i8.M APPENDIX: NON-LINEAR SIMULATION

The MATLAB script `NLPhP` simulates the non-linear acquisition behavior of a phase-locked loop (PLL). With this script we can reproduce the loci in Fig. 8.20 and obtain other enlightening results. The output is an approximation whose degree of accuracy improves as we allow more computer operations for a given problem. It is more complex than previous programs but employs the same basic techniques.

From this point on, the scripts tend to offer more flexibility and therefore require us to choose values for more variables. These variables are located near the beginning of each script. Values that exist when the script is downloaded should produce reasonable results, but not necessarily those of most interest.

8.M.1 Sampling and Simulation

Rather than performing computations on the closed-loop response, we will use MATLAB to compute the open loop response and we will close the loop explicitly on a sampled basis. That will permit us to introduce the non-linear characteristic of the PD, one sample at a time. The process is represented by Fig. 8.21.

The output phase φ_{out} is subtracted from the input phase φ_{in} and the error φ_e is processed by the non-linear PD characteristic. This modified phase error is sampled and held at a regular sampling rate. The HOLD output that depends on the value of φ_{out} at the end of sampling period j provides the input for the open-loop transfer function $G(s)$ during sampling period j+1. The methods described in Section 6.10 are used to compute the output from $G(s)$ at the end of each period. The difference between this simulation and the true case is that $G(s)$ is excited by a stair-step waveform rather than a continuous waveform but, as the sampling period shrinks, so does the inaccuracy. A good approximation of the effect of the added sampler and HOLD, for frequencies that are low compared to the sampling frequency, is that it introduces a phase shift that is linear with frequency and reaches $-\pi$ radians at the sampling frequency [Egan: 1981, pp. 123-127; 2000, pp. 304-306] . `NLPhP` computes this phase shift at ω_L and displays its value.

For reasons of accuracy, a high sampling rate must be maintained but we need not plot every point (e.g., every fifteenth computed point was plotted for the figures in this section).

Continuity of phase and frequency out of $G(s)$ requires that their values at the beginning of each period equal their values at the end of the previous period. The theory developed in Section 6.10.2 is used for this purpose.

8.M.2 Comparing Phase-Plane Plots

Fig. 8.M.1 is output from `NLPhP` under the same conditions that apply to the phase-plane plot of Fig. 8.19. Note that $\Delta\varphi = \varphi_e - \pi/2$ so the stable points in Fig. 8.19 are at $-\pi/2$ radians whereas they are at 0 in Fig. 8.M.1. Frequency in Fig. 8.M.2 is normalized to ω_n, as it is in Fig. 8.19, since we set $W_n \triangleq \omega_n = 1$. Here we have chosen to plot $\omega_{\text{out}}(\varphi_{\text{out}})$ rather than $\omega_e(\varphi_e)$ but the trajectories are the same, as can be seen from the symmetry of Fig. 8.19 or by changing variables in Eq. (6.50), (6.51) and (6.48) with $\varphi_{\text{in}} = 0$.

