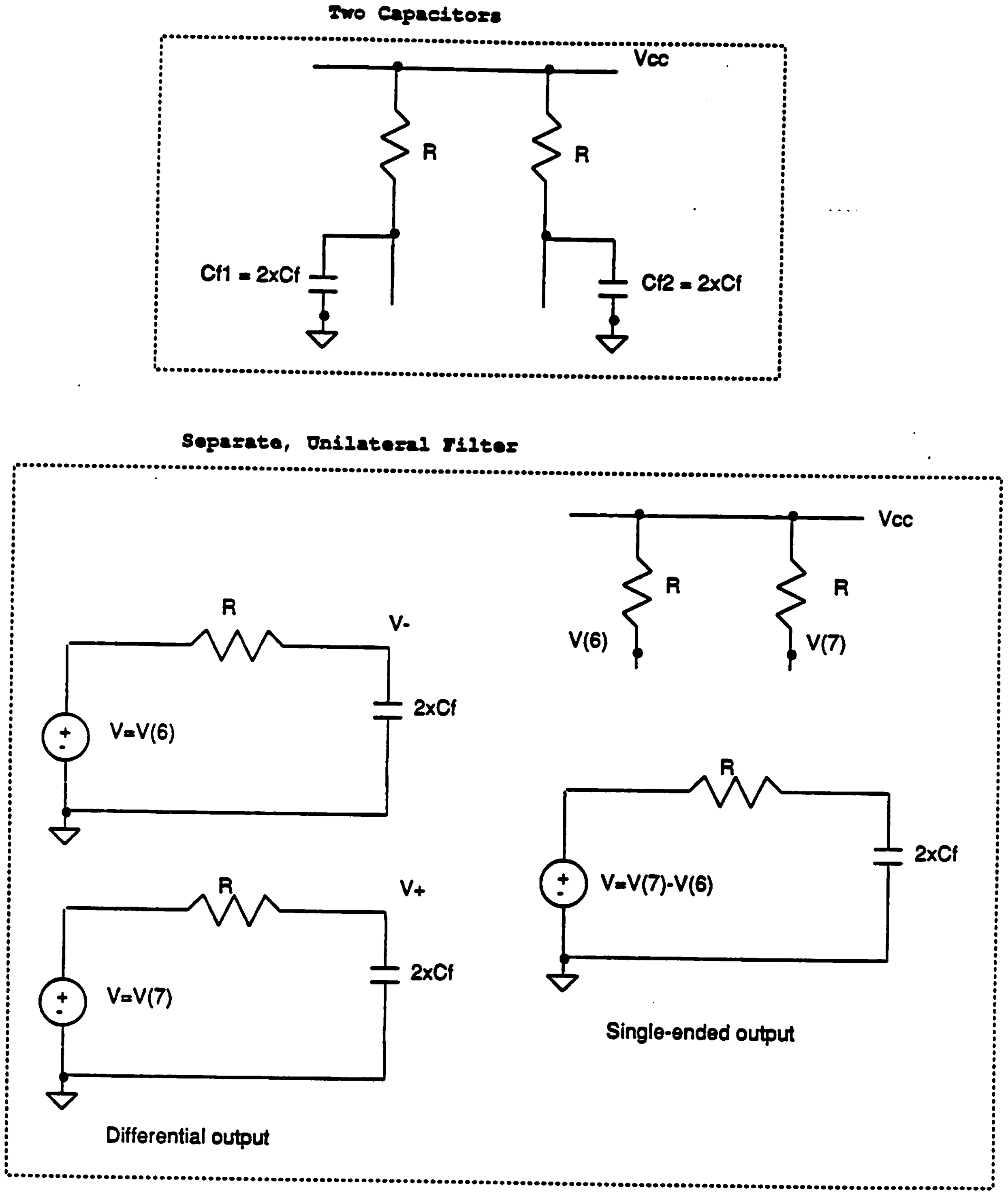
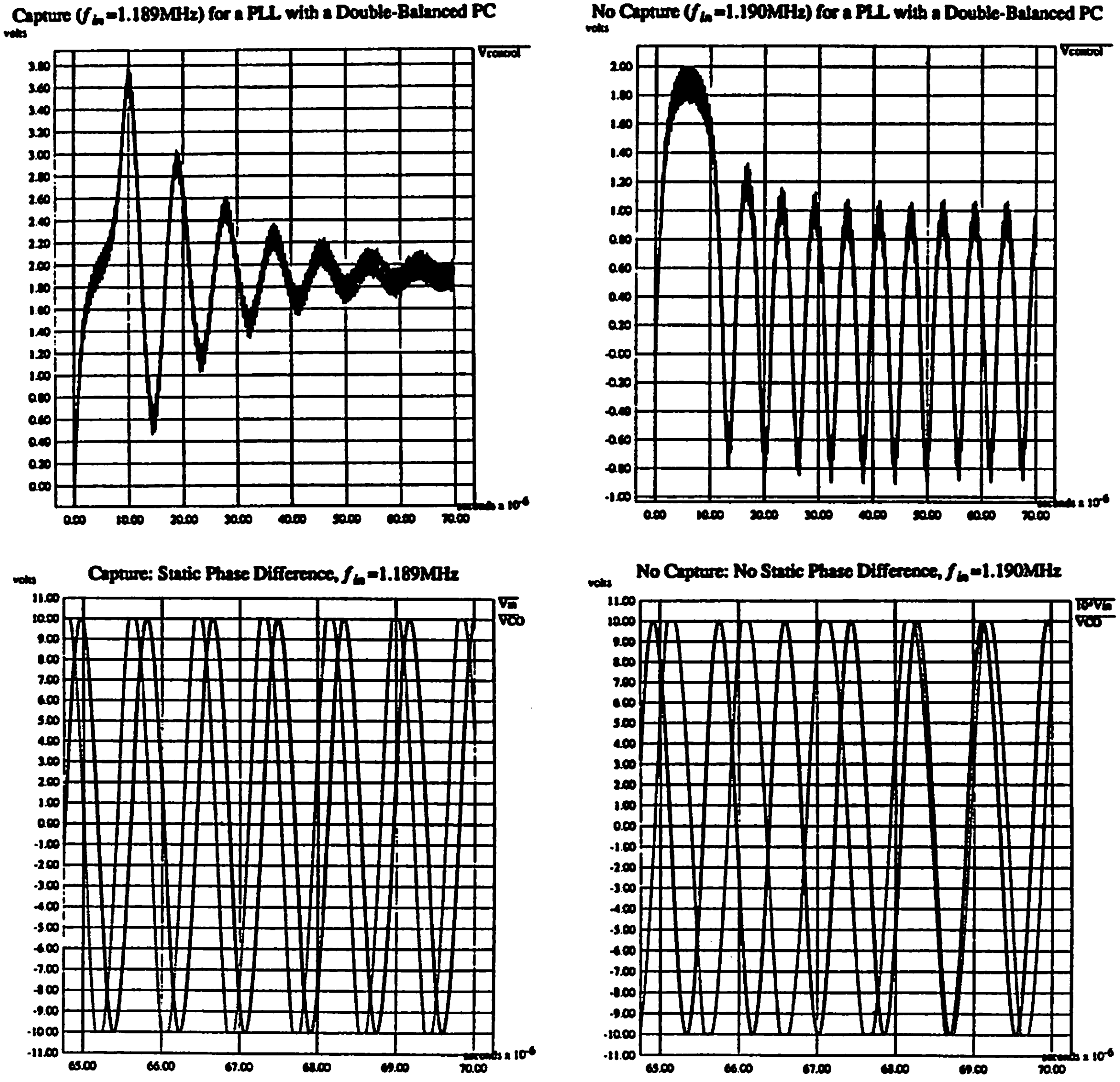


Figure 4.14: Alternative Filtering Schemes for a Double-Balanced Analog Multiplier Differential Output



Capture (f_{in}=1.189MHz) for a PLL with a Double-Balanced PC

No Capture (f_{in}=1.190MHz) for a PLL with a Double-Balanced PC

Capture: Static Phase Difference, f_{in}=1.189MHz

No Capture: No Static Phase Difference, f_{in}=1.190MHz

Figure 4.15: Upper Capture Range (1.189MHz,1.190MHz) for a PLL Macromodel with a Double-Balanced Analog Multiplier PC

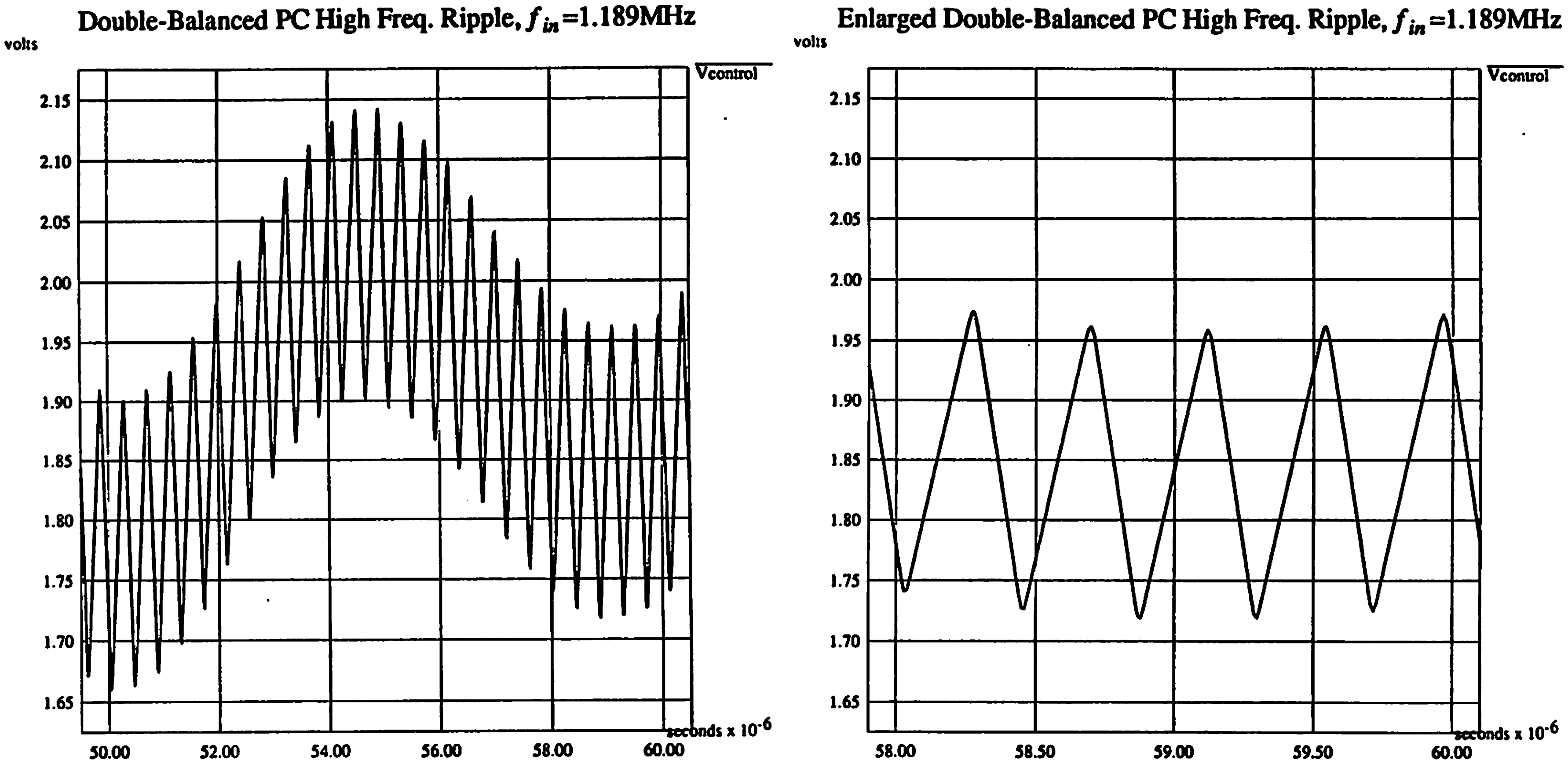
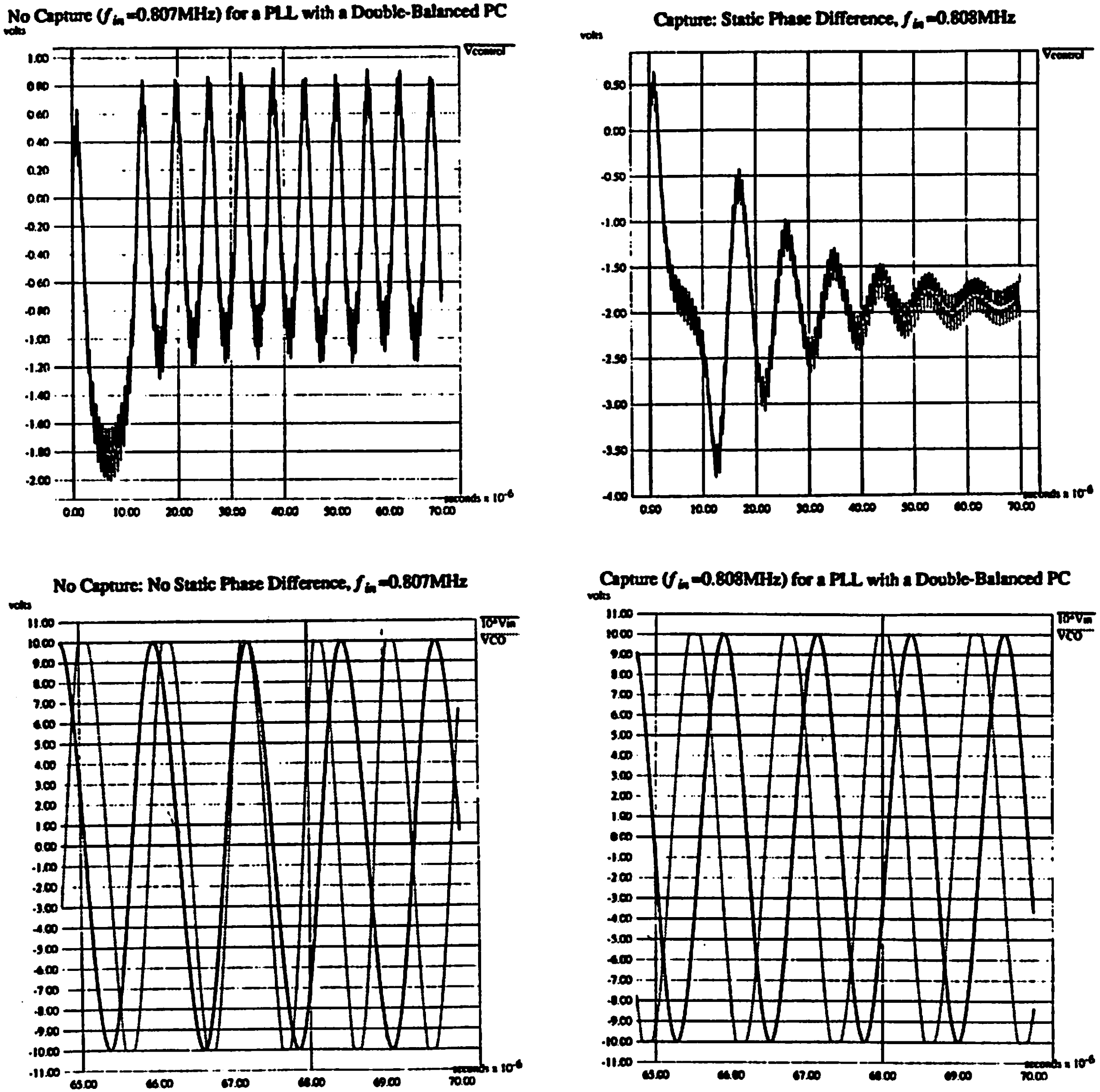


Figure 4.16: High Frequency Ripple of a PLL Macromodel with a Double-Balanced Analog Multiplier PC



No Capture (f_{in}=0.807MHz) for a PLL with a Double-Balanced PC

Capture: Static Phase Difference, f_{in}=0.808MHz

No Capture: No Static Phase Difference, f_{in}=0.807MHz

Capture (f_{in}=0.808MHz) for a PLL with a Double-Balanced PC

Figure 4.17: Lower Capture Range (0.807MHz,0.808MHz) for a PLL Macromodel with a Double-Balanced Analog Multiplier PC

VCO: Sine of Integrated Vcontrol

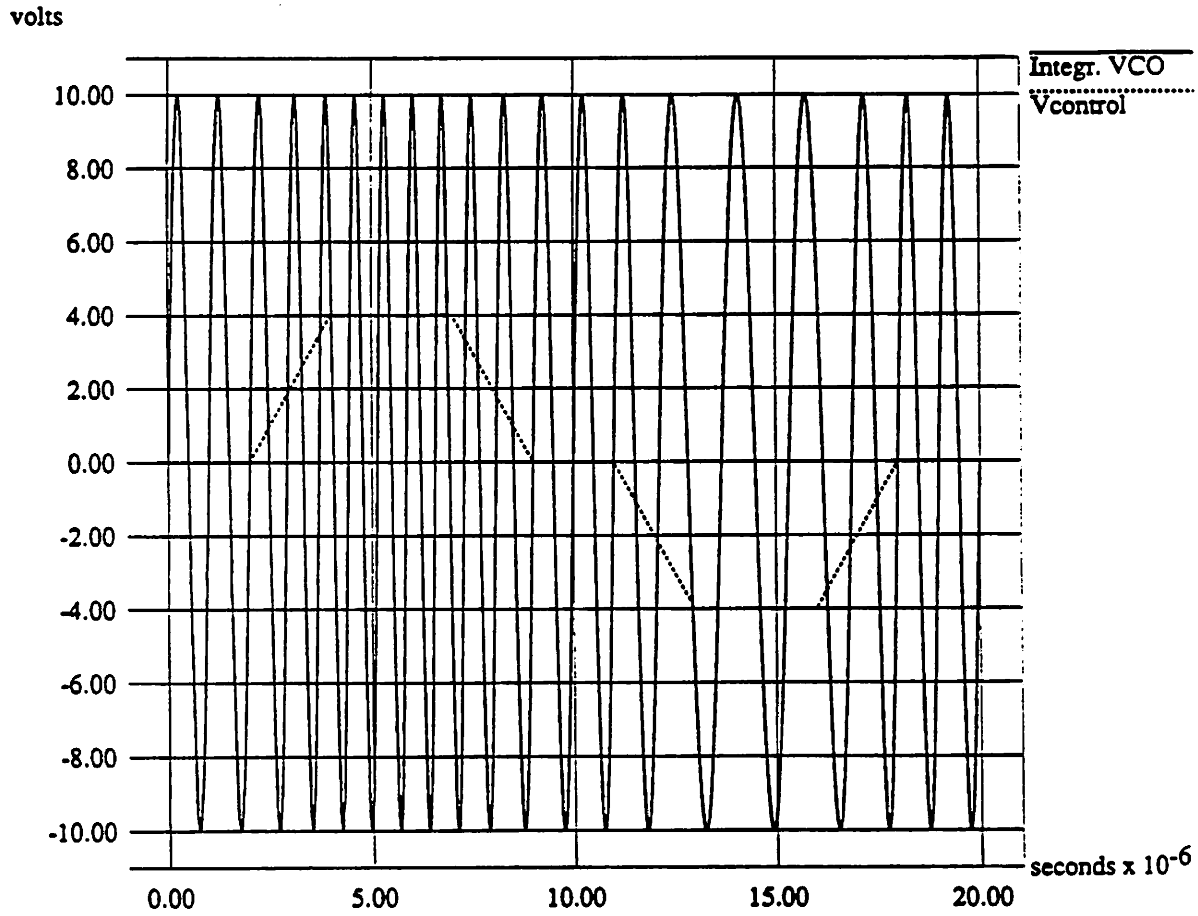
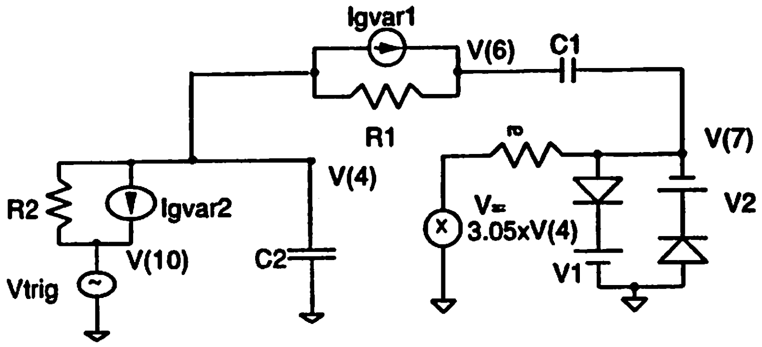


Figure 5.3: Integrating Sine VCO Output Waveform



Igvar1 = (1 / (10xR1)) x V(3) [V(4) - V(6)]

Igvar2 = (1 / (10xR2)) x V(3) [V(4) - V(10)]

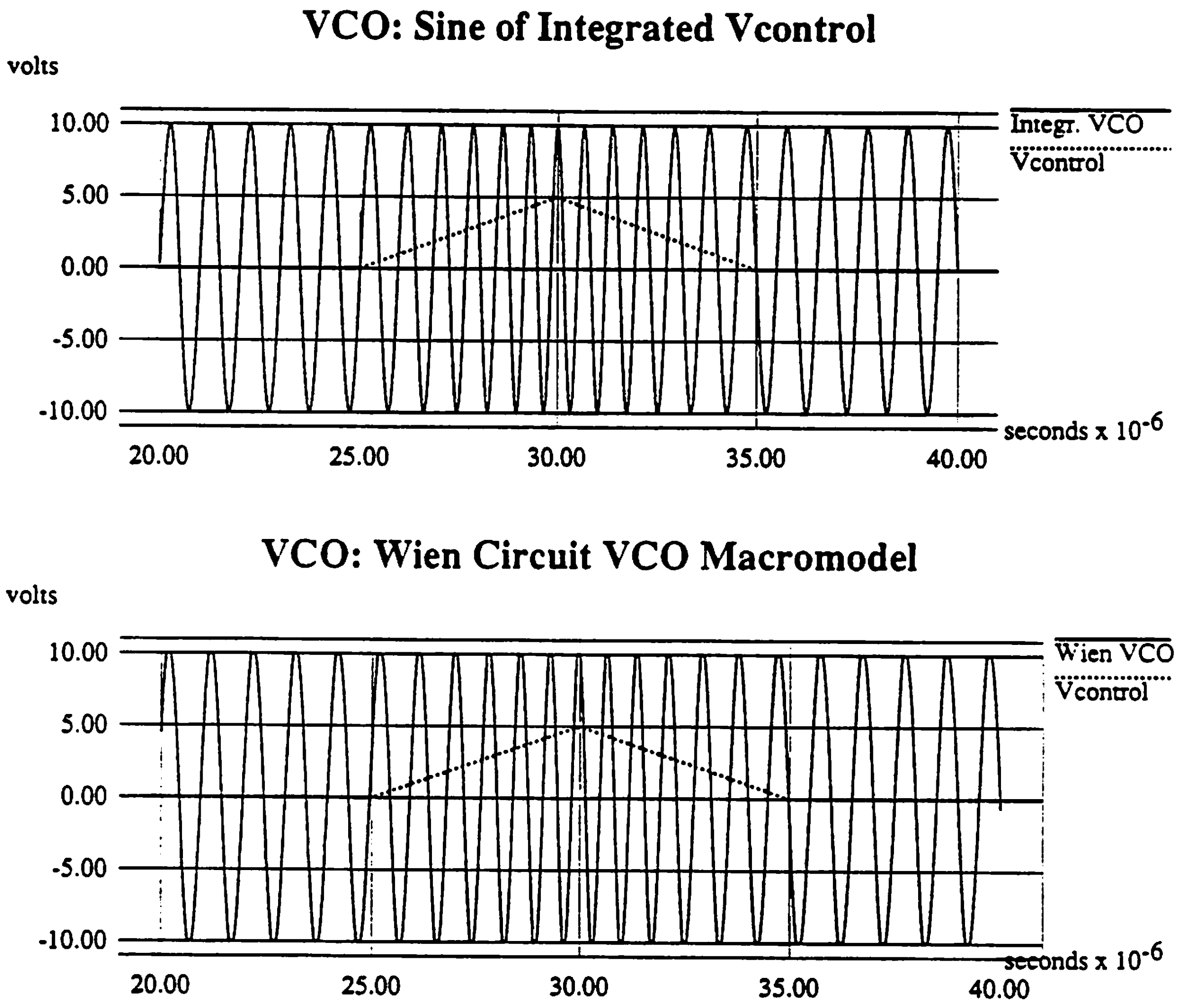
V(7) = VCO output

V(3) = Control Voltage

Figure 5.4: Schematic for the Wien VCO Macromodel

```
* Wien VCO Macromodel *
*
* Vcontrol: V(3); VCO: V(7)

*vvco is Vcontrol
*vvco 3 0 0 pwl 0 0 25u 0 30u 5 35u 0 40u 0
rin 3 0 1k

* Wien-type voltage controlled oscillator *
*  Ko=100KHz/volt   fo=1MHz   *
*
* input circuit *
r2 4 10 1k
*gvar2 4 10 poly(2) 4 10 3 0    0 0 0 0 1e-4
BVAR2 4 10 I= 1E-4*V(3)*(V(4)-V(10))
vtrig 10 0 0 pulse(0 10 0 0 0 1ns)
c2 4 0 159.1pF

*positive feed back *
r1 4 6 1k
*gvar1 4 6 poly(2) 4 6 3 0     0 0 0 0 1e-4
BVAR1 4 6 I= 1E-4*v(3)*(v(4)-v(6))
c1 6 7 159.1pF

* positive-gain block *
*egain 5 0 4 0 3.05
BGAIN 5 0 V=3.05*V(4)
ro 5 7 0.001
d1 7 8 md
v1 8 0 8.8825v
d2 0 9 md
v2 9 7 8.8825v
.model md d is=1e-16

* control cards *
.tran 0.01u 40u 20u 0.01u
.option acct
.options nopage itl5=0 limpts=20000
.options reltol=1e-6
.width out=80
.end
```

