Professional Development Short Course On:

Practical Statistical Signal Processing — using MATLAB

Instructor:

Dr. Steven Kay

ATI Course Schedule: http://www.ATIcourses.com/schedule.htm

ATI's Practical Statistical Signal Processing: http://www.aticourses.com/practical_statistical_signal.htm

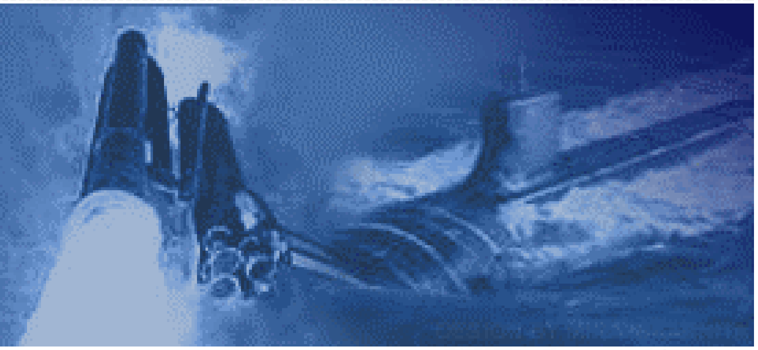
Applied Technology Institute (ATI)

Stay Current In Your Field • Broaden Your Knowledge • Increase Productivity

349 Berkshire Drive • Riva, Maryland 21140
888-501-2100 • 410-956-8805
Website: www.ATIcourses.com • Email: ATI@ATIcourses.com

Applied Technology Institute (ATI)

Stay Current In Your Field • Broaden Your Knowledge • Increase Productivity



www.ATIcourses.com

Boost Your Skills with On-Site Courses Tailored to Your Needs

349 Berkshire Drive
Riva, Maryland 21140
Telephone 1-888-501-2100 / (410) 965-8805
Fax (410) 956-5785
Email: ATI@ATIcourses.com

The Applied Technology Institute specializes in training programs for technical professionals. Our courses keep you current in the state-of-the-art technology that is essential to keep your company on the cutting edge in today's highly competitive marketplace. Since 1984, ATI has earned the trust of training departments nationwide, and has presented on-site training at the major Navy, Air Force and NASA centers, and for a large number of contractors. Our training increases effectiveness and productivity. Learn from the proven best.

For a Free On-Site Quote Visit Us At: http://www.ATIcourses.com/free_onsite_quote.asp

For Our Current Public Course Schedule Go To: http://www.ATIcourses.com/schedule.htm

References

1[*]. S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, 1993
2[*]. S. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, Prentice-Hall, 1998
3. L. Scharf, *Statistical Signal Processing*, Addison-Wesley,
Reading, MA, 1991 (more advanced treatment)
4. R.N. McDonough, A.D. Whalen, *Detection of Signals in Noise*, Academic Press, New York, 1995
5. H.L. Van Trees, *Detection, Estimation, and Modulation Theory, Vol. I*, J. Wiley, New York, 1968 (fairly involved but a classic)
6. G.M. Jenkins, D.G. Watts, *Spectral Analysis and its Applications*, Holden-Day, 1968
7. S. Kay, *Modern Spectral Estimation: Theory and Application*, Prentice-Hall, 1988
8. M.B. Priestley, *Spectral Analysis and Time Series*, Academic Press, 1981
9. R.A. Monzingo, T.W. Miller, *Adaptive Arrays*, J. Wiley, 1980
10. D.H. Johnson, D.E. Dudgeon, *Array Signal Processing*, Prentice-Hall, 1993

[*] Provided as part of course materials

Summary of Slides

MATLAB Basics

Version: 5.2 for Windows

Useful toolboxes: signal processing, statistics, symbolic

m files: script files

Fortran vs. MATLAB example:

Signal generation

Math: $s[n] = \cos(2\pi f_0 n) \quad n = 0, 1, \ldots, N-1$

Fortran:

```
      pi=3.14159
      f0=0.25
      N=25
      do 10 I=1,N
10    s(I)=cos(2*pi*f0*(I-1))
```

MATLAB:

```
f0=0.25;N=25;
s=cos(2*pi*f0*[0:N-1]');
```

Notes: pi already defined, [0:N-1]’ is a column vector, cosine of vector of samples produces a vector output, MATLAB treats vectors and matrices as elements