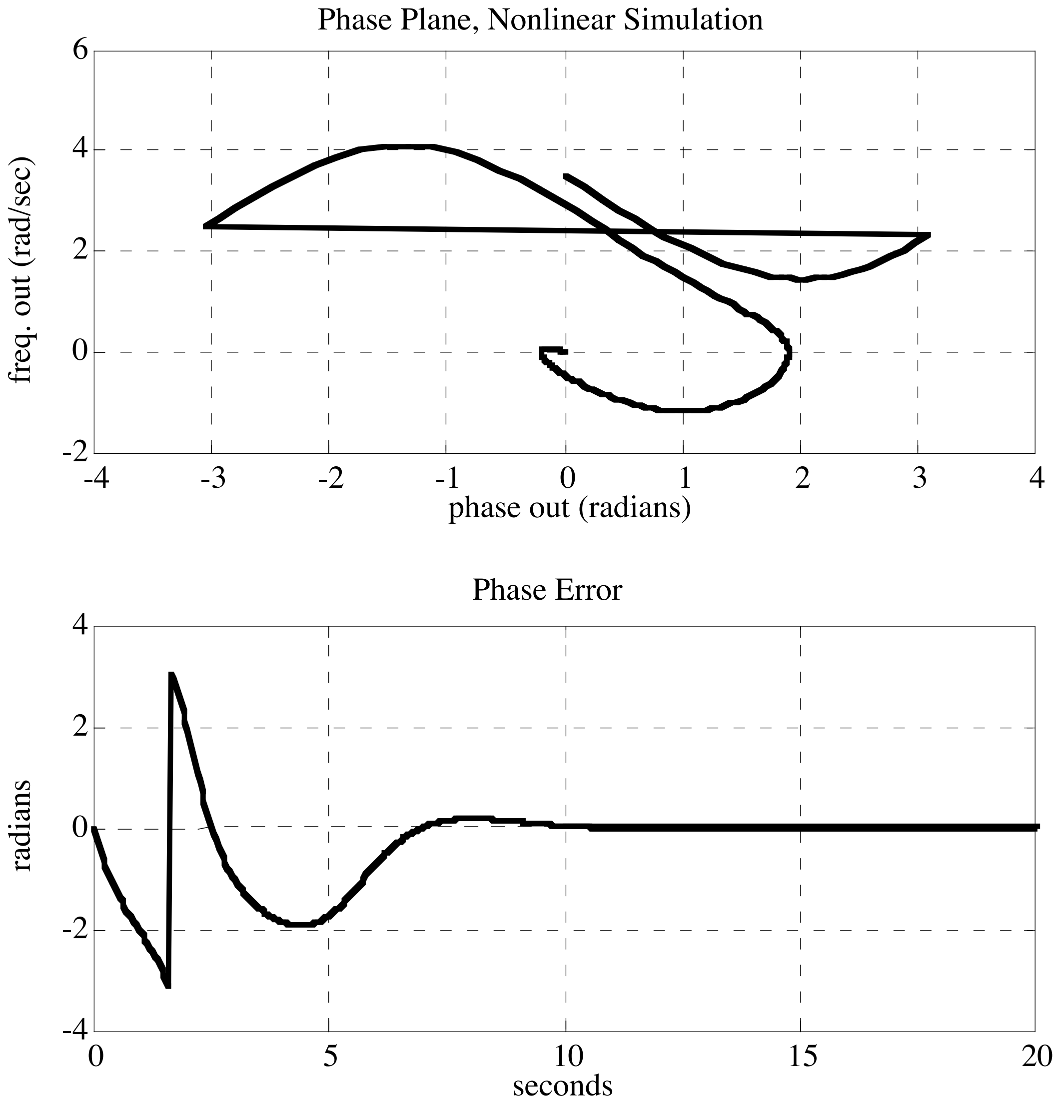


Fig. 8.M.1 Phase plane and transient, phase truncated. $\zeta = 0.707$, $\alpha = 1$, $\omega_n = 1$, 15 samples per plotted point, point each 0.05 sec.

In both plots, the locus that peaks near 4 rad/sec passes through final phase [i.e., $\varphi_e = 0$, $\Delta\varphi = -\pi/2$ rad/sec] with a frequency of about 2.8 rad/sec, goes through zero frequency at a phase of slightly less than 2 rad above the final phase, undershoots by about 1.2 rad/sec, then rapidly moves to the final phase and frequency. However, in both cases, the locus that is at final phase and 3.5 rad/sec (i.e., the starting point in Fig. 8.M.1) is above the separatrix and so skips a cycle, peaking at about 4 rad/sec in the process. (The curve drawn in Fig. 8.19 is slightly above 3.5 rad/sec at final phase. It is apparent that a curve drawn through 3.5 rad/sec would separate further from that curve while moving to the right so it would come closer to peaking at 4 rad/sec than does the existing curve.) In other words, the program output closely matches the phase-plane plot of Fig. 8.19.

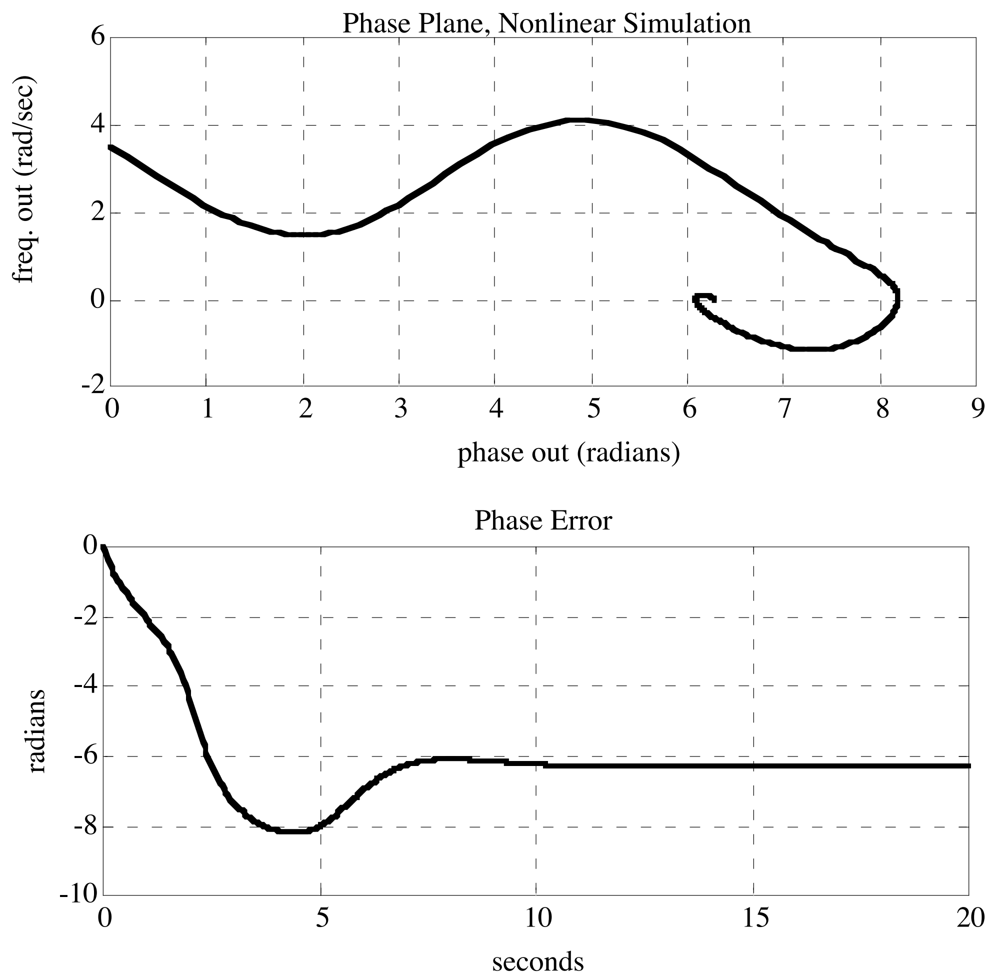


Fig. 8.M.2 Transient and phase plane, phase not truncated.
Same responses as in Fig. 8.M.1

8.M.3 Truncating Phase

We have the choice of truncating phase so it is restricted to a $\pm\pi$ range (i.e., throwing away phase changes in whole cycle increments) or of showing multiple cycles of phase. Figures 8.M.1 and 8.M.2 represent the same responses (from 3.5 rad/sec initial frequency, zero initial phase, $\zeta = 0.707$, $\alpha = 1$, zero phase input) but phase in Fig. 8.M.1 is restricted to the range $\pm\pi$ rad whereas in Fig. 8.M.2 it is not. This is only a matter of how we plot the responses; both pairs of figures represent the same transient.

