Adaptive Low-Bandwidth Tracking of Galileo and Pioneer 10 Carriers

D. A. Watola
Communications Systems Research Section

In the Deep Space Network, tracking of residual carrier phase typically occurs with a fixed-bandwidth phase-locked loop using a bandwidth sufficiently wide to prevent loss of lock under worst-case conditions of signal dynamics, received signal phase noise, and receiver phase noise. Much of the time, however, such a high bandwidth is not required and may inflict unnecessarily heavy penalties on loop signal-to-noise ratios. This article describes a technique for improving tracking performance by permitting initial tracking at narrow bandwidths and gradually widening the loop as needed. The cost is a requirement for signal buffering, which is relatively inexpensive for low data rate applications. Results based on off-line processing of recorded carrier data from Galileo and Pioneer 10 are presented, and show potential 10–16 dB gains in loop SNR over worst-case fixed-bandwidth tracking.

I. Introduction

The Block IV receiver phase-locked loops (PLL's) used by the DSN to track the carrier signals of the Galileo and Pioneer 10 spacecraft have a minimum loop bandwidth of around 1.0 Hz. The next generation receiver, the Block V, allows much narrower digital loops with potential improvement in loop signal-to-noise ratio (SNR). However, the carrier loop bandwidth is usually selected conservatively to be wide enough to prevent loss of lock even under the worst-case tracking conditions of received signal phase noise, receiver phase noise, and signal dynamics. This is because carrier phase recovery is performed in real time, making loss of carrier lock extremely undesirable since it causes unlocking and loss of data in symbol detectors and error-correcting decoders in the downstream system; resynchronization of symbols takes much longer than carrier acquisition, and can result in the loss of tens of minutes of data. In the case of Pioneer, this consideration dictates that loops as wide as 3.0 Hz be used to prevent loss of lock. Worst-case conditions are relatively rare, however, and carrier recovery can potentially be achieved at much lower bandwidths most of the time, resulting in better carrier phase estimates.

This article describes an algorithm that exploits this fact without compromising the ability to recover carrier phase in the presence of significant noise transients. In the approach described here, the carrier is sampled and recorded before tracking. However, the algorithm can be modified to operate on a delayed (rather than off-line) basis. Because buffering of data is required in either approach, loss of lock at a given bandwidth is not a catastrophic occurrence since another attempt can be made at a more suitable bandwidth. Off-line or delayed computation also permits noncausal tracking. Unfortunately, this requirement also limits applicability to low data rate mis-

sions where low sampling rates permit a practical data volume; obvious examples are Pioneer 10, Voyager 1 and 2, and the Galileo S-band missions. The results presented here, though not optimum, show significant improvement over fixed-bandwidth PLL tracking for an experimental data set taken from Galileo and Pioneer 10.

II. Experimental Setup and Data

One hour of carrier from Galileo and four hours from Pioneer 10 were recorded onto a large hard disk on an IBM PC-AT compatible platform. These data were taken at the output of the Goldstone Solar System Radar (GSSR) at DSS 14 on December 19, 1991. Using the GSSR programmable local oscillator to remove predicted Doppler, the carrier was mixed down to near-baseband (100 Hz nominal) and lowpass filtered to 512 Hz before sampling at 1.024 kHz per open loop I/Q channel with 12 bits of resolution. This experimental setup is depicted in Fig. 1.

Typical spectra from these recordings are presented in Figs. 2 and 3. Here, the carriers have been shifted to near DC, lowpass filtered to 5 Hz, and subsampled by a factor of 50. This allowed better than 0.005-Hz resolution from a 2048-point fast Fourier transform, based on subsampling a 100-second (102,400-point) segment of data. These figures indicate carrier spectral line widths much smaller than the minimum fixed-bandwidth tracking loops currently used in the DSN.

III. Basic Tracking Algorithm

The adaptive carrier-tracking algorithm is built around the digital phase-locked loop (DPLL) depicted in Fig. 4. In this system, the residual carrier is assumed to have the form

$$r(t) = \sqrt{2P_c}\sin(\omega_c t + \Phi_c) + n(t) \qquad (1)$$

where P_c is the power, ω_c is the frequency, and Φ_c is the phase of the downconverted residual carrier. The noise, $n(t)$, will be ignored in this discussion since its effects on PLL behavior are thoroughly documented elsewhere e.g., [1,2]; it is only assumed that $n(t)$ is an additive white Gaussian noise (AWGN) process with power spectral density $N_0/2$. For simplicity, this treatment only considers the case where ω_c and Φ_c are not time varying.

Noisy samples recorded at rate T_s^{-1} are first processed by an automatic gain control (AGC) to normalize the residual carrier sinusoid to unit amplitude based on estimates of the sample SNR. The output from the AGC,

$$x[n] = \sin(\omega_c n T_s + \Phi_c) \qquad (2)$$

drives the DPLL, which mixes it with the numerically controlled oscillator (NCO) output waveform. Assuming that the DPLL is operating in the locked condition, the NCO output is a cosine with frequency ω_c and phase Φ_{nco}. As long as the loop is operating in lock with $\Phi_{nco} \approx \Phi_c$, the mixer output $\Phi_e[n]$ contains a DC component nearly proportional to the phase difference:

$$\Phi_e[n] = \sin(\Phi_c - \Phi_{nco}) + \sin(2\omega_c n T_s + \Phi_c + \Phi_{nco})$$

$$\approx \Phi_c - \Phi_{nco} + \sin(2\omega_c n T_s + \Phi_c + \Phi_{nco}) \qquad (3)$$

An integrate-and-dump filter, shown in Fig. 4 as a moving average followed by a decimation by N, smooths this phase error and removes the high-frequency term from Eq. (3).