Figure 5.5: SPICE3 Input File for the Wien VCO Macromodel



VCO: Sine of Integrated Vcontrol

VCO: Wien Circuit VCO Macromodel

Figure 5.6: Integrating Sine VCO and Wien VCO Output Waveforms

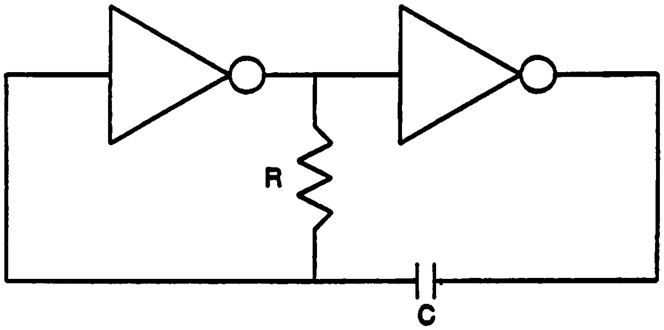


Figure 5.7: The Two-Inverter Multivibrator

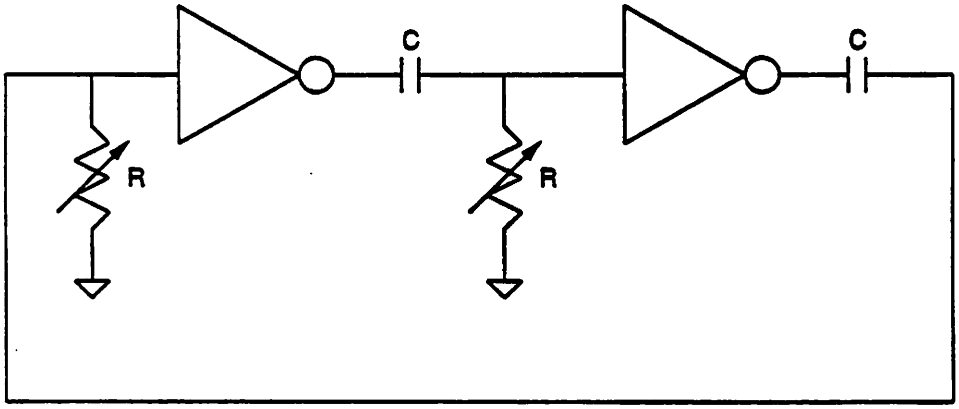


Figure 5.8: A Ring-Oscillator VCO

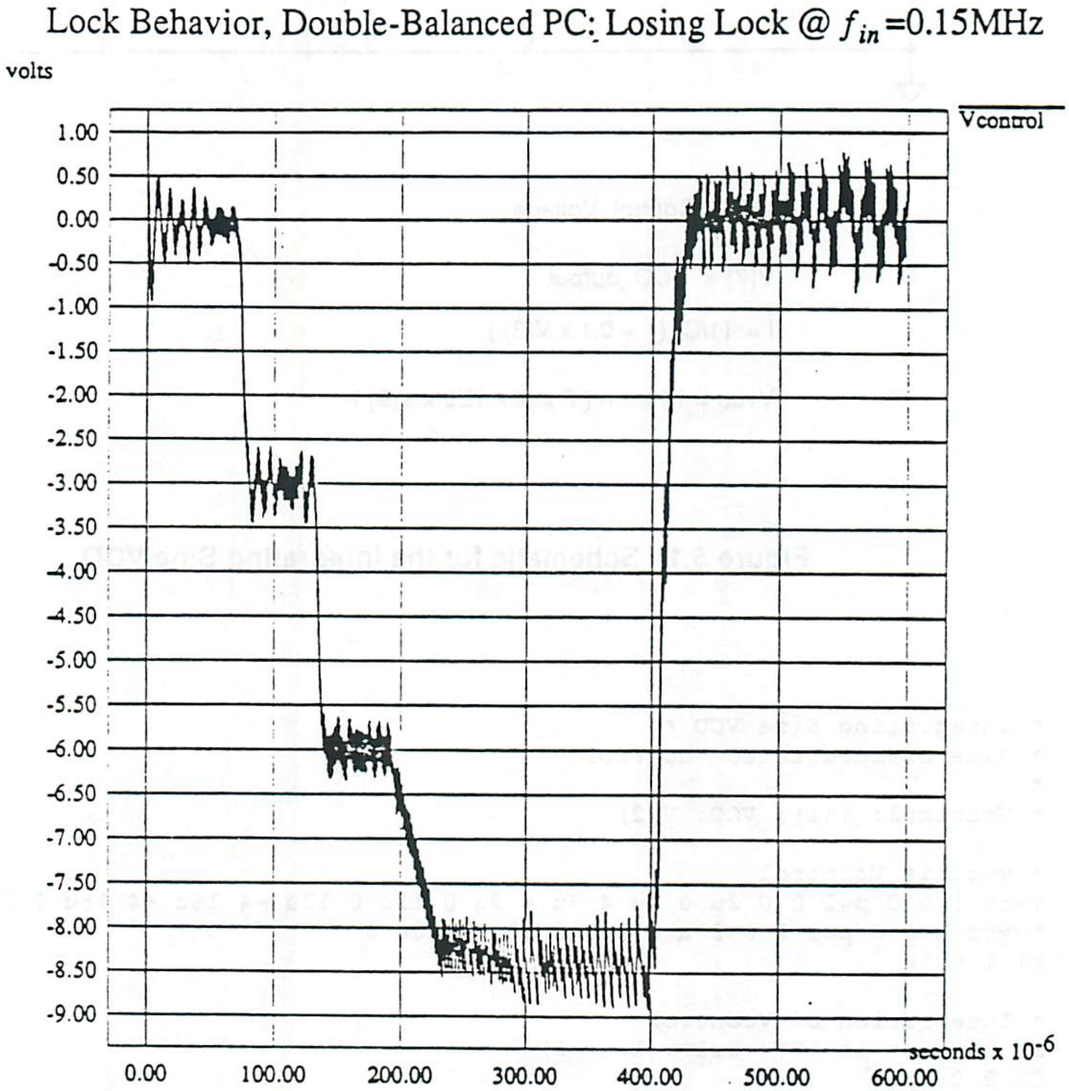
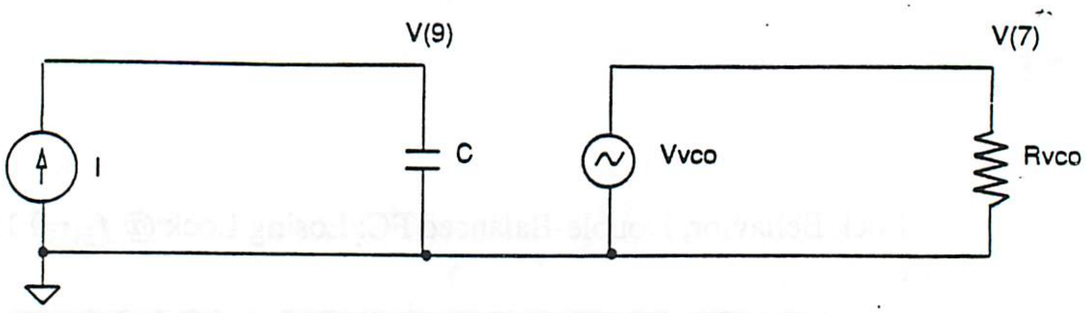


Figure 4.18: Lock Behavior for a PLL Macromodel with a Double-Balanced Analog Multiplier as the PC



V(3) = Control Voltage

V(7) = VCO output

I = (1/C) (1 + 0.1 x V(3))

Vvco = 10 x sin (2 x pi x 1E6 x V(9))

Figure 5.1: Schematic for the Integrating Sine VCO

```
* Integrating Sine VCO *
* Sine of integrated Vcontrol
*
* Vcontrol: V(1); VCO: V(2)

* vvco is Vcontrol
vvco 1 0 0 pwl 0 0 2u 0 4u 4 7u 4 9u 0 11u 0 13u -4 16u -4 18u 0 20u 0
*vvco 1 0 0 pwl 0 0 25u 0 30u 5 35u 0 40u 0
r1 1 0 1k

* Integration of Vcontrol
B1 0 9 i = 1e-12*( 0.1*v(1) + 1)
C1 9 0 1p
.ic v(9)=0

* VCO function
B2 2 0 v = 10 * sin(2e+6 * pi * (v(9)))
R2 2 0 1k

.tran .01u 40u 20u 0.01u
.options reltol=1e-6
.options nopage itl5=0 limpts=20000
.end
```

Figure 5.2: SPICE3 Input File for the Integrating Sine VCO

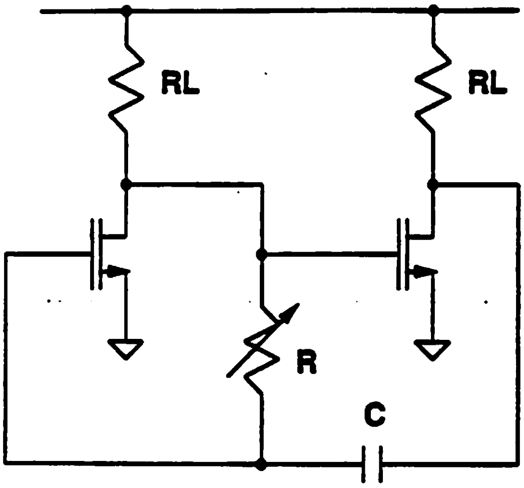


Figure 5.9: Schematic for the Resistive-Load Two-Inverter VCO

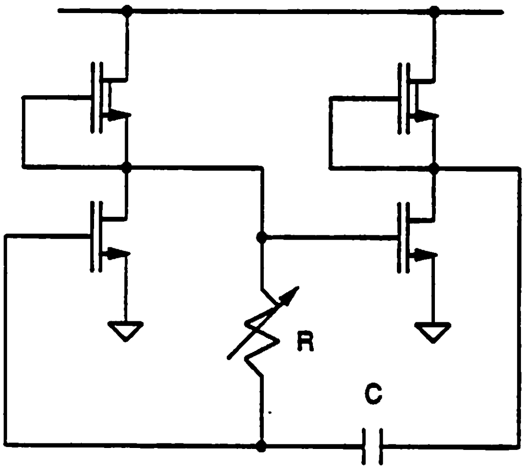


Figure 5.10: Schematic for the Depletion-Load Two-Inverter VCO

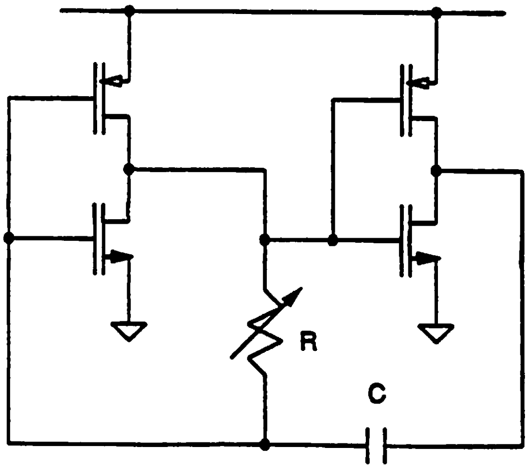


Figure 5.11: Schematic for the CMOS Two-Inverter VCO

```
Resistive-Load Inverter, 2-Inverter Multivibrator VCO
*

RD1 4 2 10K
RD2 4 3 10K
M1 2 1 0 5 MOD2 W=80u L=8u
M2 3 2 0 5 MOD2 W=80u L=8u
Vss 5 0 0v
I1 0 1 PULSE 10u 0 0 0 0 1s
C1 1 0 1f
C2 2 0 1f
R1 2 1 100K
*gvar1 2 1 poly(2) 2 1 10 0      0 0 0 0 1e-6
BVAR1 2 1 I= 1E-6*v(10)*(v(2)-v(1))
Cc 3 1 6.005p
*Cc 3 1 6p
Vcc 4 0 5V
.MODEL MOD1 PMOS vto=-0.5 KP=10u
.MODEL MOD2 NMOS vto=0.5 KP=30u
Vcontrol 10 0 0V
Rcontrol 10 0 1Meg
*.op
* control cards *
.tran 0.005u 20u 0u 0.005u
.options nopage itl5=0 limpts=20000
.options reltol=1e-6
.width out=80
.end
```

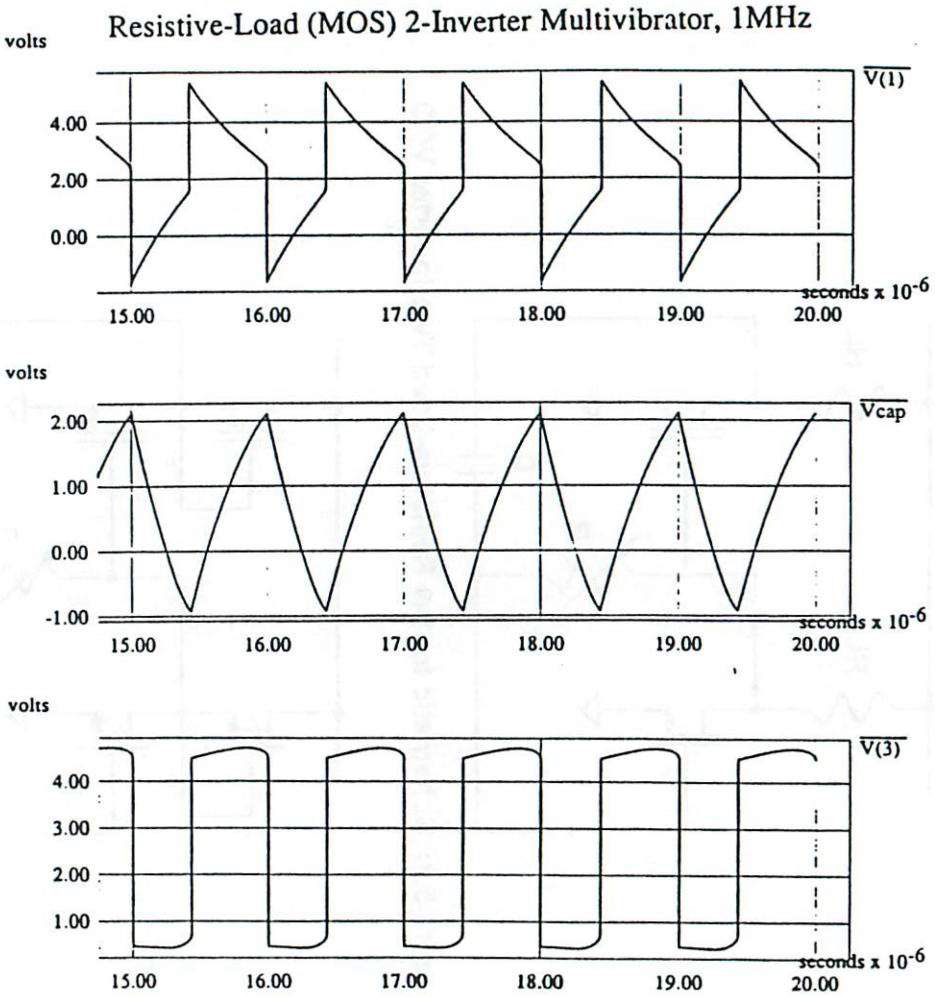


Figure 5.12: SPICE3 Input File & Output Waveforms for the Resistive-Load Two- Inverter VCO

```
Depletion-Load Inverter, 2-Inverter Multivibrator VCO

M1 2 1 0 5 MOD2 W=80u L=8u
M2 3 2 0 5 MOD2 W=80u L=8u
M11 4 2 2 5 MODd W=800u L=8u
M12 4 3 3 5 MODd W=800u L=8u
Vss 5 0 0v
I1 0 1 PULSE 10u 0 0 0 0 1s
C1 1 0 1f
C2 2 0 1f
R1 2 1 100K
*gvar1 2 1 poly(2) 2 1 10 0     0 0 0 0 0.5e-5
BVAR1 2 1 I= 0.1E-5*v(10)*(v(2)-v(1))
Vcontrol 10 0 0V
Rcontrol 10 0 1Meg
Cc 3 1 4.32p
Vcc 4 0 5V
*
* Models
.MODEL MOD1 PMOS vto=-0.5 KP=10u cbd=0 cbs=0 cj=0 cjsw=0 cgso=0 cgdo=0 cgbo=0
.MODEL MOD2 NMOS vto=0.5 KP=30u cbd=0 cbs=0 cj=0 cjsw=0 cgso=0 cgdo=0 cgbo=0
.MODEL MODd NMOS vto=-0.5 KP=30u cbd=0 cbs=0 cj=0 cjsw=0 cgso=0 cgdo=0 cgbo=0
* control cards *
.tran 0.005u 20u 0u 0.005u
.options nopage itl5=0 limpts=20000
.option acct
.options reltol=1e-6
.width out=80
.end
```

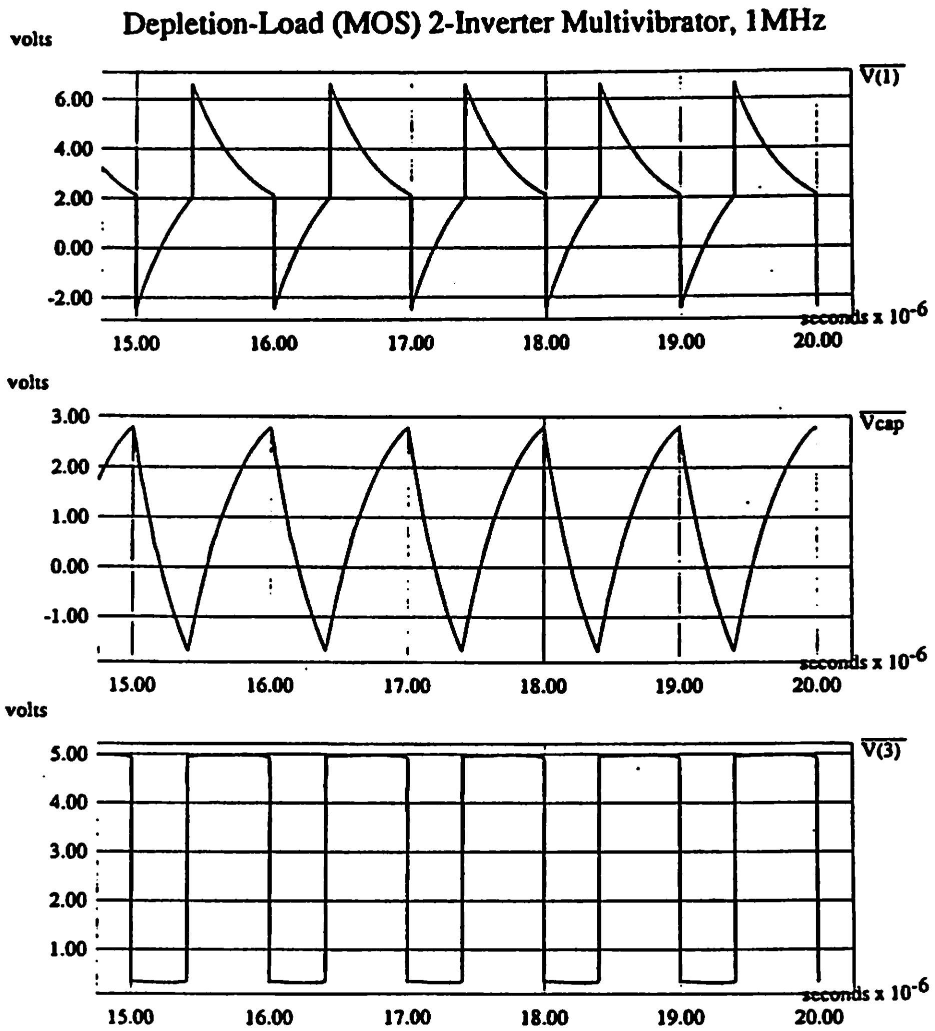


Figure 5.13: SPICE3 Input File & Output Waveforms for the Depletion-Load Two- Inverter VCO

```
CMOS Inverter, 2-Inverter Multivibrator VCO

M1 2 1 0 5 MOD2 W=80u L=8u
M2 3 2 0 5 MOD2 W=80u L=8u
M11 2 1 4 4 MOD1 W=80u L=8u
M12 3 2 4 4 MOD1 W=80u L=8u
Vss 5 0 0v
I1 0 1 PULSE 10u 0 0 0 0 1s
C1 1 0 1f
C2 2 0 1f
R1 2 1 100K
*gvar1 2 1 poly(2) 2 1 10 0     0 0 0 0 1e-6
BVAR1 2 1 I= 1E-6*v(10)*(v(2)-v(1))
Cc 3 1 4.495171p
Vcc 4 0 5V
* Models
.MODEL MOD1 PMOS vto=-0.5 KP=10u cbd=0 cbs=0 cj=0 cjsw=0 cgso=0 cgdo=0 cgbo=0
.MODEL MOD2 NMOS vto=0.5 KP=30u cbd=0 cbs=0 cj=0 cjsw=0 cgso=0 cgdo=0 cgbo=0
Vcontrol 10 0 0V
Rcontrol 10 0 1Meg
* control cards *
.tran 0.005u 20u 0u 0.005u
.options nopage itl5=0 limpts=20000
.options reltol=1e-6
.width out=80
.end
```