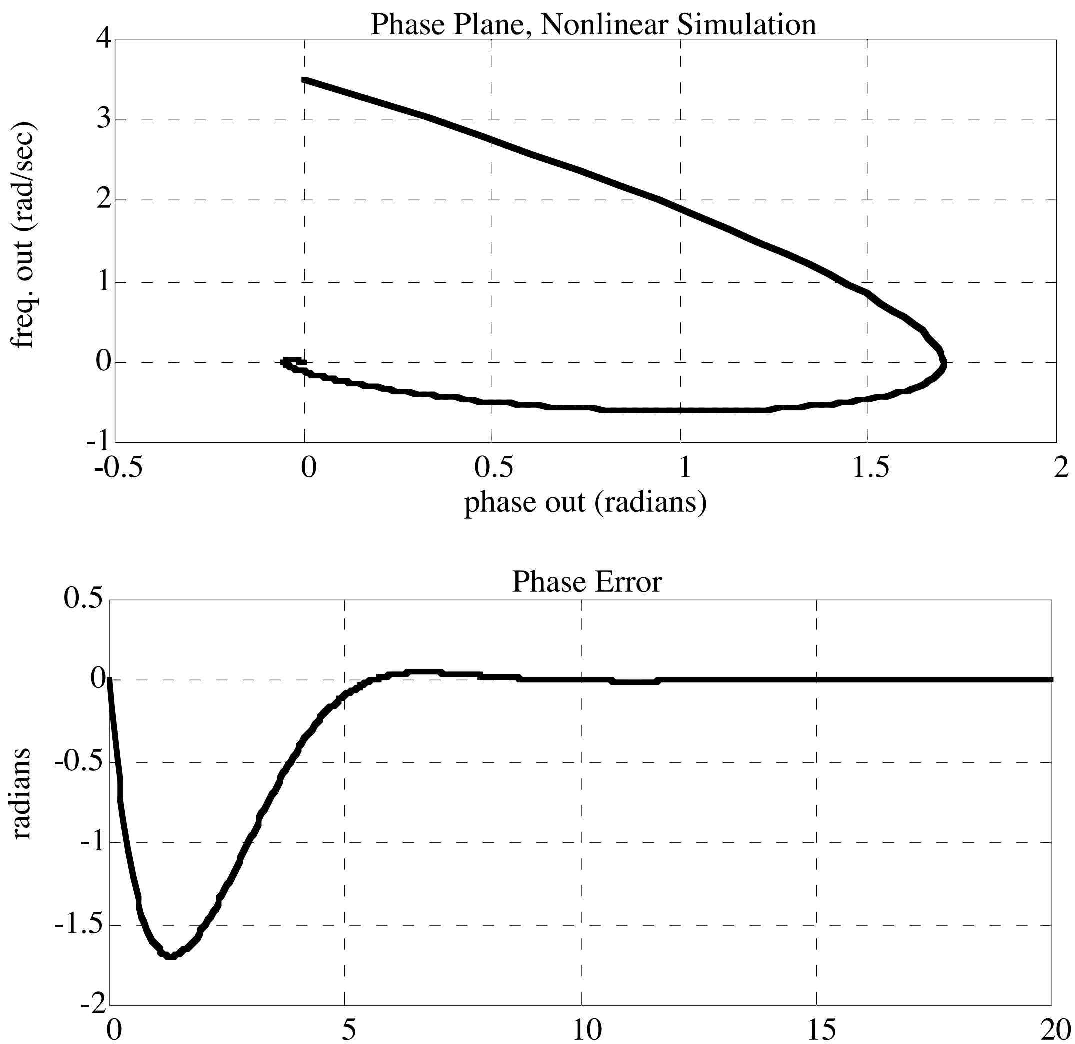


Fig. 8.M.3 Response for same conditions as for Figs. 8.M.1 and 8.M.2 but $\alpha = 0$.

8.M.4 The Effect of α

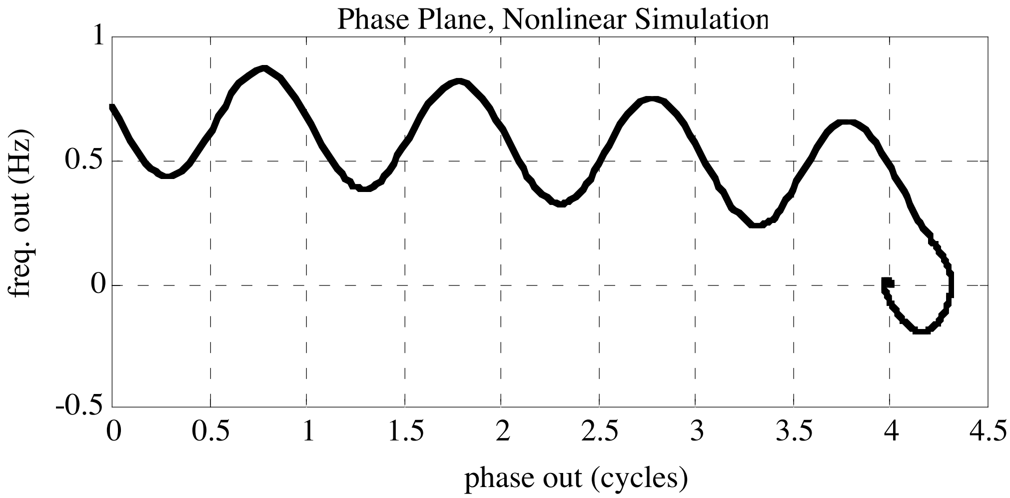
Our experimentation with linear loops showed that, if the excitation and initial conditions were the same for two loops that differed only in α, the responses were identical. Figure 8.M.3 is the response of the same loop whose response is represented by Fig. 8.M.1 except that α differs between the two. Apparently, in the non-linear case, α does make a difference. In fig. 8.M.3, with $\alpha = 0$, the phase goes just past the peak of the PD characteristic at 90° = 1.57 rad and then turns back. With $\alpha = 1$, the initial frequency error must be reduced from 3.5 to 2.8 rad/sec before the initial overshoot is the same as in Fig. 8.M.3. However, if we reduce the initial frequency offset from 3.5 to 0.35 rad/sec, so the response becomes almost linear, it looks very much like that in Fig. 8.M.3 scaled down by a factor of 10, regardless of α.