The smoothed phase error passes through two filters to form Φ_{nco}. The first is a loop filter having system response function

$$H(z) = \frac{C_2 - C_1 z^{-1}}{1 - z^{-1}} \qquad (4)$$

where

$$C_1 = 4B_A \frac{r}{r+1} \qquad (5)$$

$$C_2 = C_1 + rNT_s \left(\frac{4B_A}{r+1}\right)^2 \qquad (6)$$

Here, r is a damping parameter and B_A is the noise bandwidth of the corresponding analog filter prototype [2,3]; as long as $B_A N T_s << 0.1$, $H(z)$ should provide nearly the same loop bandwidth as B_A. The second filter is the NCO, which can be modelled by a single system function having the form

$$N(z) = \frac{\Phi_{nco}(z)}{E(z)} = \frac{NT_s}{2}\frac{z+1}{z^2(z-1)} \qquad (7)$$

as described in [2] for the DSN advanced receiver. Note that use of an integrate-and-dump filter causes the NCO

output to be updated N times slower than the input sample rate.

In addition to the mixing output, the NCO also provides a second output that is phase shifted by 90 deg. Mixing this quadrature output with the DPLL input signal produces

$$\hat{\lambda}[n] = \cos(\Phi_c - \Phi_{nco}) + \cos(2\omega_c nT_s + \Phi_c + \Phi_{nco}) \quad (8)$$

which is lowpass filtered by $L(z)$ to remove the term at $2\omega_c$, yielding a lock-detection signal

$$\lambda[n] = \cos(\Phi_c - \Phi_{nco}) \quad (9)$$

The Maclaurin series expansion of Eq. (9) shows that

$$\lambda[n] = 1 - \frac{1}{2}(\Phi_c - \Phi_{nco})^2 + \cdots \approx 1 \quad (10)$$

for small phase errors. Thus, the lock detection filter output is near unity when the loop is operating in lock as long as the AGC accurately estimates P_c. Passing $\lambda[n]$ through a hysteretic comparator with thresholds L_- and L_+ to form the final Boolean lock indication signal lends improved reliability and reduces clatter [4] in the final Boolean lock indication signal, LOCK.

IV. Adaptive Bandwidth Tracking

While a software implementation of the adaptation algorithm is complex, the method is conceptually simple. Thus, a brief description suffices here.

To make the above DPLL adaptive, a supervisory loop is added to the system shown in Fig. 4. This outer loop processes recorded carrier samples at a fixed initial bandwidth, B_0, using the LOCK signal to determine which portions of the data are successfully tracked. The segment boundaries are noted, and outputs generated during a tracked section are recorded.

After tracking attempts at this bandwidth are complete, the loop bandwidth is increased and DPLL filter coefficients are recalculated. Portions of the data that remain unlocked are reprocessed at the new bandwidth; this process continues iteratively until the carrier is tracked for the duration of the recorded samples or until some maximum acceptable bandwidth is exceeded. If the filters are implemented as direct form 1 structures (Fig. 5) [5], *only* the filter coefficients need to be recomputed; filter initial conditions are completely specified by input samples and output samples generated at lower bandwidths. Alternately, direct form 2 transposed structures (Fig. 6) can be used for better numerical stability [5] at the expense of greater computational complexity.

The untracked carrier samples are actually processed by the DPLL twice at each bandwidth—once in each direction. The forward pass uses causal filtering, and the reverse pass is anti-causal. Processing carrier data backward through time in this fashion has several advantages. First, it allows the initial segment of data to be tracked if the loop does not start in the locked state. It also allows minor transients that cause the lock detector output to drop below L_- in one direction to be processed in the other direction without a change in bandwidth. More importantly, though, it allows the effects of cycle slips to be eliminated from the carrier phase.

Traditional noncausal filtering [6] cancels phase shifts by filtering data in both directions. However, in this case once carrier phase has been successfully recovered, that segment of data is no longer filtered. This implies the existence of a phase discontinuity between adjacent segments processed in opposite directions. Fortunately, because the carrier is an extremely narrowband signal located essentially at the center of the DPLL passband, the phase discontinuity between such adjacent segments is insignificant.

A demonstration of the algorithm is presented in Fig. 7. At time 0, DPLL tracking attempts begin at loop bandwidth B_1. The loop acquires lock at time t_1 and drops lock at t_2. Lock is reacquired at t_3 and is held until the end of the data, t_{end}. Thus, after the first forward pass there are two segments, s_1 and s_2, that no longer need to be processed. Subsequently, noncausal tracking begins at t_3, where the loop is able to use conditions for $t > t_3$ to start out in lock at t_3. In this example, lock is dropped again at t_4 and not regained before t_2. Segment s_1 is skipped and tracking begins at t_1, where initial conditions allow the loop to track the carrier all the way back to time 0. Without resorting to anti-causal tracking, segments s_3 and s_4 would require a higher bandwidth. Finally, after s_4 is locked, the bandwidth is increased to B_2 and tracking is tried beginning at t_2; here, B_2 is sufficient to follow the carrier for the rest of the data. This example is slightly simplified since there is a delay through $L(z)$ as the quadrature lock detector output is smoothed. However, it is adequate for summarizing the mechanics of the bandwidth adaptation algorithm.

