NONLINEAR FILTERING WITH PIPELINE AND ARRAY PROCESSORS

R. S. Bucy

Aerospace Engineering Department
University of Southern California
Los Angeles, California 90007

Abstract

Monte Carlo analysis of the combined phase amplitude demodulator problem will be discussed. The effect of machine architectural differences on the nonlinear filtering algorithms will be emphasized. Two machines will be considered; the CDC 6600 and the Floating Point System's AP1203, which are respectively pipelined and multiprocessor arrays, in the context of numerical realization of the optimal nonlinear filter for the demodulation problem.

1. Introduction

The area of nonlinear filtering has been quite active in the past ten years with numerous papers on subjects such as Innovations, filtering on Lie Groups and a wide variety of suboptimal designs. Unfortunately, most of this work had little or no effect on the two fundamental questions which must be solved before the theory can be applied. Namely, how does one build a real-time nonlinear filter and determine the error performance of the optimal filter.

Recently in a thesis at M.I.T., Glada has made progress on the second problem as well as showing the connection between Information Theoretic Ideas and Filtering (see [1]). For some time now we have attempted to actually build nonlinear filters using digital computers (see [2]-[3]). Originally, for the two-dimensional phase demodulation problem our object was to build the filter in order to do Monte Carlo error analysis, and real-time filter operation was not considered. This was because the synthesis tool, the CDC 6600, was large, slow and expensive. Now, using the AP-1203 processor, real-time operation is possible. Of course, the Star 100 is 2.5 times faster than the array processor and, with its large memory, is an ideal tool for Monte Carlo analysis.

This research was supported in part by the United States Air Force, Air Force Systems Command, under Grant AF-AFOSR-76-1150 and Contract F46620-76-C-0085.

2. Mathematical Description of the Problem

Let \( F_n = \{P_n\} \) be the conditional density of the phase, \( \alpha_n \), phase rate, \( \beta_n \), and amplitude, \( A_n \) at time \( n \) given the observation \( \zeta_0 \ldots \zeta_n \).

\[
z_n(1) = h(A_n) \cos \alpha_n + \nu_n(1) \quad (2.1)
\]

where

\[
z_n(2) = h(A_n) \sin \alpha_n + \nu_n(2)
\]

with \( \nu \) independent Gaussian white noises of spacial density \( r \). It can be shown that

\[
P_{n+1}(x,y;\lambda) = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} a_i(\lambda,\xi) a_2(\gamma-\xi) P_n(x-\xi,y;\lambda) \, d\xi \, d\gamma
\]

\[
P_n(x,y;\lambda) = \exp \left( \frac{1}{r} h(\lambda) x \cos \lambda + h(\lambda) y \sin \lambda \right)
\]

\[
-\frac{1}{2} h^2(\lambda) P_n(x,y;\lambda)
\]

It is chosen so that \( F_n \) is a density and represents the discrete time step, and \( a_i \) is periodic of period \( 2\pi \). Note that \( F \) and \( \tilde{F} \) are periodic in their first and second arguments; of period \( 2\pi \) and \( 2\pi \) respectively. \( A_n, \alpha_n, \beta_n \) are Gaussian Markov processes--see [2] for details of the model. For the Monte Carlo runs on the Star 100, \( h \) was taken to be linear, a more physically interesting model would be exponential, as suggested by Dr. A. J. Malininckrodt. We represent the densities \( F_n \) and \( \tilde{F}_n \) as weights over a moving cube of lattice points in \( \mathbb{R}^2 \), for fixed amplitude the cross-section of the cubic lattice is a uniform grid with 16 subdivisions in phase and 96 in phase-rate. Now the fixed amplitude cross-sections are 16 in all and the configuration is centered at the best current estimate of amplitude with distances between cross-sections proportional to square root of the mean square error in estimating the amplitude.
3. Software Structure

Since the memory bandwidth of Star 100 is extremely high as long as a large number of contiguous memory locations are accessed sequentially, one can neglect memory accesses and trade storage for computation speed. By periodically extending the tensors representing F and F, the need for doing modular arithmetic can be eliminated. The detailed code can be obtained through NASA CACE.

Figure 1: Star Code

![Figure 1](image1.png)

Figure 2: AP-1208 Code

![Figure 2](image2.png)

The unusual point in the Star Code was viewing the phase convolution as the sum constants (i.e., values of the convolution kernel) times the density. This viewpoint overcomes the need to reference memory with fixed phase, which would lead to excessive timing on Star because non-contiguous memory references would be necessary, unless the memory was rearranged. Rearrangement is also time-consuming. For Star, the time necessary to accomplish one iteration of the loop pictured in Figure 2 is 269 milliseconds for the 3-dimensional problem and 5 milliseconds for the 2-dimensional problem—see [5].