8.M.5 Observing Pull-In

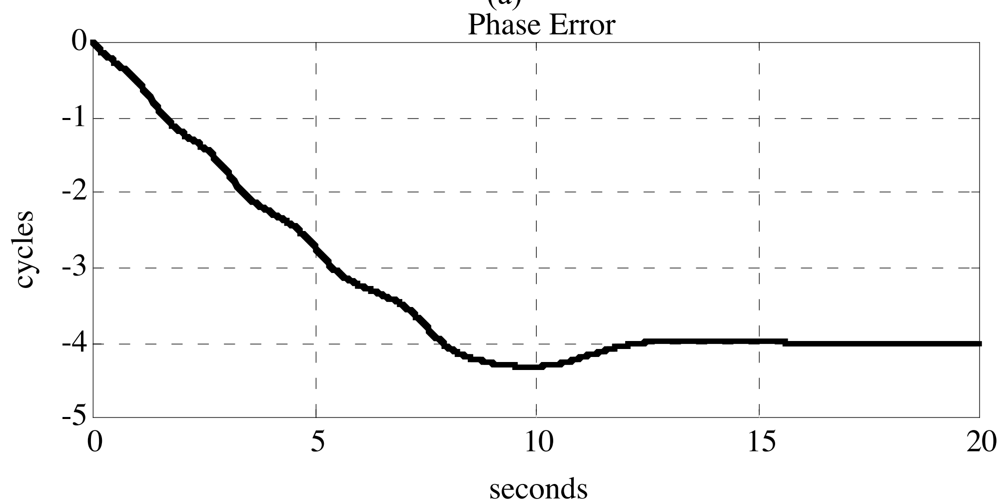
Figure 8.M.4 shows various kinds of output that are available from `NLPhP` to help us understand the pull-in process and confirm what we have learned in this chapter. The parameters here are the same as for Fig. 8.M.1 and 8.M.2 except the initial frequency error has been increased from 3.5 to 4.5 rad/sec. Note also that a different set of phase and frequency units has been selected.

In Figs. 8.M.4*a* and *b* the phase plane and the time transient of the phase error are shown without truncation. The same are shown in Figs. 8.M.4*c* and *d* with truncation.[1] The frequency and PD outputs are shown as a function of time in Figs. 8.M.4*e* and *f*.

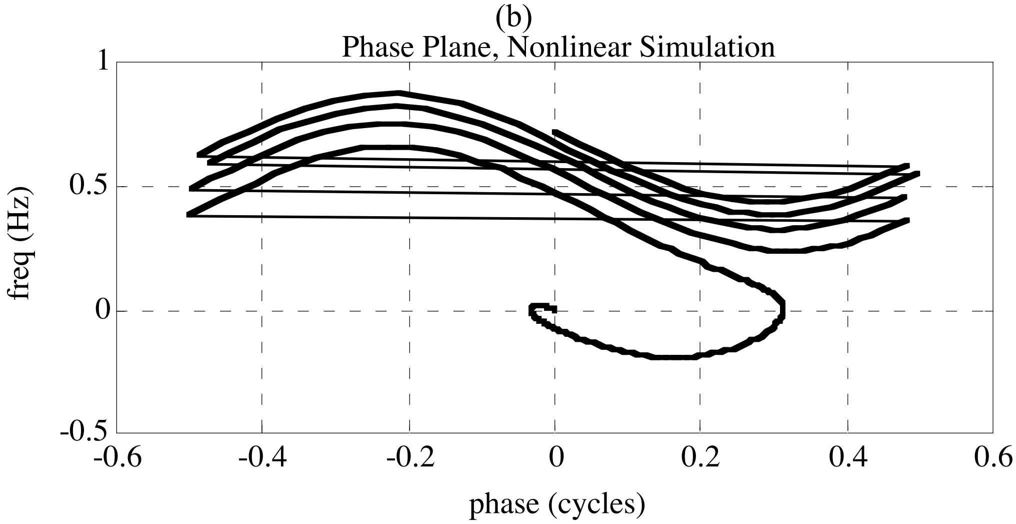
[1] These plots have been modified slightly for print. In particular, the program will not deemphasize the retrace in the phase-plane plot as has been done here. The somewhat irregular starting and ending points for the retrace in the phase plane and the extremes in the phase-versus-time plot are due to the finite time steps in the program.