Figure 7 is also useful in revealing a weakness of the algorithm. It is not truly adaptive because bandwidth is

never lowered once it is increased. It is certainly conceivable that some segment between t_2 and t_4 in Fig. 7 could be tracked at B_1 even though at the instants t_2 and t_4, a width of B_2 is needed. The algorithm described above will not discover this situation. In fact, no attempt to optimize the loop bandwidth is made at all; optimum bandwidth is never computed or examined as it is in [7]. Despite this apparent shortcoming, this method does effectively respond to transient disturbances in the data and provides good results without incurring a large computational burden. The algorithm can also be modified to attempt reduced bandwidths at the expense of a greater computational burden.

V. Measured Performance (Galileo Data)

Figures 8 through 11 give simulated results of applying the adaptive bandwidth tracking algorithm to one hour of Galileo carrier data recorded beginning at 22:41:03 UTC on DOY 353, 1991. For these simulations, $T_s = 1024^{-1}$ sec, $N = 100$, and $r = 2$. Switching thresholds in the lock detection comparator were selected at $L_- = 0.70$ and $L_+ = 0.75$ for 5 percent hysteresis. The simulations for Fig. 8 were made with various initial bandwidths ranging from 0.1 Hz to 1.0 Hz. Bandwidth increases within each simulation are in steps of 10 percent. Figure 9 is a similar plot for a starting bandwidth $B_0 = 0.025$ Hz; this proved to be a lower limit for successful carrier recovery using the current software implementation with a first-order loop filter.

These results indicate that for most of the hour, the carrier phase can be tracked with a 0.05- to 0.055-Hz loop. For nearly the entire hour, 0.1 Hz is sufficient. Only for a few brief transients is it necessary to resort to loops as wide as 0.6 to 0.7 Hz. By contrast, the minimum loop bandwidth for carrier tracking with current DSN equipment is 1.0 Hz. Since the phase estimation error variance in the linearized PLL model follows

$$\sigma^2_{\Phi_e} = \frac{P_c}{2B_A N_0} = \frac{1}{2SNR_L} \qquad (11)$$

a 13-dB decrease in phase error variance can potentially result over much of the tracking period. This translates directly to a potential 13-dB increase in loop SNR. In rare instances, as much as a 16-dB gain can be seen.

Figures 8 and 9 also show that successful tracking at lower B_0 does not necessarily outperform higher initial bandwidths over the complete span. This is reasonable since more noise is allowed into the tracking loop as the bandwidth is widened, so transients that appear at higher bandwidths may not affect narrower loops.

Figures 10 and 11 demonstrate respectively the recovered phase $\Phi[n]$ and the output from the lock detector $\lambda[n]$ for the first half hour of the experimental data. Note that each point on Fig. 9 where the bandwidth changes indicates a point where $\lambda[n]$ dipped below $L_- = 0.7$ and lock was lost. However, in Figs. 10 and 11 the phase is continuous and $\lambda[n] \geq L_-$ for all n since they are composed of results from multiple tracking passes at different bandwidths.

VI. Measured Performance (Pioneer 10 Data)

Figure 12 presents the results of applying the same process to a 700-sec segment of Pioneer 10 carrier data recorded beginning at 07:27:10 UTC on DOY 354, 1991. All simulation parameters are identical to those given above except that $r = 4$ and $B_0 = 0.5$ Hz. For Pioneer, the algorithm does not perform nearly as well as it did for Galileo. Although it does track the carrier with loops as low as 0.5 Hz, it is unable to recover the carrier over the entire time span; this is denoted in the plot by a bandwidth of 0 Hz.

The inability to completely recover the carrier from this data has not been completely explained. However, it should be noted that the quality of the data is suspect. Estimated P_c/N_0 in the data is consistently lower than 10 dB-Hz; the DSN, which successfully recovered data from Pioneer 10 during this period with $B_{LO} \approx 3$ Hz, requires approximately $P_c/2N_0B_{LO} \geq 6-8$ dB[1] or $P_c/N_0 \geq 13.8 - 15.8$ dB-Hz. It should also be noted that the software implementation of the tracking algorithm assumed that the input samples were white and Gaussian; both of these assumptions proved true for the Galileo data but neither applied well to the recordings from the Pioneer 10 spacecraft. Both of these observations suggest that the data recorded from Pioneer 10 are not as good as that used by the DSN.

VII. Design Considerations

Several design trade-offs were revealed during the implementation and testing of this algorithm. Most of them involve the selection of B_0 and its interaction with other

[1] Personal communication with T. Peng, Group Supervisor, Telecommunications Systems Section, Jet Propulsion Laboratory, Pasadena, California, June 9, 1992.

C-3

parameters. Intuitively, B_0 should be selected to be as low as possible in order to minimize $\sigma^2_{\Phi_e}$ and maximize SNR_L in Eq. (11). However, this choice can conflict with speed and reliability concerns.

One problem with choosing B_0 too low is that much time will be wasted processing untrackable carrier segments at this bandwidth and the loop will have to reprocess nearly the whole time span again at the next bandwidth; small bandwidth increments exacerbate this problem. Wiser choices for B_0 do not fragment the data into many short locked and unlocked partitions that require significant processor time to compute filter state variables and to resynchronize the loop with the data. In general, a more successful initial pass through the data will speed the overall tracking process.

As a practical example, a Sun Sparc II workstation requires approximately 250 seconds to process one hour of contiguous data at any bandwidth. At an initial bandwidth of 0.1 Hz, tracking the carrier over the entire Galileo data set involved 16 forward and backward passes through the data, for a total time of approximately 550 seconds. For $B_0 = 0.05$ Hz, nearly 2400 sec are required for 18 passes due to the computational overhead involved in resolving this fragmentation. Simulations show that total processing time is roughly proportional to the length of data. However, its variation with B_0 depends strongly on how much of the carrier can be recovered at B_0. With excessively low bandwidths, less time is consumed by restarting the entire process at a higher bandwidth instead of retaining carrier phase information gathered at B_0 because the loop rarely holds lock for segments significantly longer than the delay through $L(z)$. Once again, judicious selection of B_0 and ΔB can greatly speed processing.

