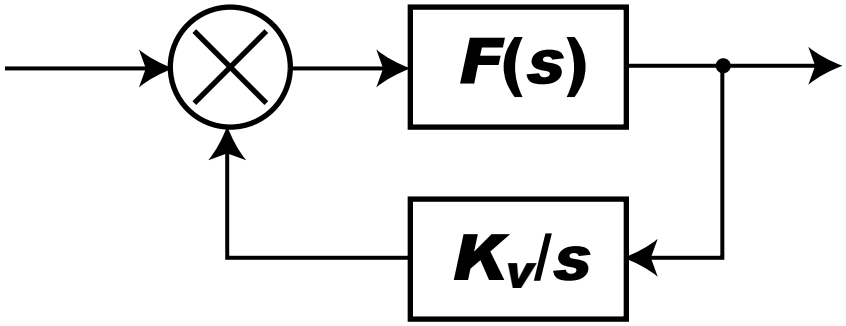
PHASE-LOCKED LOOPS WITH APPLICATIONS

ECE 5675/4675 Lecture Notes
Fall 2004



Mark A. Wickert

Chapter 1

Course Introduction/Overview

Contents

•

1.1 Lecture Outline

- This Course and the PLL Landscape
  - General PLL perspective
  - Course Topics
- Course perspective in the comm/DSP area of ECE
- The role of computer analysis/simulation tools
- Instructor policies
- Course syllabus
- Required student background
- References
  - Books
  - Reports
  - Journals
- PLL introduction and applications overview

1.2 This Course and the Phase-Locked Loop Landscape

1.2.1 General PLL Perspective

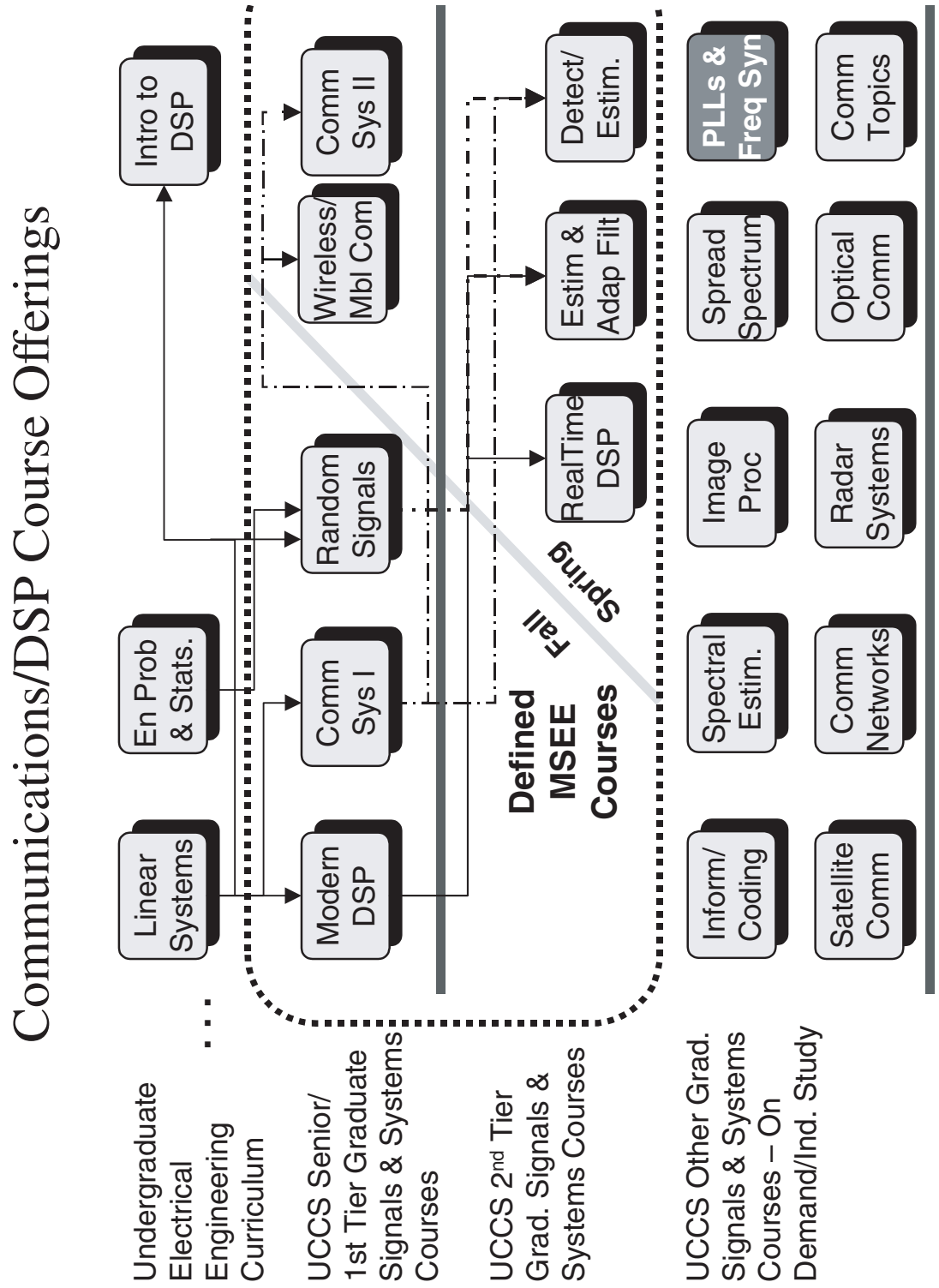
- The focus of this course is phase-lock loops (PLLs) and applications
- At first this may seem like a very narrow course of study, but the PLL has many applications and many implementation variations
- The use of PLLs for frequency synthesis, i.e., creating a stable yet tuneable local oscillator for radio transmitters and receivers is one traditional application area
- In communication systems in general the PLL is widely used
  - Carrier phase and frequency tracking
  - Symbol (bit) synchronization
  - Chip synchronization is spread-spectrum systems (this includes GPS receivers)
- Clock recovery (same class of problems as symbol sync)
- The implementation may be:
  - All analog electronics (microwave/RF/baseband)
  - A hybrid of analog and digital electronics
  - A hybrid of analog and software
  - Pure software
- The implementation technology may be:
  - Board level using RF and baseband devices
  - Single chip with a few off-chip or maybe no off-chip parts

  - Custom ASIC or FPGA
  - A combination of RF and baseband analog with the remainder in software via a real-time digital signal processing
  - Entirely real-time DSP approach if signal samples are acquired somewhere else

1.2.2 Course Topics

- PLL fundamentals
  - Loop components
  - Loop response
  - Loop stability
  - Transient response
  - Modulation response
- Acquisition
  - Unaided
  - Aided
- Analog PLL lab experiment
- Performance in noise
  - Input noise
  - Phase noise
  - Nonlinear behavior and cycle slipping
- Digital PLLs
- Communication applications

1.3 Course Perspective in the Comm/DSP Area of ECE



1.4 The Role of Computer Analysis/Simulation Tools

- In working homework problems pencil and paper type solutions will work for most problems
- Often times solutions can be enhanced through the use of computer analysis and simulation tools
- At least one computer project will be assigned which involves both mathematical modeling and simulation
- Most often MATLAB will suffice for modeling and simulation
- The use of Spice with behavioral level modeling capabilities may also be useful, e.g., XSpice or PSpice
- Block level simulation environments are particularly convenient for PLL simulations
  - MATLAB Simulink: very powerful block simulation environment, most capable for PLL work when expanded with DSP blockset and the Comm block set; limited PLL specific blocks even in Comm block set
  - VisSim/Comm (included with the student version of Mathcad: A friendly block simulation environment, which with the Comm library is very good at complex baseband simulation; PLL components are also included
  - Elanix SystemView (used the last time the course was taught): free student version which has a RF and Comm block set with some PLL specific oriented blocks

1.5 Instructor Policies

- Working homework problems will be a very important aspect of this course
- Each student is to his/her own work and be diligent in keeping up with problem assignments
- If work travel keeps you from attending class on some evening, please inform me ahead of time so I can plan accordingly, and you can make arrangements for turning in papers
- The course web site

  `http://eceweb.uccs.edu/wickert/ece5675/`

  will serve as an information source in between weekly class meetings
- Please check the web site updated course notes, assignments, hints pages, and other important course news

1.6 Course Syllabus

ECE 5675/4675 Phase-Locked Loops, Synchronization, and Frequency Synthesis

Fall Semester 2004

Instructor: Dr. Mark Wickert **Office:** EB-226 **Phone:** 262-3500
wickert@eas.uccs.edu **Fax:** 262-3589
http://eceweb.uccs.edu/wickert/

Office Hrs: Tue. 10:45 am–12:00 pm, 3:15–4:15 pm and after 7:05 pm as needed, others by appointment. Note: These hours may be adjusted if needed.

Required Texts: W. F. Egan, *Phase-Lock Basics*, Wiley, 1998. Optional text: B. Razavi, *Monolithic Phase-Locked Loops and Clock Recovery Circuits*, IEEE Press 1996.

Optional Software: MATLAB Student Version 7.x, Simulink 5.x, and Symbolic Math Toolbox (no matrix size limits). An interactive numerical analysis, data analysis, and graphics package for Windows/Linux/Mac OSX $99.95. The signal processing and control toolboxes will be helpful at $29.95 each. Order both from `www.mathworks.com/student`. Note: The ECE PC Lab has the full version of MATLAB and Simulink for windows (ver. 7.0) with many toolboxes. Other tools of interest include VisSim/Comm (`www.vissim.com/`)

Grading:
1.) Graded homework assignments totaling 40%.
2.) Mid-term Exam worth 25%.
3.) Analog PLL Laboratory 10%.
4.) Final Project/Exam worth 25%.

Topics	Text Chapters	Session (wks)
1. Introduction/Overview	1, 10	1.0
2. Phase-Locked Loop Fundamentals (including basic synthesizers)	2, 3, 4, 5, 6, 7	3.0
3. PLL Tracking Performance in Noise (including phase noise)	11, 12, 13, 14, 15, 16, 17, 18, 19	2.0
4. Unaided and Aided Acquisition	8, 9	2.0
5. Analog PLL Lab Experiment	Handout	2.0
6. Digital signal processing Based PLLs	Notes only	2.0
7. Real-time DSP PLL Experiment (if time permits or just simulation)	Handout	1.0
8. Specific synchronization techniques in Comm Systems (DSP based?)	Notes, 10	0.5

Requirements: A background in basic communication theory, probability and random variables, and basic digital signal processing, i.e. sampling theory would be desired. Please contact Dr. Wickert if you are considering this course, but are in doubt as to whether you have adequate background

1.7 Required Student Background

- Basic linear systems theory is a must
- Random variables is needed for noise analysis
  - A brief introduction to random processes will be provided if needed
- Basic modulation theory is also assumed
- A knowledge of digital communication systems is desirable
- A basic understanding of digital signal processing would be helpful, but is not required
- Knowledge of sampling theory is needed for digital loop concepts
- Knowledge of z-domain concepts is desirable, but again a brief introduction will be given if needed
- The ability to program using MATLAB is important for simulation aspects
- The ability to build Spice circuit models is also desirable

1.8 References

The following list of references is not exhaustive by any means, but is a list core of books I have in my library or have been recommended to me. Some of these books are hard to find since they are now out-of-print.

General PLL

1. Heinrich Meyr and Gerd Ascheid, *Synchronization in Digital Communications*, Volume 1, John Wiley, 1990.
   - This text is basically concerned with analog PLLs (including charge-pump) starting from the very basic concepts all the way through very detailed nonlinear analysis with noise
   - The text also includes material on automatic frequency control (AFC) and automatic gain control (AGC)
   - The book is clearly telecommunications based since PLL synthesizers are not considered at all
   - Used as the course text the last time the course was taught
2. Roland E. Best, *Phase-Locked Loops, Theory Design, and Applications*, fourth edition, McGraw Hill, 1999.
3. Alain Blanchard, *Phase-Locked Loops: Application to Coherent Receiver Design*, Wiley, New York,1976. Original course text, but now unavailable.
4. Floyd M. Gardner, *Phaselock Techniques*, 2nd ed., Wiley, New York, 1979.
5. Dan Wolaver, *Phase-Locked Loop Circuit Design*, Prentice Hall, New Jersey, 1991.
6. Jack K. Holmes, *Coherent Spread Spectrum Systems*, John Wiley, 1982.
7. Jacob Klapper and John T. Frankle, *Phase-Locked and Frequency-Feedback Systems*, Academic Press, New York, 1972.
8. William C. Lindsey and Marvin K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

9. A. J. Viterbi, *Principles of Coherent Communications*, McGraw-Hill, New York, 1966.

10. Rodger E. Ziemer and Roger L. Peterson, *Digital Communications and Spread Spectrum Systems*, Macmillan, New York, 1985.

Frequency Synthesizers

1. William F. Egan, *Frequency Synthesis by Phase Lock*, second edition, Wiley, New York, 2000.

2. Venceslav F. Kroupa, *Frequency Synthesis: Theory, Design, and Applications*, Wiley, New York, 1973.

3. Vadim Manassewitsch, *Frequency Synthesizers: Theory and Design*, 3rd ed., Wiley, New York, 1987.

4. W. P. Robbins, *Phase Noise in Signal Sources (Theory and Applications)*, Peter Peregrinus Ltd., London, UK., 1982.

5. Ronald C. Stirling, *Microwave Frequency Synthesizers*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

Synchronization

1. Umberto Mengali and Aldo N. D'Andrea, *Synchronization Techniques for Digital Receivers*, Plenum Press, New York, 1997.

2. Heinrich Meyr, Marc Moeneclaey, and Stefan Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, Prentice Hall, New Jersey, 1998. This is volume II of of Meyr and Ascheid.

Reports

1. L. Bogusch, "Digital Communications in Fading Channels: Tracking and Synchronization," AFWL-TR-90-15, Weapons Laboratory, Kirkland AFB, NM, 1990.