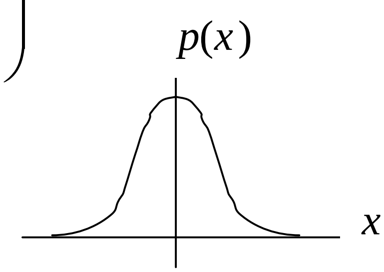
Noise Generation

Simplest model for observation noise is white Gaussian noise (WGN)

Definition: zero mean, all samples are uncorrelated, power
spectral density (PSD) is flat, and first order probability density function (PDF) is
Gaussian

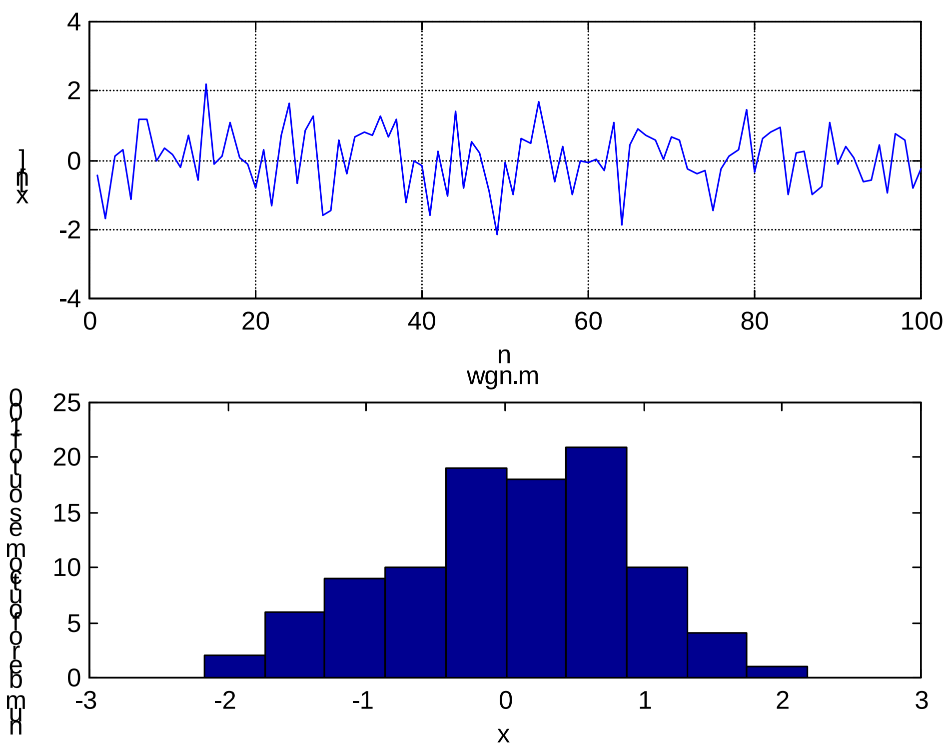
PDF:

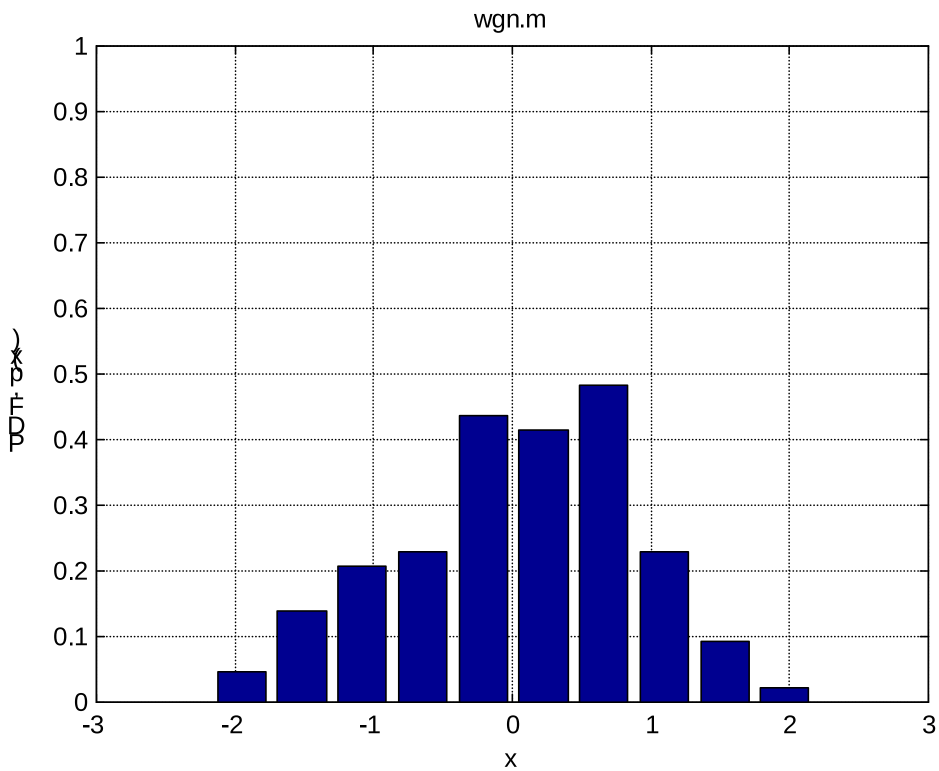
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} x^2\right)$$