(a)



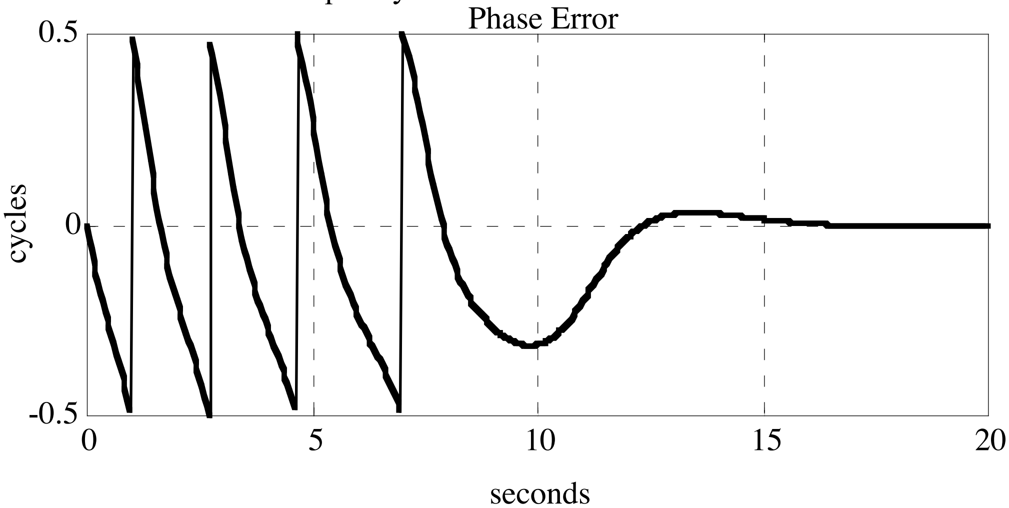
(b)



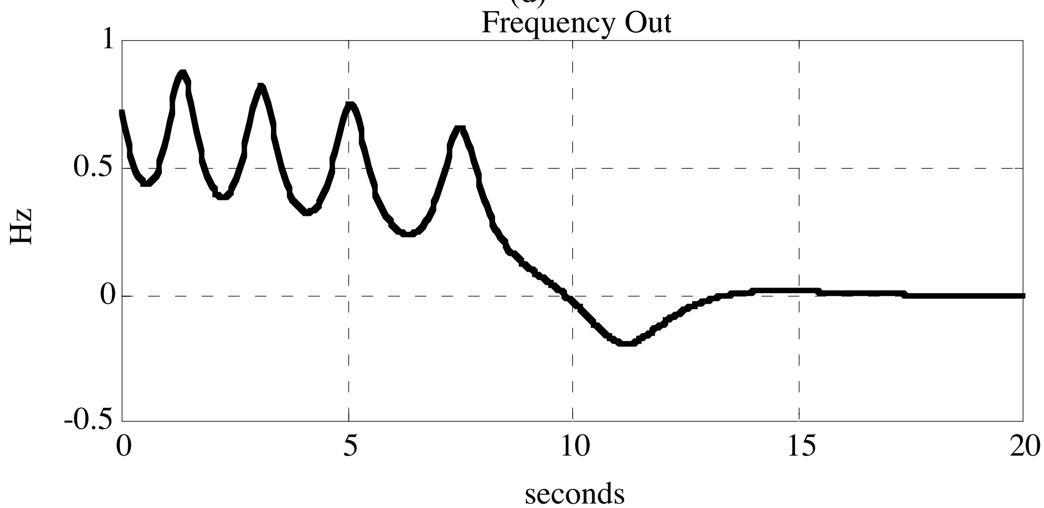
(c)

Fig. 8.M.4 (*a*)-(*f*) Phase and frequency plots.

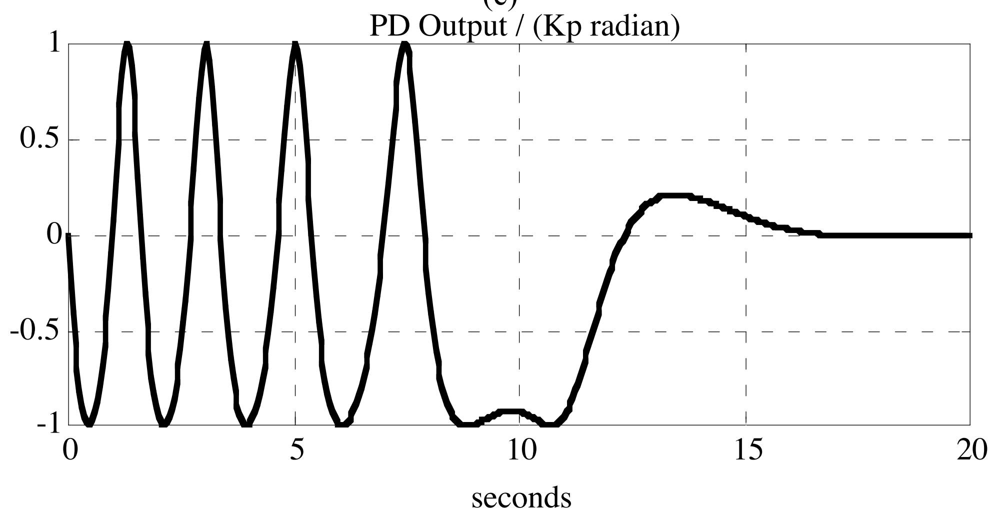
Initial frequency error increased from 3.5 to 4.5.



(d)



(e)



(f)

Note how the beat frequency (Fig. 8.M.4*f*) slowly decreases at first and how the distortion of the beat note develops an average value in the loop filter that causes the output frequency to decrease toward lock (Fig. 8.M.4*e*). Observe what happens 10 seconds into the transient. The plot in Fig. 8.M.4*d* shows that the phase just passes a point of minimum PD output at -0.25 cycle when it turns around and moves toward the lock value. This can be seen in the PD output (Fig. 8.M.4*f*), which takes a slight inverted dip before going again to a minimum on the way to its final value. Note how the frequency error (Fig. 8.M.4*e*) goes to zero at 10 seconds, corresponding to the zero slope of the phase (8.M.4*d*) and of the PD output (Fig. 8.M.4*f*). We can also correlate this minimum phase error (8.M.4*d*) at 10 seconds with the loop-back at freq = 0 and phase = 0.31 cycles (8.M.4*c*).