Journals

1. IEEE Transactions on Communications.

2. IEEE Journal on Select Areas in Communications.

1.9 PLL Introduction

The history of the phase-locked loop dates back to as early as 1932. According to author Dr. Rolland Best, the French engineer de Belle-size is the inventor of coherent communication. Current applications of the phase-locked loop (PLL) in-clude more than just coherent communications. Frequency synthesis is currently a very important PLL application area. The emphasis of this course will however be for the most part on basic PLL theory and telecommunication applications. A discussion of various synthesizer types will also be included at some point.

1.9.1 Classification of PLLs

In this course the intention is to discuss four basic classes of PLLs. Using the notation of Best[1] these classes are:

- Analog or Linear PLL (LPLL)
- Digital PLL (DPLL)
- All digital PLL (ADPLL)
- Software PLL (SPLL)

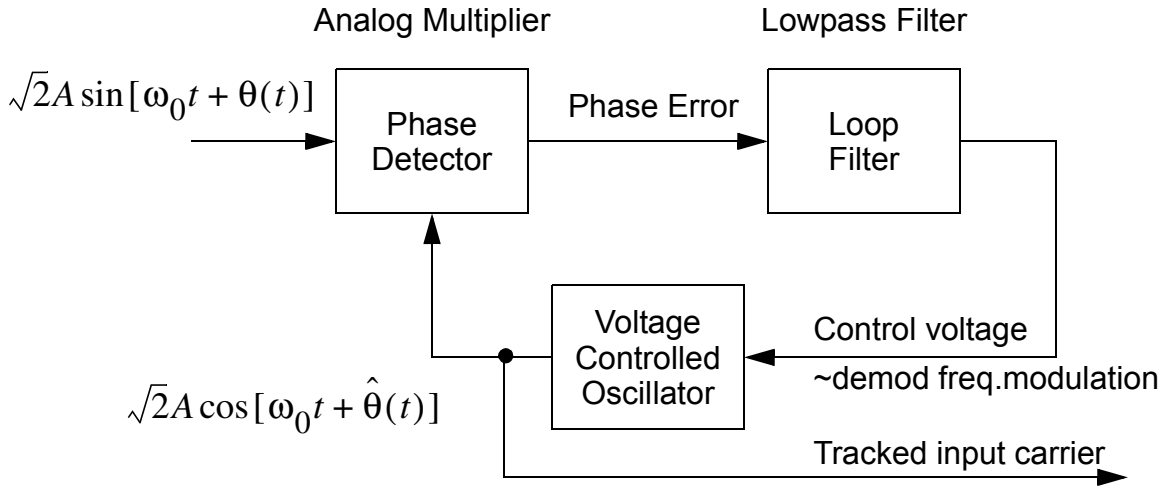
1.9.2 LPLL

The LPLL (Best) or analog PLL is the classical form of PLL. All components in the LPLL operate in the continuous-time domain. A LPLL block diagram is shown below:

[1] Roland E. Best, Phase Locked-Loops: Theory, Design, and Applications, fourth edition, McGraw Hill, 1999

- The phase detector is typically some form of analog multiplier, either a double-balanced mixed (DBM) or an active four-quadrant multiplier
- The phase error function is of the form

$$\phi(t) = \underbrace{K_m K_1 A}_{K_D} \sin\left[\theta(t) - \hat{\theta}(t)\right] \overset{\text{small error}}{\approx} K_D\left[\theta(t) - \hat{\theta}(t)\right]$$



Classical analog PLL

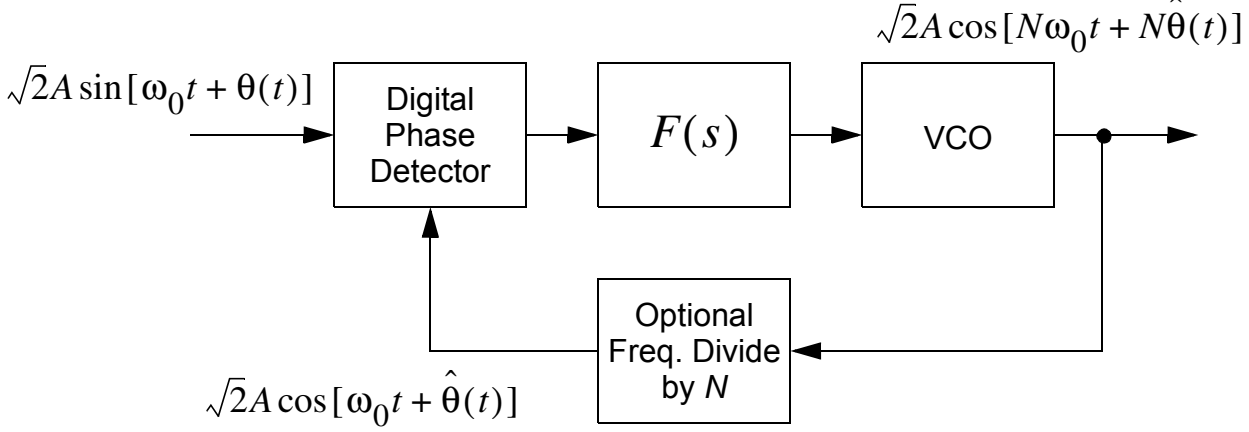
- The loop filter may be active or passive, but it typically results in the loop being either first-order or second-order
- The design/analysis of the loop filter makes use of the Laplace transform
- Loop filter system functions, $F(s)$, include:

Loop Order	Filter $F(s)$
1	1
2(perfect integrator)	$\frac{1+s\tau_2}{s\tau_1}$
2(imperfect integrator)	$\frac{1+s\tau_2}{1+s\tau_1}$
2(lag or lowpass)	$\frac{1}{1+s\tau}$

1.9.3 DPLL

The digital PLL is really just an analog PLL with a digital phase detector.

- The DPLL is really a hybrid system
- The DPLL is very popular in synthesizer applications
- In the above figure the optional digital divider, and variations on it, are used in frequency synthesis applications
- Popular types of digital phase detectors include:
  - Exclusive or gate (EXOR)
  - Edge-triggered JK-flipflop
  - Phase frequency detector (PFD)



Classical digital PLL (digital phase detector)

1.9.4 ADPLL

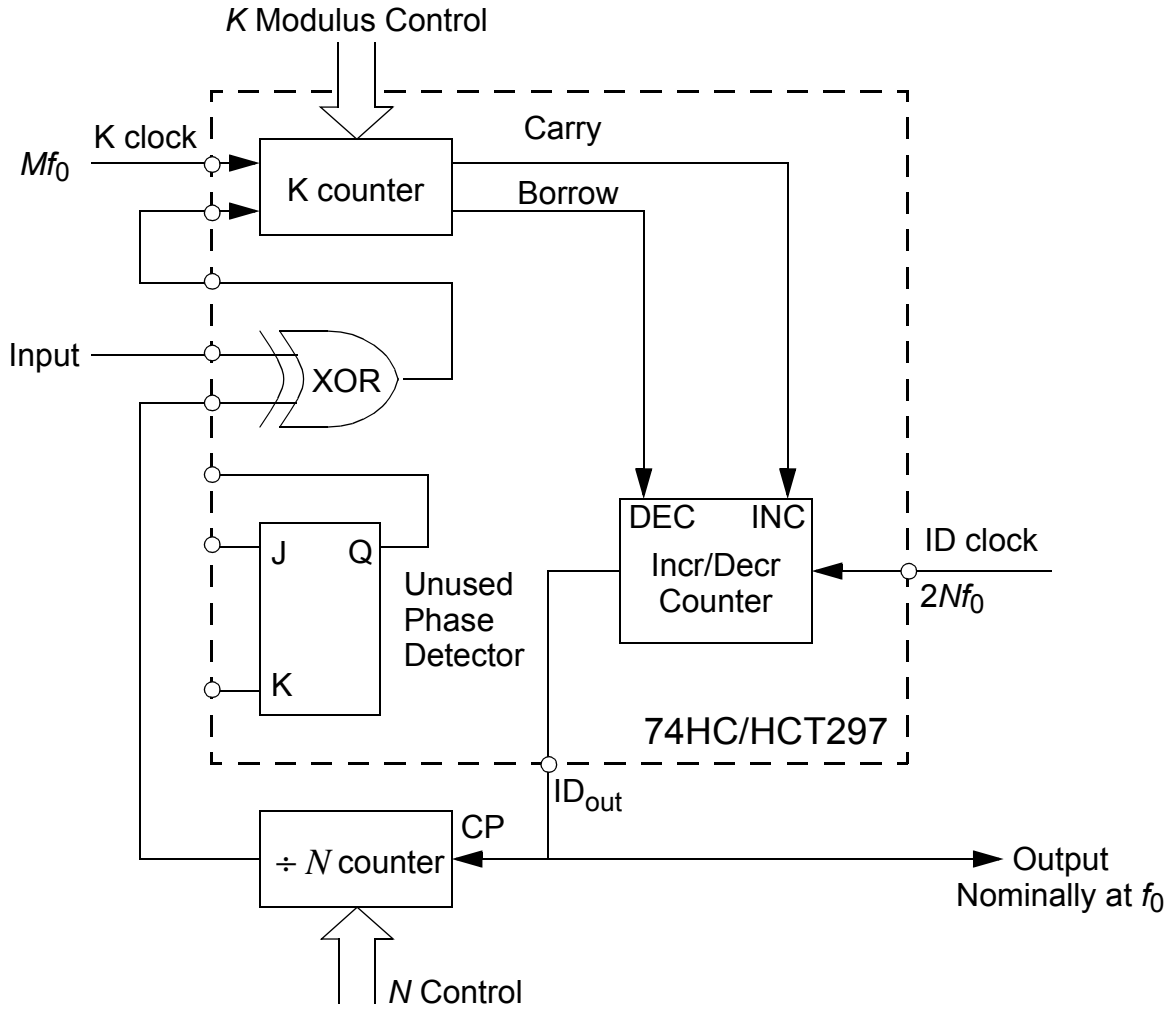
The all-digital PLL (classical all-digital) is distinctly different from the first two PLLs mentioned thus far.

- The ADPLL is a digital loop in two senses:

- All digital components
- All digital (discrete-time) signals

- There are many ADPLL building blocks, and many variations on putting them together
  - The ADPLL is also closely related to the SPLL

- The VCO is replaced by a digitally controlled oscillator (DCO) or also called a numerically controlled oscillator (NCO)

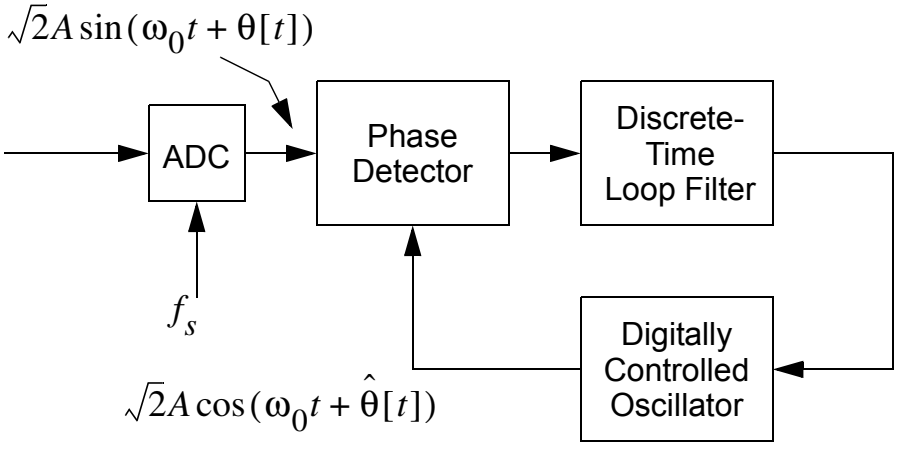


All-digital PLL as implemented in the part 74xx297

1.9.5 SPLL

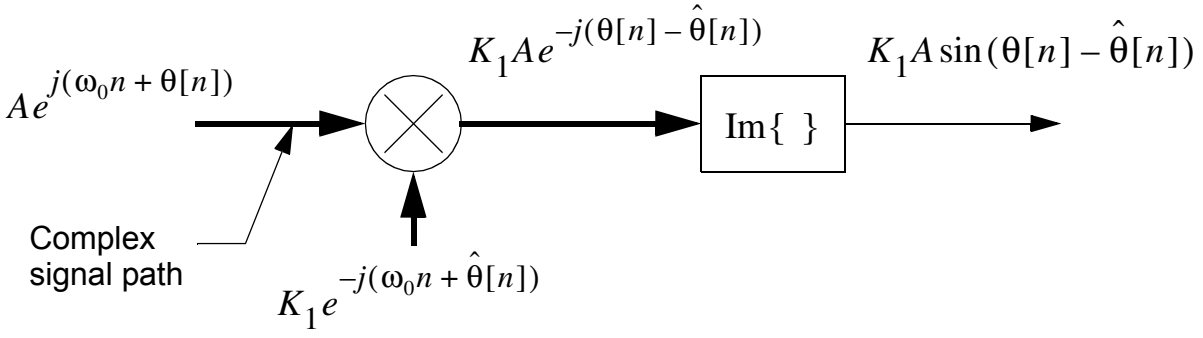
The SPLL can be viewed as a discrete-time implementation of either the LPLL or the DPLL. The block diagram of a generic SPLL is shown below.