Low initial bandwidths can also impact the effectiveness of the lock detection circuit. For very narrow loops, the lock detector output tends to hover just above L_- for long intervals. It is natural for $\lambda[n]$ to waver between L_- and unity while in lock, but an affinity for values near L_- suggests that performance may be marginal even though the loop is considered locked according to the LOCK indicator criterion. This is especially a problem if B_0 is just at the threshold of trackability and bandwidth steps are very small. Under these conditions, SNR_L computed by Eq. 11 is likely to be an overestimate, and phase error variance will be underestimated.

These two phenomena suggest that some criterion be specified for determining if a given B_0 is acceptable. The current implementation requires that a single locked region of at least 100 seconds be found in the data. Not only is such a length helpful in preventing these problems, but the long segment also acts as a "seed" providing initial conditions for tracking in either direction with wider loops. Additionally, it may be helpful to require that the initial pass through the data result in a certain minimum percentage of the recorded span being tracked. This method would be particularly useful in a near-real-time implementation where the loop can be widened when this running average drops below the threshold.

VIII. Summary

For applications in which data recording and postprocessing are viable, an adaptive, noncausal tracking algorithm that increases loop bandwidth to accommodate transients in the data can provide significantly higher loop SNR's than currently achieved in the DSN. Simulations of the algorithm described here show that the Galileo carrier phase can be recovered with DPLL's as narrow as 0.025 Hz by using only a simple first-order loop filter. This is a potential 16-dB gain in loop SNR over the DSN Block IV receiver. Most of the time, however, improvements in the range of 10 to 13 dB are possible. Results for Pioneer 10 are inconclusive but may be useful when more experimental data become available.

References

[1] F. M. Gardner, *Phaselock Techniques*, 2nd ed., New York: John Wiley and Sons, 1979.

[2] S. Aguirre and W. J. Hurd, "Design and Performance of Sampled Data Loops for Subcarrier and Carrier Tracking," *TDA Progress Report 42-79*, vol. July–September 1984, Jet Propulsion Laboratory, Pasadena, California, pp. 81–95, November 15, 1984.

[3] S. Aguirre, W. J. Hurd, R. Kumar, and J. Statman, "A Comparison of Methods for DPLL Loop Filter Design," *TDA Progress Report 42-87*, vol. July–September 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 114–124, November 15, 1986.

[4] R. C. Tausworthe, "Design of Lock Detectors," *JPL Space Programs Summary 37-43*, vol. 3, Pasadena, California: Jet Propulsion Laboratory, pp. 71–75, January 31, 1967.

[5] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

[6] J. Kormylo and V. Jain, "Two Pass Recursive Digital Filter with Zero Phase Shift," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-22, pp. 384–387, October 1974.

[7] V. A. Vilnrotter, W. J. Hurd, and D. H. Brown, "Optimized Tracking of RF Carriers with Phase Noise, Including Pioneer 10 Results," *TDA Progress Report 42-91*, vol. July–September 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 141–157, November 15, 1987.

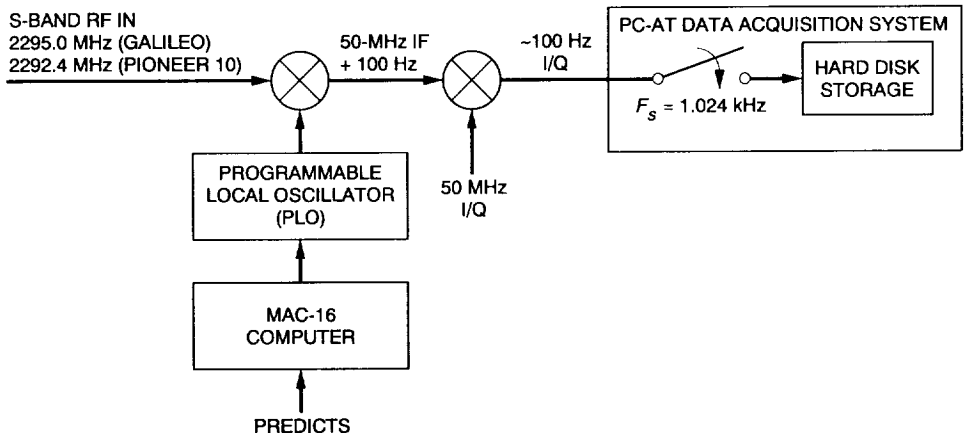

Fig. 1. Block diagram of the experimental setup.