Note the similarity between Figs. 8.M.4*e* and *f* and Fig. 8.3. Understandable difference can be seen, due to the effect of the low-pass filter associated with Fig. 8.3 and the integrator-and-lead filter with Fig. 8.M.4*c*, in the ripple magnitude as the beat frequency decreases.

These plots have been for one set of loop parameters and initial conditions. An infinite number of other combinations can be obtained using `NLPhP`.

8.M.6 Introducing a Phase Offset

`NLPhP` permits us to introduce an offset at the PD output so we can see how acquisition proceeds when the final PD output is not zero. This differs from an input phase step. For example, with $\alpha = 1$ (integrator and lead), the input to $G(s)$ must settle to zero. With an offset, the corresponding final phase error will be established somewhere on the sinusoidal characteristic where the slope (gain) is less than maximum whereas, with a phase step input (and no offset), the final value from the PD would be at the maximum gain point.

8.M.7 Introducing a Frequency Step

A phase step is provided by giving φ_{in} a steady value during the simulation and a frequency step is provided by increasing φ_{in} at a constant rate. These functions can be present when initial conditions are specified but, when a step response (`SR`) is selected[2], the initial conditions are overridden so the loop can begin at steady state. Other driving functions can be provided at the input by following a similar procedure.

A phase step has no discernible effect on the final state of a loop (unless it causes the loop to break lock and remain out of lock). The output phase changes by the same amount as the input phase. But, with a type-1 loop, a frequency step changes the final phase error. The response can be affected by both the initial phase error and the final phase error, since both affect the region of operation in the nonlinearity. We can choose a combination of phase offset and frequency step to simulate any initial and final state for a frequency step.

8.M.8 Customizing the Nonlinearity

The nonlinearity has been explicitly stated (i.e., broken out) at two places in `NLPhP` to make it easy to find so we can replace it with other nonlinearities. For example, we might create a saw-tooth PD characteristic by replacing the sine nonlinearity by its argument and using truncation, like `NLPhP` employs on the variable `e` (see the region marked by "<<<TRUNCATION").

Fig. 8.M.5 shows the results of changing to a sawtooth PD characteristic in `NLsaw`.[3] The initial frequency error has been increased to 8 rad/sec because of the greater seize range with the sawtooth PD. The sampling period and period between outputs has also been halved because that happens to allow us to see an anomaly, which is marked by the dashed line in the phase-plane plot. It results from an unfortunate lack of synchronization between the occurrence of the sharp nonlinearity in the sawtooth PD characteristic and the occurrence of truncation in the plot (the anomaly would not be apparent in untruncated plots). The plot truncation occurs as soon as 0.5 cycle is exceeded but the frequency does not respond until the end of the next sample period, during which the value from the HOLD circuit reflects the severe change in PD output. Moreover, the

[2] Other requirements, in order to obtain this type of simulation: Initial phase and frequency zero and a nonzero input phase or frequency.

[3] The PD output display (lower right) has also been changed to be more suitable to the new PD characteristic, but it is the same as the phase error plot when phase is displayed in cycles. We can see that the displayed time has been halved. The period between samples has also been halved.

display of the frequency change is delayed 14 samples (for this simulation) until the next displayed point. Usually this anomaly occurs at one of the 14 undisplayed points between displayed points, in which case it is not seen. In this case, however, the phase plane shows one occurrence of a step in phase due to truncation followed by a step in frequency due to the loop response, rather than showing both at the same time. This anomalous display could be prevented at the cost of program complexity.