- The implementation of the loop filter is typically a difference equation
- The design/analysis of the loop filter is done using the z-transform
- The SPLL is feasible due to the availability of digital signal processing (DSP) oriented microprocessors
  - A DSP processor features a Harvard architecture (separate data and program memories)
  - Pipelined instruction capabilities
- The computer hardware requirements are relatively high, but the flexibility is also very great
- By designing the SPLL to closely match say an LPLL we can effectively simulate analog PLL designs using software tools



A software or discrete-time signals based PLL

- The SPLL is very popular in digital communication applications
- For complex baseband systems a popular phase detector is again the sinusoidal phase detector, but now it takes the following form

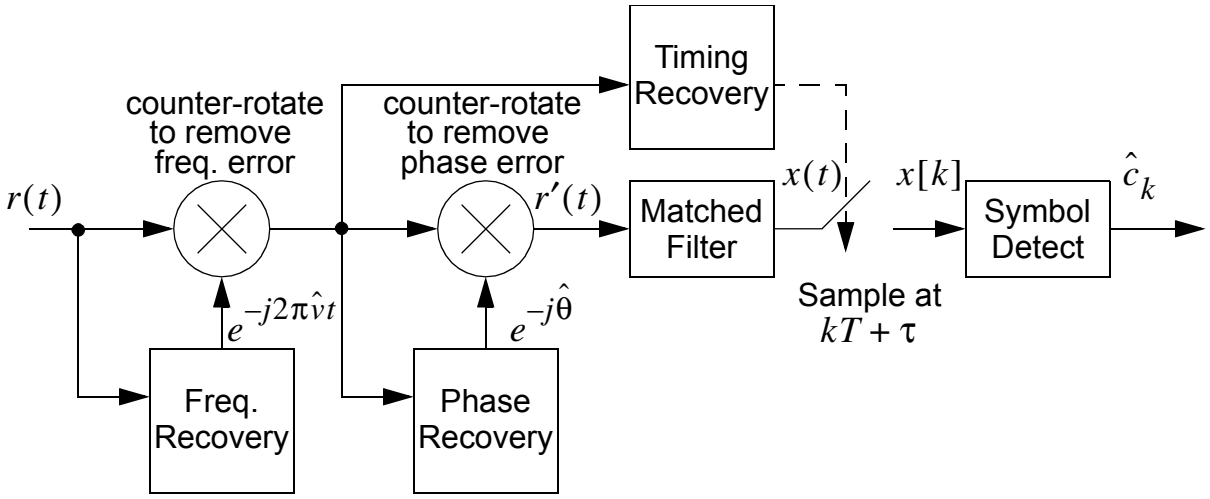


Complex signal form of SPLL

1.10 Applications Overview

1.10.1 Synchronization for Digital Communications

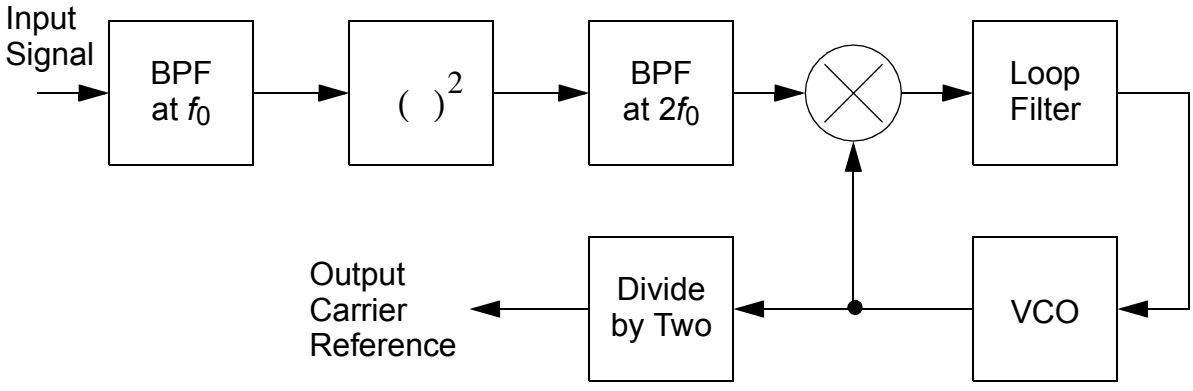
- In digital communications various levels of synchronization are required
- Systems that only operate at baseband typically require fewer total function than those at passband or RF
- The synchronization problem can viewed as one of parameter estimation from waveforms
- The three parameters most often referred to are, carrier frequency, f_0, carrier phase, θ, and symbol timing, τ
- Taken jointly they form a parameter vector with three components



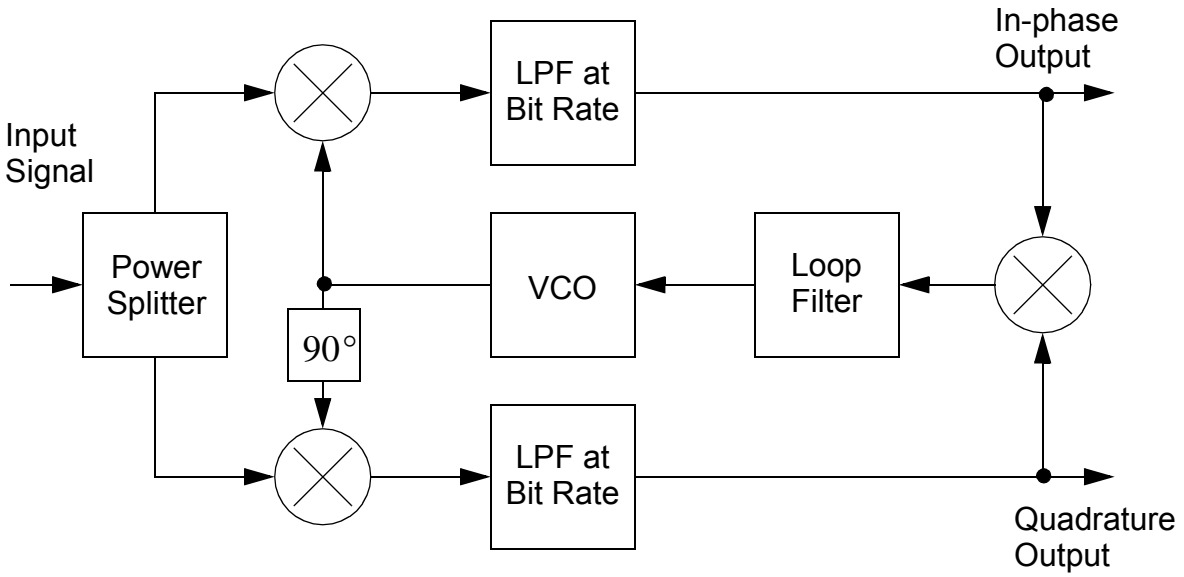
Synchronization in digital communications using complex signals

- The figure above depicts analog signal processing, but in practice combinations of analog and digital signal processing are required
- Parameter tracking can be implemented using feedback, as in PLL based techniques
- Feed-forward methods are also possible
- Some block diagrams of synchronization systems are shown next

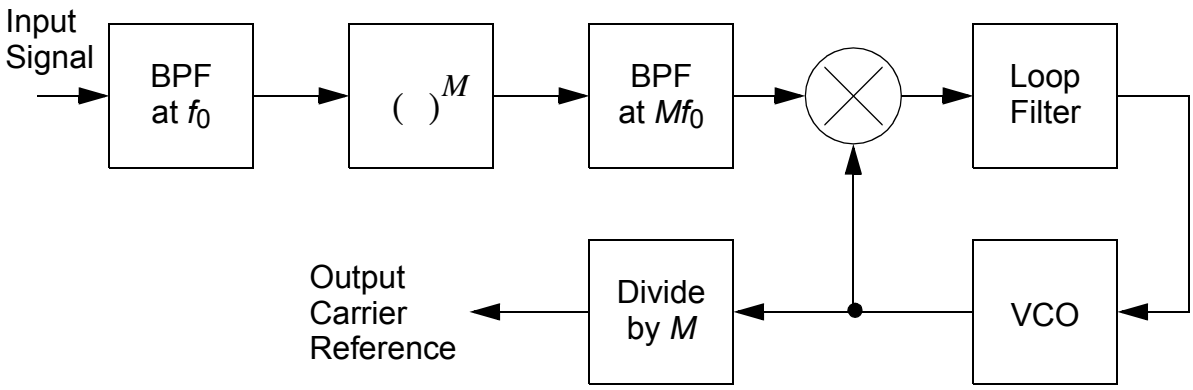
Coherent Carrier Reference Generation



Squaring loop for suppressed carrier modulation



Costas loop which performs at baseband the operation of the squaring loop



Mth-power loops for M-ary phase shift keying

A DSP Based Mth Power and Decision Directed Approach

- In bandpass digital communication systems the carrier frequency is generated from a local timing reference
- At the receiver coherent demodulation requires that the same carrier frequency (or a translated version) and phase be used for demodulation

- A coherent carrier recovery system, which tracks the frequency and phase of the carrier at the receiver, is used for this purpose

- The received signal (noiseless) is of the form

$$x(t) = \mathrm{Re}\left\{s(t)e^{j[\omega_c t+\theta(t)]}\right\}$$

- Consider a complex baseband information carrying waveform given by (here pulse amplitude modulation PAM)

$$y(t) = e^{j[\omega_c t+\theta(t)]} \sum_{m=-\infty}^{\infty} A_m p(t - mT)$$

  where A_m is the data, $p(t)$ is the transmit pulse shape, T is the symbol period, and $q(t)$ is the frequency offset and phase jitter

- We need to form an estimate of the received carrier phase $q(t)$ so that the above signal can be demodulated with

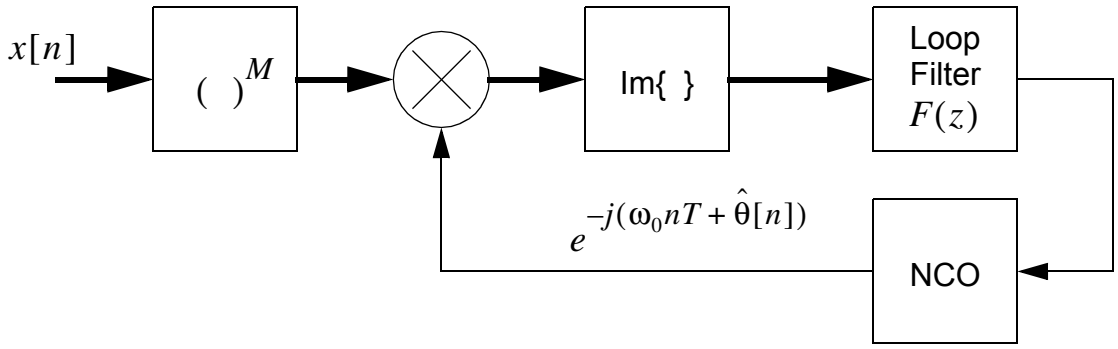
$$e^{-j[\omega_c t+\hat{\theta}(t)]}$$

- In a digital (DSP) based receiver we may sample the signal at the symbol rate (or a submultiple), the product of the carrier estimate with the input to obtain

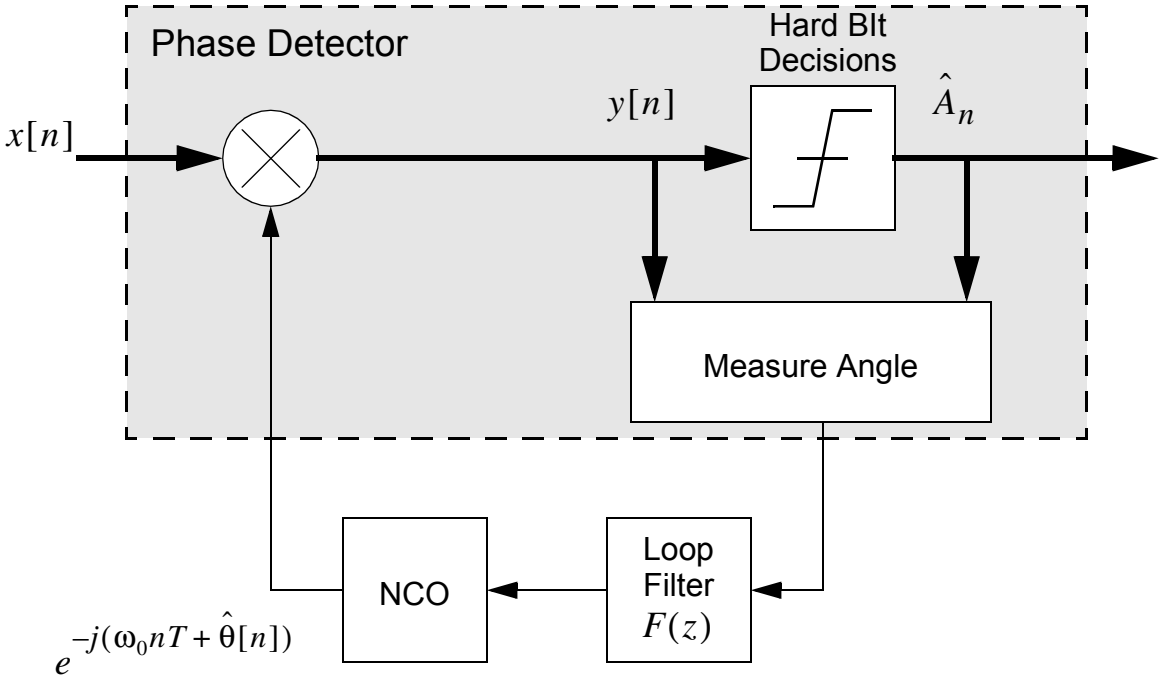
$$y[n] = e^{j[\theta(nT)-\hat{\theta}(nT)]} \sum_{m=-\infty}^{\infty} A_m p(nT - mT)$$