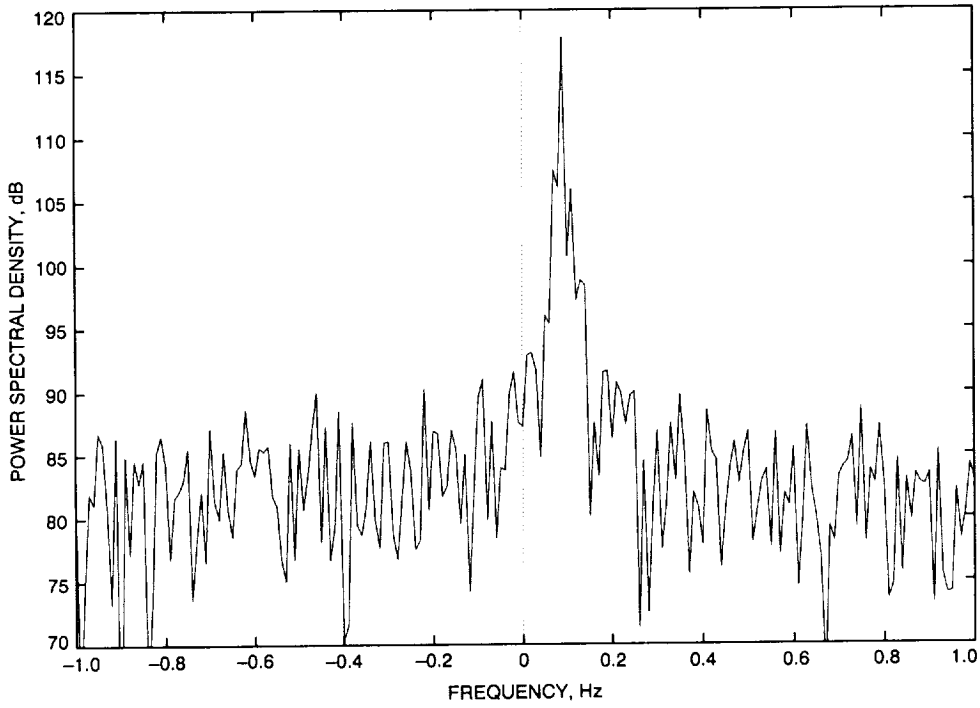

Fig. 2. Observed typical power spectrum (at baseband) for the Galileo carrier.

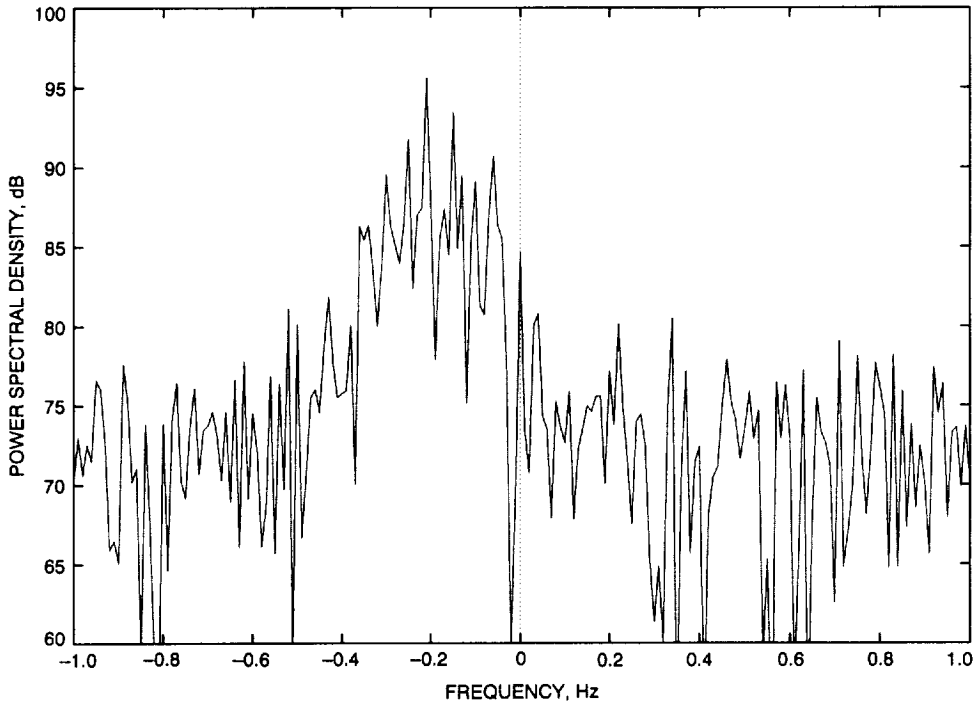

Fig. 3. Observed typical power spectrum (at baseband) for the Pioneer 10 carrier.