CMOS 2-Inverter Multivibrator, 1MHz

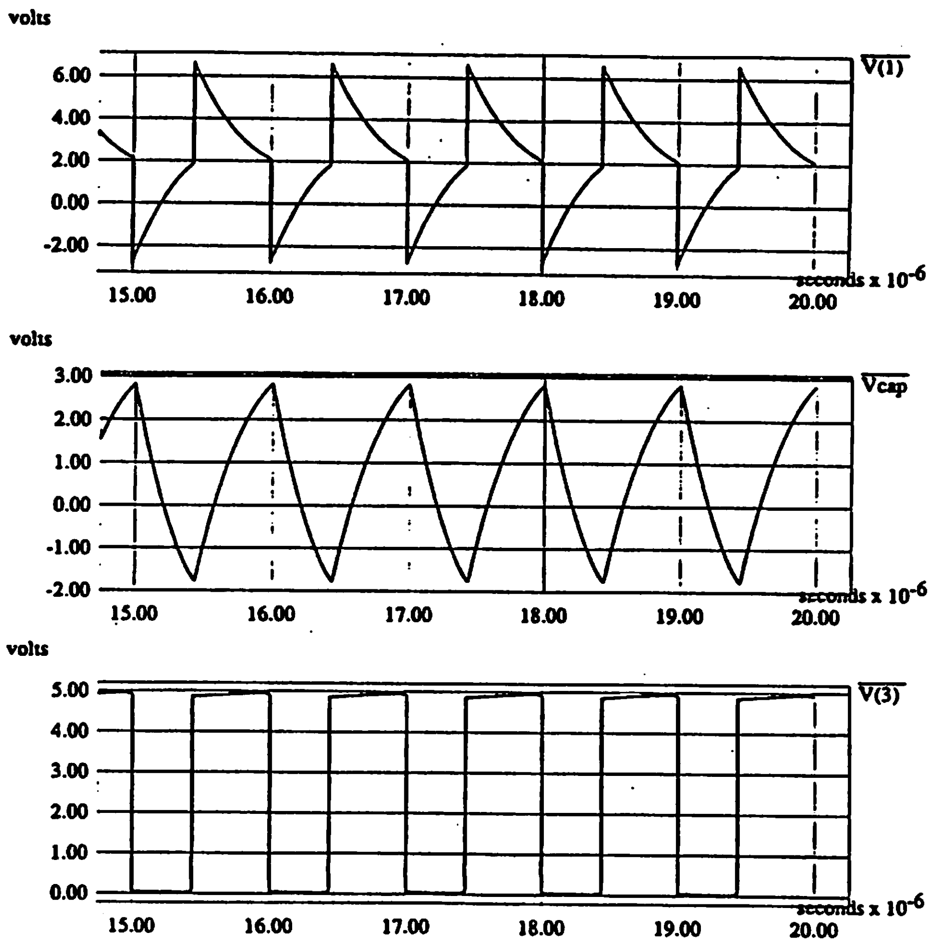


Figure 5.14: SPICE3 Input File & Output Waveforms for the CMOS Two-Inverter VCO

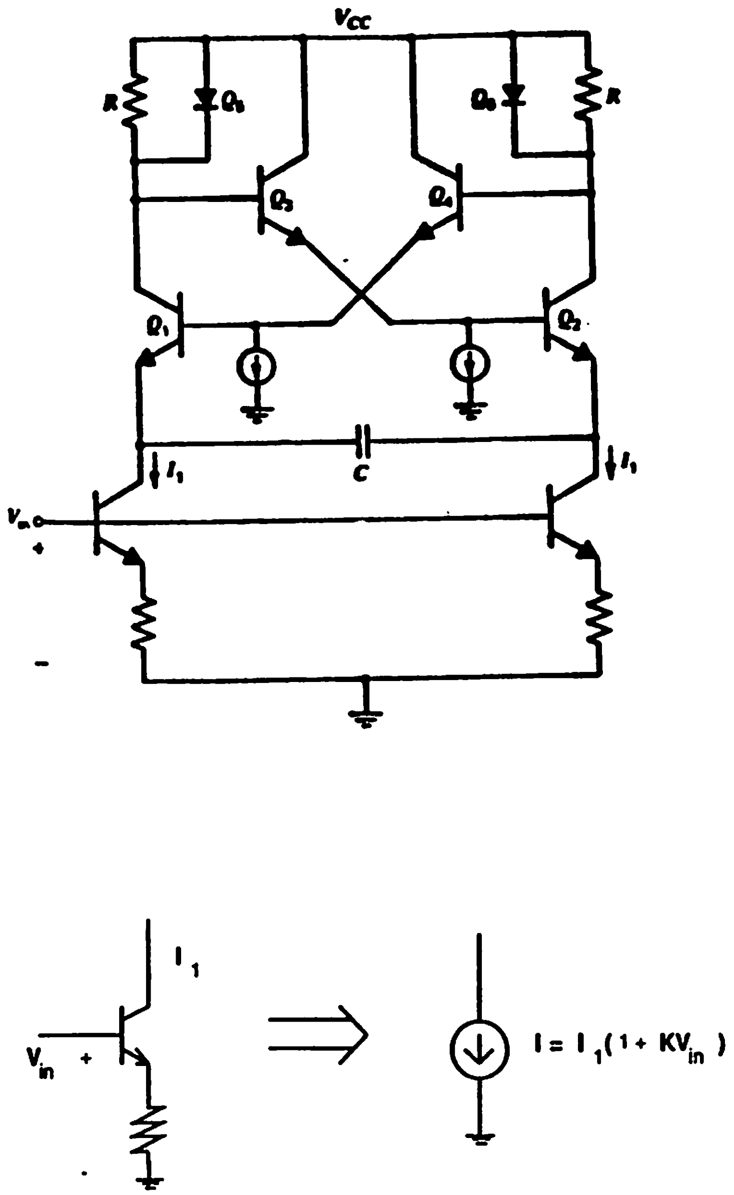
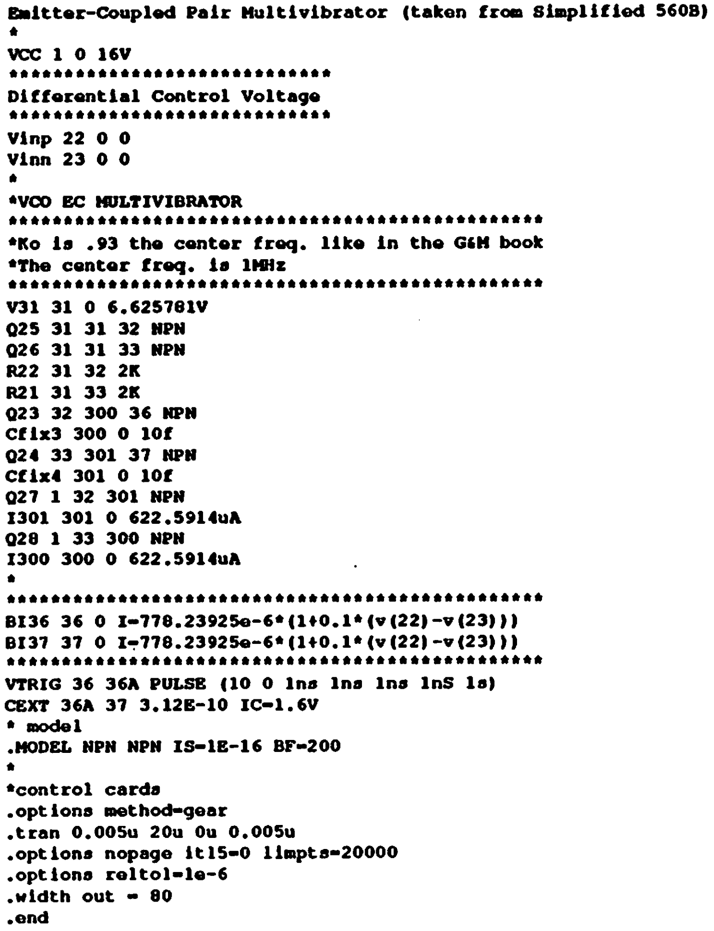
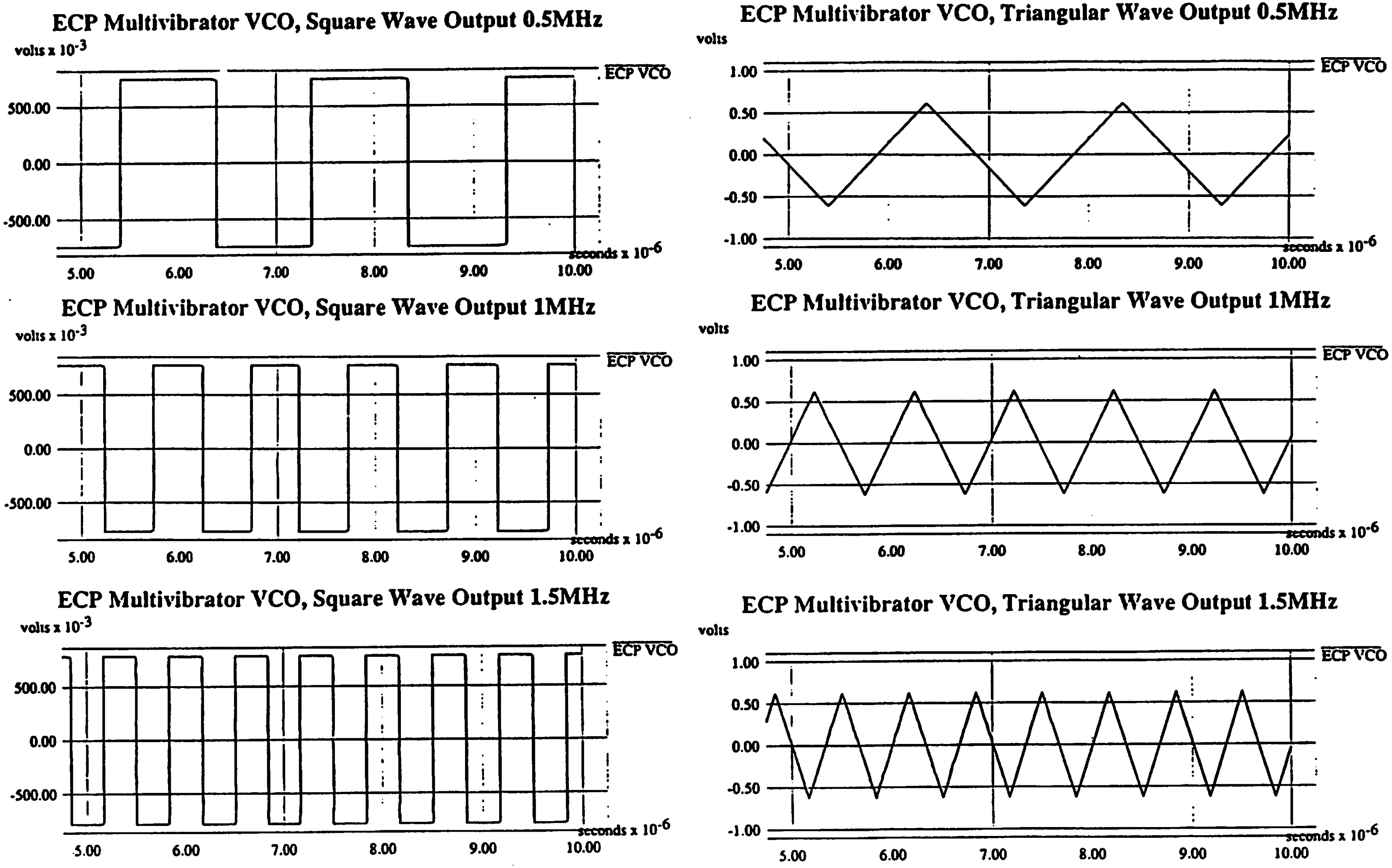


Figure 5.15: An Emitter-Coupled Pair Multivibrator as a VCO



```
Emitter-Coupled Pair Multivibrator (taken from Simplified 560B)
*
VCC 1 0 16V
******************************
Differential Control Voltage
******************************
Vinp 22 0 0
Vinn 23 0 0
*
*VCO EC MULTIVIBRATOR
**************************************************
*Ko is .93 the center freq. like in the G&M book
*The center freq. is 1MHz
**************************************************
V31 31 0 6.625781V
Q25 31 31 32 NPN
Q26 31 31 33 NPN
R22 31 32 2K
R21 31 33 2K
Q23 32 300 36 NPN
Cfix3 300 0 10f
Q24 33 301 37 NPN
Cfix4 301 0 10f
Q27 1 32 301 NPN
I301 301 0 622.5914uA
Q28 1 33 300 NPN
I300 300 0 622.5914uA
*
**************************************************
BI36 36 0 I=778.23925e-6*(1+0.1*(v(22)-v(23)))
BI37 37 0 I=778.23925e-6*(1+0.1*(v(22)-v(23)))
**************************************************
VTRIG 36 36A PULSE (10 0 1ns 1ns 1ns 1nS 1s)
CEXT 36A 37 3.12E-10 IC=1.6V
* model
.MODEL NPN NPN IS=1E-16 BF=200
*
*control cards
.options method=gear
.tran 0.005u 20u 0u 0.005u
.options nopage itl5=0 limpts=20000
.options reltol=1e-6
.width out = 80
.end
```

Figure 5.16: SPICE3 Input File for the ECP Multivibrator VCO


ECP Multivibrator VCO, Square Wave Output 0.5MHz

ECP Multivibrator VCO, Triangular Wave Output 0.5MHz

ECP Multivibrator VCO, Square Wave Output 1MHz

ECP Multivibrator VCO, Triangular Wave Output 1MHz

ECP Multivibrator VCO, Square Wave Output 1.5MHz

ECP Multivibrator VCO, Triangular Wave Output 1.5MHz

Figure 5.17: Emitter-Coupled Pair Multivibrator VCO Output Waveforms (Square & Triangular)

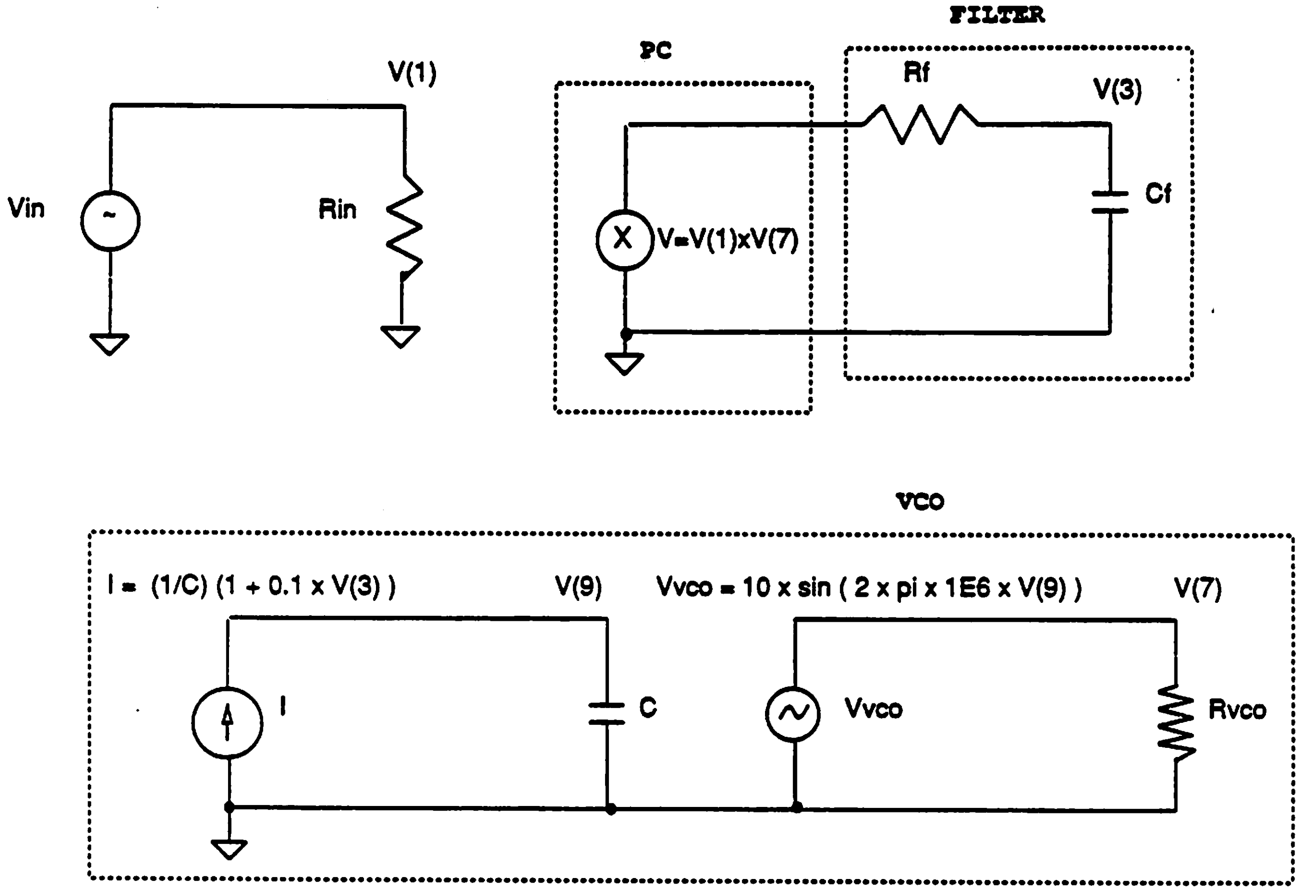

FILTER
PC
V(1)
Rf
V(3)
Vin
Rin
Cf
V=V(1)xV(7)
VCO
I = (1/C) (1 + 0.1 x V(3))
V(9)
Vvco = 10 x sin (2 x pi x 1E6 x V(9))
V(7)
I
C
Vvco
Rvco

Figure 6.1: Proposed PLL Macromodel (1)

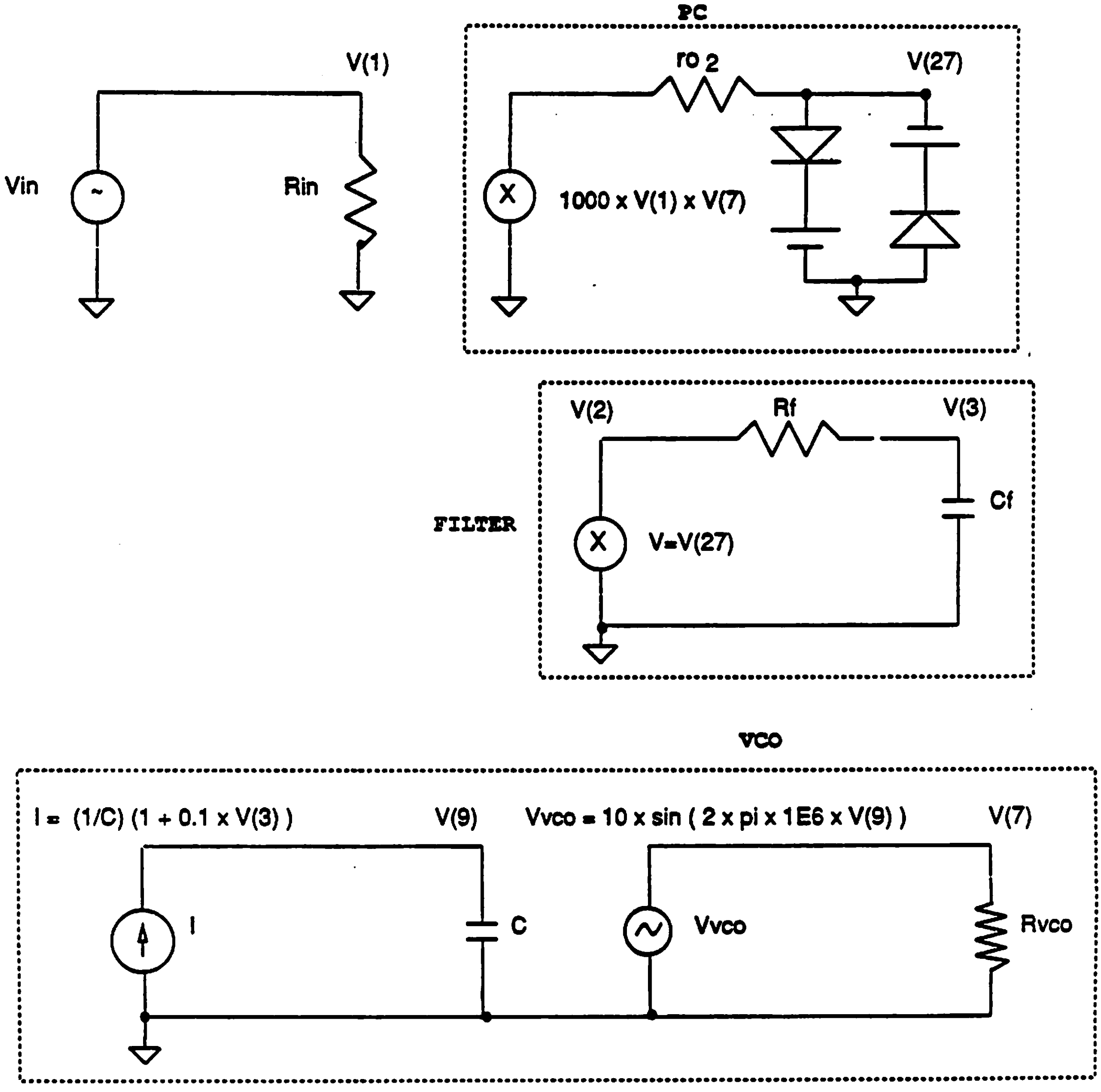


Figure 6.2: Proposed PLL Macromodel (2)

```
* Proposed PLL Macromodel(1)
* VCVS PC, simple RC filter, Integrating Sine VCO
*
* Vin: V(1); VCO: V(7) Vcontrol: V(3)
* Vin Variable Freq. Control: V(63)

* Fixed Input Frequency
*---------------------------
*vin 1 0 sin(0 1 1.125meg)
*vin 1 0 sin(0 1 1.13meg)
*vin 1 0 sin(0 1 0.860meg)
*vin 1 0 sin(0 1 0.865meg)
vin 1 0 sin(0 1 1.000meg)
rin 1 0 1k
*---------------------------

* Variable Input Frequency
*-----------------------------------------------
*vin 63 0 pwl 0 1 40u 1 240u 4.75 400u 4.75
*vin 63 0 pwl 0 1 40u 1 240u -4.75 400u -4.75
*vin 63 0 pwl 0 1 40u 1 300u -5 400u -5
*vin 63 0 pwl 0 1 40u 1 370u 5.0 400u 5.0
*rin 63 0 1k

* Integration of Vcontrol
*B1in 0 69 i = 1e-12*( 0.1*v(63) + 1)
*C1in 69 0 1p
*.ic v(69)=0

* VCO function
*B2in 1 0 v = 1 * sin(2e+6 * pi * (v(69)))
*R2in 1 0 1k
*-----------------------------------------------

* Phase comparator multiplier *
*---------------------------
* Kc = Km Vi Vosc / 2 = 5 V/rad
* emult 2 0 poly(2) 1 0 7 0   0 0 0 0 1
bmult 2 0 v=v(1)*v(7)
*---------------------------
```

