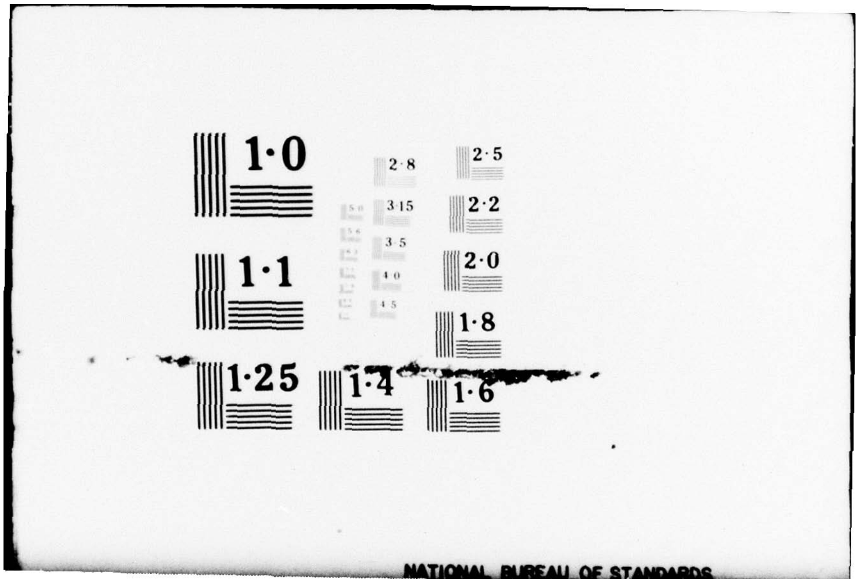
AD-A046 463 UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES DEPT O--ETC F/G 12/1
NONLINEAR FILTERING WITH PIPELINE AND ARRAY PROCESSORS.(U)
SEP 77 R S BUCY
F44620-76-C-0085
UNCLASSIFIED
AFOSR-TR-77-1275
NL

1 OF 1
ADA
046463

END
DATE
FILMED
12-77
DDC


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6
NATIONAL BUREAU OF STANDARDS

# NONLINEAR FILTERING WITH PIPELINE AND ARRAY PROCESSORS[1]

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 Star 100 and the Floating Point System's AP120B, which are respectively pipeline and multi-processor array, 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 has 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., Glados 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] - [5]). 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-120B processor, real-time operation is feasible. 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.

## 2. Mathematical Description of the Problem

Let $F_n \infty \{P_n\}$ be the conditional density of the phase, $X_n$, phase rate, $Y_n$, and amplitude, $A_n$ at time $_n$ given the observation $\underline{z}_n \ldots\ldots \underline{z}_0$ $\{\underline{z}_{n-1} \ldots\ldots \underline{z}_0\}$, where

$$\begin{aligned} z_n(1) &= h(A_n)\cos x_n + v_n^{'} \\ z_n(2) &= h(A_n)\sin x_n + v_n^{2} \end{aligned} \qquad (2.1)$$

with $v^i$ independent Gaussian white noises of spacial density $r$. It can be shown that

$$P_{n+1}(x,y,A) = \int_{-\infty}^{\infty}\int_{-\pi/\Delta}^{\pi/\Delta} a_1(A-\ell)a_2(y-\xi)F_n(x-\Delta\xi,\xi,\ell)\, d\xi d\ell \qquad (2.2)$$

$$F_n(x,y,A) = \frac{\exp\{}{K} \frac{1}{r}\left(h(A)z_n^{'}\cos x + h(A)z_n^{2}\sin x - \frac{1}{2}h^2(A)\right\} P_n(x,y,A)$$

$K$ is chosen so that $F_n$ is a density $\Delta$ represents the discrete time step, and $a_i$ is periodic of period $2\pi/\Delta$. Note that $F$ and $P$ are periodic in their first and second arguments; of period $2\pi$ and $2\pi/\Delta$ respectively. $A_n$, $x_n$, $y_n$ are Gauss-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 expodential, as suggested by Dr. A. J. Mallinckrodt. We represent the densities $P_n$ and $F_n$ as weights over a moving cube of lattice points in $R^3$, 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.