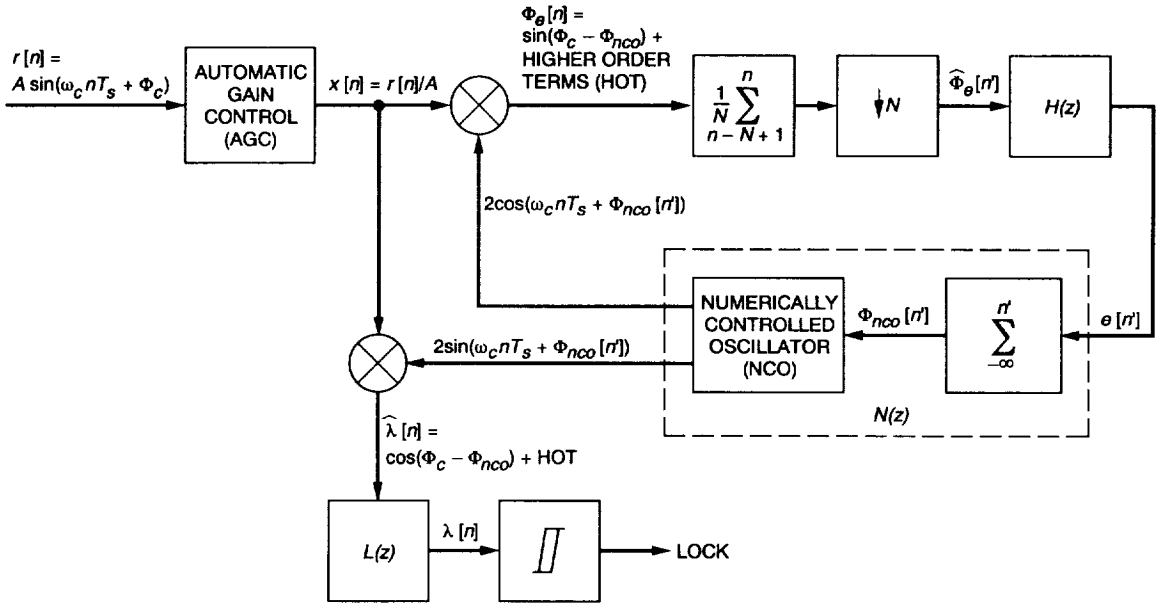

Fig. 4. Digital phase-locked loop (DPLL) with lock detector.

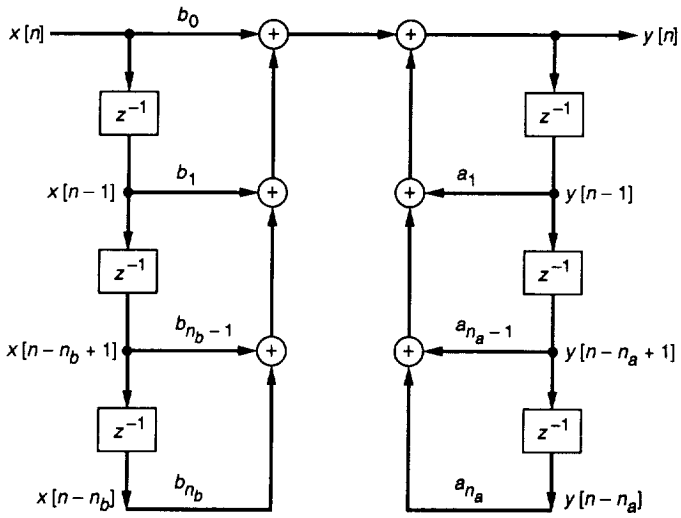

Fig. 5. Direct form 1 filter structure.

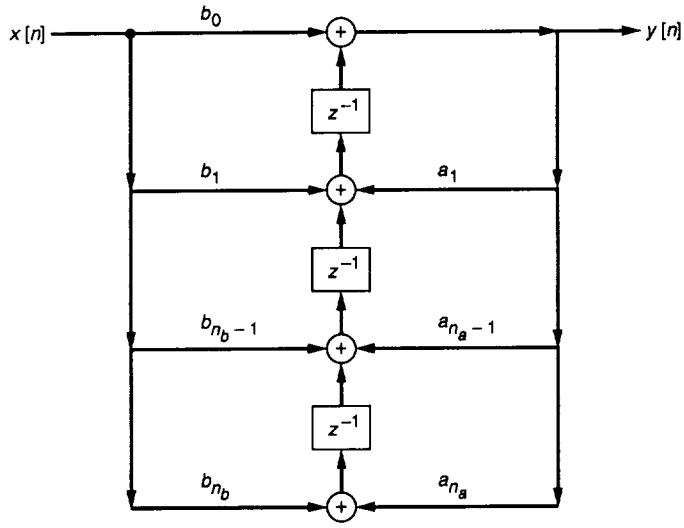

Fig. 6. Direct form 2 transposed filter structure.

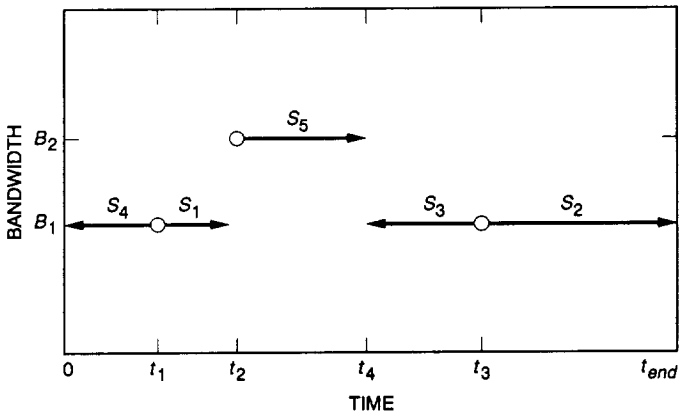

Fig. 7. Demonstration of algorithm operation.