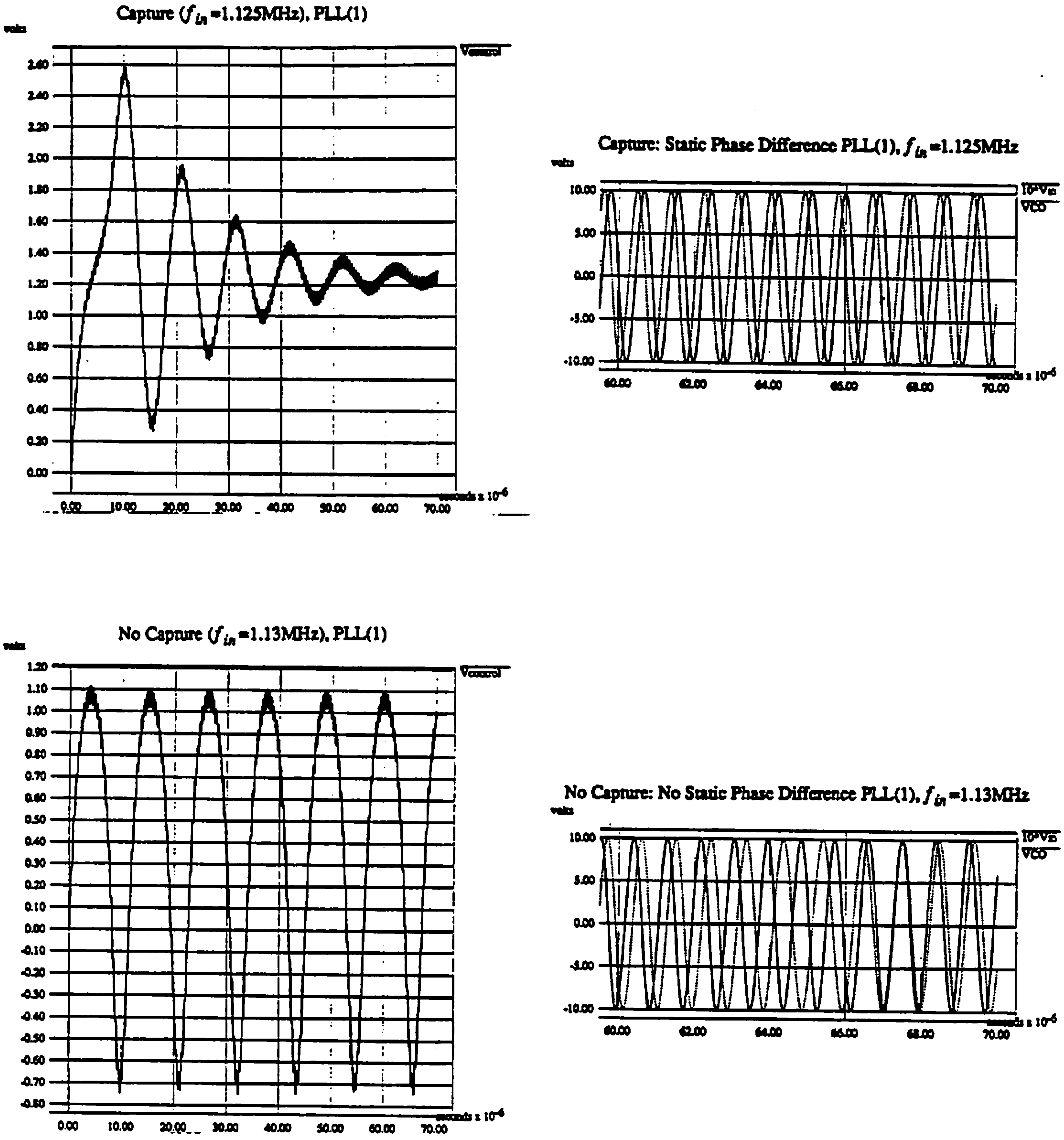
```
* Low-pass Filter *
*---------------------------
* BW=20kHz
rf 2 3 1k
cf 3 0 7.958nF
*---------------------------

* Integrating Sine VCO
*---------------------------
* Integration of Vcontrol
B1 0 79 i = 1e-12*( 0.1*v(3) + 1)
C1 79 0 1p
*.ic v(79)=0

* VCO function
B2 7 0 v = 10 * sin(2e+6 * pi * (v(79)))
R2 7 0 1k
*---------------------------
.ic v(79)=0

*ANALYSIS
*
.TRAN .07U 100U 0U 0.01U
.OPTION ACCT
.OPTIONS NOPAGE ITL5=0 LIMPTS=1000000
.OPTIONS VNTOL=0.001 ABSTOL=1.0E-6 reltol=1E-6 METHOD=GEAR
*.OPTIONS reltol=1E-6 METHOD=GEAR
*.OPTIONS reltol=1E-6
.WIDTH OUT = 80
.END
```

Figure 6.3: SPICE3 Input File for the Proposed PLL Macromodel(1)

```
*Proposed PLL Macromodel(2)

*PC output is amplified & limited, separate RC filter,
*Integrating Sine VCO
*
* Vin: V(1); VCO: V(7) Vcontrol: V(3)
* PC Output: V(27)
* Vin Variable Freq. Control: V(63)


* Variable Input Frequency
*-------------------------------------------------
*Vvar 63 0 1 pwl 0 -1 40u -1 280u -8.25 300u -8.25
*Vvar 63 0 1 pwl 0 1 40u 1 280u 9.5 300u 9.5
*Rvar 63 0 1K

* Integration of Vcontrol
*Bvar1 0 69 i = 1e-12*( 0.1*v(63) + 1)
*Cvar1 69 0 1p
*.ic v(69)=0

* Vin function
*Bvar2 1 0 v = 1 * sin(2e+6 * pi * (v(69)))
*Rvar2 1 0 1k
*-------------------------------------------------

* Fixed Input Frequency
*---------------------------
*vin 1 0 sin(0 1 1.16meg)
*vin 1 0 sin(0 1 1.165meg)
*vin 1 0 sin(0 1 0.825meg)
vin 1 0 sin(0 1 1.000meg)
rin 1 0 1k
*---------------------------

* Phase comparator multiplier *
*---------------------------
* Kc = Km Vi Vosc / 2 = 5 V/rad
* emult 2 0 poly(2) 1 0 7 0   0 0 0 0 1
bpd 23 0 v=1000*v(1)*v(7)
ropd 23 27 0.001
d1pd 27 28 md2
v1pd 28 0 8.66v
```

```
d2pd 0 29 md2
v2pd 29 27 8.66v
.model md2 d is=1e-16
*
bpdout 2 0 v=v(27)
*---------------------------

* Low-pass Filter *
*---------------------------
* BW=20kHz
rf 2 3 1k
cf 3 0 7.958nF
*---------------------------

* Integrating Sine VCO
*---------------------------
* Integration of Vcontrol
B1 0 79 i = 1e-12*( 0.1*v(3) + 1)
C1 79 0 1p
*.ic v(79)=0

* VCO function
B2 7 0 v = 10 * sin(2e+6 * pi * (v(79)))
R2 7 0 1k
*---------------------------
.ic v(79)=0

*ANALYSIS
*
.TRAN .07U 100U 0U 0.01U
*.TRAN .01U 70U 0U 0.01U
.OPTION ACCT
.OPTIONS NOPAGE ITL5=0 LIMPTS=1000000
.OPTIONS VNTOL=0.001 ABSTOL=1.0E-6 reltol=1E-6 METHOD=GEAR
*.OPTIONS reltol=1E-6 METHOD=GEAR
*.OPTIONS reltol=1E-6
.WIDTH OUT = 80
.END
```

Figure 6.4: SPICE3 Input File for the Proposed PLL Macromodel(2)



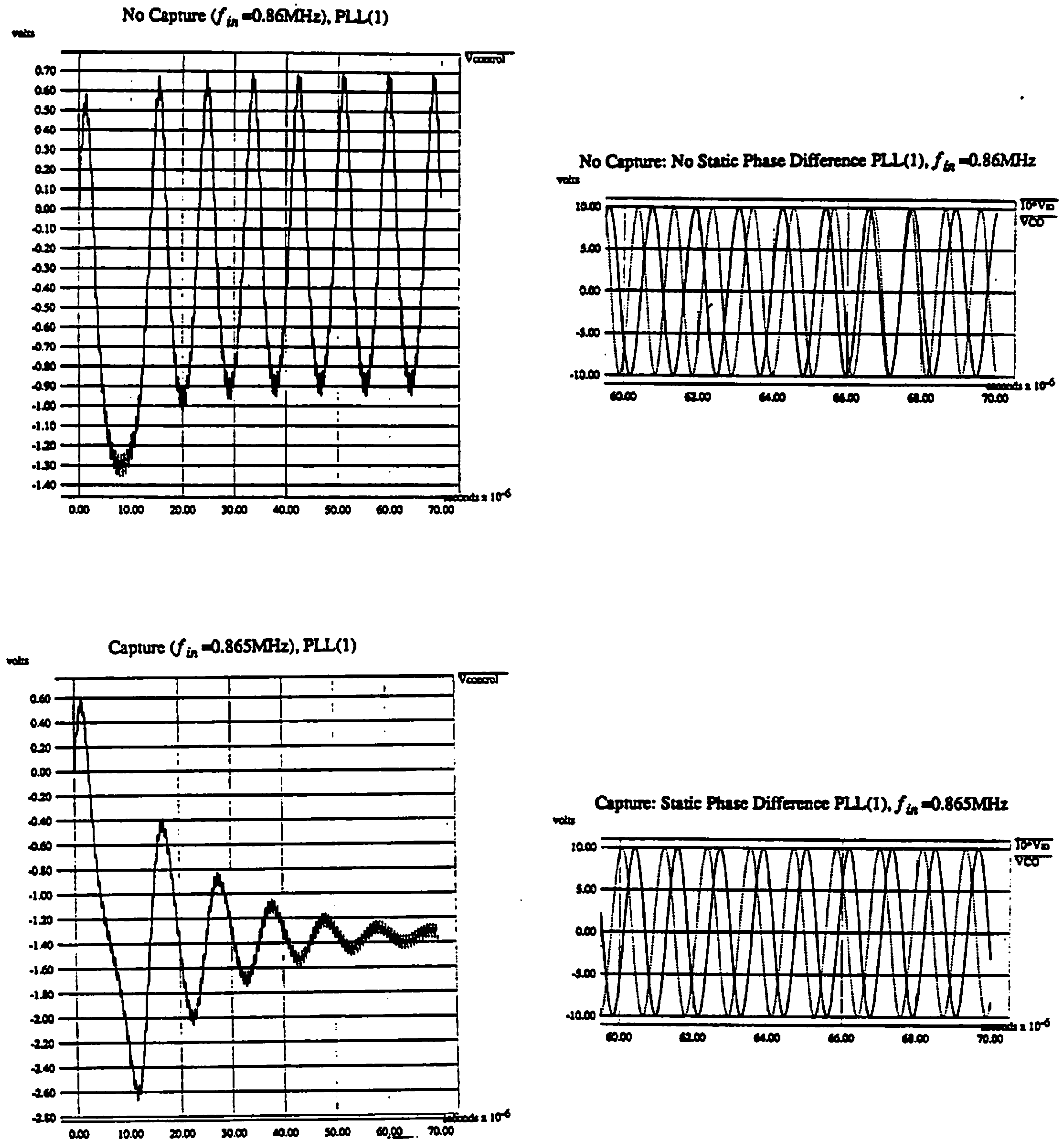
Capture (f_{in} =1.125MHz), PLL(1)

Capture: Static Phase Difference PLL(1), f_{in} =1.125MHz

No Capture (f_{in} =1.13MHz), PLL(1)

No Capture: No Static Phase Difference PLL(1), f_{in} =1.13MHz

Figure 6.5: Upper Capture Range (1.125MHz,1.13MHz) for PLL(1)



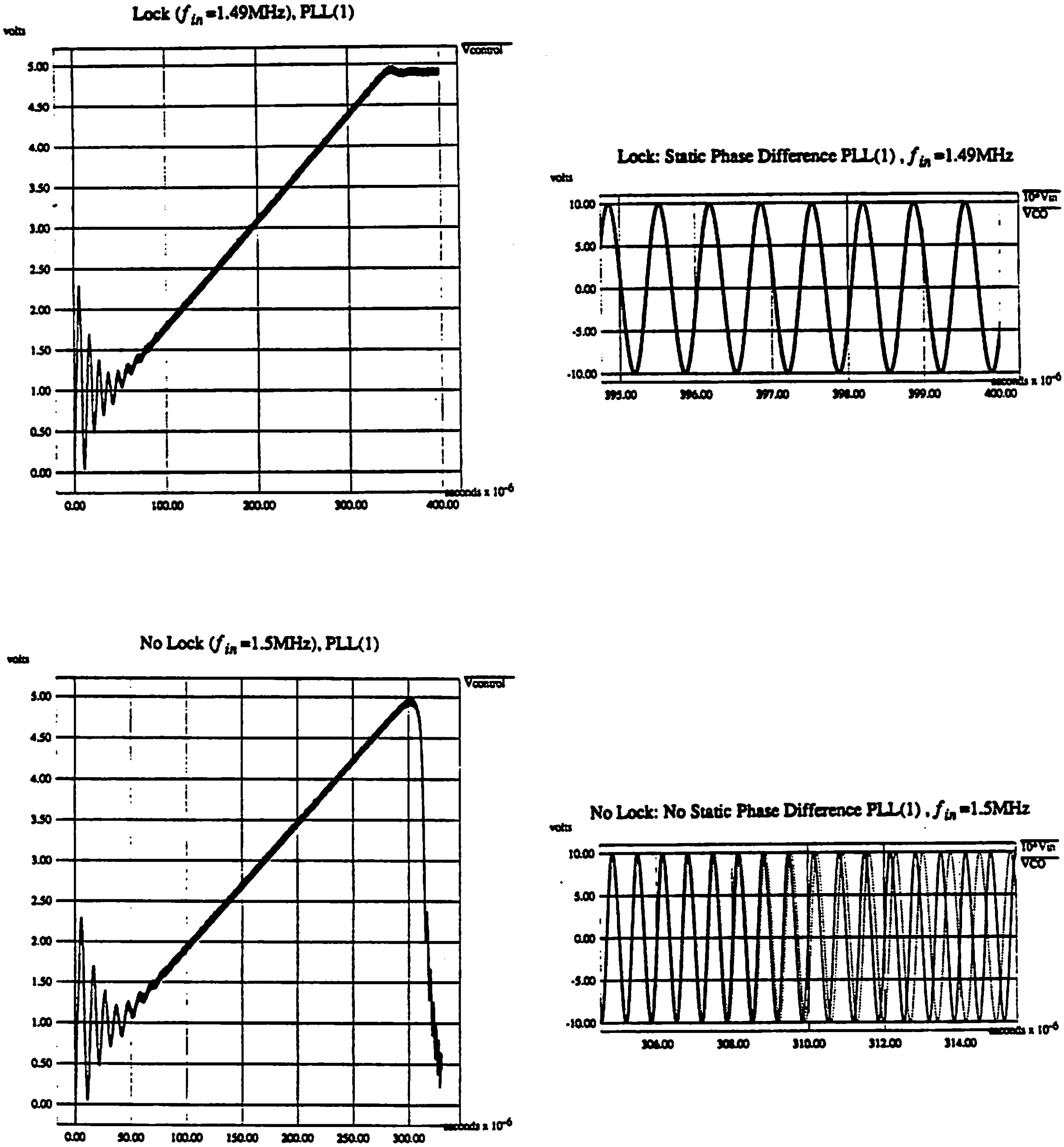
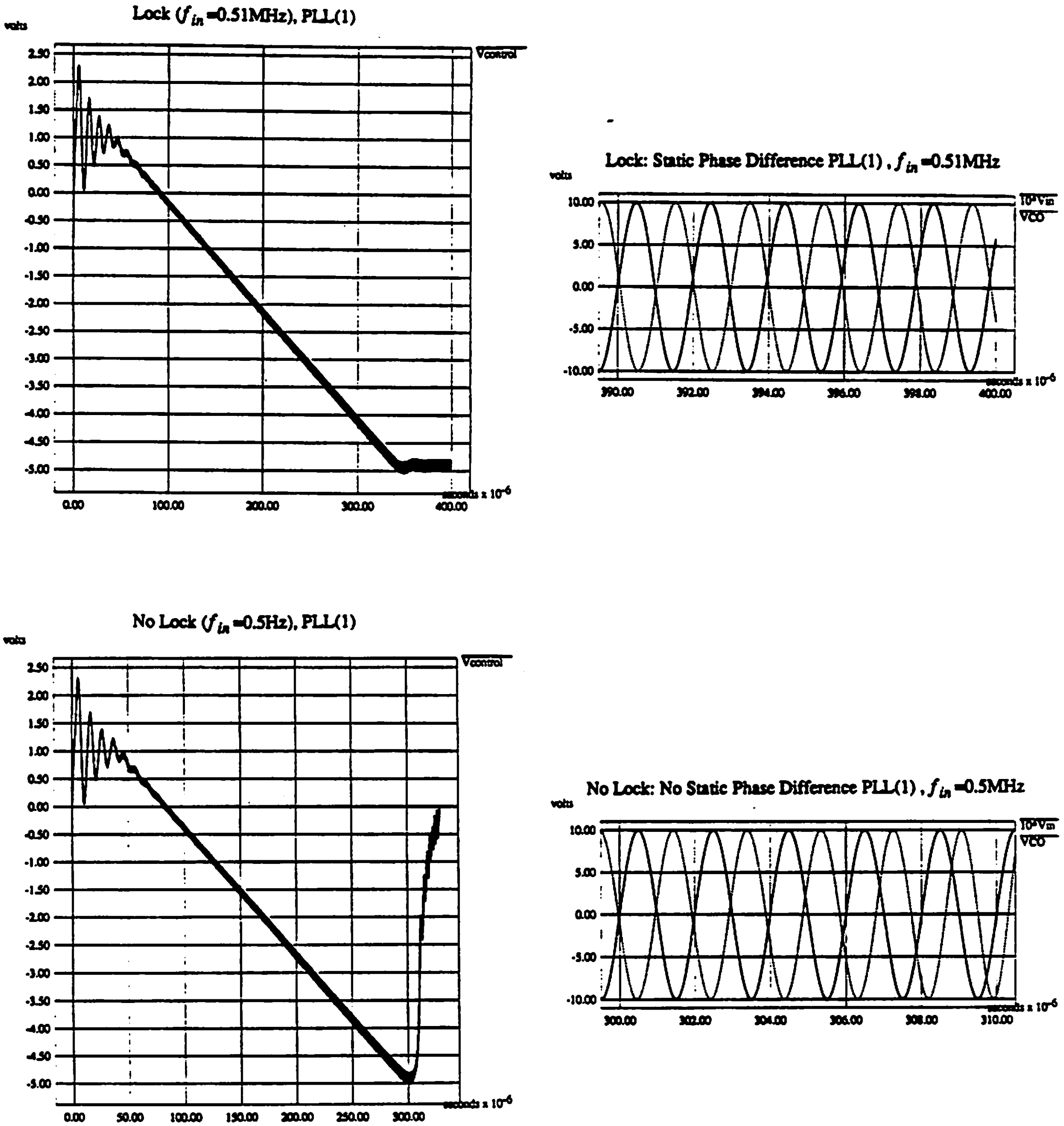


Figure 6.7: Upper Lock Range (1.49MHz,1.5MHz) for PLL(1)



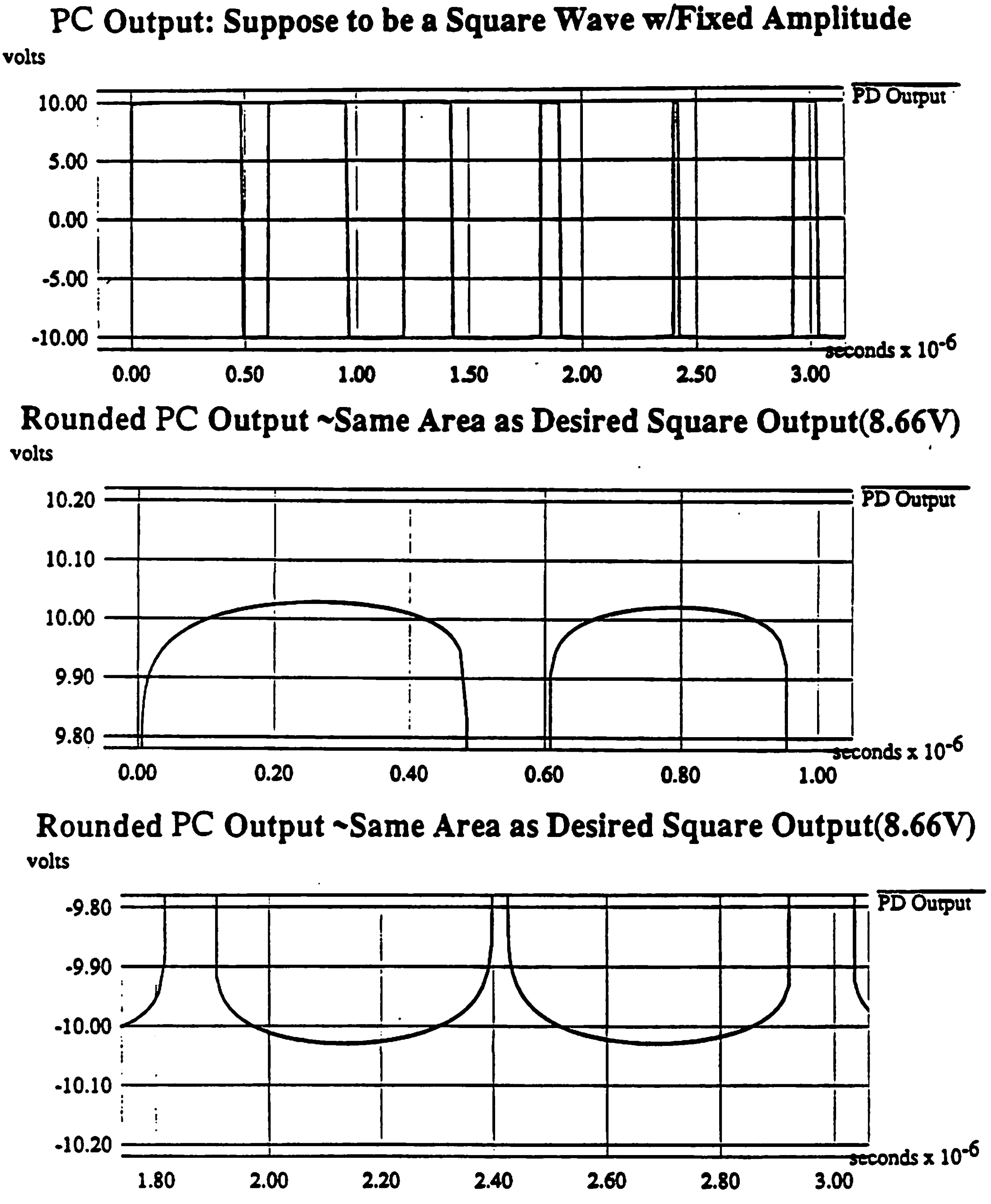
Lock (f_{in}=0.51MHz), PLL(1)

Lock: Static Phase Difference PLL(1), f_{in}=0.51MHz

No Lock (f_{in}=0.5Hz), PLL(1)

No Lock: No Static Phase Difference PLL(1), f_{in}=0.5MHz

Figure 6.8: Lower Lock Range (0.51MHz,0.5MHz) for PLL(1)