---

[1] This research was supported in part by the United States Air Force, Air Force Systems Command, under Grant AF-AFOSR-[redacted] and Contract F44620-76-C0085.
76-3100

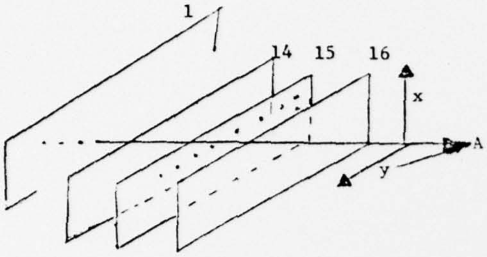


FIGURE 1

Notice that the grid structure moves only in the A direction. There are 24,576 grid points in all. Notice that (2.2) can be viewed as doing 16 computations of the two-dimensional phase demodulation problem, followed by a convolution over amplitude--see [3], [24] - [26]. This observation is the basis of the software realization for both the Star 100 and the Floating Point Processor. In order to explain the global structure of both the software for the Star 100 and the Array Processor, it will be useful to replace (2.2) by a series of simpler operations as;

scrambling: $F_n(x,y,A) \rightarrow F_n(x-\Delta y,y,A)$ (2.3)

{Notice that as F is periodic, modular arithmetic is involved.}

interpolation: $F_n(x_i-y_j\Delta,y_j,A_k) \rightarrow \alpha F_n(x^H_{i,j},y_j,A_k)$
$+(1-\alpha)F_n(x^L_{i,j},y_j,A_k)$ (2.4)

where $x^H_{ij}$ $\{x^L_{ij}\}$ is the grid point directly above {below} the value $x_i-y_j\Delta$;

$$\alpha = \frac{|x^L_{i,j} - (x_i-\Delta y_j)|}{|x^L_{ij} - x^H_{ij}|} ;$$

phase convolution: $F^I_n \rightarrow \sum_j a_2(\cdot,y_j)F^I_n(x_i,y_j,A_k)$
(2.5)

where $F^I$ is the image under the interpolation map.

amplitude convolution: $F^C_n \rightarrow \sum_\ell a_1(\cdot,\ell)F^C_n(x_i,y_j,\ell)$
(2.6)

new data: The second mapping in (2.2) with $K = 1$.

norming: Finding the sum of the weights, K .

normalization: Division by K.

## 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 P, the need for doing modular arithmetic can be eliminated. The detailed code can be obtained through ICASE, attention Dr. Bob Voight, NASA Langley, Hampton, Virginia. The original code was written by Dr. H. Youssef of Lockheed Aircraft, Burbank, following the 2D phase lockloop code, written by Dr. K. D. Senne. Youssef's code was debugged and corrected by the author. $F_n(x,y,A)$ is carried by Star as a vector whose $i+(M)(j-1)+MN(k-1)$ is the weight associated with the $i^{th}$ phase grid point and $j^{th}$ phase rate grid point and $k^{th}$ amplitude grid point. The structure of the program is given in Figure 2 below.

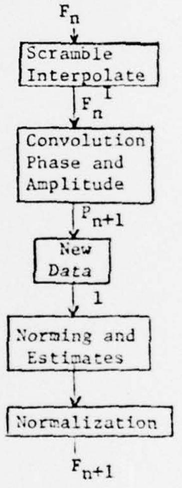


FIGURE 2 - STAR CODE

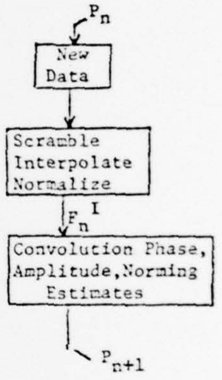


FIGURE 3 - AP-120B CODE

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 289 milliseconds for the 3-dimensional problem and 5 milliseconds for the 2-dimensional problem--see [5].