- Assuming that $p(nT - mT) = \delta(n)$ the samples we obtain are of the form

$$y[n] = e^{j(\theta[n]-\hat{\theta}[n])} A_n$$

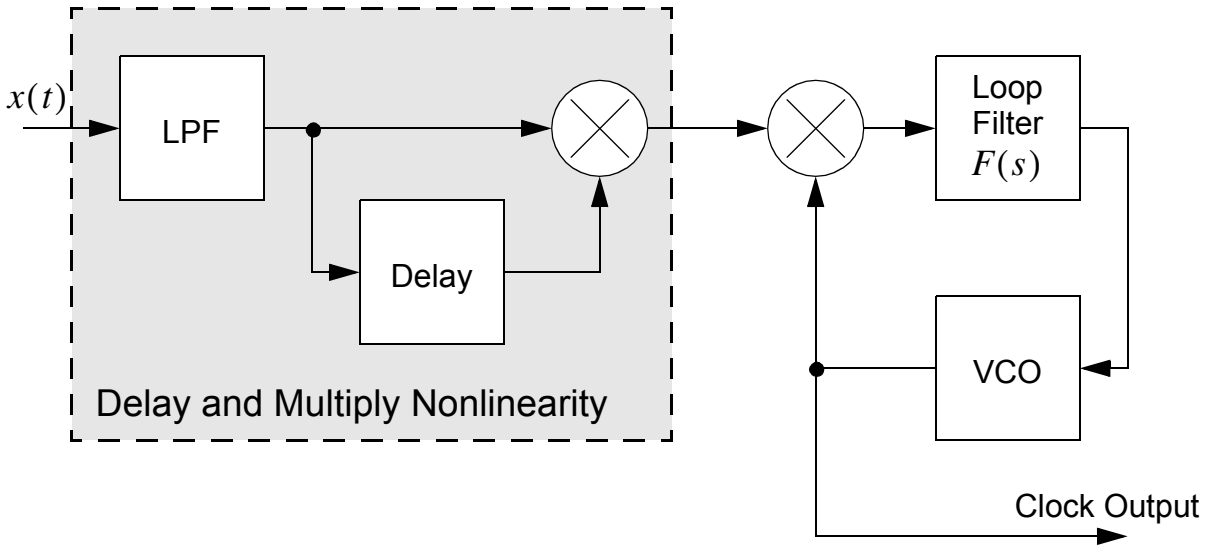


Mth Power Loop

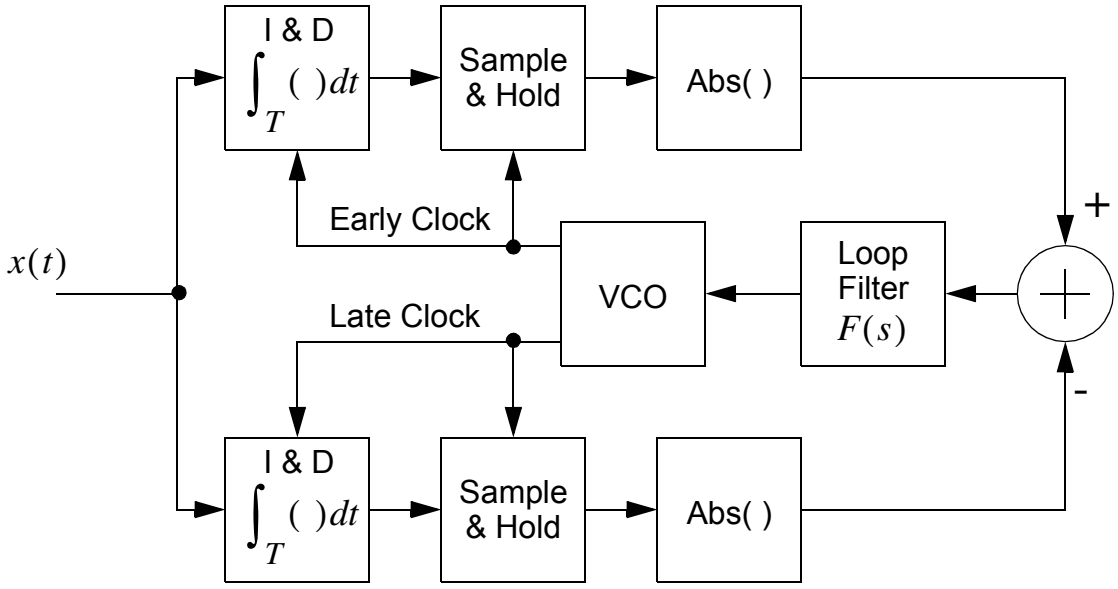


Decision Directed Loop

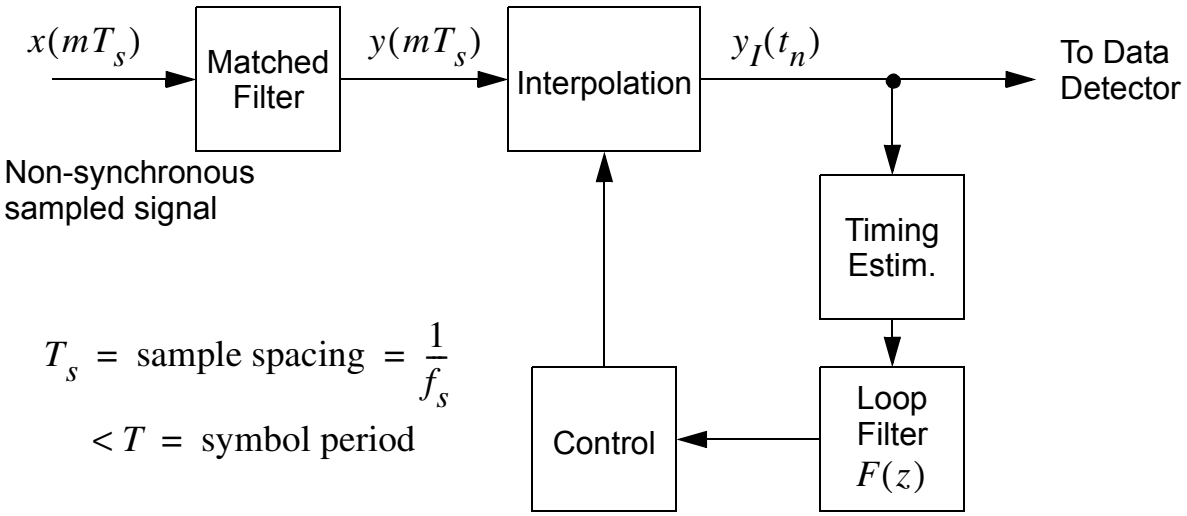
Symbol/Bit Synchronization



Nonlinear filter clock recovery using a delay and multiply circuit

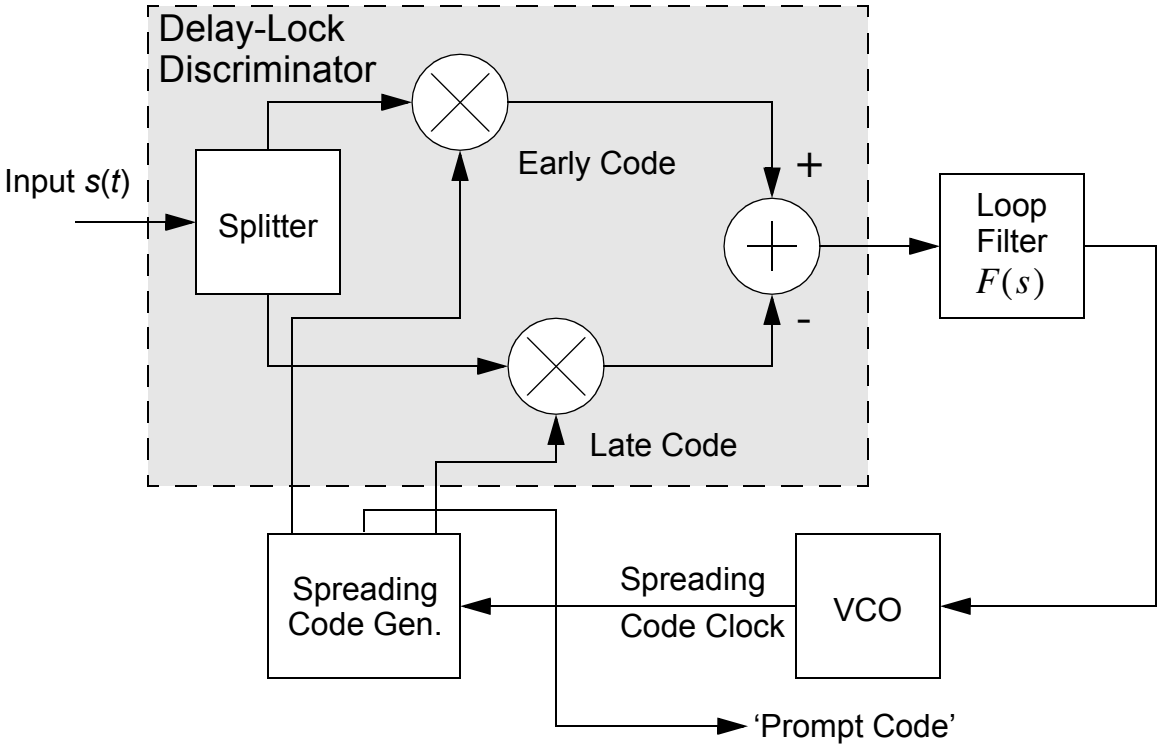


Early-late gate timing recovery



A DSP based non-synchronous clock recovery loop

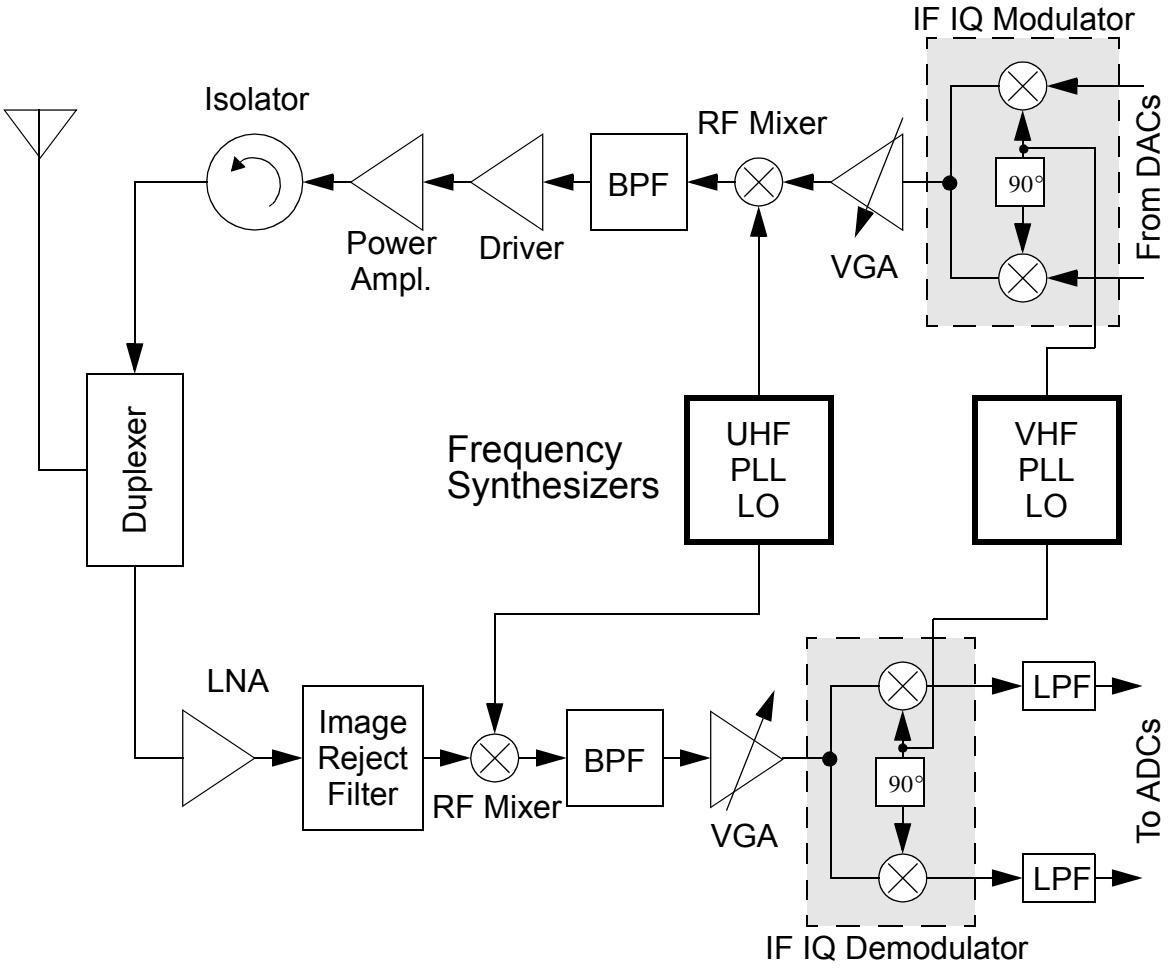
Spread Spectrum Code Synchronization



Delay-lock Loop

1.10.2 Frequency Synthesis Applications

- In wireless applications frequency synthesizers provide local oscillators for up and down conversion of modulated signals
- Any radio based electronics that operates over multiple frequencies, likely incorporates a frequency synthesizer
- The transmitter and receiver of a cellular telephony handset is shown below



Wireless handset transmitter and receiver RF signal processing

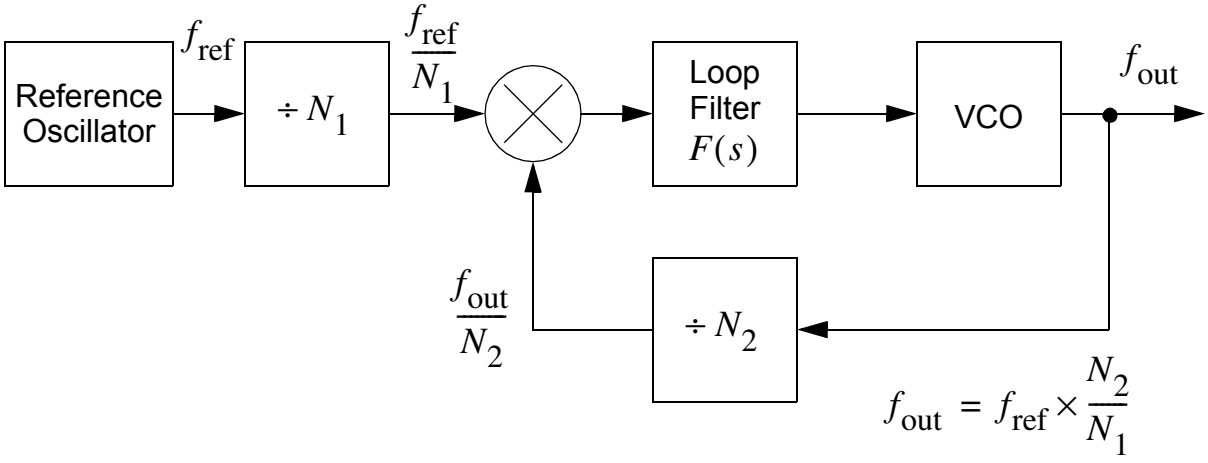
Direct Synthesis

- Use mixing (multiplication) to build up the desired frequency and a division of a single reference frequency