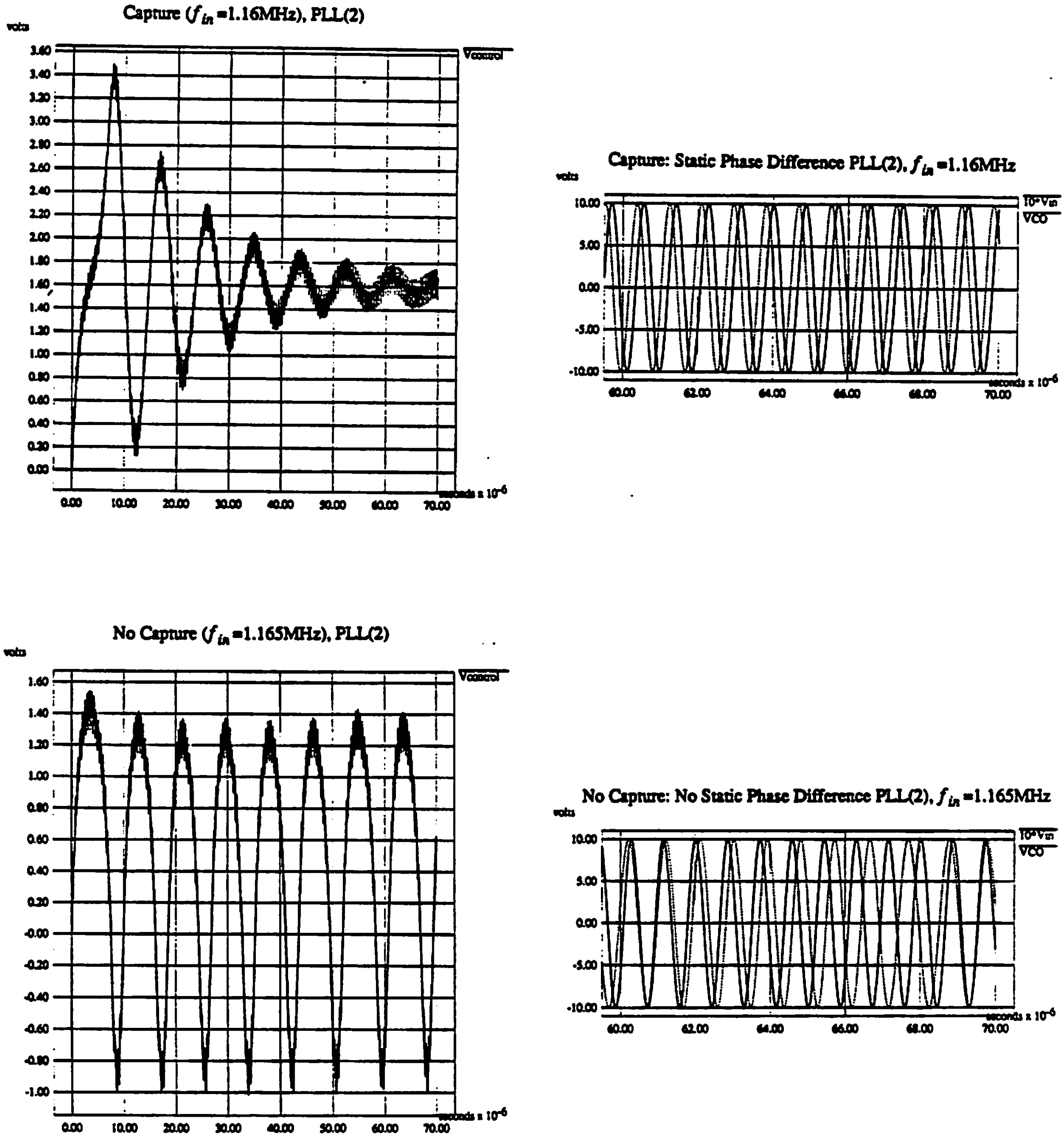


PC Output: Suppose to be a Square Wave w/Fixed Amplitude

Rounded PC Output ~Same Area as Desired Square Output(8.66V)

Rounded PC Output ~Same Area as Desired Square Output(8.66V)

Figure 6.9: Rounded PC Output has ~Same Area as Desired Square Wave



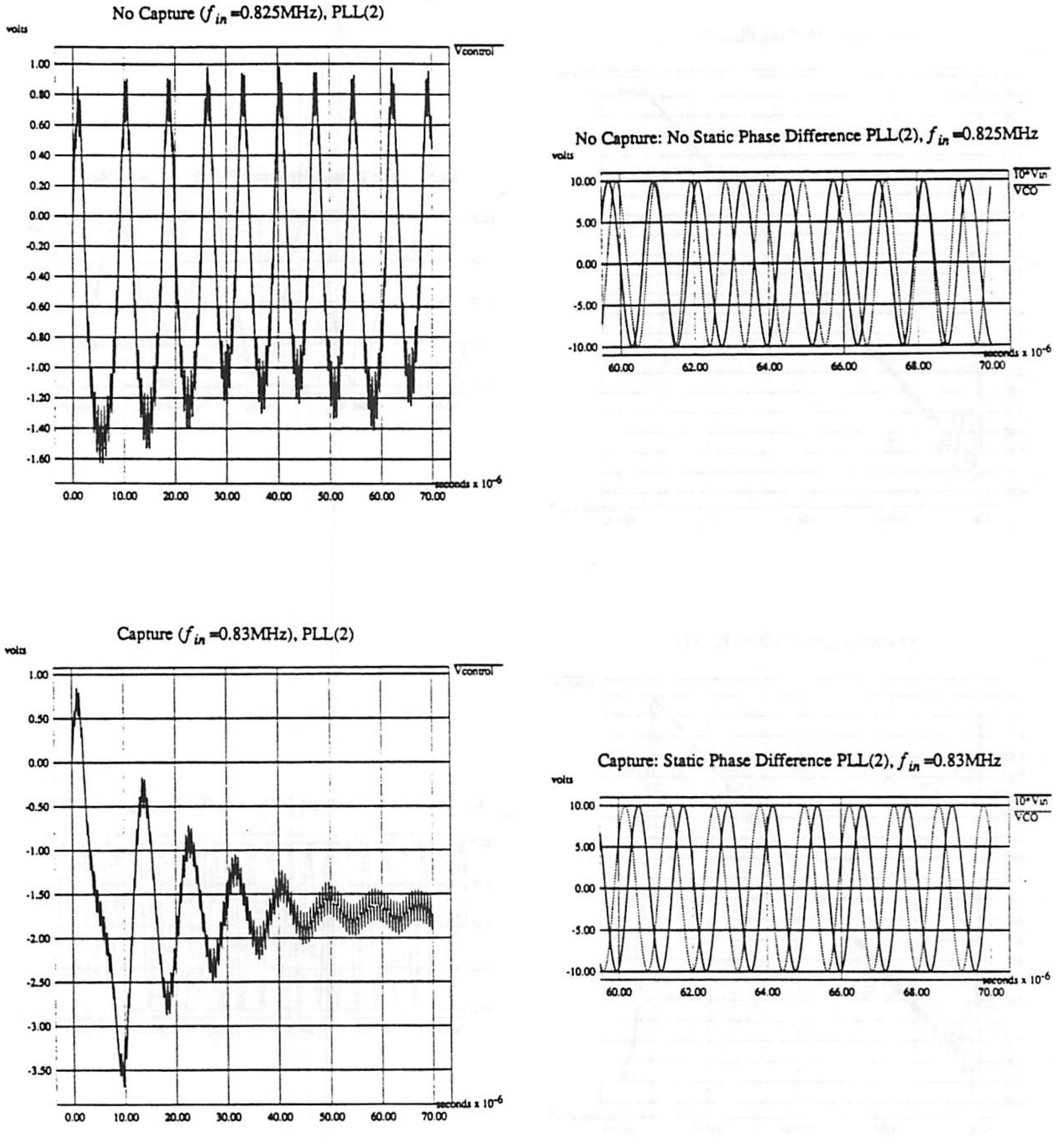
Capture (f_{in} =1.16MHz), PLL(2)

Capture: Static Phase Difference PLL(2), f_{in} =1.16MHz

No Capture (f_{in} =1.165MHz), PLL(2)

No Capture: No Static Phase Difference PLL(2), f_{in} =1.165MHz

Figure 6.10: Upper Capture Range (1.16MHz,1.165MHz) for PLL(2)



No Capture (f_{in} =0.825MHz), PLL(2)

No Capture: No Static Phase Difference PLL(2), f_{in} =0.825MHz

Capture (f_{in} =0.83MHz), PLL(2)

Capture: Static Phase Difference PLL(2), f_{in} =0.83MHz

Figure 6.11: Lower Capture Range (0.825MHz,0.83MHz) for PLL(2)

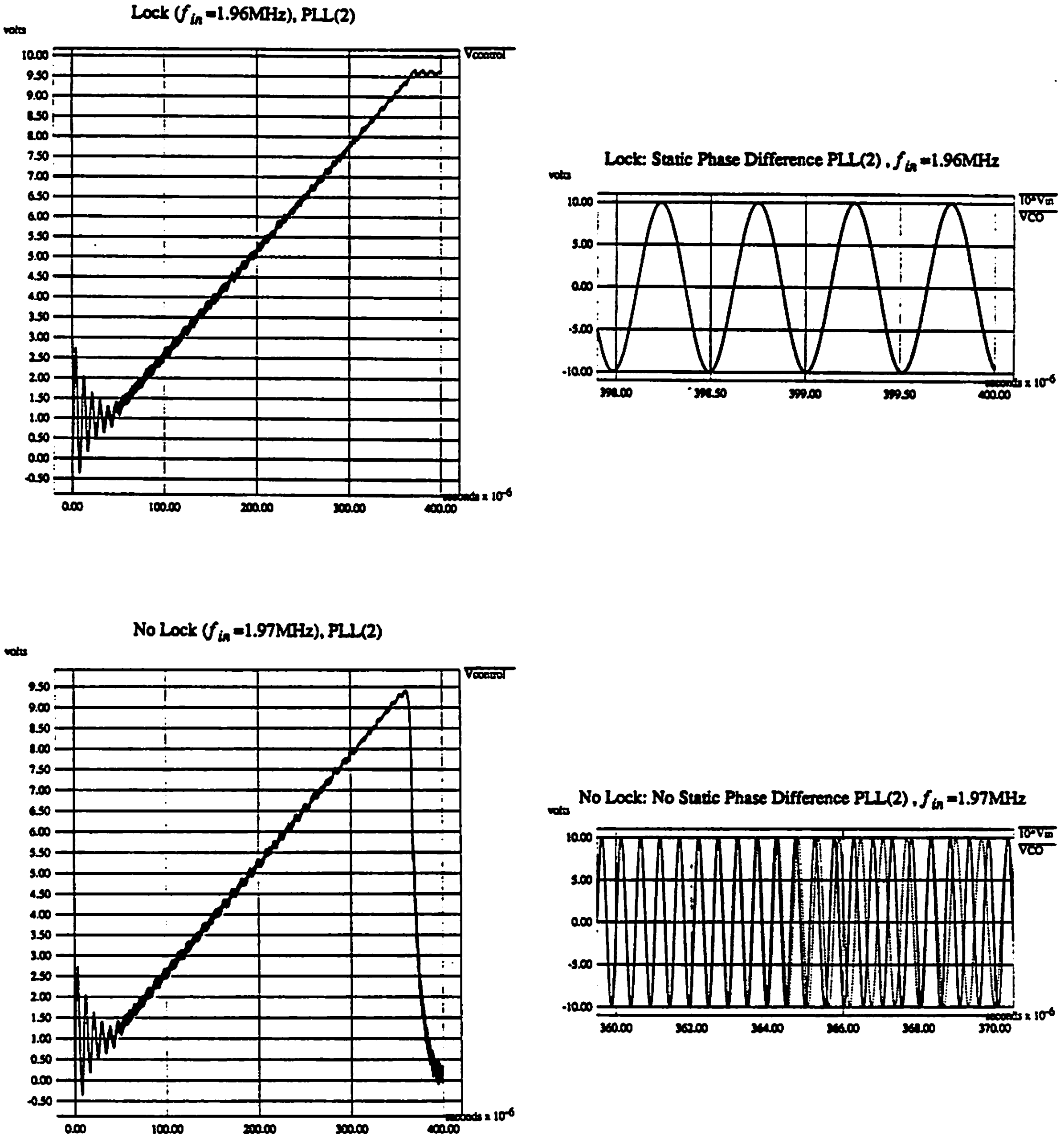
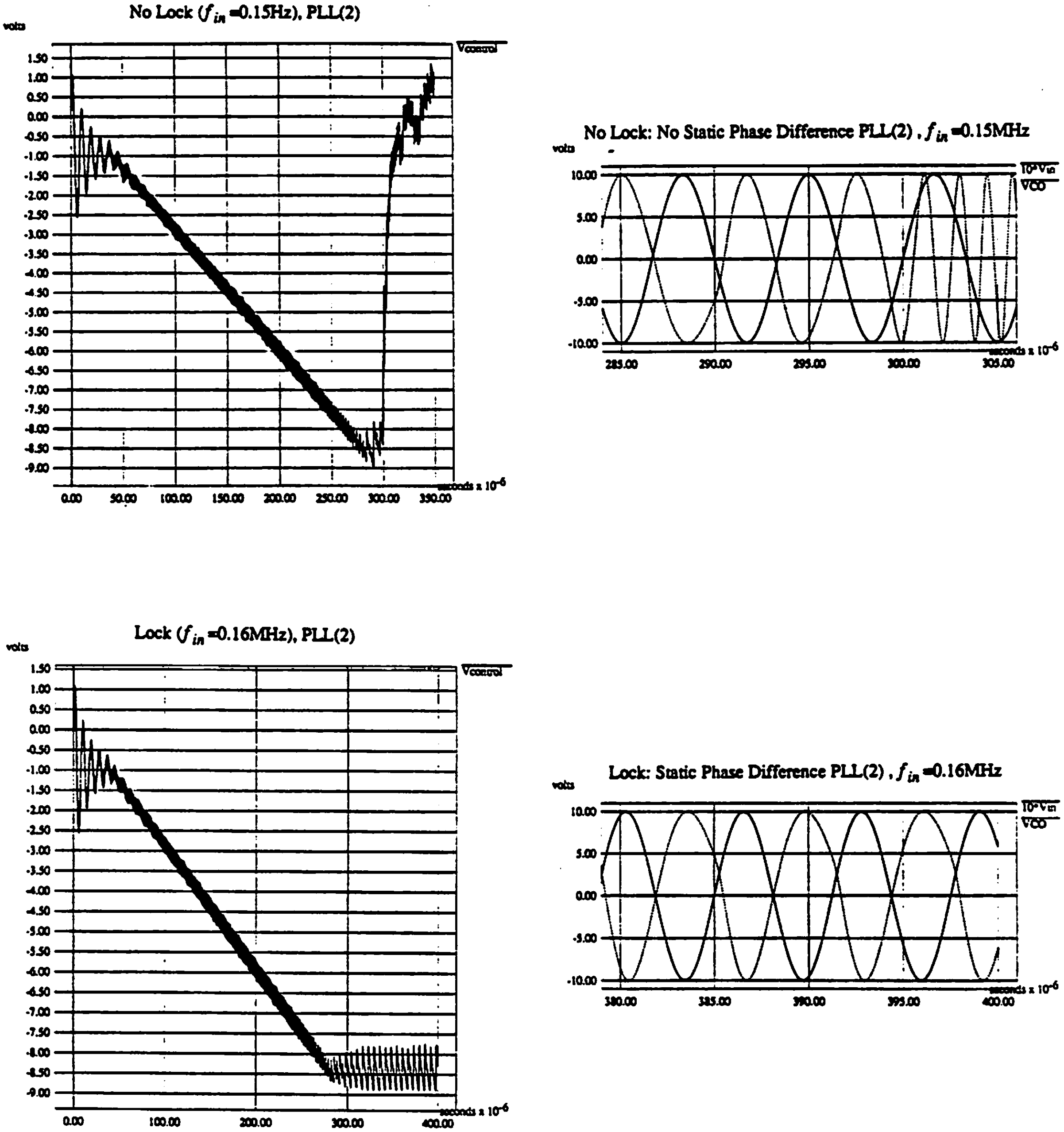


Figure 6.12: Upper Lock Range (1.96MHz,1.97MHz) for PLL(2)



No Lock (f_{in} =0.15Hz), PLL(2)

No Lock: No Static Phase Difference PLL(2) , f_{in} =0.15MHz

Lock (f_{in} =0.16MHz), PLL(2)

Lock: Static Phase Difference PLL(2) , f_{in} =0.16MHz

Figure 6.13: Lower Lock Range (0.15MHz,0.16MHz) for PLL(2)

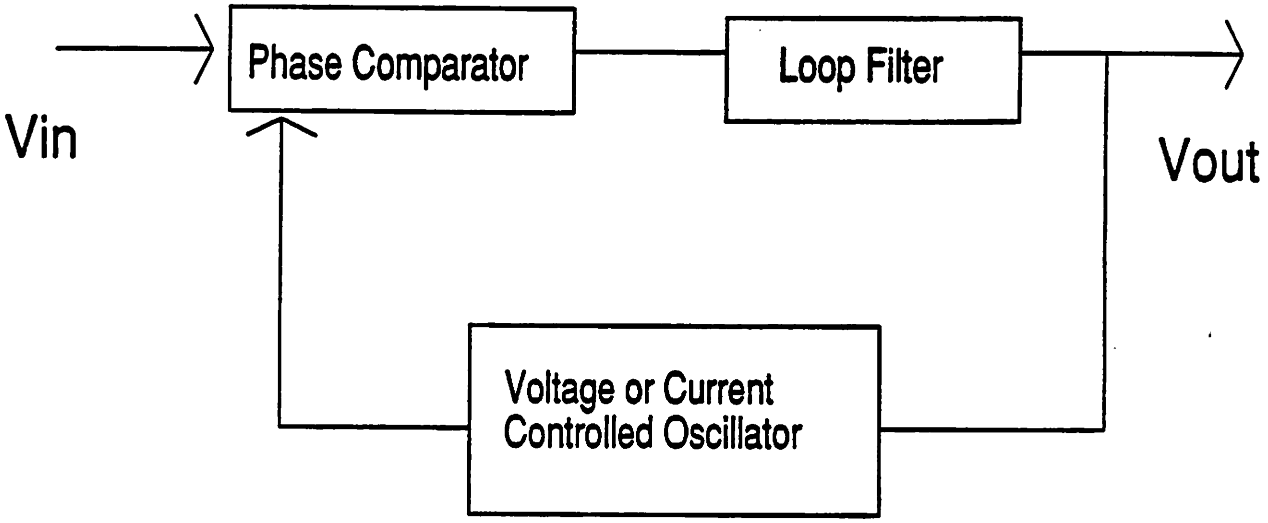


Figure A.1: The Basic PLL Configuration

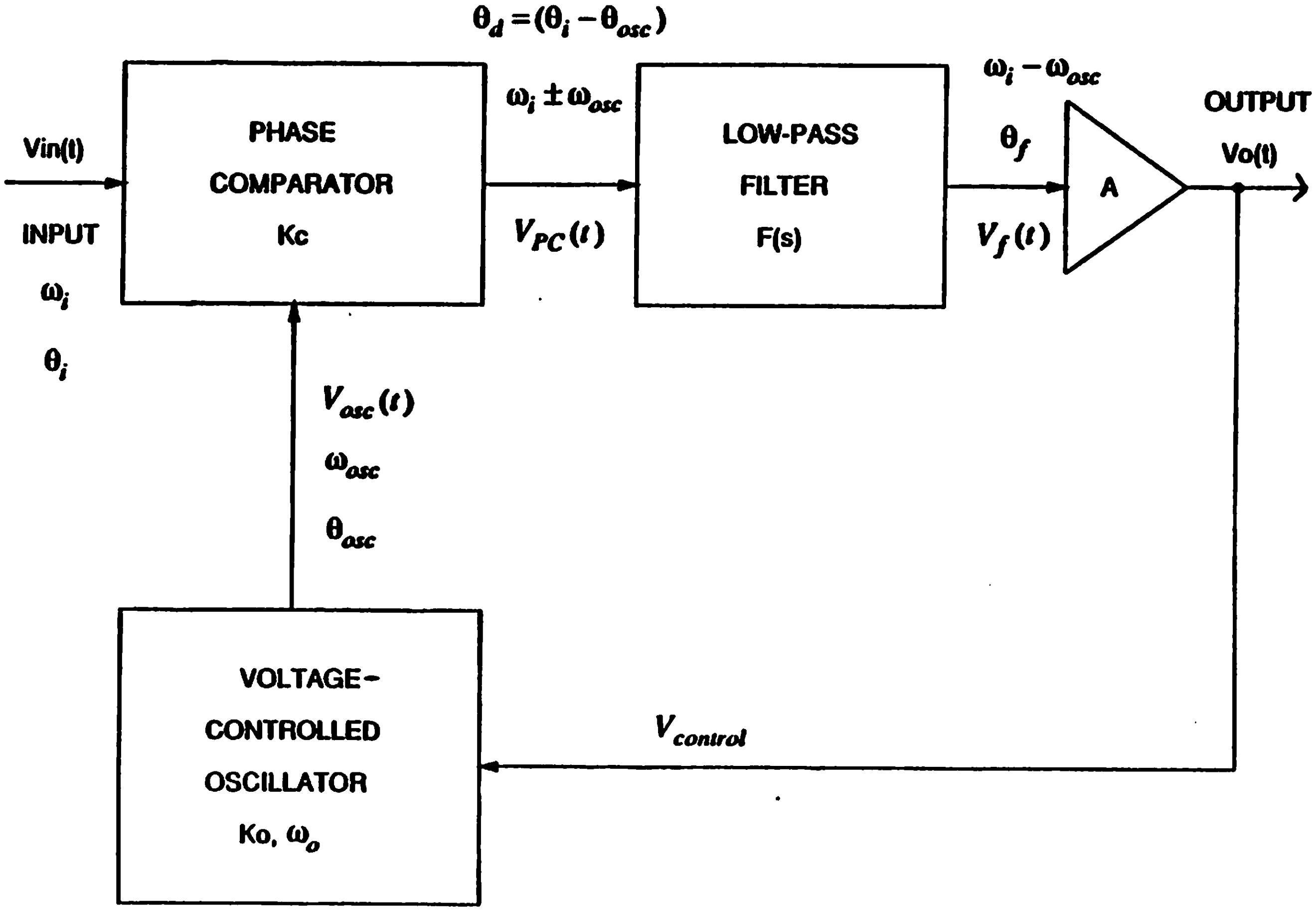


Figure A.2: A Block Diagram of a PLL

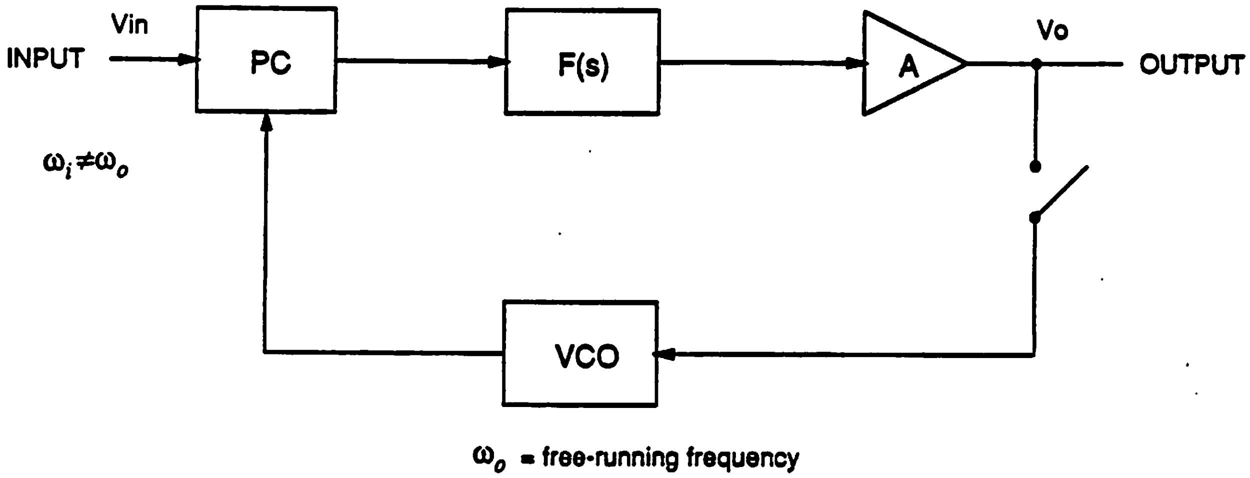


Figure A.3: A PLL with Loop Open; Input Frequency Different from VCO Free-Running Frequency