The assembly language code for the AP-120B 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, $P_n$, rather than the fitter, $F_n$.
A significant feature of the AP120B code is the parsimonious use of memory, which is a must for effective real-time synthesis. In fact, Star Code requires 3.5L* versus 1.2L memory locations for the AP120B Code. The Star Code runs 2.61 times faster than the AP120B Code, which is quite close to the ratio of add plus multiply times of the two machines, which is 2.78. The structure of the AP120B loop was suggested by Dr. K. D. Senne of M.I.T. Lincoln Laboratory and the coding of the convolution loop by Dr. A. J. Mallinckrodt of the Communications Research Laboratory, and the interpolation and scrambling loop by F. Ghovanlou of U.S.C.

## 4. Monte Carlo Programs

We developed a restartable Monte Carlo program for the Star 100. Each Monte Carlo run consisted of 130 consecutive estimates in time. Our program allowed us to do only a selected number of Monte Carlo runs at a time with a file written to allow us to continue later where we had left off, i.e., the state of the random number 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 100 CPU time. When $h(A) = A$ and the kernel $a_1(A,\ell)$ 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 $a_1$ is Gaussian with variance $q_3 = .1$ mean 1, we have done Monte Carlo runs at output signal-to-noise ratios $R = -3$ and $-1.8$ 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 AP120B is connected to a PDP11-55 computer with the multi-user operating system RX11M version 3. At the moment the Floating Point Processor has 16K word memory; in the near future we will expand the memory to 64K words 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 2-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 .008 db and allow very precise statements concerning the db-improvement of the optimal phase demodulator over the traditional phase-lock design. Actual estimate production time is due 50% to the AP120B and 50% to the overhead of communications between the AP120B and the 11-55 and PDP11 tasks. The only tasks currently done in the 11-55 are generation of observations and signal process, Monte Carlo statistics and evaluation of the sensor (i.e., evaluation of the expodential in the second formula of (2.2)). The large contribution of the PDP-11 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 AP120B should be assigned all the floating point computation tasks. The system programming described in this section was done by Tom Bleakney of U.S.C.

## 5. Real Time Capability

The Floating Point Processor AP120B, 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 inputed to the AP120B 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 100 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 AP120B 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 Hofler 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

[1] J. I. Glados, Thesis, Department of Electrical Engineering, M.I.T., Cambridge, Mass.,1975.

[2] R. S. Bucy, K. D. Senne, H. M. Youssef, "Pipeline, Parallel and Serial Realizations of Phase Demodulators," article in Stochastic Control, eds. E. Roxin and R. Sternberg, New York: Marcel Dekker, 1977.

[3] R. S. Bucy, C. Hecht, K. D. Senne, "New Methods for Nonlinear Filtering," Revue Francais d'Automatique, Informatique,Recherche Operationelle, J-1, 1973, 3-54.

[4] R. S. Bucy, K. D. Senne, "New Frontiers in Nonlinear Filter Computation," to appear.

[5] R. S. Bucy, K. D. Senne,"Nonlinear Filtering Algorithms for Parallel and Pipeline Machines," to appear in Proceedings Conference Parallel Mathematics, Parallel Computing, Munich, April 1977, North Holland Press, Amsterdam, 1977.

[6] Software Development Package Manuals, Floating Point Systems, Portland, Oregon, 1976.

[7] Star-100 Computer System Reference Manual, Control Data Corporation Publication No.60256000, 1975.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

(18) (19) **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS
BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>AFOSR-TR-77-1275 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>(6) NONLINEAR FILTERING WITH PIPELINE AND ARRAY PROCESSORS. | | 5. TYPE OF REPORT & PERIOD COVERED<br>(9) Interim rept., |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>(10) R. S./Bucy | | (15) 8. ~~CONTRACT OR GRANT NUMBER(s)~~<br>F44620-76-C-0085,<br>AFOSR-76-3100 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>University of Southern California<br>Aerospace Engineering Dept<br>Los Angeles, California 90007 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>61102F (17) A1<br>(16) 2304/A1 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Air Force Office of Scientific Research/NM<br>Bolling AFB DC 20332 | | 12. REPORT DATE<br>(11) Sep 77 |
| | | 13. NUMBER OF PAGES<br>4 |
| 14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*<br>(12) 5p. | | 15. SECURITY CLASS. *(of this report)*<br>UNCLASSIFIED |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

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 AP120B, which are respectively pipeline and multi-realization of the optimal nonlinear filter for the demodulation problem.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

402 019

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

LB