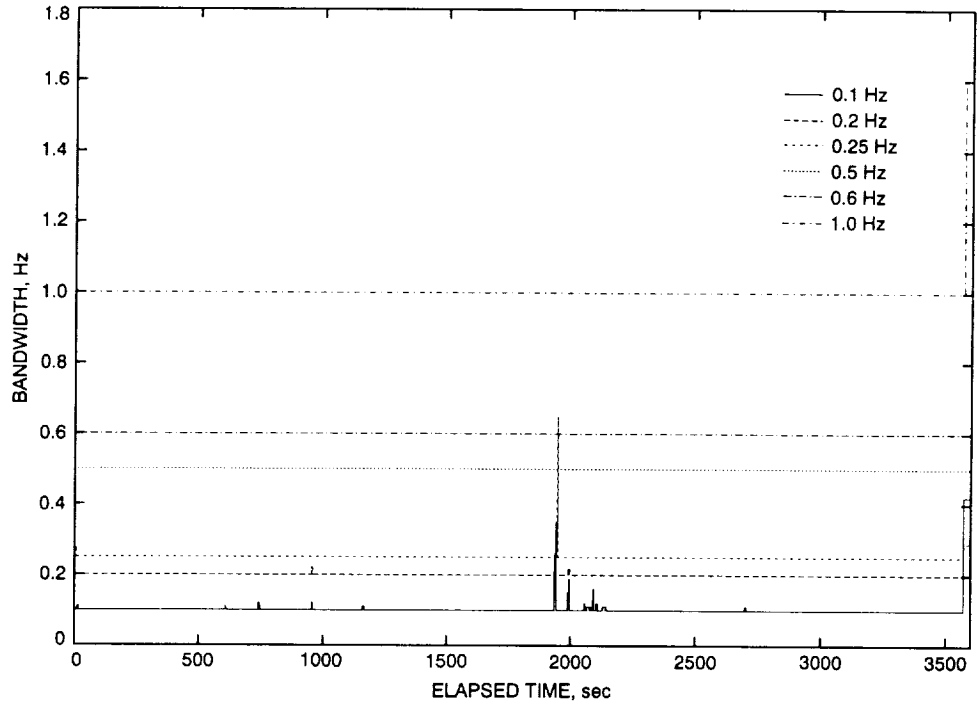

Fig. 8. Loop bandwidth for Galileo carrier phase recovery at various B_0. The events near ~3570 sec are due to a switch from one-way to two-way tracking.

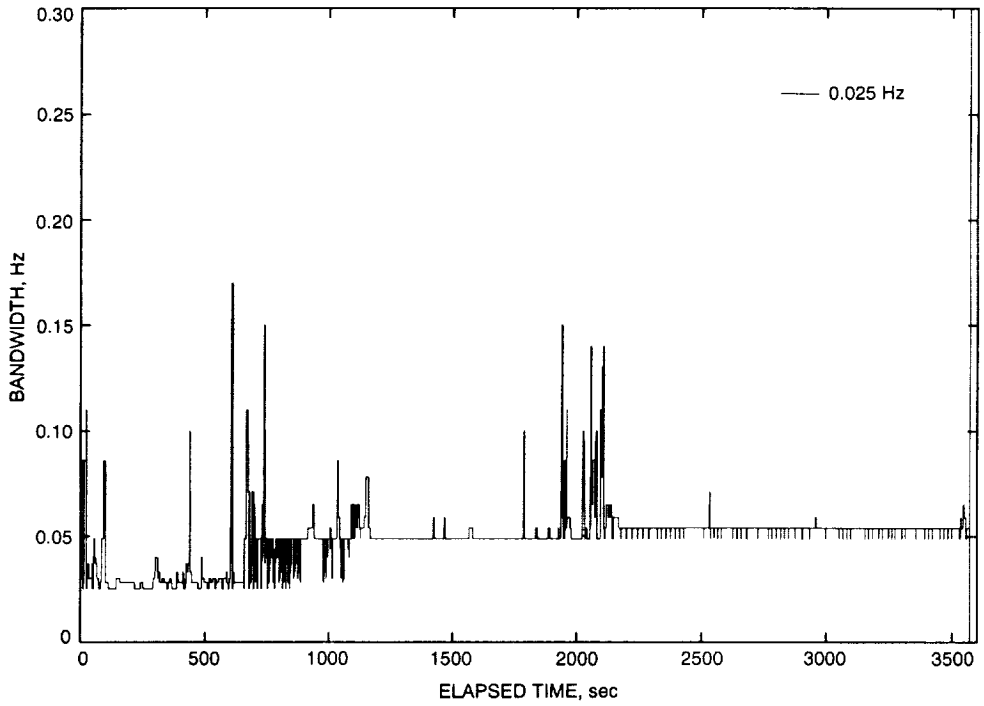

Fig. 9. Loop bandwidth for Galileo carrier phase recovery, B_0 = 0.025 Hz.

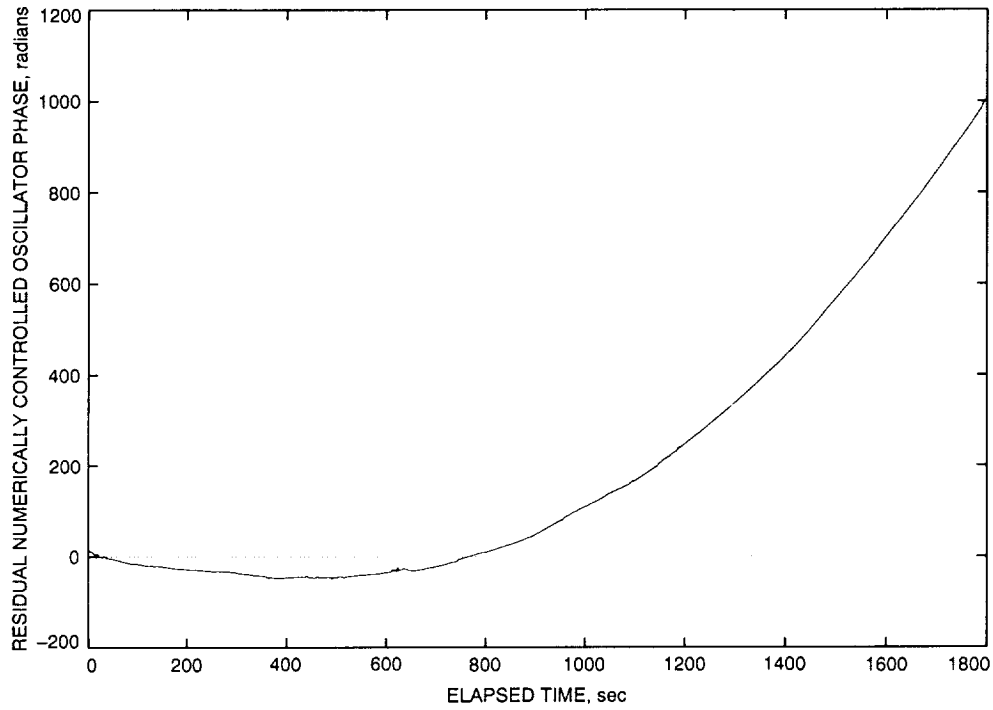

Fig. 10. Recovered phase from the first half hour of the experimental Galileo data set with initial bandwidth B_0 = 0.1 Hz.