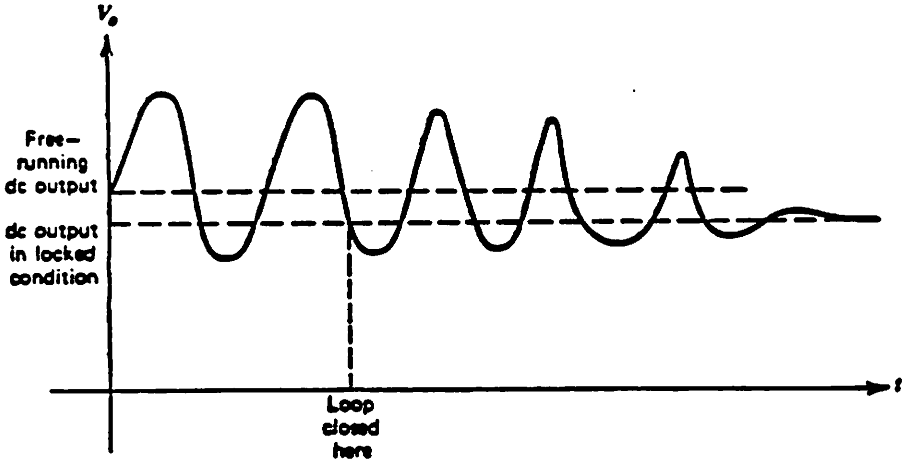
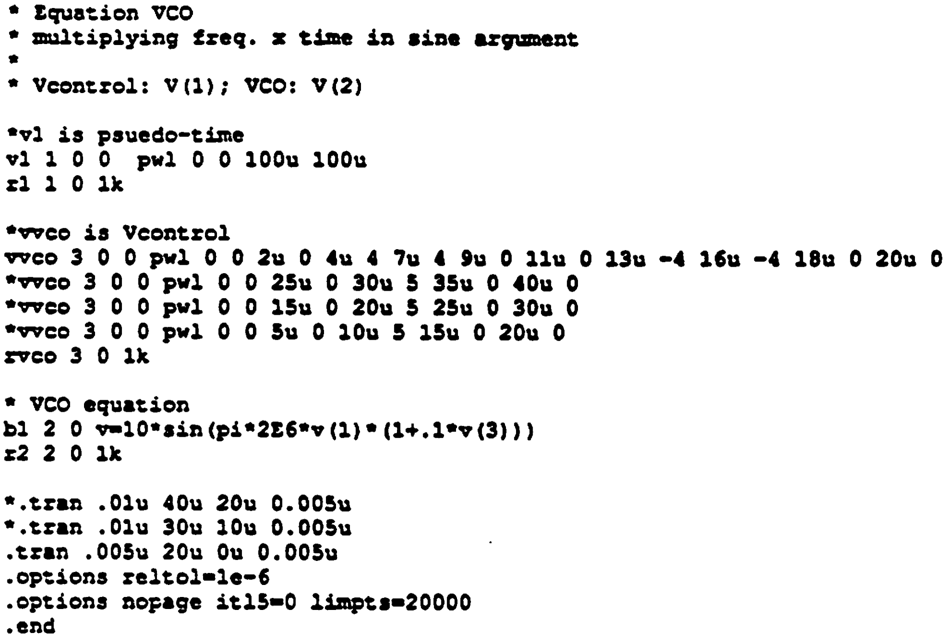


Figure A.4: The Non-Linear Behavior of the Capture Process



```
* Equation VCO
* multiplying freq. x time in sine argument
*
* Vcontrol: V(1); VCO: V(2)

*v1 is psuedo-time
v1 1 0 0  pwl 0 0 100u 100u
r1 1 0 1k

*vvco is Vcontrol
vvco 3 0 0 pwl 0 0 2u 0 4u 4 7u 4 9u 0 11u 0 13u -4 16u -4 18u 0 20u 0
*vvco 3 0 0 pwl 0 0 25u 0 30u 5 35u 0 40u 0
*vvco 3 0 0 pwl 0 0 15u 0 20u 5 25u 0 30u 0
*vvco 3 0 0 pwl 0 0 5u 0 10u 5 15u 0 20u 0
rvco 3 0 1k

* VCO equation
b1 2 0 v=10*sin(pi*2E6*v(1)*(1+.1*v(3)))
r2 2 0 1k

*.tran .01u 40u 20u 0.005u
*.tran .01u 30u 10u 0.005u
.tran .005u 20u 0u 0.005u
.options reltol=1e-6
.options nopage itl5=0 limpts=20000
.end
```

Figure C.1: SPICE3 Input File for the Equation VCO

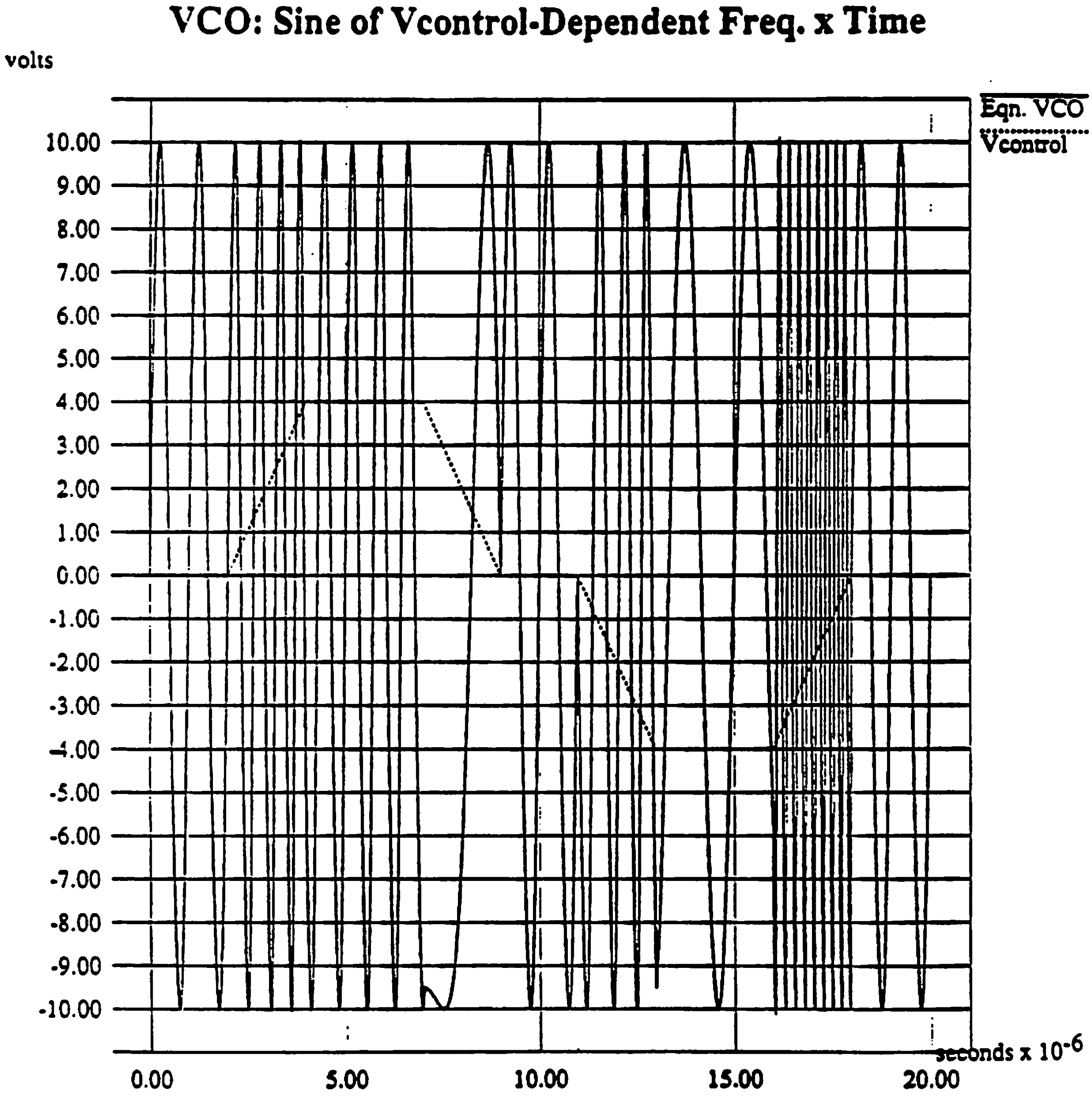


Figure C.2: Equation VCO Output Waveform for a Changing Control Voltage

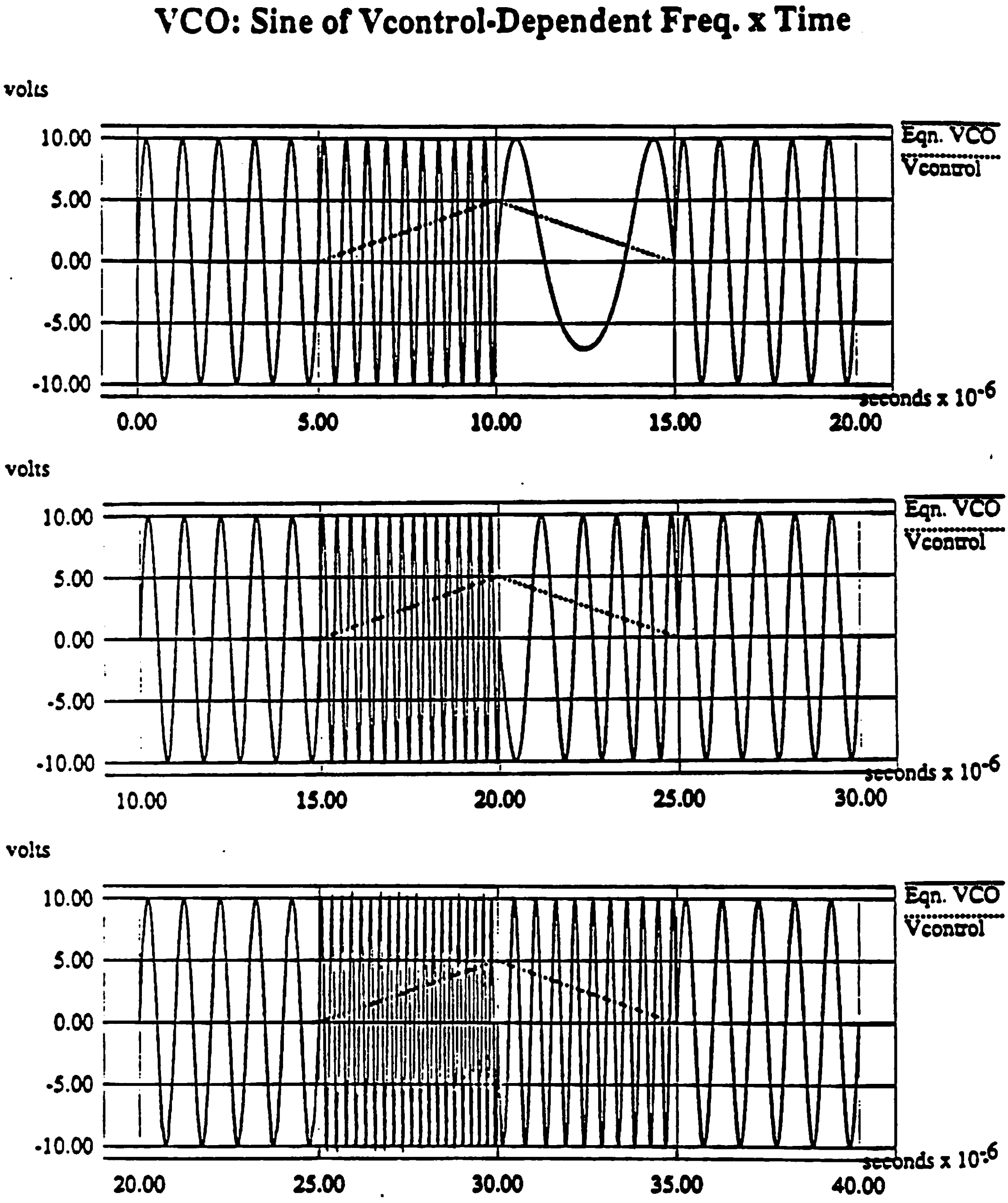
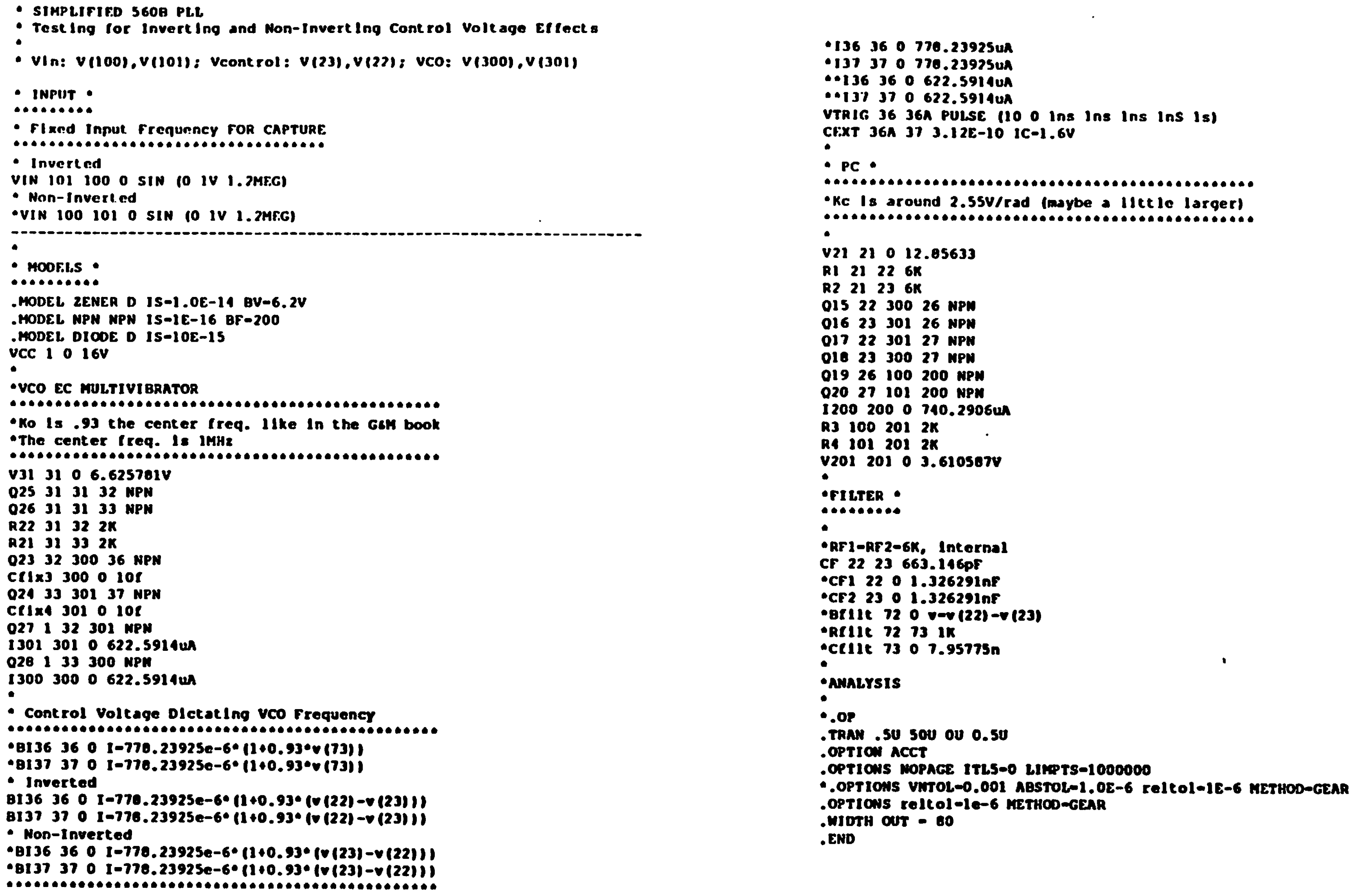


Figure C.3: Equation VCO Output Waveform for Delayed Versions of Changing Control Voltage



```
* SIMPLIFIED 560B PLL
* Testing for Inverting and Non-Inverting Control Voltage Effects
*
* Vin: V(100),V(101); Vcontrol: V(23),V(22); VCO: V(300),V(301)

* INPUT *
*********
* Fixed Input Frequency FOR CAPTURE
***********************************
* Inverted
VIN 101 100 0 SIN (0 1V 1.2MEG)
* Non-Inverted
*VIN 100 101 0 SIN (0 1V 1.2MEG)
----------------------------------------------------------------------
*
* MODELS *
**********
.MODEL ZENER D IS=1.0E-14 BV=6.2V
.MODEL NPN NPN IS=1E-16 BF=200
.MODEL DIODE D IS=10E-15
VCC 1 0 16V
*
*VCO EC MULTIVIBRATOR
************************************************
*Ko is .93 the center freq. like in the G&M book
*The center freq. is 1MHz
************************************************
V31 31 0 6.625781V
Q25 31 31 32 NPN
Q26 31 31 33 NPN
R22 31 32 2K
R21 31 33 2K
Q23 32 300 36 NPN
Cfix3 300 0 10f
Q24 33 301 37 NPN
Cfix4 301 0 10f
Q27 1 32 301 NPN
I301 301 0 622.5914uA
Q28 1 33 300 NPN
I300 300 0 622.5914uA
*
* Control Voltage Dictating VCO Frequency
************************************************
*BI36 36 0 I=778.23925e-6*(1+0.93*v(73))
*BI37 37 0 I=778.23925e-6*(1+0.93*v(73))
* Inverted
BI36 36 0 I=778.23925e-6*(1+0.93*(v(22)-v(23)))
BI37 37 0 I=778.23925e-6*(1+0.93*(v(22)-v(23)))
* Non-Inverted
*BI36 36 0 I=778.23925e-6*(1+0.93*(v(23)-v(22)))
*BI37 37 0 I=778.23925e-6*(1+0.93*(v(23)-v(22)))
************************************************
```

```
*I36 36 0 778.23925uA
*I37 37 0 778.23925uA
**I36 36 0 622.5914uA
**I37 37 0 622.5914uA
VTRIG 36 36A PULSE (10 0 1ns 1ns 1ns 1nS 1s)
CEXT 36A 37 3.12E-10 IC=1.6V
*
* PC *
************************************************
*Kc is around 2.55V/rad (maybe a little larger)
************************************************
*
V21 21 0 12.85633
R1 21 22 6K
R2 21 23 6K
Q15 22 300 26 NPN
Q16 23 301 26 NPN
Q17 22 301 27 NPN
Q18 23 300 27 NPN
Q19 26 100 200 NPN
Q20 27 101 200 NPN
I200 200 0 740.2906uA
R3 100 201 2K
R4 101 201 2K
V201 201 0 3.610587V
*
*FILTER *
*********
*
*RF1=RF2=6K, internal
CF 22 23 663.146pF
*CF1 22 0 1.326291nF
*CF2 23 0 1.326291nF
*Bfilt 72 0 v=v(22)-v(23)
*Rfilt 72 73 1K
*Cfilt 73 0 7.95775n
*
*ANALYSIS
*
*.OP
.TRAN .5U 50U 0U 0.5U
.OPTION ACCT
.OPTIONS NOPAGE ITL5=0 LIMPTS=1000000
*.OPTIONS VNTOL=0.001 ABSTOL=1.0E-6 reltol=1E-6 METHOD=GEAR
.OPTIONS reltol=1e-6 METHOD=GEAR
.WIDTH OUT = 80
.END
```

Figure D.1: SPICE3 Input File for the Simplified 560B PLL with both Inverting and Non-Inverting Control Voltage Connections

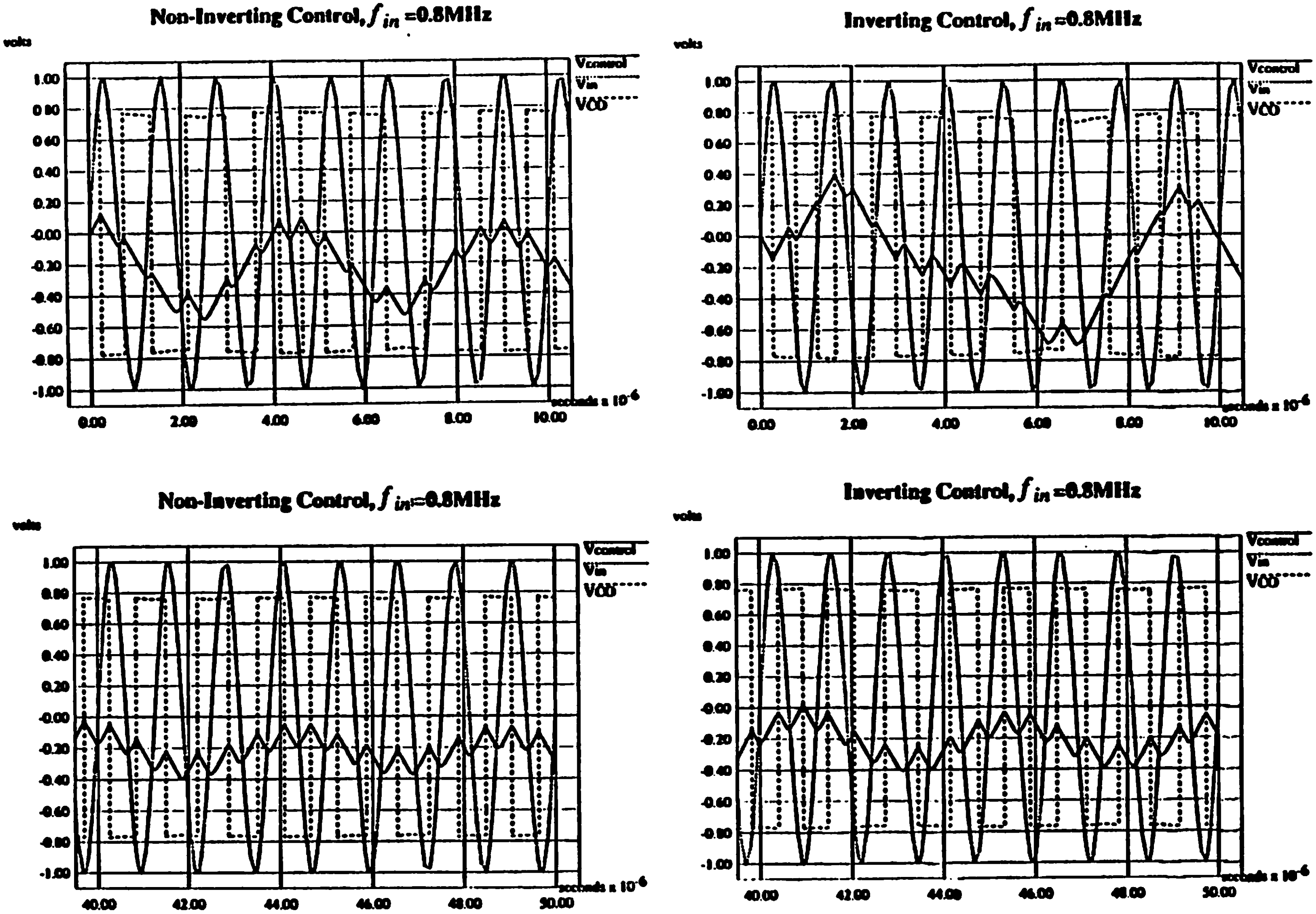
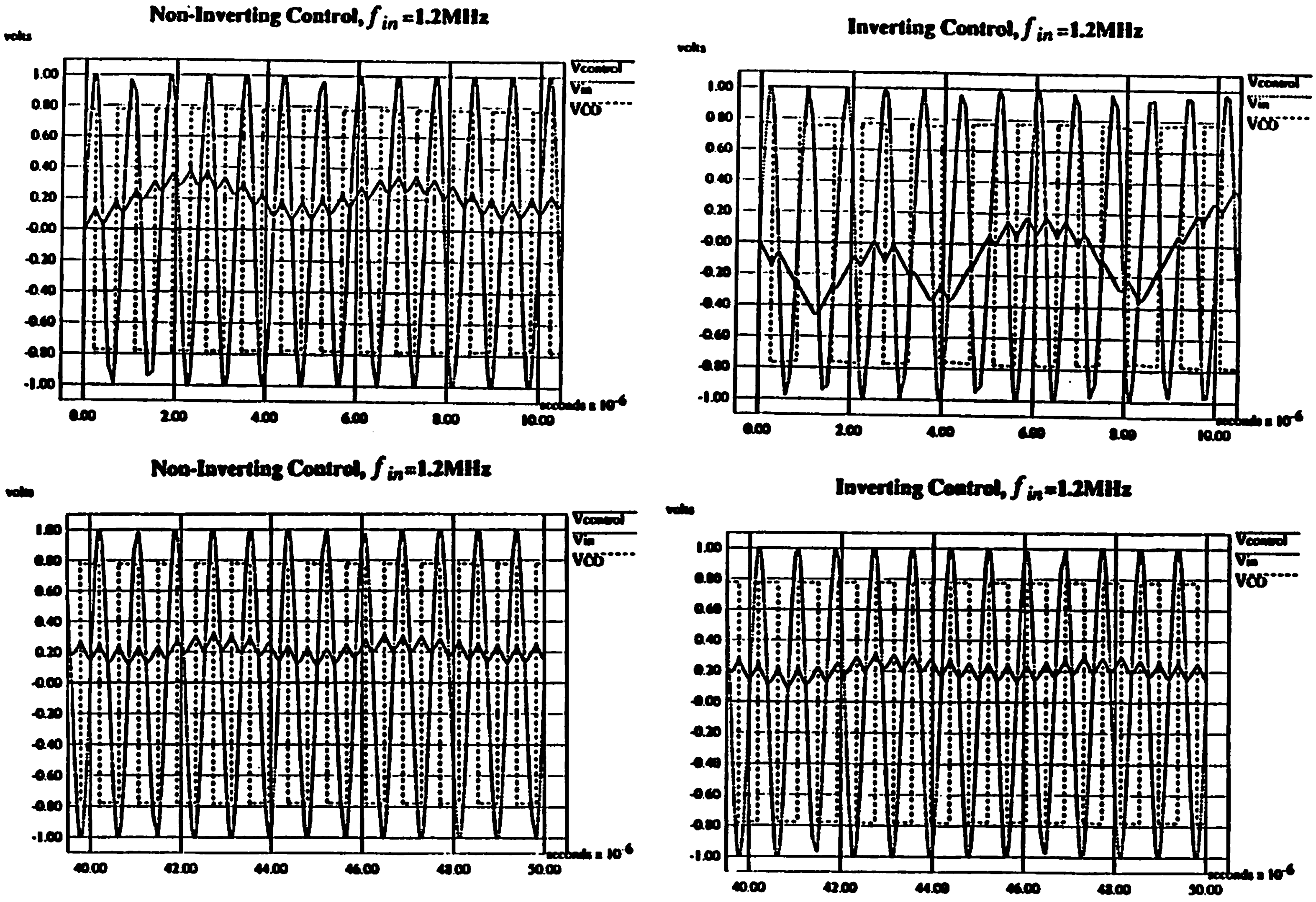