- Rapid switching time is one of the features
- Bulky rf hardware (switches, mixers, filters) is a disadvantage
- Spurious frequency generation can be a big problem

Indirect Synthesis

- Indirect synthesis is another name for PLL frequency synthesis
- A basic PLL synthesizer is the following



General indirect synthesis using a DPLL

- With indirect synthesis all by itself, large divide ratios are required to obtain fine resolution
- Being able to set the divide ratio to a fractional, non-integer value would solve this problem
- The basis of the *fractional-N* method is to alter the divide ratio N_2 between two values

Direct Digital Synthesis (DDS)

- DDS is a DSP based method of using an N-bit accumulator to generate a phase ramp corresponding to one clock cycle

– The accumulator increments the phase by an equivalent amount of $\Delta\theta$ each clock cycle

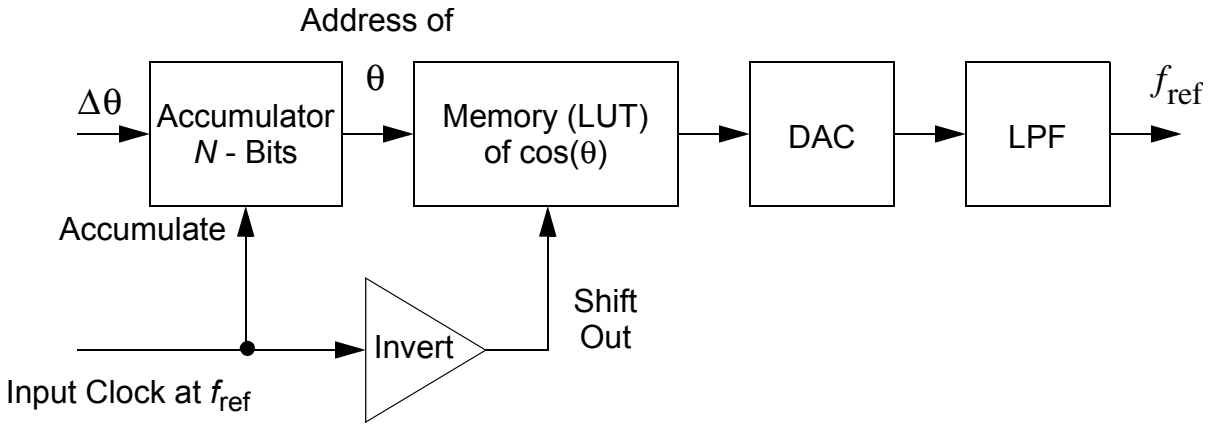
– The output frequency is given by

$$f_{\text{out}} = \frac{N_i}{2^N} f_{\text{ref}}$$

where N_i corresponds to the phase step size $\Delta\theta$

– The minimum frequency change is

$$\Delta f = \frac{f_{\text{ref}}}{2^N}$$



A simple direct digital synthesis scheme

Hybrid Methods

- To meet various design goals combinations of the above method may be employed

  – Several PLL synthesizers can be combined to create a multi-loop synthesizer

  – DDS and a PLL can be combined to achieve fine step sizes, yet the wide tuning range of a PLL

1.11 Simulation Examples

In this section several working PLL based circuits/systems will be demonstrated using a variety of software tools. The intent is to give a flavor of what options are available in the way of simulation approaches for PLL based designs.

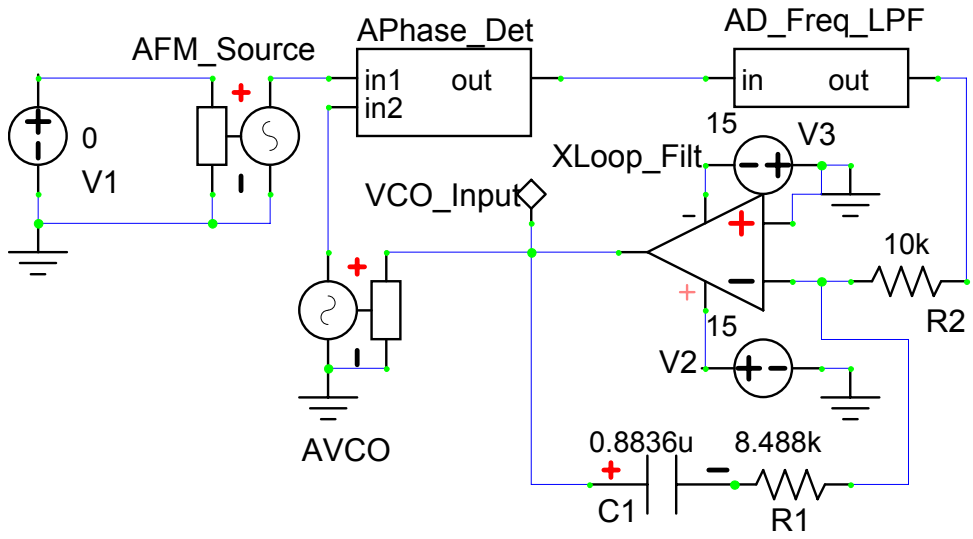
Example 1.1: **Behavioral Level Modeling using XSpice**

Analog circuit designers are well acquainted with Spice. The challenge with PLL designs is to effectively make use of behavioral modeling capabilities of modern Spice implementations. One such extension to Spice is XSpice, developed at Georgia Tech.

- We will soon learn that the full simulation of a PLL is numerically intensive because waveform fidelity at the VCO level is required, yet the bandwidth of the closed loop tracking system is much smaller than the VCO quiescent frequency

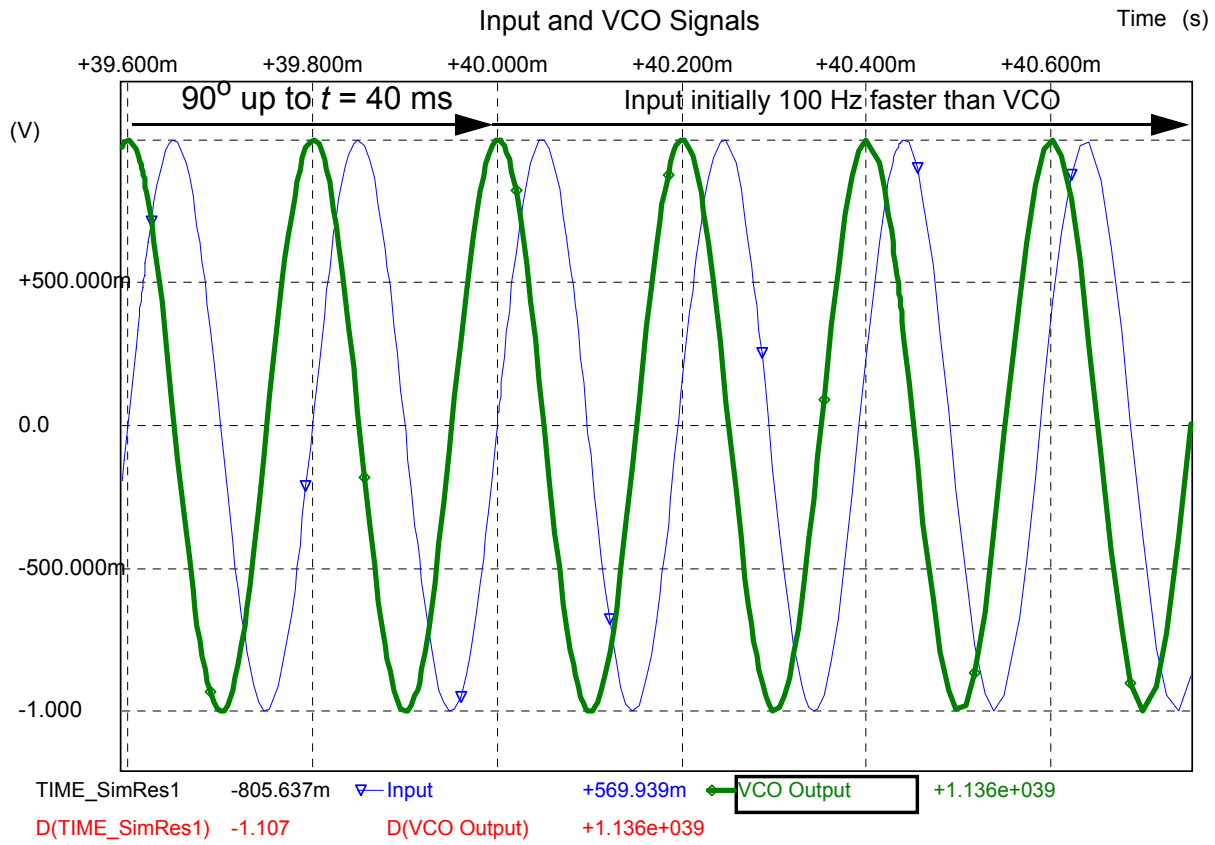
- In this example full waveform fidelity is preserved, but behavioral models, as opposed to detailed circuit models, are used for most of the loop

- The VCO quiescent frequency is chosen to be just 5000 Hz and the PLL closed-loop noise equivalent bandwidth is 100 Hz



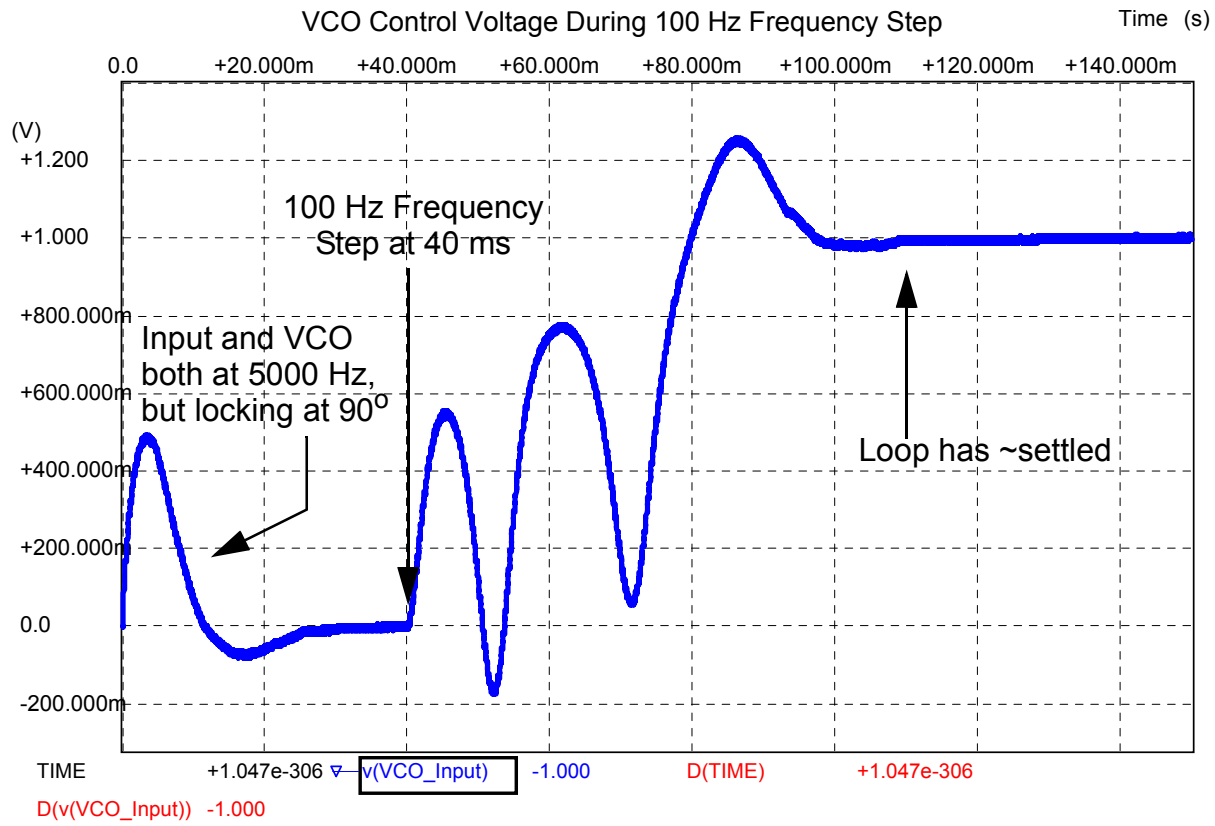
Analog PLL (LPLL) schematic using XSpice behavioral modeling

- The input signal source and VCO are both created using a voltage controlled sinusoid generator
- The phase detector is an ideal multiplier and to remove the double frequency term following the phase detector, a transfer function behavioral level model is used
- The transfer function associated with the block `AD_FREQ_LPF` is $H(s) = 1000/(s + 1000)$, a lowpass filter with cutoff frequency $f_c = 1000/(2\pi) = 159.15$ Hz
- A loop filter of the form $F(s) = (1 + s\tau_2)/(s\tau_1)$ is implemented in circuit form using an Op-amp
- The input sinusoid is initially at 5000 Hz and the VCO quiescent frequency is also at 5000 Hz
- At 40 ms the input frequency is stepped to 5100 Hz, a 100 Hz frequency step, (the VCO gain is $K_v = 100$ Hz/v)



The input sinusoid and the VCO output near $t = 40$ ms

- The loop breaks lock and acquires in both frequency and phase-lock via the nonlinear *pull-in process*
- In about 70 ms the loop has settled ($t = 110$ ms)
- In addition to transient or time domain modeling, Spice can be used in the AC analysis mode to characterize open and closed loop magnitude and phase response of linearized loops
- The AC analysis mode also provides for detailed noise analysis and the incorporation of *phase noise* models of the input source oscillator and VCO
- Ultimately closed-loop phase noise performance can be obtained using actual circuit elements mixed with behavioral level models