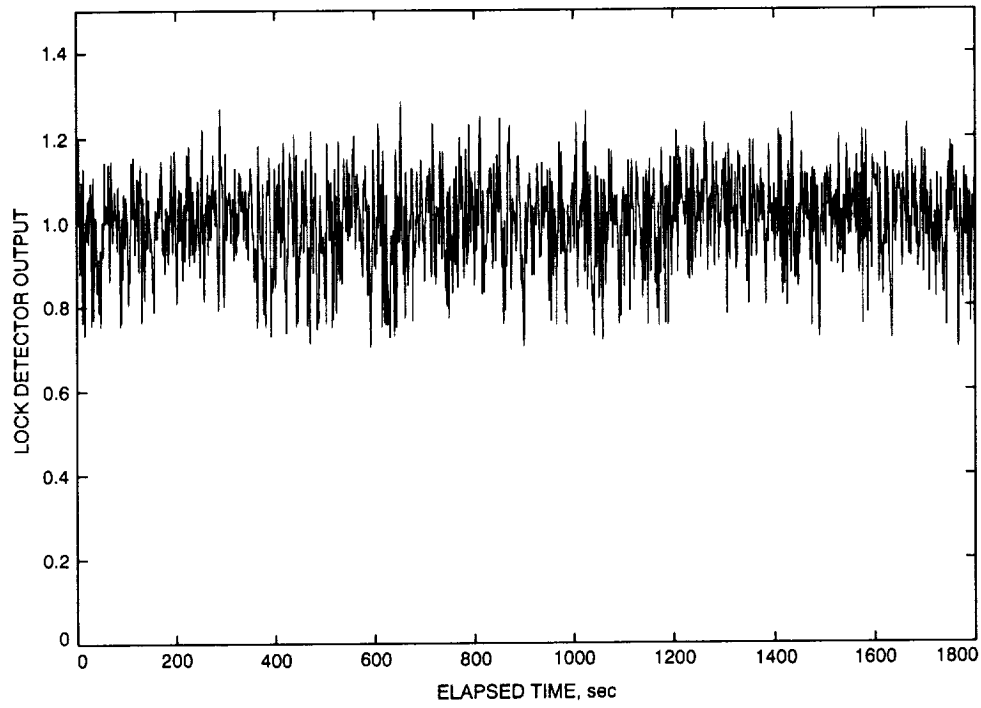

Fig. 11. Output from lock detection filter for the Fig. 10 simulation. Lower and upper hysteresis thresholds are 0.70 and 0.75.

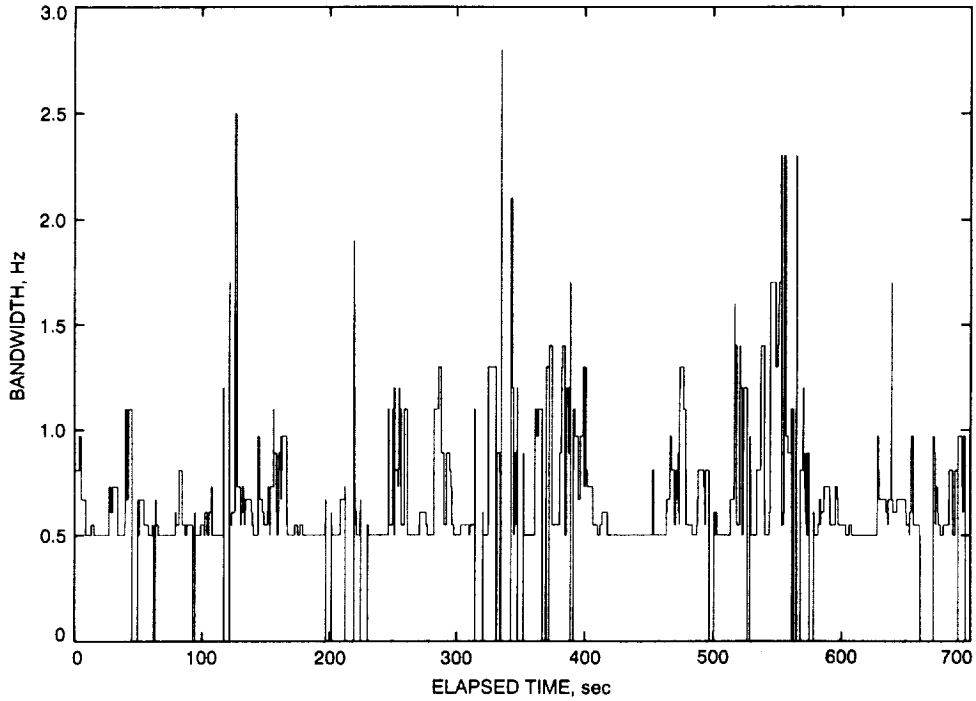

Fig. 12. Loop bandwidth for the Pioneer 10 carrier phase recovery, B_0 = 0.5 Hz.