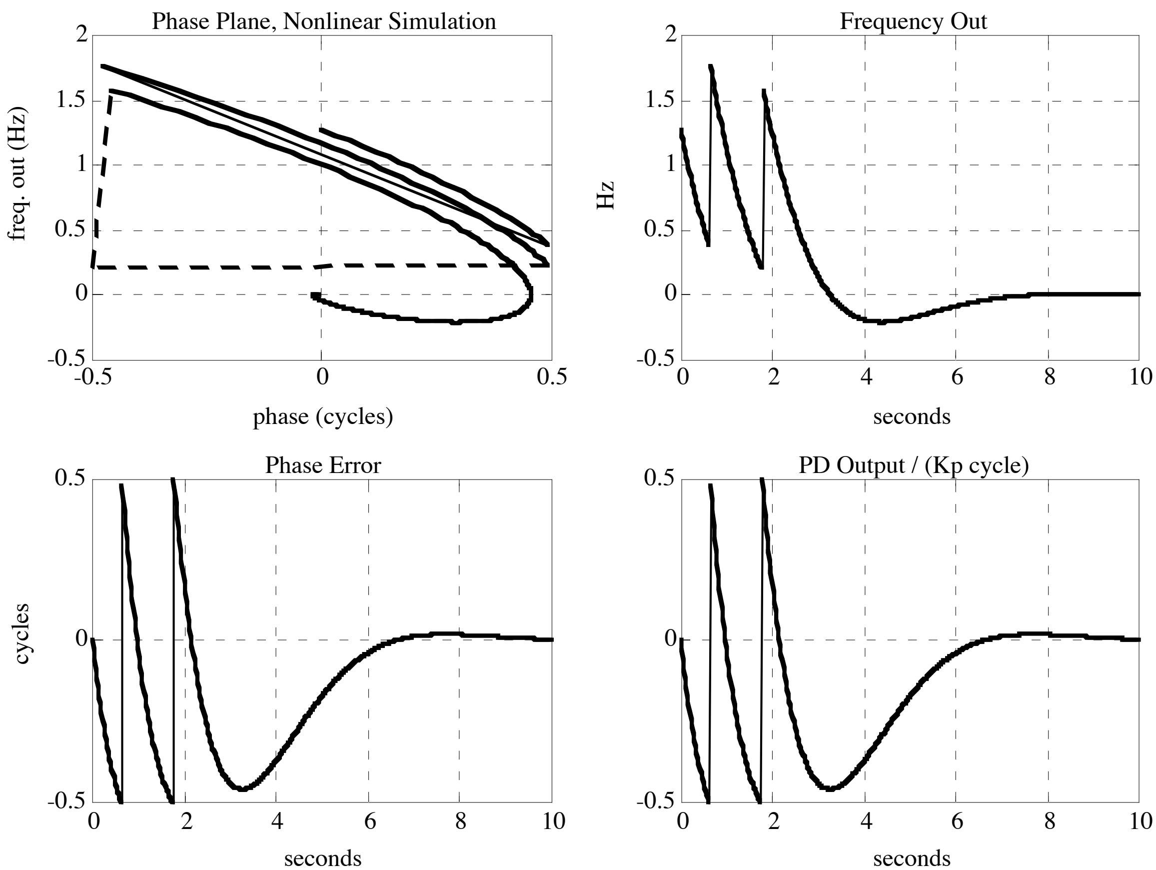


Fig. 8.M.5 Simulation of a PLL having a sawtooth PD. $\alpha = 1$, $\zeta = 0.707$, $\omega_n = 1$ rad/sec. Dashed lines in phase plane illustrate plotting anomaly associated with truncation.

8.M.9 Verifying Acquisition Equations

`NLPhP` gives us a tool to verify the acquisition formulas in Section 8.3 and their limits of applicability. After modifying the nonlinearity, we can do the same for loops employing other PD characteristics [Egan: 1981, pp. 211-220; 2005, pp. 408-418].

8.M.10 Some Experiments

Here are some suggested experiments using `NLPhP`. The spreadsheets in Appendix 8.S can be helpful in computing theoretical acquisition parameters for comparison to simulations.

8.M.10.1 Type-1 Loop with Low-Pass Filter, Pull-In Use `SR = Winit = Ap = alpha = Offset = 0, Wn` = 1, `z` = 0.2, `OutInc` = 0.05/`Wn`, `ending` = 40/`Wn`, and `SmpPerOut` = 15. `Winit` = 0 means that the output frequency is initially zero, the steady-state frequency at the center of the PD characteristic. `Ap` = 0 means that the input phase is not stepped. We will set certain values for `Phinit`, the initial output phase, and certain values for `Af`, implying a reference frequency other than zero, that is, mistuning.

If we had chosen `Af` = 0 and `Winit` ≠ 0, we would be moving from some mistuning to zero mistuning. The responses would be different because our locus on the PD characteristic and the corresponding K_p would change with time in a different manner in the two cases.

`Ap` = 0 and `Phinit` = *k* gives the same results as `Ap` = -*k* and `Phinit` = 0. A step response (`SR = 1`) with `Ap` = -*k* would also give the same results when `alpha` is zero since the initial output will not be influenced by a step when the filter is low-pass.

What is the theoretical pull-in frequency? Try a simulation with `Af` = 1.15 and `Phinit` = 3.14. Do the results agree with the theoretical prediction? Change `Af` to 1.1. Try various values for `Phinit`. Is the predicted value as accurate as you had expected?

8.M.10.2 Type-1 Loop with Low-Pass Filter, Hangup Use `SR = Winit = Ap = alpha = Offset = 0, Wn` = 1, `z` = 0.2, `Phinit` = 3.14, `OutInc` = 0.1/`Wn`, `ending` = 25/`Wn`, and `SmpPerOut` = 5. Observe the very slow start of the transient. What would happen if `Phinit` were set closer to π?

8.M.10.3 Type-2 Loop, Integrator-and-Lead Filter, Seize and Speed Use `SR = Ap= Af = Offset = 0, alpha = 1, z` = 2, `OutInc` = 0.05/`Wn`, `ending` = 40/`Wn`, and `SmpPerOut` = 15, `Winit` = -10.

Since the DC gain is infinite, `Winit` = k produces the same results as `Af` = $-k$. In either case the steady-state phase error at start and end of transient is zero.

We computed pull-in frequency for the previous loop. For this type-2 loop we will compute and measure seize frequency and pull-in time. (Recall that a loop with a low-pass filter has no seize frequency and a type-2 loop has theoretically infinite pull-in frequency.)

We will measure seize frequency by looking for the highest initial frequency error from which the loop locks without skipping a cycle at any initial phase. We will be simulating the case in which the PD output is initially zero because there is no reference and then a reference of a given frequency and phase suddenly appears.

We can also estimate seize frequency from the phase plane plot. When that plot goes through a phase of zero or π radians, the frequency error is the same as is the mistuning when a reference is suddenly applied, if the phase of that reference is such as to produce zero PD output.[4] The seize frequency, which is for arbitrary phase, occurs between the

[4] Justification for the seize frequency being between the next-to-last zero-phase crossings and the subsequent π radians point in the phase plane of a Type-2 Loop: In the phase-plane, the mistuning can only be observed at zero or π radians phase error in a Type-2 loop. No current flows in the filter at zero PD output so the state is determined entirely by phase and frequency or, equivalently, filter capacitor charge. If the reference were removed the PD output would not change, nor would the output frequency, so the frequency error under this condition is the mistuning, the value of frequency error before a connection is made.

We identify the last time when the phase error is zero before lock as T_{-1} and the next previous such time as T_{-2}. All phases occur between T_{-2} and T_{-1}. Moreover, since stable equilibrium occurs at zero phase error, as the locus leaves such a point ($T_{-2}+$), the filter capacitor charges in such a direction as to reduce the error. When π error is reached (at T_{π}), it will begin charging in the opposite direction and continue to do so until T_{-1} but the minimum error will have occurred at T_{π} Therefore, between T_{-2} and T_{-1} the capacitor will always have a charge between its values at T_{-2} and T_{π} so the corresponding mistuning is between the mistunings at T_{-2} and at T_{π}. Thus, the locus between T_{-2} and T_{-1} contains all possible phases and each point corresponds to a mistuning between the frequency at the next-to-last crossing of zero phase and the frequency at the subsequent point where the phase is π, so it will be somewhat uncertain. (If the resolution of the plotted points were fine enough, the frequency at π radians could be read where the retrace crosses zero phase. When the step size of the plotted phase is significant, that point must be estimated. The estimate could be helped by choosing not to truncate the phase [`truncatePh` ⇒ 0] so a smooth curve exists between points on either side of π radians.) By observing this range from the phase plane produced under various initial conditions we could estimate the seize frequency.