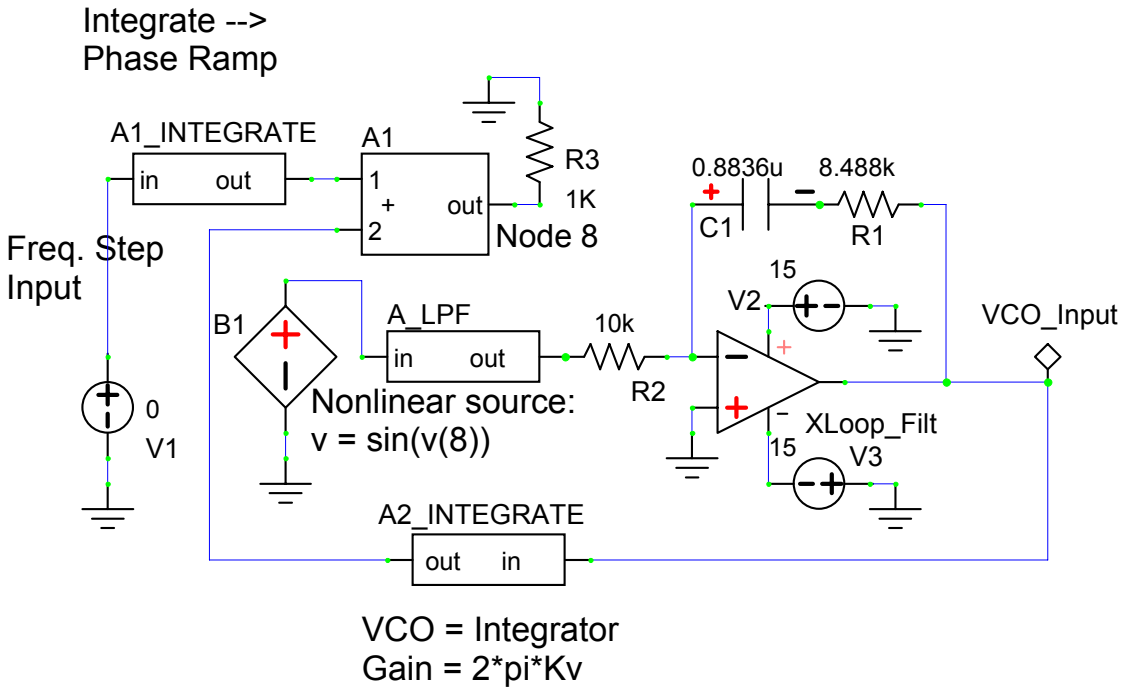
VCO control voltage in response to a 100 Hz frequency step at $t = 40$ ms

Example 1.2: **Baseband PLL in XSpice**

In this example high frequency modeling, e.g., the individual waveform cycles of the input and VCO are removed by abstracting to what is known as a baseband PLL model. This model will be developed in Chapter 2 of the course notes.

- The multiplier phase detector is replaced by the phase difference block followed by a nonlinearity, here a sin function
- To model a frequency step an integrator at the input converts frequency to phase with a 2π factor times a frequency deviation constant

- The VCO, which is like a frequency modulator, is modeled as an integrator with gain constant $2\pi \times K_v$, where $K_v = 100$ Hz/v in this case

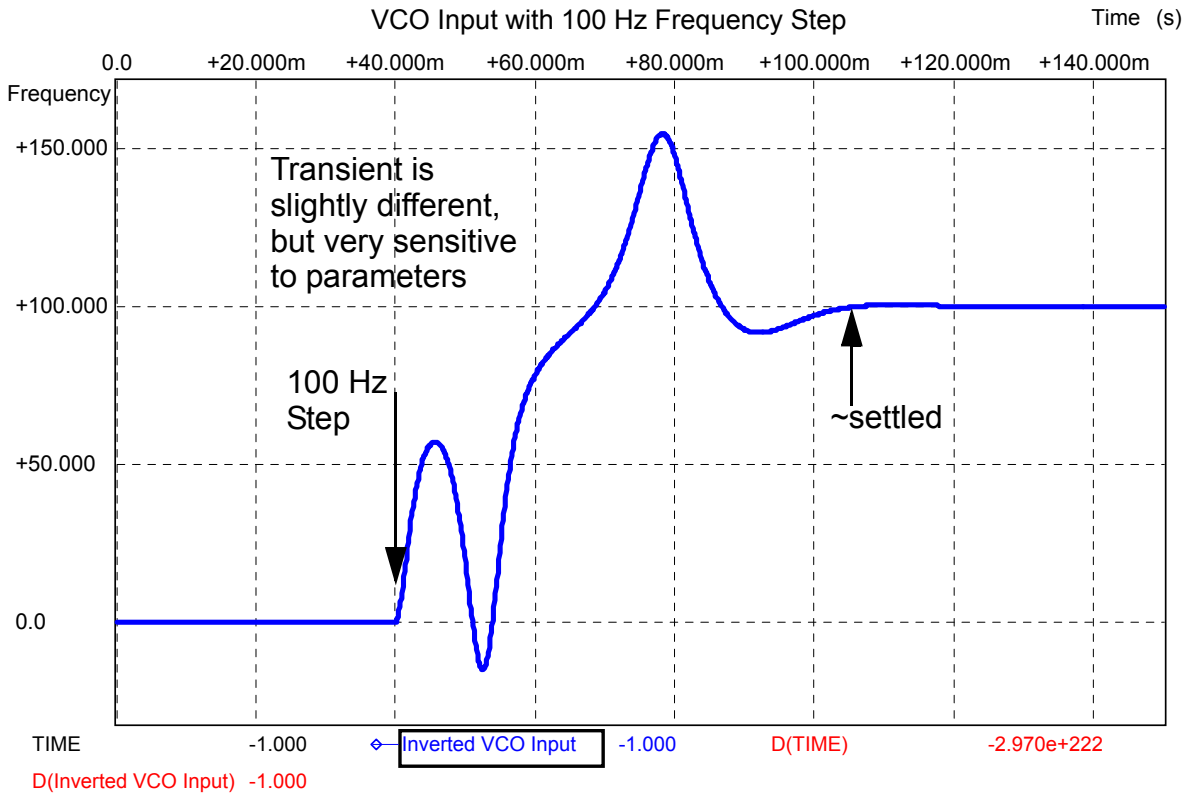


Baseband analog PLL schematic using XSpice behavioral modeling

- A 100 Hz frequency step is applied at the input at $t = 40$ ms as in the previous example

- Loop transient 'pull-in' response is slightly different than before, but the total time required to settle is very similar

- The waveform plotted is the VCO control voltage inverted to account for the '+' sign in the summer block, and then scaled by 100 to scale the waveform to instantaneous frequency deviation in Hz



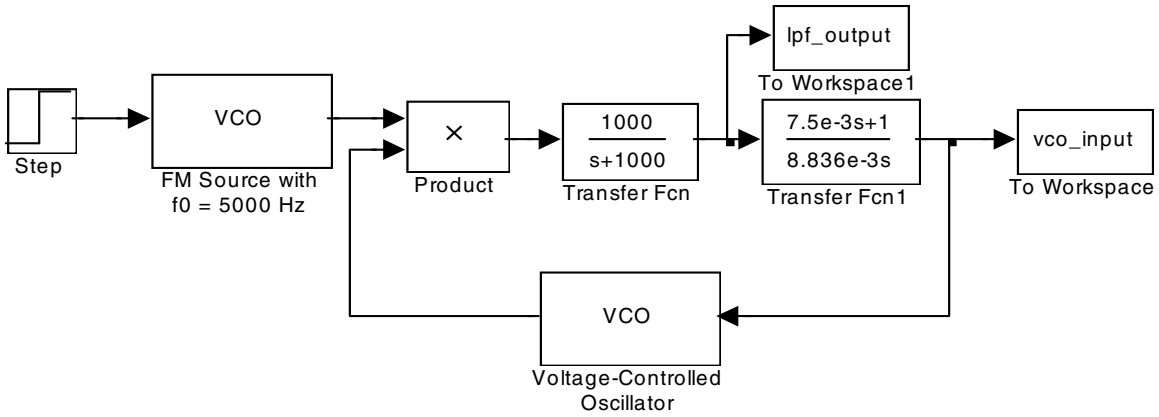
VCO input in response to a 100 Hz frequency step

Example 1.3: **Analog PLL Modeling in MATLAB Simulink**

Spice is useful for modeling certain aspects of PLL circuits, but there are times when a complete abstraction to the system level is the most efficient means to validate a design. In this example Simulink, which is an odd-on to MATLAB (included with the student version) is used.

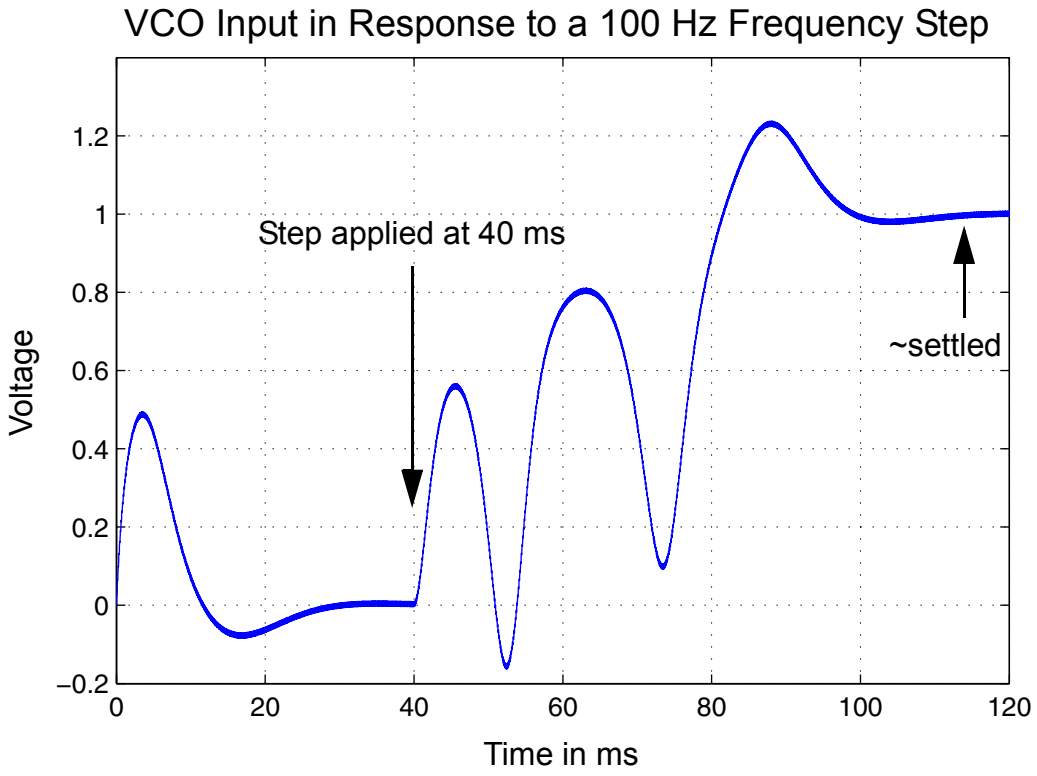
- The Egan text includes MATLAB m-files for complete command line nonlinear PLL simulation
- Simulink is a block diagram simulation environment, so PLL simulation here is more visual

- Simulink alone does not provide direct support for PLL's and synchronization
  - If you add to Simulink the Signal Processing toolbox, the DSP blockset, the Communication toolbox, and finally the Communication blockset, you have some basic PLL building blocks, e.g. a VCO
  - With the basic Simulink library that comes with the student version of Simulink PLL simulation requires a bit more work, but is possible
- In the example given here Simulink is used to model a bandpass analog PLL with 5 kHz center frequency and 100 Hz loop bandwidth, as in Example 1.1



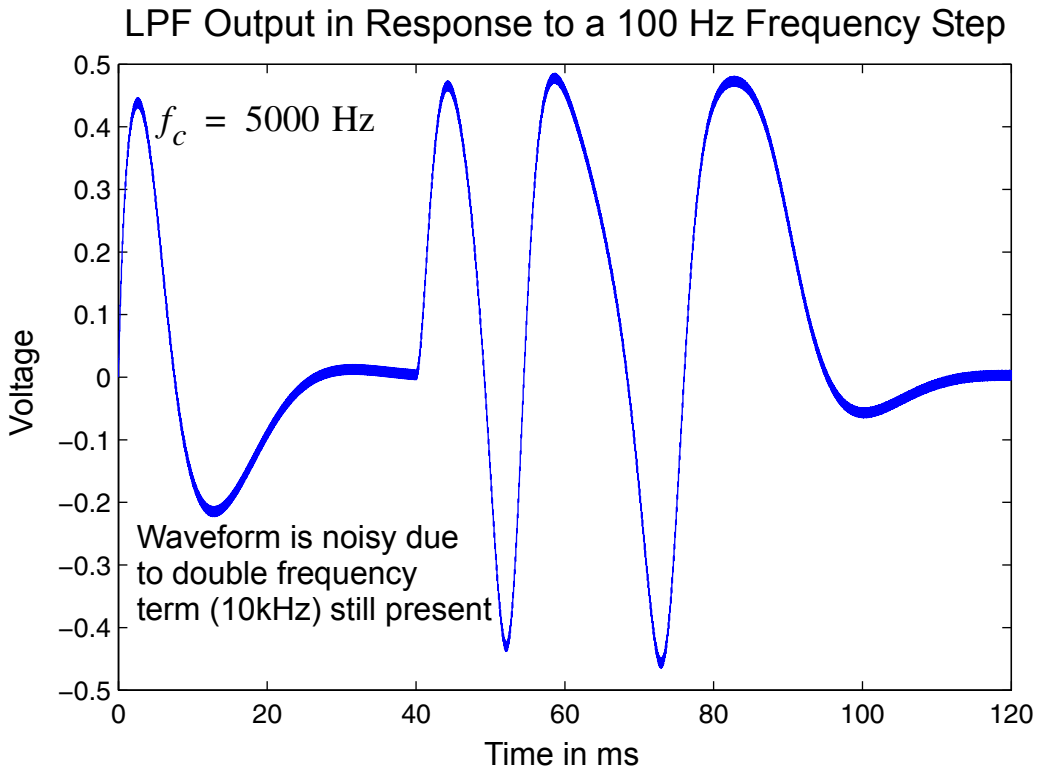
MATLAB Simulink model of an analog PLL at 5 kHz