The assembly language code for the AP-1208 was arranged in order to minimize the number of reads and writes from memory, as memory bandwidth is critical. The operations were arranged...
in two groups; phase rate fixed, scrambling and interpolation and phase fixed convolution. In each group a read and write are necessary for lightly more words than the number of grid points. Notice that the loop cycles on the one-step predictor, $\Phi_n$, rather than the fitter, $\Phi$. A significant feature of the API2OB code is the allowance of very precise statements concerning the double-precision code requires 3.5* versus 1.2L memory locations estimate production time for the AP1203 of the AP120B loop was suggested by Dr. K. D. Senne Monte Carlo statistics and evaluation of the close to the ratio of add plus multiply times of only tasks currently done in the I-I-53 are predictor, $P_\alpha$, rather than the fitter, $F_\alpha$.

Each group a read and write are necessary for noise ratios. Each run then computes the mean with left off, i.e., the state of the run is stored. In this way at each signal-to-noise ratio the requisite 200 Monte Carlo runs could be accomplished in pieces of 40 runs which would take around 30 minutes of Star 103 CPU time. When $h(A) = A$ and the kernel $q_1(A, t)$ approaches a delta function, it is clear that (2.2) becomes closer and closer to the update for the 2-dimensional phase demodulation problem studied in [3]. Using this fact, the 3-dimensional problem was debugged by comparing the results to the 2-dimensional situation. When $h(A) = A$ and as is Gaussian with variance $q_1 = 1$ mean 1, we have done Monte Carlo runs at output signal-to-noise ratios $B = -1$ and $-1.9$ dB. The resulting error variances were extremely close to those which we found in [3] for the corresponding 2-dimensional problem. In the future we intend to investigate the problem where $h(A) = e^A$ by Monte Carlo analysis as this may be a more realistic model of the real world problem.

The Floating Point System's API2OB is connected to a PDP11-55 computer with the multiuser operating system UNIX version 3. At the moment the Floating Point Processor has 16K word memory; in the near future we will expand the memory which will allow us to run the problem described here. In the interim, a background task has been installed which is a restartable Monte Carlo task for the 3-dimensional phase demodulation problem—see [3]. This task has allowed the evaluation of phase error statistics for extremely large numbers of Monte Carlo runs, 30,000, at various output signal-to-noise ratios. Each run then computes the mean square phase estimation on the basis of 3,900,000 estimates. These runs produce a one-sigma confidence interval of length 0.088 dB and allow very precise statements concerning the db improvement of the optimal phase demodulator over the traditional phase-locked design. Actual estimate production time is due 50% to the AP1203 and 50% to the overhead of communications between the AP1203 and the I-I-55 and PDP11 tasks. The only tasks currently done in the I-I-55 are generation of observations and solution of the sensor (i.e., evaluation of the exponential in the second formula of (2.2)). The large contribution of the PDP11-I-I to the estimate time in view of the limited tasks it performs is disturbing, and we are currently looking into where the time is used by timing various parts of the driver program.

Because of its 50-100 to one floating point computation speed advantage in either a real-time or a simulation environment, the API2OB should be assigned all the floating point computation tasks. The system programming described in this section was done by Tom Skealney of U.S.C.

5. Real Time Capability

The Floating Point Processor API2OB, because of its size, speed and cost, makes possible now a real-time nonlinear phase demodulator when used in conjunction with the PDP11-55. With direct memory access, the observations are sent to the PDP11-55 and the observations are input to the API2OB while it is computing the previous density (i.e., the update of the density is overlapped with the data acquisition and the estimate acquisition). It appears that the data rate could be 160 per second. Even higher data rates could be achieved in the near future with improved hardware now available in the same multiprocessor array configuration.

6. Conclusions

We have attempted to demonstrate how software was developed for nonlinear filter realization which took advantage of the machine architecture of the Star 100 and the API2OB array processor. We feel that the revolutions in machine design, speed and size have made real-time nonlinear filter construction possible. In the near future it is clearly feasible to design real systems whose operation is governed by nonlinear filters.

*L represents the number of grid points for the density representation.
Acknowledgments

We wish to thank NASA Langley Research Center and Ed Foudriat of NASA for granting us remote access to and computing time on the CDC Star 100. We received extensive systems from Richard Helston and programming assistance from Illona Howser of NASA Langley. Our Air Force Grant supported the telephone charges to do the remote computing on the Star 100.

Professor Richard Kaplan of U.S.C. provided us with appreciated help in interfacing the AP120B to the PDP11-55 and writing command files which automated a number of software development tasks.

References

**Report Title:**
Nonlinear Filtering with Pipeline and Array Processors.

**Performing Organization Name and Address:**
University of Southern California
Aerospace Engineering Dept
Los Angeles, California 90007

**Contract or Grant Number:**
AFOSR-76-3

**Keywords:**
Monte Carlo analysis of the combined phase amplitude demodulator problem will be discussed. The effect of machine architectural differences on the nonlinear filtering algorithms will be emphasized. Two machines will be considered; the CDC Star 100 and the Floating Point System's AF120E, which are respectively pipeline and multi-realization of the optimal nonlinear filter for the demodulation problem.