where σ^2 = variance

MATLAB Example: $\sigma^2 = 1$





<u>Note</u>: randn(‘state’,0) sets random number generator to default seed and thus generates the same set of random numbers each time the program is run.

<u>MATLAB code</u>:

```
% wgn.m
%
%  This program generates and plots
the time series, histogram, and
```

```
%  estimated PDF for real white
Gaussian noise.
randn('state',0)
x=randn(100,1);
subplot(2,1,1)
plot(x)
xlabel('n')
ylabel('x[n]')
grid
subplot(2,1,2)
hist(x)
xlabel('x')
ylabel('number of outcomes out of
100')
title('wgn.m')
figure
pdf(x,100,10,-3,3,1)
xlabel('x')
ylabel('PDF, p(x)')
title('wgn.m')


% pdf.m
%
function
pdf(x,N,nbins,xmin,xmax,ymax)
%
```

```
%  This function subprogram computes
and plots the
%  PDF of a set of data.
%
%  Input parameters:
%
%    x    - Nx1 data array
%    N    - number of data points
%    nbins - number of bins (<N/10)
%    xmin,xmax,ymax - axis scaling
%
  [y,xx]=hist(x(1:N),nbins);
  delx=xx(2)-xx(1);
  bar(xx,y/(N*delx))
  grid
  axis([xmin xmax 0 ymax]);
```

Complex White Gaussian Noise

Definition: $x[n] = w_1[n] + jw_2[n]$

where $w_1[n]$ and $w_2[n]$ are independent of each other and
each one is real WGN with variance of $\sigma^2 / 2$

Mean: $E(x[n]) = 0$

Variance: $\text{var}(x[n]) = \text{var}(w_1[n]) + \text{var}(w_2[n]) = \sigma^2$

MATLAB code:

```
% cwgn.m
%
% This program generates complex
white Gaussian noise and
% then estimates its mean and
variance.
%
N=100;
varw=1;
x=sqrt(varw/2)*randn(N,1)+j*sqrt(varw
/2)*randn(N,1);
muest=mean(x)
varest=cov(x)
```

NonGaussian Noise

Generation: transform WGN using a nonlinear memoryless
transformation

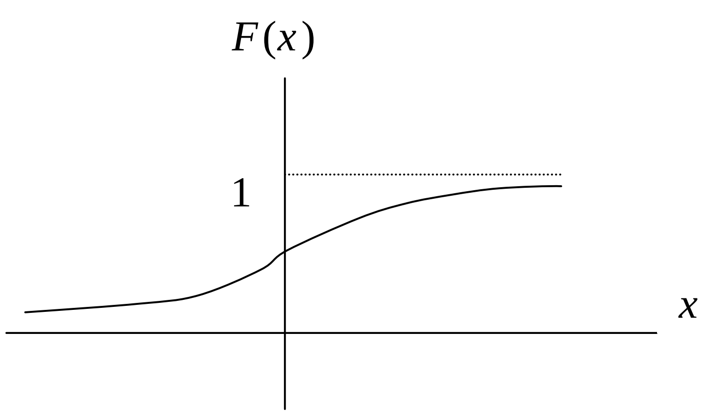
Example: Laplacian noise

$$p(x) = \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\sqrt{\frac{2}{\sigma^2}} |x| \right)$$