Start with `Phinit` = 3.14 (rad). At this phase, determine the boundary between a value of `Winit` that will allow lock without cycle skipping ("seizing") and one that will not. `Winit` = 5.2 (rad/sec) is suggested as a starting point. With Winit set to that highest value, try several other phases to show that the loop will lock without skipping at the highest frequency at which it did so with `Phinit` = 3.14 rad. Does this frequency compare well with the predicted seize frequency? Are the conditions of validity for Eq. (8.23) met?

Set `Winit` = -10 and estimate the seize frequency from the phase plane. Start with `Phinit` = 1.57. Observe also the pull-in time, estimating the time at which the frequency error equals the seize frequency for the end of the pull-in period. Now try `Phinit` = -1.57. Notice how much the time to seize can vary. The longest time at any phase is the required value. You can try some other initial phases. How does the pull-in time obtained here compare with theory. How does the seize frequency compare between the two methods and with theory. Are the conditions of Eq. (8.26) and (8.27) well met?

8.M.10.4 Type-1 Loop, Lag-Lead Filter, Seize Use `Winit = Phinit =Offset = 0, SR = 1, Af` = 9, `Wn` = 1.0005, `alpha` = 0.99701, `z` = 1.6725, `OutInc` = 0.05/`Wn`, `ending` = 50/`Wn`, and `SmpPerOut` = 30.

Set `Ap` to 3.14 and various other phases. Show that the seize frequency is between 4.45

---

mistuning at T_{-2} and T_{π}, both of which can be read from the phase-plane plot. Therefore, the mistuning that will allow lock without cycle skipping regardless of phase lies between the mistunings at T_{-2} and at T_{π}, both of which can be read from the phase plane plot. Therefore, the mistuning that will allow lock without cycle skipping regardless of phase lies between the mistuning at T_{-2} and at $T_{-\pi}$, which are observable in the phase plane.

and 4.5 rad/sec. To do this, show that `Af` = 4.45 (rad/sec) seizes at all phases and 4.5 fails at some initial phase. What is the theoretical value and how well are the conditions met?

Observe the phase planes with `Ap` = 0 and various values of `Phinit`. How would an estimate of seize frequency based on the phase plane, as for $\alpha = 1$, compare to the value obtained by trying various frequency steps. The theory that supports the equivalence for $\alpha = 1$ does not do so for $\alpha < 1$.[5] Nevertheless, if we take the pull-in time to end when the average frequency over a cycle equals the seize frequency, the frequency at zero phase is a good estimate of that average, at least in this case. What is the estimated pull-in time from the plots? How do these values compare to theory and how well are conditions met?

8.M.10.5 Type-1 Loop, Lag-Lead Filter, Offset For an offset of 0.15 radians, Eq. (8.76) gives an optimum value of $x = 0.0225$. By Eq. (8.24), this implies a ratio of pole to zero frequency of 0.045. Choose $\omega_p = 4.5$ rad/sec, $\omega_z = 100$ rad/sec, and $\omega_L = 1000$ rad/sec. Using `Acqpz`, we obtain, from these parameters, $K = 22{,}112$ sec^{-1}, $\zeta = 1.584351299$, $\alpha = 0.995497962$, $\omega_n = 315.444$ rad/sec and $\Omega_{PI} = 6559$ rad/sec. Enter `alpha, z,` and `Wn` in `NLPhP`.

The development of the equations of section 8.7 is based on the idea that an offset amounts to a mistuning and thus reduces the (additional) mistuning allowed for lock to be assured. Unfortunately, a simulation to show pull-in range can be very time consuming because the pull-in time grows as the pull-in limit is reached. Instead, we will observe that the offset is effectively added to the initial mistuning. We will do this by obtaining the same pull-in time for a given mistuning with zero offset and for a mistuning that is reduced by the effective mistuning when an offset exists. Initially, set Offset = 0, `Ap` = -1.57, and `Af` equal to 80% of the pull-in frequency, 5247 rad/sec. Also set `SR` = 1, `Phinit = Winit = 0, OutInc =` 0.05/`Wn`, `ending` = 200/`Wn`, and `SmpPerOut` = 7. Determine how long the loop takes to seize. By Eq. (8.75), ω_{PI} will be reduced by $\omega_L \times 0.15$ rad$/(2|\varphi_{os}|) = 3333$ rad/sec in the presence of the offset. Reduce the initial frequency by 3333 rad/sec to give `Af` = 1914 rad/sec. Rerun `NLPhP` with `Offset` = -0.15 and the reduced value of `Af`. How close is the time to seize to what was obtained before.

[5] Zero phase does not imply steady state. The frequency does not uniquely identify the state. Even with zero phase error, the filter capacitor could be discharging.

8.M.11 Simulation Using Approximate Method

The MATLAB program `NLPhx` performs the same function as `NLPhP` but uses the approximate algorithm described in Section 6.11 for the repeated computations. In some tests, outputs appeared to be the same as obtained with `NLPhP` and the simulations were faster, but only by about 9%. Probably its main value would be for use when the matrix manipulation programs, such as are contained in MATLAB, are not available. (Because they are available, MATLAB's matrix mathematics are used where appropriate in `NLPhx`).