- The first-order lowpass filter following the multiplier phase detector is used to remove the double frequency term at 10 kHz
- A 100 Hz frequency step is applied to the input at 40 ms
- Results very similar to the XSpice model of Example 1.1 are obtained



VCO input in response to a 100 Hz frequency step

- A small amount of noise can be seen on the VCO control signal
- This noise in reality is the double frequency term, at 10 kHz, leaking through the loop filter
- This leakage will modulate the VCO and produce annoying FM sidebands or spurs
- This leakage is even more evident at the output of the lowpass filter



Phase detector lowpass filter output showing 10 kHz leakage

Example 1.4: **Analog PLL Modeling in VisSim**

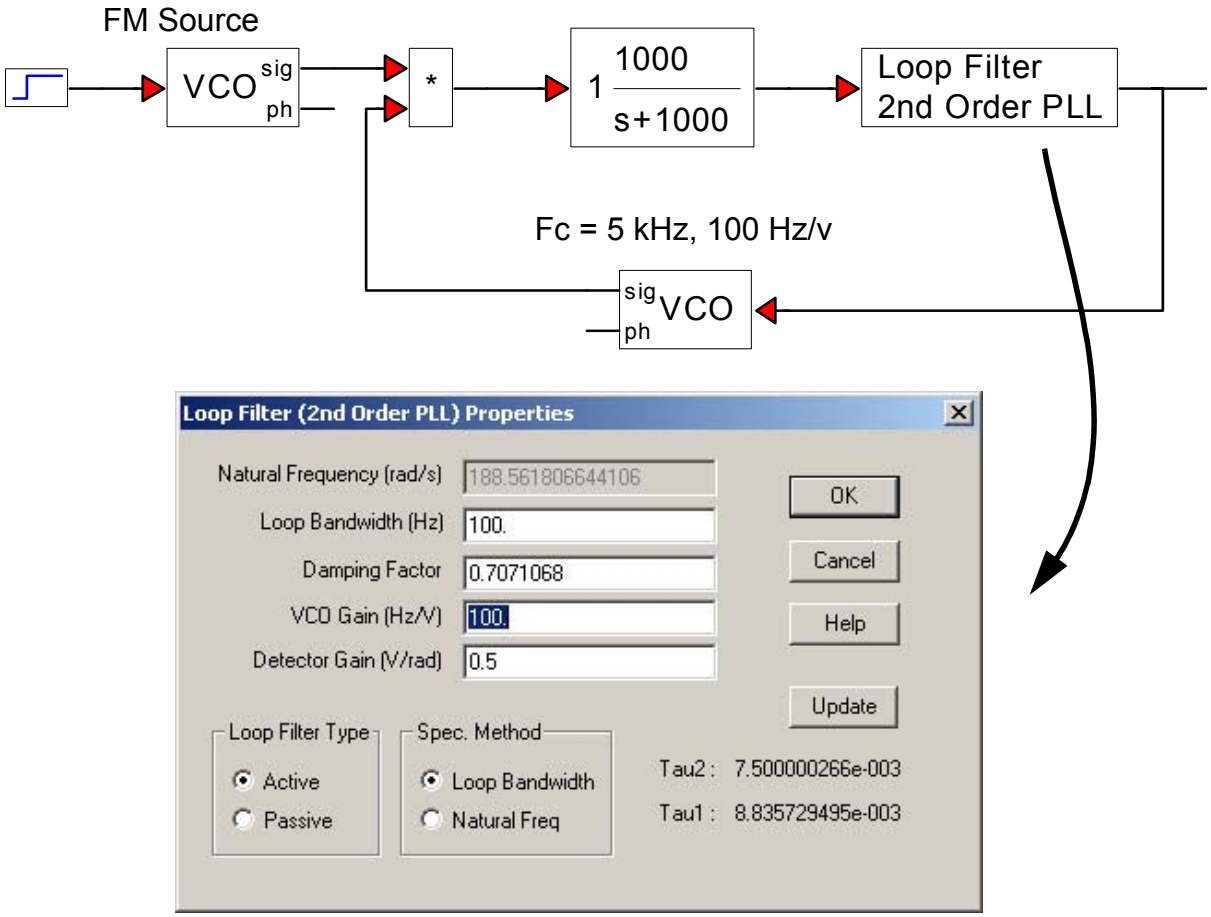
In this example we consider the block diagram modeling tool VisSim/Comm from Visual Solutions Inc. With the readily available Communications library support for PLL simulation is very convenient.

- VisSim is similar to Simulink in that it is a block environment for dynamic system simulation
- With the VisSim/Comm many powerful communications blocks are available, and complete *complex baseband* simulation blocks are also defined

- VisSim also has an interface with MATLAB, so results can be directly imported in to MATLAB, or MATLAB can perform calculations

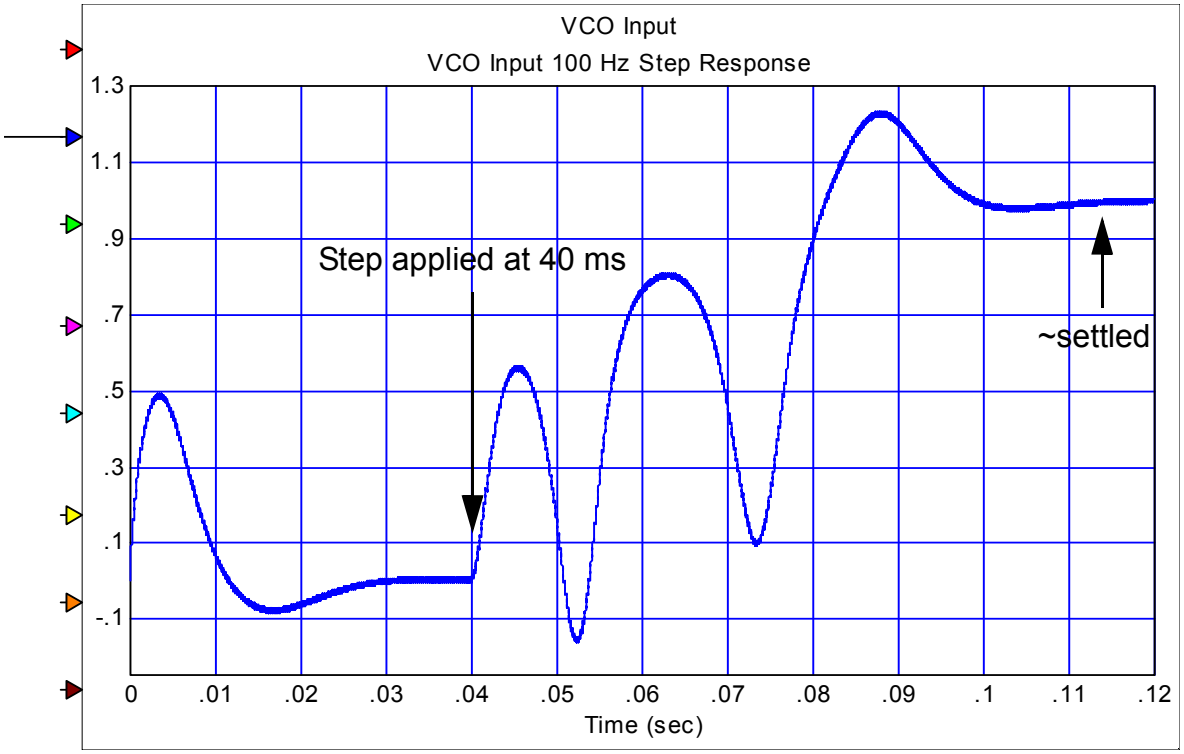
- Standard loop filters can be defined using loop filter blocks which directly calculate the required time constants, e.g., τ_1 and τ_2 from system level specifications, e.g., loop bandwidth and damping factor

- In this first VisSim example we repeat the original XSpice example of an analog PLL at center frequency of 5000 Hz and loop bandwidth of 100 Hz



VisSim/Comm model of an analog PLL at 5 kHz

- The VCO input signal (voltage) is measured when a 100 Hz frequency step is applied at 40 ms
- The results are consistent with the previous XSpice and Simulink results
- Note also that in VisSim the block diagram and graph windows are 'wired' together



VCO input in response to a 100 Hz frequency step

Example 1.5: **Complex Baseband PLL Modeling in VisSim**

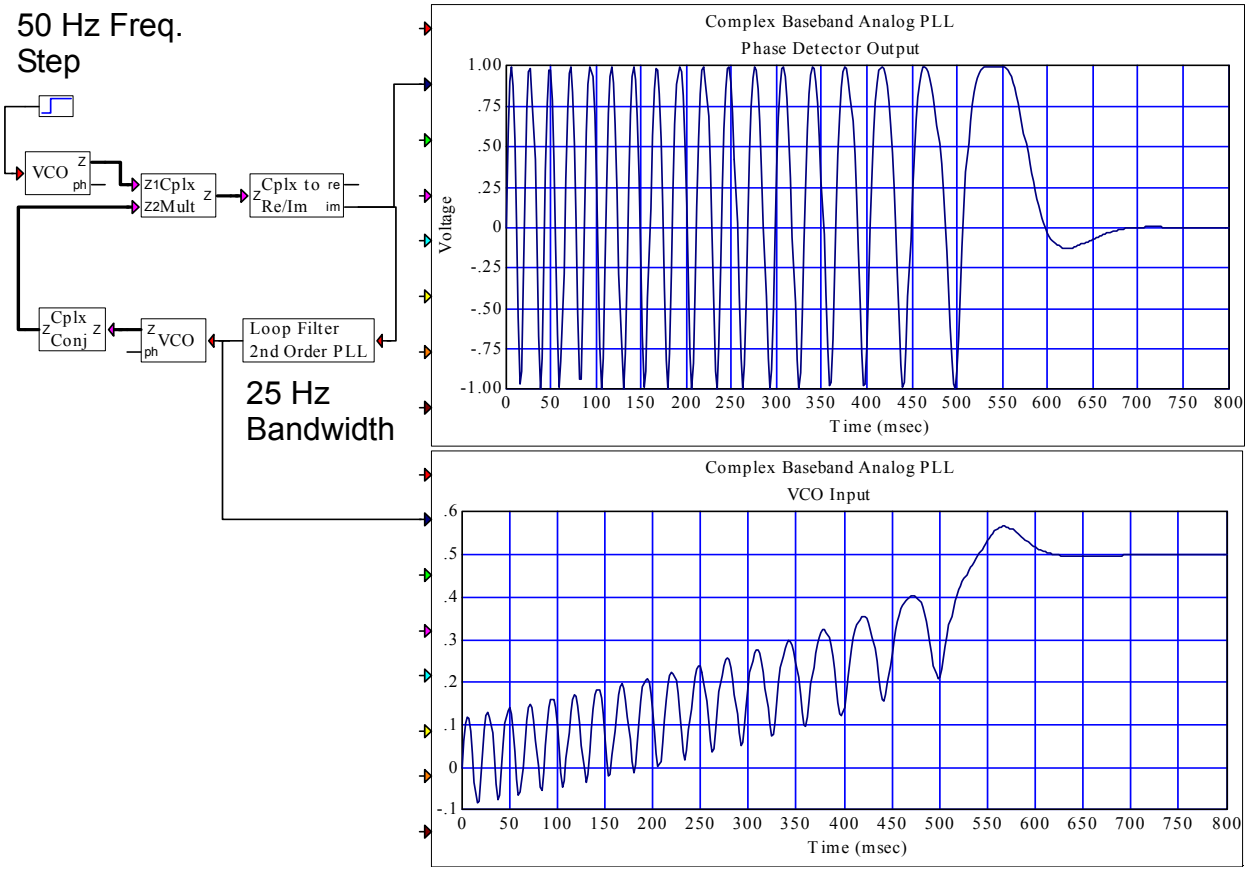
In digital signal processing based implementations, common place in digital communications, the PLL is likely top operate on complex signals, i.e., complex baseband. VisSim supports this via a complex VCO block and other complex processing blocks. In this example a complex baseband analog PLL is constructed.