Figure D.2: Output Waveforms for Non-Inverting and Inverting Control Voltages (Input 0.8MHz)



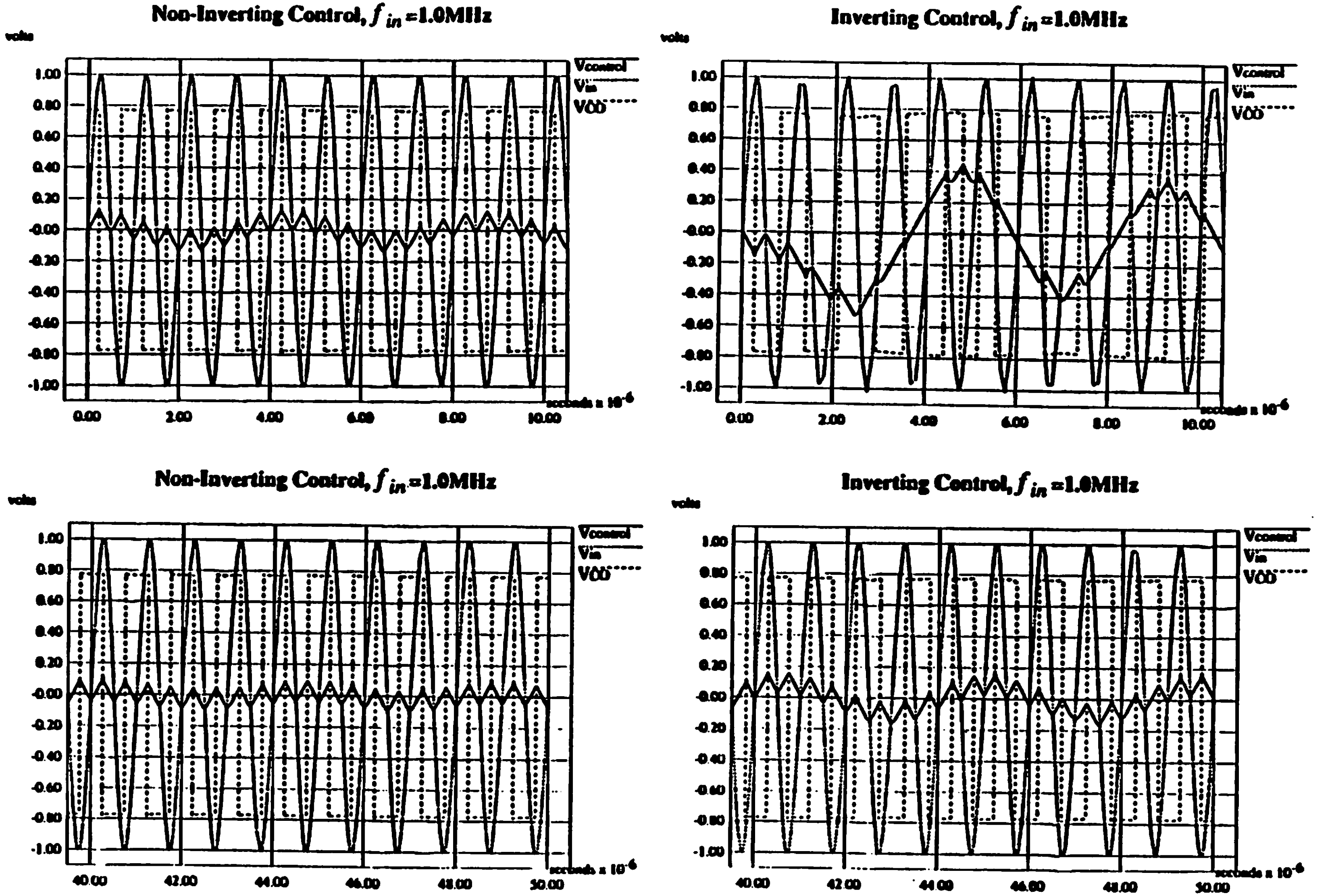
Non-Inverting Control, f_{in}=1.2MHz

Inverting Control, f_{in}=1.2MHz

Non-Inverting Control, f_{in}=1.2MHz

Inverting Control, f_{in}=1.2MHz

Figure D.3: Output Waveforms for Non-Inverting and Inverting Control Voltages (Input 1.2MHz)



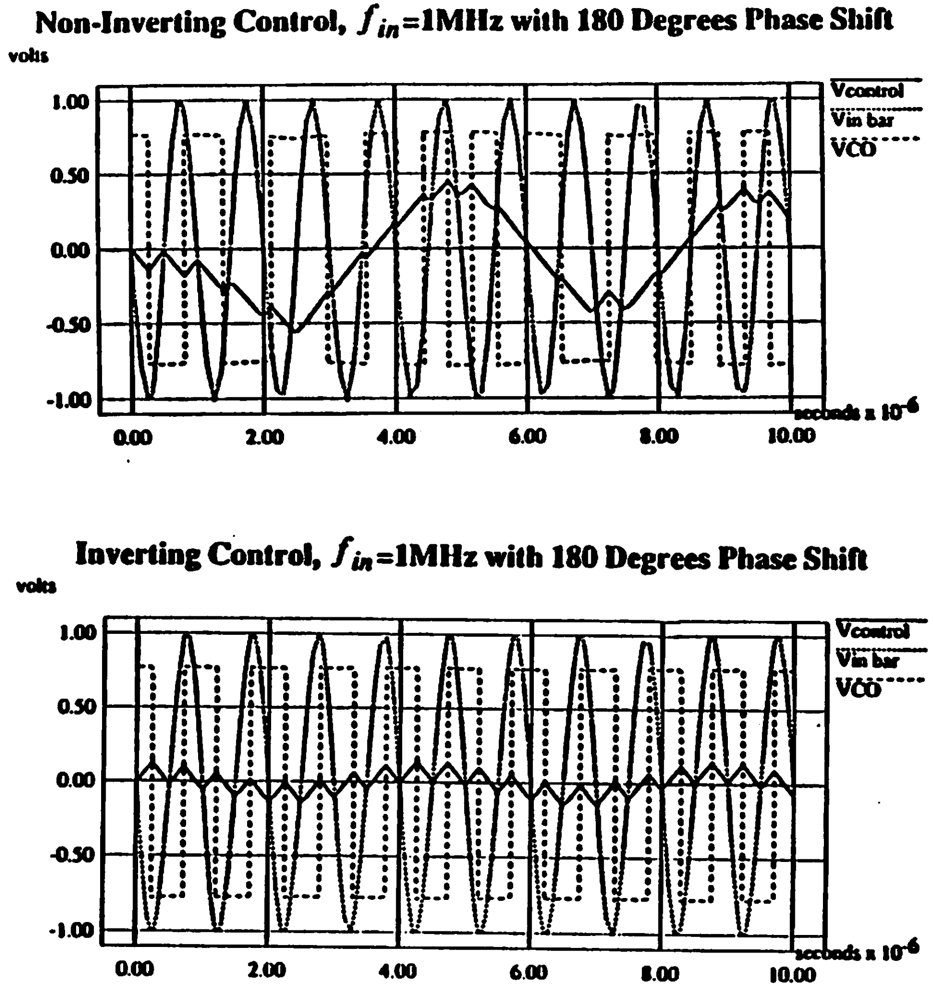
Non-Inverting Control, f_{in} =1.0MHz

Inverting Control, f_{in} =1.0MHz

Non-Inverting Control, f_{in} =1.0MHz

Inverting Control, f_{in} =1.0MHz

Figure D.4: Output Waveforms for Non-Inverting and Inverting Control Voltages (Input 1MHz)


Non-Inverting Control, f_{in}=1MHz with 180 Degrees Phase Shift

Inverting Control, f_{in}=1MHz with 180 Degrees Phase Shift

Figure D.5: Output Waveforms for Non-Inverting and Inverting Control Voltages (Input 1MHz with 180 Degrees Phase Shift)

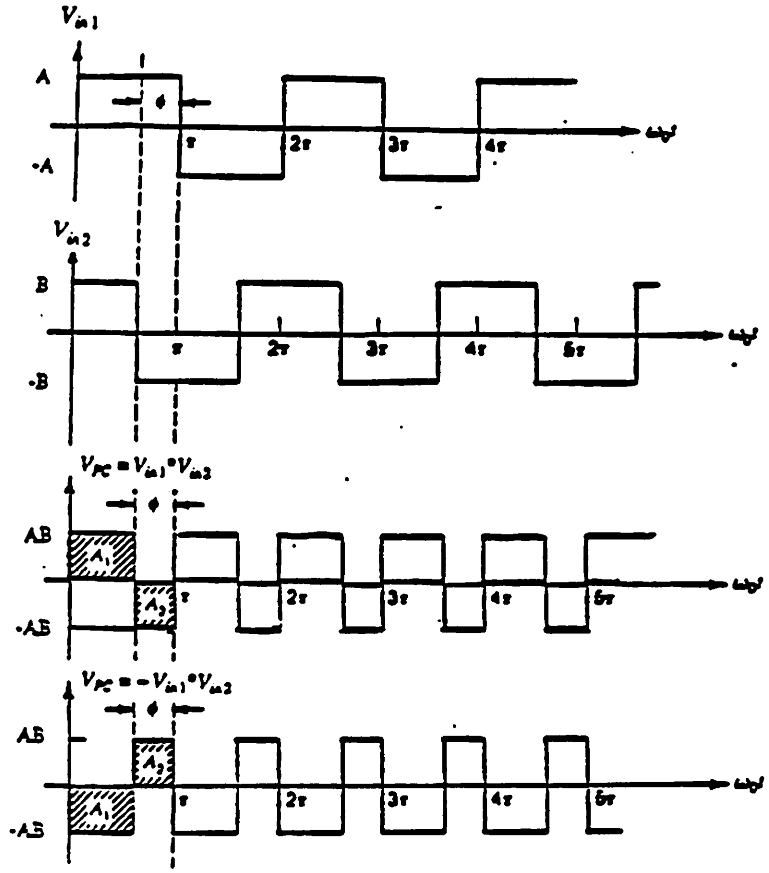


Figure E.1: Square Wave Inputs and Outputs of a Non-Inverting and Inverting Analog Multiplier PC

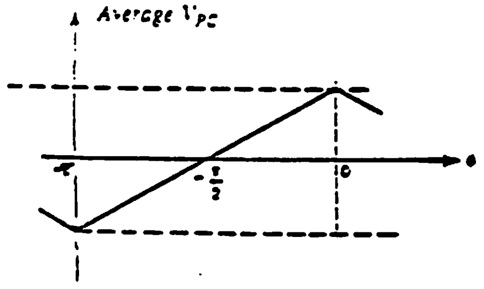


Figure E.2a:
Average PC Output vs Phase Difference for Non-Inverting Multiplier

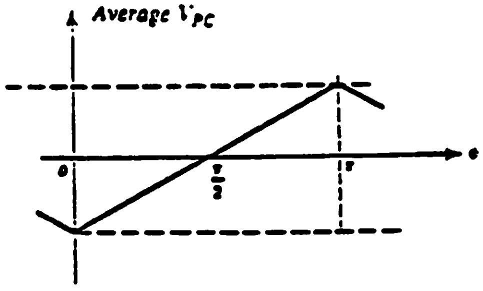


Figure E.2b:
Average PC Output vs Phase Difference for Inverting Multiplier

APPENDIX A

THE BASICS OF PHASE-LOCKED LOOPS

The phase-locked loop (PLL) is a negative feedback system that contains three basic components as shown in Figure A.1: a phase comparator [PC], a loop filter [F(s)], and a voltage-controlled or current-controlled oscillator [VCO/CCO]. The phase comparator (also called phase detector [PD]) is typically an analog multiplier or a digital edge-detection circuit; the loop filter is a low-pass filter, usually of first or second order; and the controlled oscillator is commonly a voltage-controlled multivibrator circuit. The voltage-controlled oscillator is used much more than the current-controlled oscillator in textbooks and in practice; as such, the VCO is used exclusively for the controlled oscillator function in this report. In addition to these three components, there is often an amplifier following the filter to boost the loop gain of the feedback loop, as shown in Figure A.2. The importance of a large loop gain becomes apparent during the presentation of the PLL capture and lock processes. These PLL components work together in the feedback loop to enable the VCO to match and track the frequency of the input signal. The steady-state phase relationship between the input and the VCO makes the *phase functions*[*] of the VCO and input match; thus the name 'phase-locked loop'.

[*] *Phase function* [$\theta(t)$], not frequency, is the proper term to use. If $\theta(t) = \omega_x t$, the signal has a constant frequency. If the signal is not a constant frequency source, $\theta(t)$ is an arbitrary continuous function of time. For this latter case, no 'frequency' can be associated with the signal; only a phase function properly describes the signal. However, because 'frequency' is a much more familiar term than 'phase function', both are used in this report.

Before any analysis of the PLL can be done, some general PLL parameters, illustrated in Figure A.2, must be introduced. The VCO has a free-running (or center) frequency ω_o, which is the frequency of VCO operation with a zero (or quiescent) controlling input voltage. ω_{osc} is the instantaneous frequency of the VCO, which depends on the value of the VCO control voltage. The VCO also has a parameter K_o: $K_o = \frac{\omega_{osc}}{V_o} \frac{rad}{V-sec}$ and is the VCO transfer constant. The phase comparator has a parameter K_p: $K_p = \frac{V_{PC}}{\theta_d} \frac{V}{rad}$ and is the phase-comparator transfer constant. F(s) is the lowpass filter function, and A is an added amplifier for the feedback loop. Another useful parameter is the loop gain of the feedback loop, K_L, which is simply the product of K_p, K_o, A, and $|F(j\omega)|$.

For an overall understanding of how the PLL works, a qualitative analysis may be useful. When operating in the proper frequency range, the PLL uses the current phase function difference between the incoming signal and the VCO output signal to generate a phase difference that, in turn, generates a corrective control voltage. The corrective control voltage modifies the VCO frequency to follow the incoming input frequency, provided that the input frequency is within the 'capture range' set by the PLL parameters. The capture range is described in detail later. When the phase function difference is driven to a constant value, the input and VCO have the same phase functions: they have the same frequency or go through the same frequency changes. The PLL has then reached steady state or equilibrium. The steady-state phase difference between the two signals may or may not be zero, however; the steady-state phase difference insures that the proper control voltage is provided to the VCO to make the phase functions match. This process of the initially uncorrelated VCO frequency matching the input frequency is called the 'capture process'; the frequency range within which capture can occur is the previously alluded to 'capture range'.

Once the input frequency is 'captured' by the PLL, any slight change in the input frequency appears as a change in phase function between the input signal and the VCO output. This phase change triggers the negative feedback loop to once again cause the VCO frequency to match the input frequency. In this way, the PLL tracks changes in the input frequency by adjusting the VCO frequency accordingly. This tracking of the input frequency once the input signal has been 'captured' is referred to as the 'lock process'; the corresponding frequency range within which lock can occur is called the 'lock range'. It is shown later that the lock range is usually larger than the capture range.

The above description introduces the two most important processes of the PLL: capture and lock. For a look closer at the capture and lock processes, and at their corresponding ranges, the operation of the PLL is followed in more detail, mathematically. The capture process is first examined and the capture range is derived. As the capture process is a basic nonlinear process, the following linear analysis is a crude approximation to the actual process; however, the analysis provides insight into the operation of the PLL and the importance of the parameters of the PLL components.

Assume the feedback loop of a PLL is open at the input to the VCO as shown in Figure A.3. With a zero (or quiescent) control voltage, the VCO is operating at the VCO's free-running frequency, ω_o. Let the input signal be of a fixed frequency, ω_i, different than the VCO's free-running frequency. Let the PC of the PLL be an ideal analog multiplier with multiplication constant K_m. Assume the inputs to the PC are sinusoidal:

$$V_i(t)=V_i sin(\omega_i t)$$

$$V_{osc}(t)=V_{osc}sin(\omega_o t)$$

Any initial phases are conveniently neglected because the starting time reference can be changed to an instant of time where there are no phases associated with the signals. Since $\omega_i \neq \omega_o$ and the PLL has not yet captured the input frequency, any initial phase difference is only an initial condition. Though initial phase differences greatly affect the time to capture, it is not important to the analysis of the capture range.

The output of the PC is of the form:

$$V_{PC}(t)=K_m V_i V_{osc}\sin(\omega_i t)\sin(\omega_o t) \quad [A.1]$$

With the trigonometric identities,

$$cos(a+b)=cosa\ cosb-sina\ sinb$$

$$cos(a-b)=cosa\ cosb+sina\ sinb$$

Equation [A.1] can be rewritten as:

$$V_{PC}(t)=\frac{K_m V_i V_{osc}}{2}[cos(\omega_i-\omega_o)t-\cos(\omega_i+\omega_o)t] \quad [A.2]$$

When passed through an appropriate low-pass filter, the sum-frequency is attenuated, leaving principally the difference frequency portion of the signal. Again, since a constant frequency is a special instance of a phase function, the frequencies can be rewritten as $\theta_i(t)=\omega_i t$ and $\theta_{osc}(t)=\omega_o t$. The difference frequency component can then be rewritten as $cos(\omega_i-\omega_o)t=\cos(\theta_i(t)-\theta_{osc}(t))=\cos(\theta_d(t))$, i.e., the control signal is a function of the phase difference between the input signal and the VCO signal.

As is the case of Figure A.2, there is an amplifier following the filter to boost the correcting control signal. For the assumed linear case, the control voltage of the VCO is:

$$V_o(t) \approx \frac{K_m V_i V_{osc}}{2} A \mid F[j(\omega_i - \omega_o)] \mid cos[(\omega_i - \omega_o)t + \phi] \qquad [A.3]$$

where ϕ is the phase of $F[j(\omega_i - \omega_o)]$, and the sum-frequency term is assumed to be completely filtered out.

This is the input to the VCO when the loop is closed (see Figure $A.4^1$) and is the voltage that modifies the frequency of the VCO. With the negative feedback loop closed, the VCO's frequency deviates by $\pm\Delta\omega$ from the VCO's free-running frequency, ω_o, due to the control signal from the filter/amplifier. The frequency deviation is proportional to the correcting control signal, $\frac{K_m V_i V_{osc} A}{2} \mid F[j(\omega_i - \omega_o)] \mid$, which passes through the low-pass filter. That control signal is a changing signal, since the frequency of the VCO changes from ω_o; therefore, in the transient case, the generated VCO frequency is also a changing signal. Sometimes, depending on the specific instantaneous phase difference, the VCO frequency moves away from the input frequency, and other times, the VCO frequency moves closer to the input frequency. If the oscillator frequency moves closer to the input frequency, the frequency difference (phase function difference) decreases, and the PC output becomes a slowly-varying function of time. If the oscillator frequency moves away from the input frequency, the frequency difference increases, and the PC output becomes a rapidly-varying function of time. This nonlinear process is illustrated in Figure $A.4^1$. Due to this phenomena, the output (control signal) waveform is an asymmetric waveform that contains a DC value. The DC value, which is always passed by the low-pass filter, pulls the average VCO frequency towards the input frequency. If the magnitude of the input V_i is large enough, and if signal saturation does not occur, eventually the VCO's frequency equals the input frequency, i.e., $\omega_{osc} = \omega_i$, which is

$\omega_o \pm \Delta\omega$.

The frequency of the VCO is described by

$$\omega_{osc} = \omega_o + K_o V_o \qquad [A.4]$$

where K_o is the VCO transfer factor, the conversion constant relating the VCO's output frequency to the applied input voltage.

In order for capture to occur, the magnitude of the voltage applied to the VCO input must be

$$|V_o| = \frac{(\omega_i - \omega_o)}{K_o} \qquad [A.5]$$

where Equation [A.4] is solved for V_o, and ω_{osc} is replaced with ω_i, the desired end result.