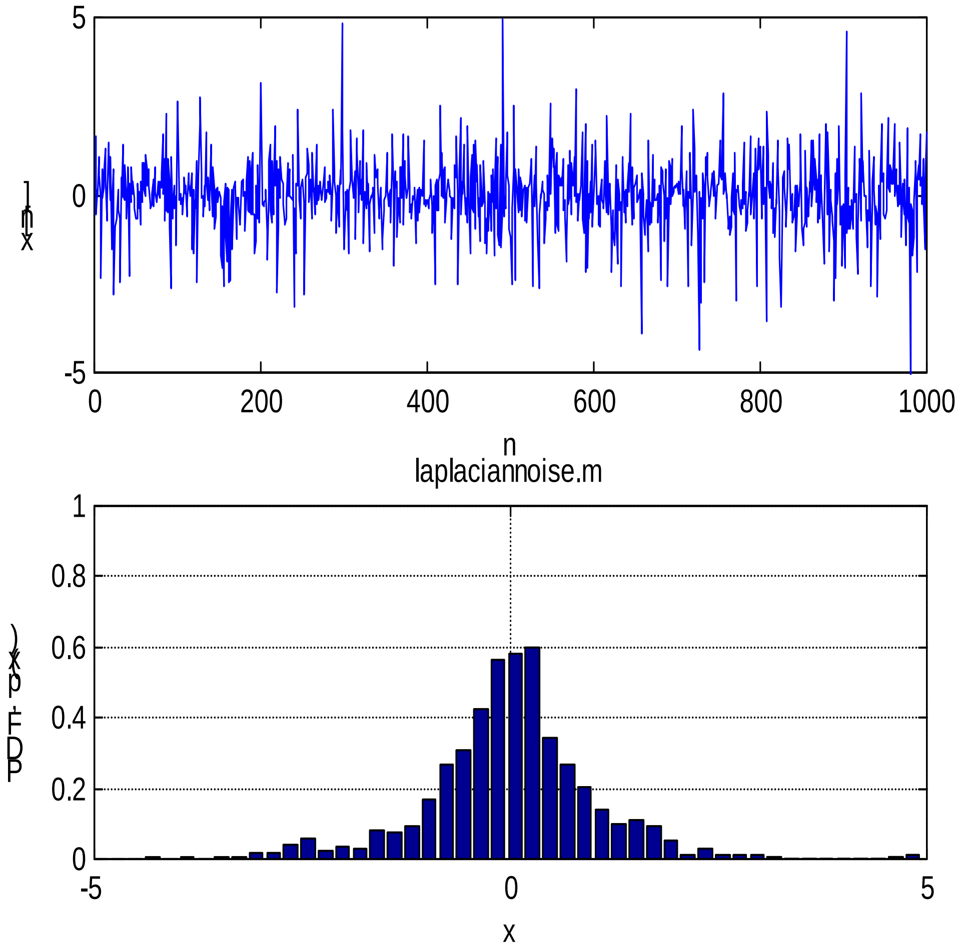
Use the transformation

$$x = F^{-1}(w)$$

where w is uniform random variable on the interval [0,1]
and F is the cumulative distribution function of the Laplacian
PDF.



Example: $\sigma^2 = 1$



MATLAB Code:

```
% laplaciannoise.m
%
% This program uses a memoryless
transformation of a uniform
% random variable to generate a set
of independent Laplacian
% noise samples.
%
  rand('state',0)
  varx=1;N=1000;
   u=rand(N,1);
   for i=1:N
     if u(i)>0.5

x(i,1)=sqrt(varx)*(1/sqrt(2))*log(1/(
2*(1-u(i))));
     else

x(i,1)=sqrt(varx)*(1/sqrt(2))*log(2*u
(i));
     end
     end
  subplot(2,1,1)
  plot(x)
  xlabel('n')
  ylabel('x[n]')
```

```
axis([0 1000 -5 5]);
subplot(2,1,2)
pdf(x,N,50,-5,5,1)
title('laplaciannoise.m')
```

Solving Parameter Estimation Problems

Approach:

1. Translate problem into manageable estimation problem

2. Evaluate best possible performance (bounds)

3. Choose optimal or suboptimal procedure

4. Evaluate actual performance

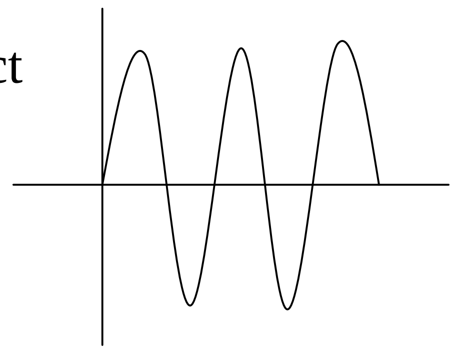
   a. Analytically – exact or approximate
   b. By computer simulation

Radar Doppler Estimation
(Step 1)

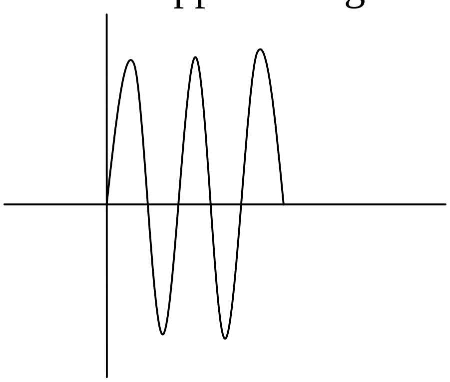
Problem: Given radar returns from automobile, determine speed to within 0.5 mph

Physical basis: Doppler effect

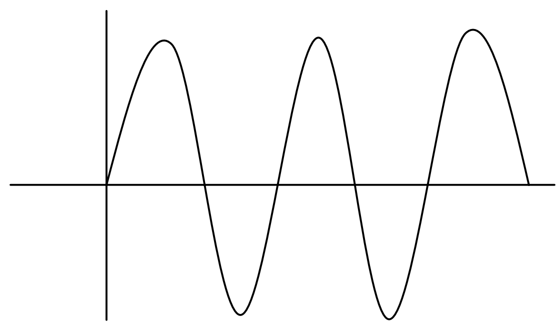
transmit



receive –
approaching



receive-
moving away



Received frequency is

$$F = F_0 + \underbrace{\frac{2v}{c} F_0}_{F_D}$$

where v = velocity, c= speed of light, F_0 = sinusoidal transmit
frequency

To measure the velocity use

$$v = \frac{c}{2} \frac{F - F_0}{F_0}$$

and estimate the frequency to yield

$$\hat{v} = \frac{c}{2} \frac{\hat{F} - F_0}{F_0}$$

Modeling and Best Possible Performance
(Step 2)

Preprocessing: first demodulate to baseband to produce the
sampled complex envelope or

$$\tilde{s}[n] = (A/2)\exp(j2\pi F_D n\Delta + \varphi)$$
$$\left(F_D = \frac{2v}{c}F_0\right)$$

Must sample at $F_s = 1/\Delta > 2F_D = 2\left(\frac{2v_{\max}}{c}F_0\right)$

Example: $v_{\max}$=300 mph, F_0=10.5 Ghz (X-band),
$c = 3\text{x}10^8$ m/s

$$F_{D-\max} = \frac{2v_{\max}}{c}F_0 \approx 9388\,\text{Hz}$$

$$\Rightarrow F_s > 18,776 \text{ complex samples/sec}$$

How many samples do we need?

Spec: error must be less than 0.5 mph for

$$\text{SNR} = 10\log_{10}\frac{(A/2)^2}{\sigma^2} > -10\,\text{dB}$$

Cramer-Rao Lower Bound for Frequency

- tells us the minimum possible variance for estimator – very useful for feasibility studies

$$\text{var}(\hat{f}_D) \geq \frac{6}{(2\pi)^2\eta N(N^2-1)} \qquad (*) \qquad \text{(see [Kay 1988])}$$