- A complex baseband PLL has the advantages of the baseband PLL of Example 1.1, but takes complex signal inputs of the form

$$s(t) = Ae^{j]2\pi f_c t + \theta(t)]}$$

not phase inputs

- In this example a 2nd-order loop is constructed with center frequency at $f_c = 0$ (DC) having a loop bandwidth of 25 Hz
- A frequency step of 50 Hz is applied at $t = 0$ and both the sinusoidal phase detector output and VCO input signals are observed
- The entire VisSim simulation block diagram and plots are shown below

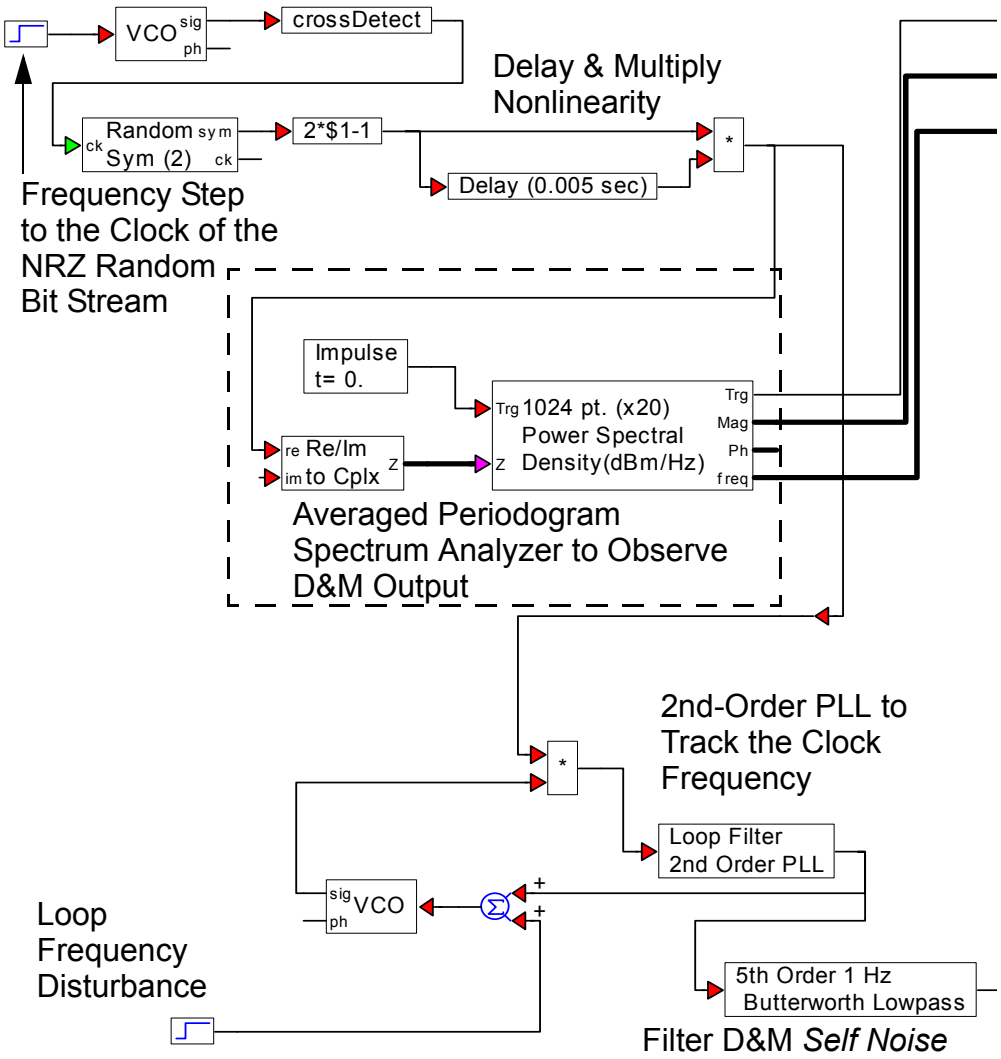


Complex baseband PLL with a 50 Hz frequency step input

Example 1.6: **Bit Synch Modeling in VisSim**

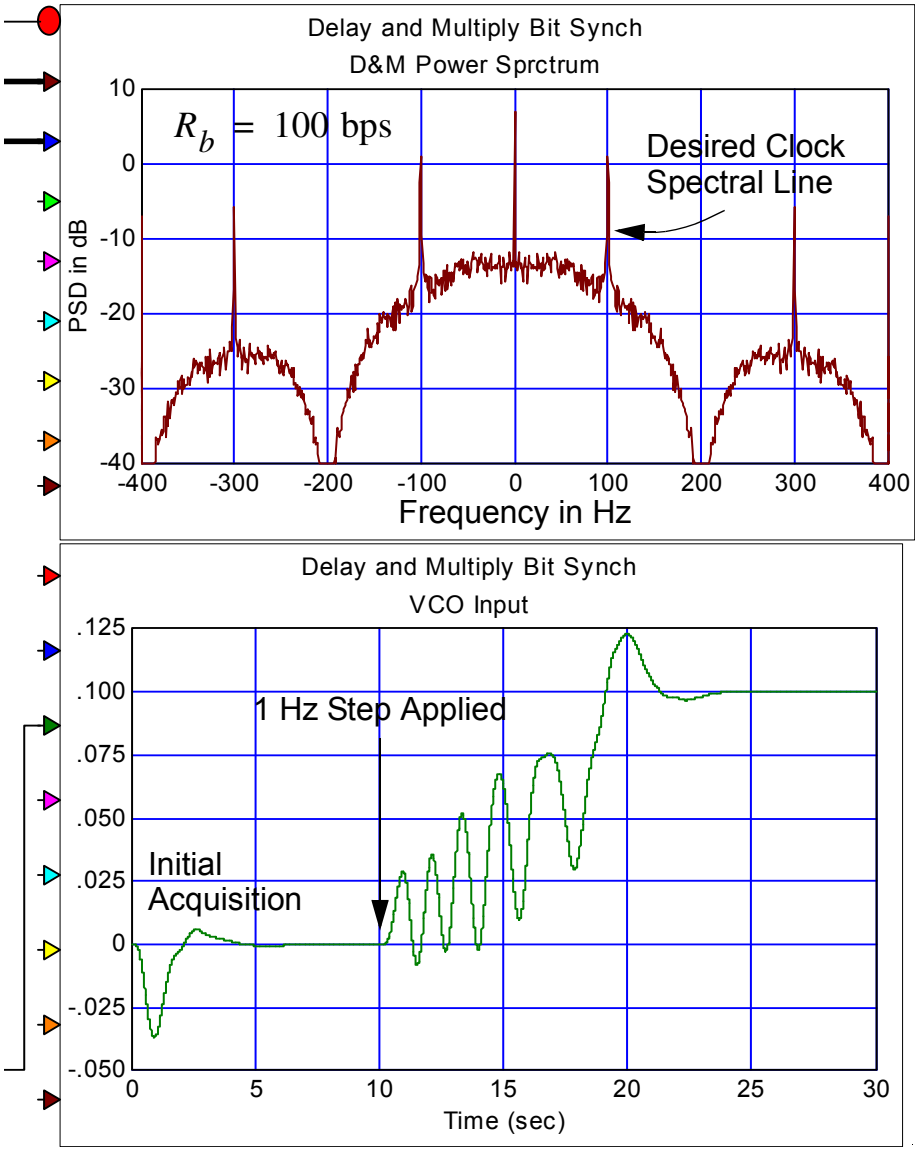
Clock recovery in baseband digital communications frequency employs a PLL.

- In this simulation example the assumed bit rate is 100 bps
- A delay and multiply (D&M) circuit is used to generate a spectral line from the NRZ (non-return-to-zero) binary bit stream
- The power spectrum of the D&M output is also computed



Delay-and-multiply bit synch block diagram

- A 2nd-order PLL with loop bandwidth of 1 Hz is configured with the D&M output as the input signal source
- To test the acquisition and tracking of the bit synch a 1 Hz frequency step is applied as a loop disturbance

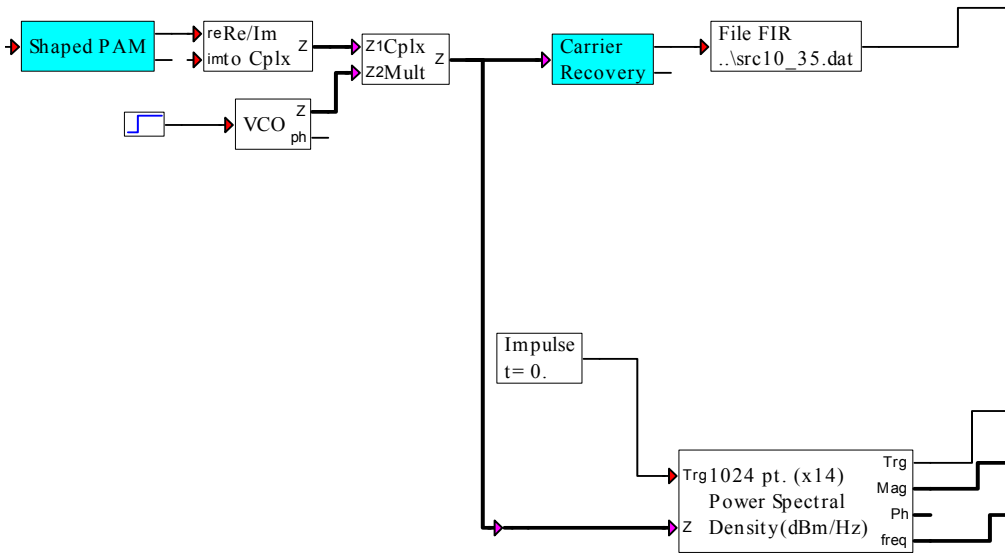


D&M output PSD and VCO input for a frequency step

Example 1.7: **Carrier Tracking in VisSim**

In this example a complex baseband implementation of carrier phase tracking is described.

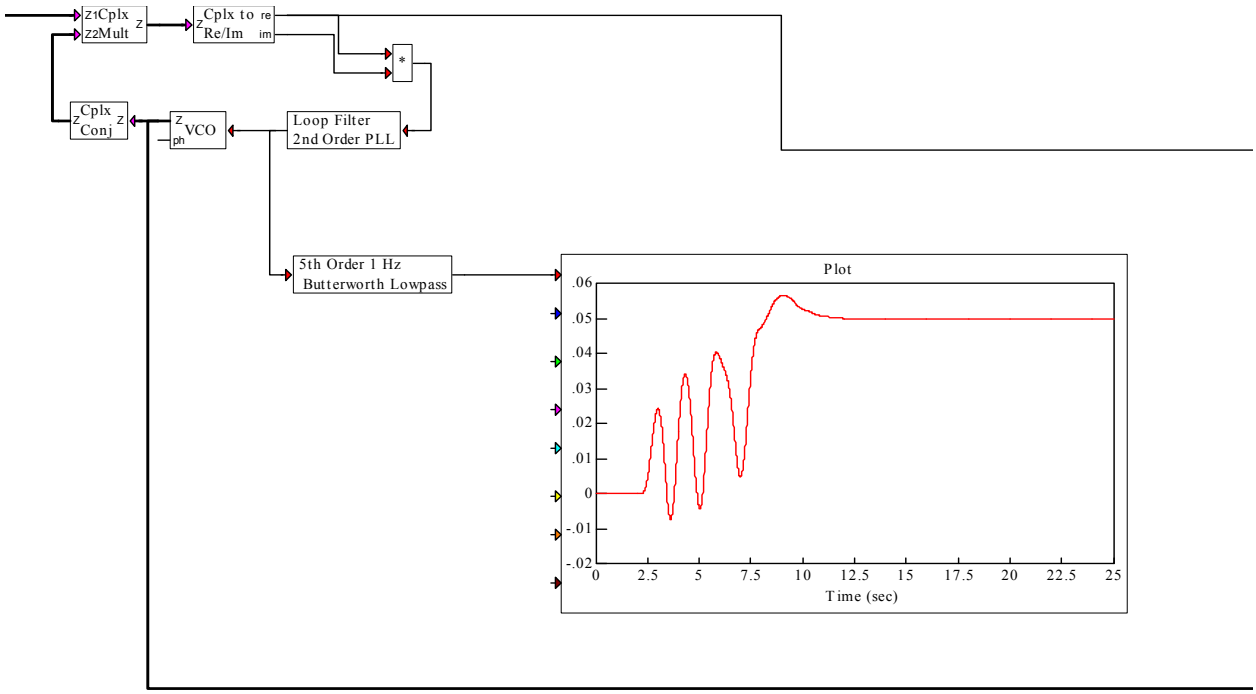
- The modulation scheme is binary phase-shift keying (BPSK) with square-root raised cosine (SRC) pulse shaping
- Pulse shaping filter FIR coefficients are imported from MATLAB, but could also be designed directly in the VisSim/Comm environment
- The bit rate is 10 bps and the carrier frequency is nominally 0 Hz
- To eliminate clutter a hierarchical block diagram is created



Shaped BPSK with Costas carrier phase tracking

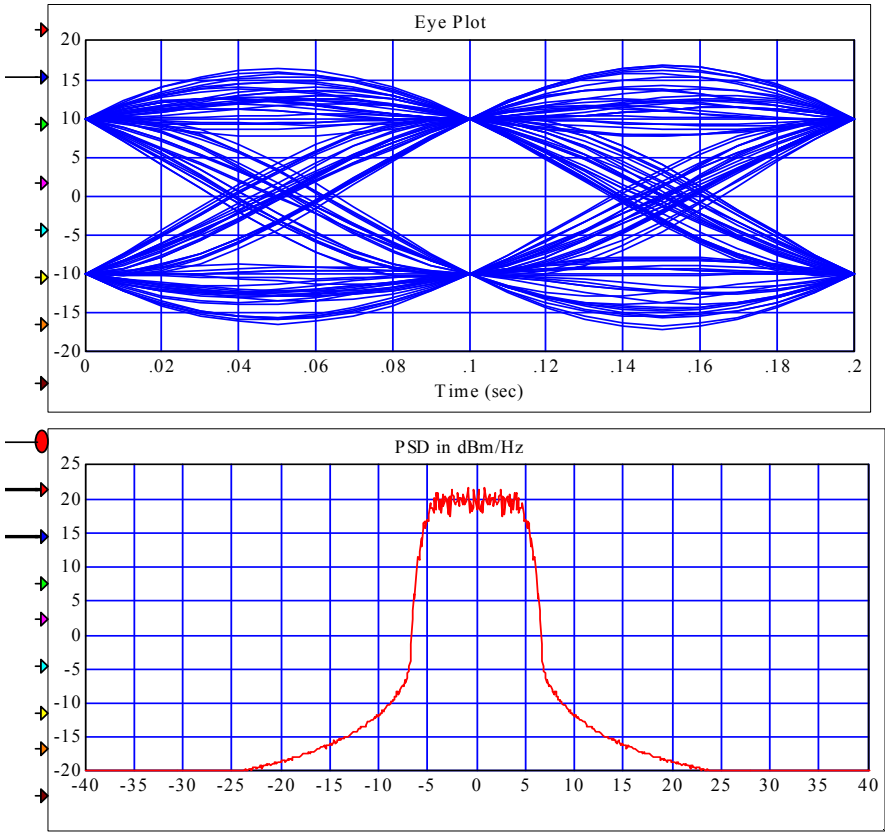
- The modulator sub-block (Shaped PAM) is created to handle the pulse shaping

- The Carrier Recovery block houses a complex baseband Costas loop having a 2nd-order loop filter with loop bandwidth of 0.5 Hz
- A 0.5 Hz frequency step is applied 2 seconds into the simulation
- The carrier recovery block contains a graphical display showing how the loop acquires lock following the frequency step



Carrier recovery block showing the VCO input signal

- A matched filter having an identical SRC filter coefficients are applied to the received signal following carrier recovery
- Shown below is an eye plot at the output of this filter
- The transmitted power spectrum is also computed



Eye plot of the received BPSK and the transmit PSD