The capture range can be estimated by setting Equation [A.5] as the upper bound for the capture range, i.e., $\omega_c = \omega_i - \omega_o$ is the upper bound of the capture range. Equating [A.3] and [A.5] gives a good estimate of the capture range:

$$\omega_c \leq \frac{K_m V_i V_{osc}}{2} A \, |F[j(\omega_i - \omega_o)]| \, K_o \qquad [A.6]$$

A subtle point to notice is the use of K_m instead of the familiar K_p of Figure A.2 and various texts on PLLs. K_p is a contrived constant to make the PLL fit the mold of a conventional feedback loop configuration. If the definition $K_p = \frac{K_m V_i V_{osc}}{2}$ is made, the equations describing the capture process match the derivations in many of the books on PLLs. However, K_p is not necessarily a constant. Recall Equation [A.2], the output of the PC. Even ignoring the sum-frequency term, there is not a simple transfer function relationship between $V_{PC}(t)$ and $\theta_d(t)$. If $\cos(\theta_d)$ is rewritten as $\sin(\theta_d+\phi)$, and the constant phase ϕ is ignored, $\sin(\theta_d) \approx \theta_d$ for small θ_d; only then is $K_p = \frac{V_{PC}}{\theta_d} \frac{V}{rad} = \frac{K_m V_i V_{osc}}{2}$. For some PCs, no approximation is needed for the correct definition of K_p; it depends on the type and characteristic of PC used. But for consistency and simplicity, the K_p substitution is made. Then, the capture range is:

$$\omega_c \leq K_p A \,|F[j(\omega_i - \omega_o)]|\, K_o$$

$$\leq K_L \textit{ (loop gain) evaluated at the difference frequency} \qquad [A.7]$$

For a simple passive, one-pole RC filter,

$$|F(j(\omega_i - \omega_o))| = \frac{1}{\sqrt{1 + \frac{(\omega_i - \omega_o)^2}{\omega_{rc}^{\ 2}}}} \qquad [A.8]$$

where $\omega_{rc} = \frac{1}{R_f C_f}$ is the bandwidth of the RC filter.

In order for capture,

$$\omega_c = \omega_i - \omega_o \leq \frac{K_p K_o A}{\sqrt{1 + \frac{(\omega_i - \omega_o)^2}{\omega_{rc}^{\ 2}}}} \qquad [A.9]$$

With the substitutions $W = (\omega_i - \omega_o)^2$ and $K_l = K_p K_o A$ (DC loop gain for a passive filter), the capture range can be solved more easily:

$$(1 + \frac{W}{\omega_{rc}^{\ 2}})W \leq K_l^2$$

$$(\omega_{rc}{}^2+W)W \le K_l^2\omega_{rc}{}^2$$

$$W^2+\omega_{rc}{}^2W-K_l^2\omega_{rc}{}^2\le 0$$

The roots of the quadratic equation are

$$W=-\frac{\omega_{rc}{}^2\pm\sqrt{\omega_{rc}{}^4+4K_l^2\omega_{rc}{}^2}}{2}$$

$$=-\frac{\omega_{rc}{}^2}{2}\pm\frac{\omega_{rc}{}^2}{2}\sqrt{1+\frac{4K_l^2}{\omega_{rc}{}^2}}$$

$$=\frac{\omega_{rc}{}^2}{2}(-1\pm\sqrt{1+\frac{4K_l^2}{\omega_{rc}{}^2}})$$

Recalling that $W=(\omega_i-\omega_o)^2$ and solving for ($\omega_i-\omega_o$) yields:

$$\omega_c\le\omega_i-\omega_o=\frac{\omega_{rc}}{\sqrt{2}}\sqrt{-1+\sqrt{1+\frac{4K_l^2}{\omega_{rc}{}^2}}}$$

If $K_l \gg \omega_{rc}$,

$$\omega_c\approx\sqrt{K_l\,\omega_{rc}}=\sqrt{K_pK_oA\,\omega_{rc}}\;\frac{rad}{s} \qquad [A.10]$$

This result is only an approximation for the capture range because the difference frequency between the input and VCO is constantly changing; thus, the filter function magnitude (Equation [A.8]) is constantly changing also, whereas this analysis assumed that the difference frequency stayed at the initial difference frequency for the entire time. As such, the capture range is larger than that calculated; as the difference frequency component gets smaller (begins to capture), the filter magnitude (Equation [A.8]) gets larger, and thus the capture potential gets stronger.

Since $K_p = \frac{K_m V_i V_{osc}}{2}$, the capture range is controlled by various constants of the PLL (K_m, K_o, A), by the input and VCO amplitudes, and by the filter's cutoff frequency. The input signal cannot be captured if the input amplitude is too small to drive the VCO, or if the input frequency is beyond the dynamic range of the PLL, in which case the filter attenuates the control signal to a negligible level.

The same type of analysis can be performed to examine the lock process and to obtain the lock range. For the lock process, however, there is the additional condition that $\omega_{osc} = \omega_i$, i.e., the initial condition is one of a captured input signal. With this observation, a slightly modified set of equations is obtained:

$$V_i(t) = V_i sin(\omega_i t)$$

$$V_{osc}(t) = V_{osc} sin(\omega_i t + \theta_d)$$

The output of the PC then becomes:

$$V_{PC}(t) = \frac{K_m V_i V_{osc}}{2}[\cos(\theta_d) - \cos(2\omega_i + \theta_d)t] \qquad [A.11]$$

With the high-frequency portion assumed to be successfully filtered out of the control signal, what remains is a DC quantity that is the control signal needed to make the VCO frequency match the input frequency. The control signal is:

$$V_o(t) = \frac{K_m V_i V_{osc}}{2} A\, |F[j(0)]|\, cos(\theta_d) \qquad [A.12]$$

If the input frequency now changes to ω_i', the desired change in frequency is $\omega_{osc} = \omega_i'$.

$$\omega_{osc}=\omega_o+K_o V_o=\omega_i' \qquad \text{[A.13]}$$

In order for lock to occur,

$$|V_o|=\frac{\omega_i'-\omega_o}{K_o} \qquad \text{[A.14]}$$

The lock range can then be estimated by setting Equation [A.14] as the upper bound for the lock range, i.e., $\omega_l=\omega_{i'}-\omega_o$ is the upper bound of the lock range. If Equations [A.12] and [A.14] are now solved simultaneously, a good estimate of the lock range is:

$$\omega_l \le \frac{K_m V_i V_{osc}}{2} A\, |F[j(0)]|\, K_o \qquad \text{[A.15]}$$

With the substitution $K_p=\dfrac{K_m V_i V_{osc}}{2}$, a lock range similar to the capture range is obtained:

$$\begin{aligned} \omega_l &\le K_p A\, |F[j(0)]|\, K_o \\ &\le K_L\ (\textit{loop gain})\ \textit{evaluated at DC} \end{aligned} \qquad \text{[A.16]}$$

Both the capture range and the lock range are controlled by the loop gain of the PLL; for a larger range of PLL operation, a larger loop gain is desired (hence the need for a gain block A to boost the loop gain). The lock range, however, depends on the magnitude of the loop gain evaluated at DC, not at the $\omega_i-\omega_o$ difference frequency as in the capture range. Thus the lock range is more readily controlled, as the value of $|F[j(0)]|$ is well known. However, the the value of the initial frequency difference which governs the value of $|F[j(\omega_i-\omega_o)]|$ is not known. Since the only difference between the lock and capture ranges is the magnitude of the filter function, the lock range is generally larger than the capture range for an ordinary low-pass filter, since the magnitude of the filter function at DC is usually the largest.

From the above observations, the capture process and capture range are not very well predicted. Furthermore, the capture process is very non-linear in nature. The lock range, however, is more predictable, and the initial state of the loop is well known. For the lock process to be successful, the input frequency is only allowed to change within a certain range of the VCO's captured frequency. With these restricted frequency changes, and a large Ko, the assumption is that there are small phase differences. Upon examination, the correction control signal at the output of the PC is of the form:

$$V_o(t)=K\ sinusoid[\theta_d(t)]\approx K\,\theta_d(t)\ \text{for}\ small\ \theta_d(t) \qquad [A.17]$$

where K is the product of known quantities. Here, the system is indeed a near-linear one for small phase differences, in which case, linear analysis techniques are valid.

For a near-linear system as in the lock process, the feedback configuration of the PLL lends itself to well-known feedback analysis techniques. These techniques are only valid for the PLL operating as a linear system, so the analysis is not valid for the capture process. With this in mind, the conventional feedback loop equations that describe the PLL of Figure A.2 can be written:

$$V_{PC}(s)=K_p[\theta_i(s)-\theta_{osc}(s)] \qquad [A.18]$$

$$V_o(s)=AF(s)V_{PC}(s) \qquad [A.19]$$

$$\theta_{osc}(s)=\frac{K_o}{s}V_o(s) \qquad [A.20]$$

By re-arranging Equations [A.18]-[A.20], the familiar transfer functions of the loop can be derived:

$$\frac{\theta_{osc}(s)}{\theta_i(s)}=\frac{K_oK_pAF(s)}{s+K_oK_pAF(s)}=\frac{K_L}{s+K_L} \qquad [A.21]$$

$$\frac{\theta_i(s)-\theta_{osc}(s)}{\theta_i(s)}=\frac{\theta_d(s)}{\theta_i(s)}=\frac{s}{s+K_oK_pAF(s)}=\frac{s}{s+K_L} \qquad [A.22]$$

$$\frac{V_o(s)}{\theta_i(s)}=\frac{sK_pAF(s)}{s+K_oK_pAF(s)}=\frac{s\frac{K_L}{K_o}}{s+K_L} \qquad [A.23]$$

and the definition $K_L=loop\ gain=K_pK_oAF(s)$ is made.

From Equation [A.21], only when a DC signal (s=0) is passed by the filter does the phase function of the input and the VCO match. Similarly, from Equation [A.22], the phase difference $\theta_d(0)=0$. When a low-frequency signal is passed by the filter, these linear equations can be used to help evaluate and understand the lock process better.

The linear feedback transfer functions can also be used to analyze the steady-state of the lock process. If the final value theorem is used along with Equation [A.22],

$$\lim_{t\to\infty}\theta_d(t)=\lim_{s\to 0}\frac{s^2\theta_i(s)}{s+K_pAK_oF(s)} \qquad [A.24]$$

If from lock, there is a step change in input phase, $\theta_i(s)=\frac{\delta\theta}{s}$:

$$\lim_{t\to\infty}\theta_d(t)=0 \qquad [A.25]$$

In this case, regardless of F(s), the constant phase difference is 0.

If from lock, there is a step change in input frequency, $\theta_i(s)=\frac{\delta\omega}{s^2}$:

$$\lim_{t\to\infty}\theta_d(t)=\lim_{s\to 0}\frac{\delta\omega}{s+K_pAK_oF(s)}=\frac{\delta\omega}{K_L}\,at\,DC \qquad [A.26]$$

With θ_d limited to $\pm\pi$, and in most cases, limited to $\pm\frac{\pi}{2}$ or $\pm\frac{\pi}{4}$, there is a maximum $\delta\omega$ for the PLL to stay in lock. The particular limit on $\delta\omega$ depends on the limitations of the PC. For this case, the value of $|F(0)|$ directly affects the maximum $\delta\omega$ allowable. An active filter (a filter composed of active elements like transistors, thus capable of gain) with $|F(0)|$=A is much preferable to a passive filter (a filter composed of only passive elements, not capable of gain) with $|F(0)|=1$.

If from lock, there is an input frequency that linearly changes with time, i.e., $\theta_i(s)=\frac{\delta\dot{\omega}}{s^3}$:

$$\lim_{t\to\infty}\theta_d(t)=\lim_{s\to 0}\frac{\delta\dot{\omega}}{s(s+K_pAK_oF(s))} \qquad [A.27]$$

In this case, if F(0) is finite, then there is no steady-state θ_d, as θ_d approaches infinity; obviously, lock is lost. If an active filter (with the characteristic F(s)=$\frac{1}{s}$) is used, however, θ_d approaches $\frac{\delta\dot{\omega}}{K_pAK_o}$ in the steady-state, which is finite. Again, this places a limit on the maximum value of $\delta\dot{\omega}$ while still maintaining lock.

APPENDIX B

EFFECTS OF Ro ON THE WIEN VCO FREQUENCY

The characteristic equation of the Wien VCO macromodel (see Figure 5.4) is:

$$s^2+\frac{R_1C_1+(R_2+r_o)C_2+R_1C_2-AR_1C_2}{R_1(R_2+r_o)C_1C_2}s+\frac{1}{R_1(R_2+r_o)C_1C_2}=0 \qquad [B.1]$$

where $R_1=R1\,||\,\frac{V(6)-V(4)}{Igvar1}$, $R_2=R2\,||\,\frac{V(4)}{Igvar2}$, and A=3.05.

On the surface, this slight change appears to translate to an $R_{2eq}=R_2+r_o$. However, the output resistance corrupts the VCO output because r_o is fixed while R_2 is variable.

With $C_1=C_2=C$ and $R_1=R_2=R_{eq}=\frac{1}{G_{eq}}=\frac{R}{1+kV_cR}$ (k is a chosen constant),

$$\omega_{osc}=\frac{1}{\sqrt{R_{eq}C\,[R_{eq}+r_o]C}} \qquad [B.2]$$

Substituting in the actual expression for $\frac{1}{R_{eq}C}$ and factoring yields:

$$\omega_{osc}=\frac{1}{C}\frac{1}{\sqrt{\frac{R}{1+kV_cR}(\frac{R}{1+kV_cR}+r_o)}}$$

$$=\frac{1}{C}\frac{1}{\sqrt{\frac{R}{1+kV_cR}\frac{(R+r_o)+kV_cRr_o}{1+kV_cR}}}$$

$$=(\frac{1}{RC}+\frac{k}{C}V_c)\frac{1}{\sqrt{\frac{R+r_o}{R}+kV_cr_o}}$$

$$=(\frac{1}{RC}+\frac{k}{C}V_c)\frac{1}{\sqrt{1+r_o(\frac{1}{R}+kV_c)}} \qquad [B.3]$$

In Equation [B.3], the original variable frequency is factored out, showing an extra frequency reduction factor of:

$$\frac{1}{\sqrt{1+r_o(\frac{1}{R}+kV_c)}} \qquad [B.4]$$

This frequency reduction makes the VCO frequency depend on r_o and vary nonlinearly for different V_c and R values.

APPENDIX C

GENERATING A SINUSOID WITH A VOLTAGE-CONTROLLED ARGUMENT

An apparent sinusoidal equation VCO can be implemented in SPICE as:

$$BVCO\ \ 7\ \ 0\ \ V = A * \sin(\omega_o(1 + k_o V_c)t) \qquad [C.1]$$

BVCO indicates a controlled source; A is the fixed amplitude of the oscillator; ω_o is the free-running frequency of oscillation; k_o is the scaled VCO gain factor; V_c is the control voltage; and t is time.

Unfortunately, this type of controlled source does not exist in SPICE2; it does exist in PSPICE as well as SPICE3. In PSPICE, time is a usable variable; however, in SPICE3, it is not. Thus, a dummy variable of constant slope must be used as a pseudo-time source in place of the real time. An Equation VCO comparable to the Wien VCO macromodel for SPICE3 simulation is:

```
BVCO 7 0 V=10*sin( 2*pi*1e+6* (1+0.1*v(3)) *v(1) )
V1 1 0 PWL 0 0 100u 100u                              [C.2]
```

where v(1) is the pseudo-time function, and v(3) is the control voltage.

Such an Equation VCO appears to be a very simple and viable substitute for the Wien VCO macromodel. However, there is a subtle problem with the way this VCO is defined, namely that it works with frequency and not with a phase function. What is really desired is not frequency changing linearly with a control voltage, but the phase function changing linearly with a control voltage. By definition,

$$\frac{d\theta(t)}{dt}=\omega(t) \qquad [C.3]$$

The term 'frequency' is the special case of a constant $\omega(t)$:

$$\omega(t)=\omega_{fixed}$$

and

$$\frac{d\theta(t)}{dt}=\omega_{fixed}$$

$$\theta(t)=\int \omega_{fixed}\,dt=\omega_{fixed}\,t \qquad [C.4]$$

For a PLL, however, this special case does not apply. Consequently, the VCO macromodel must work with phase functions if it is to behave as desired. In Equation [C.1], the phase function is strictly a multiplication of time and a frequency function, namely, $\theta(t)=\omega(t)t$. In order for Equation [C.3] to hold,

$$d\theta(t)=d\,[\omega(t)\,t]=\omega(t)dt \qquad [C.5]$$

which is only true if $\omega(t)$ is independent of t.

Figure C.1 shows the SPICE3 input file for this Equation VCO that multiplies a frequency term and time. Figure C.2 is the corresponding output for a changing control voltage, which is the equivalent of a PLL input changing frequency. For a constant control signal (constant frequency), the output is correct. However, once the control voltage changes, $\omega(t)$ is not constant and an undesired change in phase function relating to initial conditions and slope occurs. Because the phase function must be continuous, the VCO works to achieve this continuity under the forced constraint of the 'frequency' and time product. Figure C.3 shows that for the same input, but time delayed, the output frequency is different. The system is thus time-variant. In addition, static phase jumps arise to keep the continuity, as also illustrated in Figure C.3.

Instead of multiplying a control voltage with time, integrating the voltage over time produces the correct linear phase function. This can be achieved in SPICE by forcing a voltage-controlled current source though a capacitor (see Figure 5.1):

$$I = K\,(1 + k_o V_c)\ ;\ \ C\frac{dV}{dt} = I \quad (K \textit{ is a chosen constant}) \tag{C.6}$$

$$C\frac{dV}{dt} = K\,(1 + k_o V_c)$$

$$V = \frac{K}{C}\int (1 + k_o V_c)\,dt \tag{C.7}$$

If K=C, the voltage V satisfies Equation [C.3] and can be used as the desired phase function

$$\theta(t) = V = \int \omega(t)dt \tag{C.8}$$

where $\omega(t) = 1 + k_o V_c$ for the VCO macromodel.

The Integrating Sine VCO can be implemented in SPICE3 as follows:

```
BVCO1 7 0 V=10*sin(2*pi*1e+6*V(9))
Ccap 9 0 1pF
Bcap 0 9 I=1e-12* (1+0.1V(3))                    [C.9]
```

where V(3) is the control voltage.

APPENDIX D

INVERTING AND NON-INVERTING CONTROL VOLTAGES

The control voltage to the VCO is generated by the average value of the PC output. In the case of the 560B circuits, the control voltage depends on the negative average product of the PC inputs, while in all the other macromodels, the control voltage depends on the positive average product of the PC inputs. Both cases are valid. The inverting control-voltage case simply centers around the input phase leading the VCO phase by 90^o, while the non-inverting control-voltage case centers around the input phase lagging the VCO phase by 90^o. In other words, the inverting control-voltage configuration has a PC characteristic centered around $+90^o$ while the non-inverting control-voltage configuration has a PC characteristic centered around -90^o (see Figures E.2a and E.2b). This phase relationship is shown in Figures D.2 through D.5, which are the outputs for the Simplified 560B Macromodel with both inverting and non-inverting connections. The SPICE3 input file for the Simplified 560B macromodel is shown in Figure D.1.

Figures D.2 though D.4 show the output waveforms for the two configurations mentioned above. The initial phase conditions of all the simulations have the input phase lagging the VCO phase. Though the control voltage finally settles to approximately the same value for both configurations, the inverting case shows much more fluctuation initially. Figure D.5 shows a case where the initial phase condition has the input phase leading the VCO phase; the non-inverting case now shows more fluctuation. A non-inverting configuration will have large fluctuations for all input frequencies if the initial phase condition has the input phase leading the VCO phase, and vice versa. This is shown in Figures D.2 though D.4 for the inverting case. If a non-inverting configuration has an initial condition with the input phase lagging the VCO phase, the control voltage will very quickly reach the steady-state voltage for input frequencies higher than the VCO free-running frequency (Figure D.3), but will fluctuate with greater amplitude for input frequencies lower than the VCO free-running frequency (Figure D.2). This fluctuation for input frequencies lower than the VCO free-running frequency can help capture input signals at the low input-frequency end, causing an uneven capture range.

For the PLL macromodels that are simulated in this report, the inverting control-voltage PLLs are given initial conditions in which the input phase leads the VCO phase, and vice-versa.

APPENDIX E

THE TRIANGULAR PHASE-COMPARATOR CHARACTERISTIC

Figure E.1 shows the inputs and outputs of an analog multiplier PC with $V_{PC}=V_{in1}*V_{in2}$ and $V_{PC}=-V_{in1}*V_{in2}$. The PC inputs are two fixed-amplitude rectangular waves of the same frequency (for simplicity) but of differing phase. The PC outputs are rectangular waves with an amplitude of A*B. The average value of the PC output is the DC term passed by the loop filter of the PLL, which is the control signal to the VCO. This DC component of the waveform for the non-inverting PC ($V_{PC}=V_{in1}*V_{in2}$) is:

$$V_{control}=\frac{1}{2\pi}\int V_{PC}(t)d(\omega_o t) \qquad [E.1]$$

$$=\frac{1}{\pi}(A_1-A_2) \qquad [E.2]$$

where A_1 and A_2 are the areas indicated in Figure E.1. Thus,

$$V_{control}=[A*B\frac{(\pi-\phi)}{\pi}-A*B\frac{\phi}{\pi}] \qquad [E.3]$$

$$=-\frac{2A*B}{\pi}(\phi-\frac{\pi}{2})$$

$$=\frac{2A*B}{\pi}(\phi+\frac{\pi}{2}) \qquad [E.4]$$

The triangular PC characteristic is plotted in Figure E.2a. K_p, the positive slope, is $\frac{2(A*B)}{\pi}$ and operates around a phase difference of $-\frac{\pi}{2}$.

If the analog multiplier PC is inverting, i.e., $V_{PC} = -V_{in1} * V_{in2}$, the control voltage is inverted. The same calculation for control voltage produces:

$$V_{control} = \frac{2A*B}{\pi}(\phi - \frac{\pi}{2}) \qquad [E.5]$$

with K_p being $\frac{2A*B}{\pi}$ and operating around $\frac{\pi}{2}$ as plotted in Figure E.2b. For the purposes of this report, K_p refers to the magnitude of the slope, and the corresponding operating point is denoted by specifying the PC as producing a non-inverting or inverting control voltage.

From Equations [E.4] and [E.5], a triangular PC characteristic depends on the PC output being rectangular (for the simple amplitude*phase area) and on the value of the PC rectangular output amplitude. If the inputs to the PC are amplified and limited to produce fixed amplitude rectangular waves, the situation is the one illustrated above, which has a triangular PC characteristic. Alternatively, if the PC output is amplified and limited, the PC output is also rectangular with a fixed amplitude, and a triangular PC characteristic also results. Both these methods are viable ways of achieving a triangular PC characteristic.

With a rectangular PC waveform, the maximum $V_{control}$ possible is $\frac{\pi}{2} K_p = A*B$. Substituting this new maximum control voltage into the lock and capture range equations of Appendix A yields:

$$\omega_c \leq \frac{\pi}{2} K_p A \,|F[j(\omega_i - \omega_o)]|\, K_o$$

$$\leq \frac{\pi}{2} K_L (loopgain) \textit{evaluated at the difference frequency} \qquad [E.6]$$

and

$$\omega_l \le \frac{\pi}{2} K_p A \,|F[j(0)]|\, K_o$$

$$\le \frac{\pi}{2} K_L \textit{ (loop gain) evaluated at DC} \qquad [E.7]$$

For a simple one-pole RC filter with bandwidth ω_{RC} and $\frac{\pi}{2} K_p K_o A \gg \omega_{rc}$, the approximate capture range is:

$$\omega_c \approx \sqrt{\frac{\pi}{2} K_p K_o A \,\omega_{rc}} \; \frac{rad}{s} \qquad [E.8]$$

REFERENCES

[1] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 2nd edition, John Wiley & Sons, Inc. 1984.

[2] D. O. Pederson and K. Mayaram, *Analog ICs for Communications: Principles, Simulation & Design*, Kluwer Academic Publishers, 1990.

[3] F. M. Gardner, *Phaselock Techniques*, 2nd edition, John Wiley & Sons, Inc. 1979.

[4] *Applications of PLL*, Signetics, 1974.

[5] for example, see Program ELDO by Anacad, and Program SABER by Analogy.