where $f_D = F_D / F_s$, N = number of complex samples, η=linear SNR

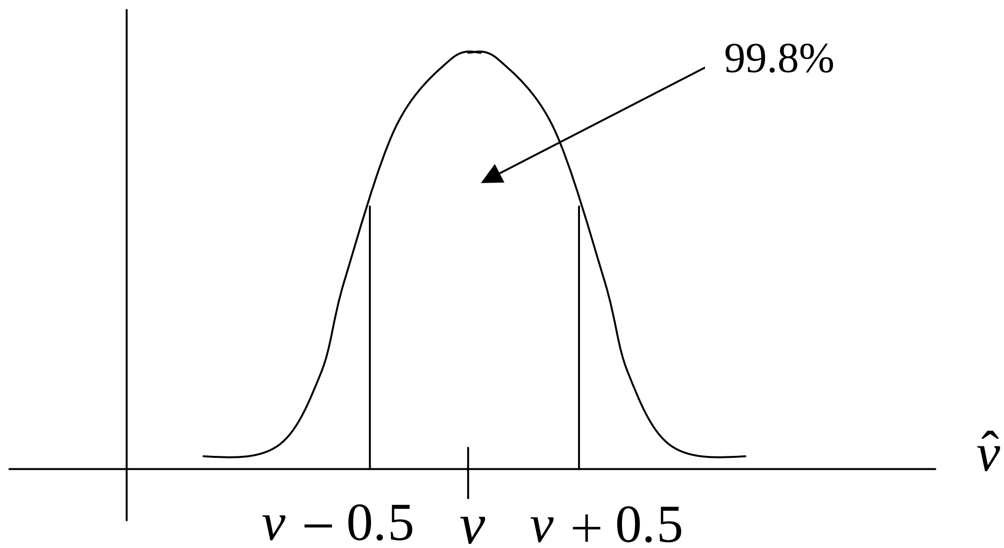
Since $F_D = (2v/c)F_0 \Rightarrow v = \dfrac{cF_s}{2F_0} f_D$

and we can show that

$$\text{var}(\hat{v}) = \left(\frac{cF_s}{2F_0}\right)^2 \text{var}(\hat{f}_D)$$

For an error of 0.5 mph (0.22 m/s) set

$$3\sqrt{\mathrm{var}(\hat{v})} = 0.22 \;\Rightarrow\; \mathrm{var}(\hat{f}_D) = 7.47\mathrm{x}10^{-8}$$



and finally we have from (*) that

$$N > \left[\frac{6}{(2\pi)^2 \eta \,\mathrm{var}(\hat{f}_D)} \right]^{1/3} \approx 272 \text{ samples}$$

Descriptions of MATLAB Programs

1. **analogsim** – simulates the action of an RC filter on a pulse

2. **arcov** - estimates the AR power spectral density using he covariance method for AR parameter estimation for real data.

3. **arexamples** - gives examples of the time series and corresponding power spectral density for various AR models. It requires the function subprograms: gendata.m and armapsd.m.

4. **armapsd** - computes a set of PSD values, given the parameters of a complex or real AR or MA or ARMA model.

5. **arpsd** - plots the AR power spectral density for some simple cases. The external subprogram armapsd.m is required.

6. **arpsdexample** - estimates the power spectral density of two real sinusoids in white Gaussian noise using the periodogram and AR spectral estimators.
It requires the functions subprograms: per.m and arcov.m.

7. **arrivaltimeest** - simulates the performance of an arrival time estimator for a DC pulse. The estimator is a running correlator which is the MLE for white Gaussian noise.

8. **avper** - illustrates the effect of block averaging on the periodogram for white Gaussian noise.

9. **classicalbayesian** - demonstrates the difference between the classical approach and the Bayesian approaches to parameter modeling.

10. **cwgn** - generates complex white Gaussian noise and then estimates its mean and variance.

11. **DClevelhist** - generates Figures 1.4, 1.5 in "Fundamentals of Statistical Signal Processing: Detection Theory", S. Kay

12. **DCleveltime** - generates a data set of white Gaussian noise only and also a DC level A in white Gaussian noise

13. **discretesinc** – plots the graph in linear and dB quantities of a discrete sinc pulse in frequency

14. **estperform** - compares the frequency estimation performance for a single complex sinusoid in complex white Gaussian using the peak location of the periodogram and an AR(1) estimator.

15. **Fig35new** - computes Figure 3.5 (same as Figure 4.5) in "Fundamentals of Statistical Signal Processing: Detection Theory", S. Kay. The function subprograms Q.m and Qinv.m are required.

16. **Fig39new** - computes Figure 3.9 in "Fundamentals of Statistical Signal Processing: Detection Theory", S. Kay. The function subprograms Q.m and Qinv.m are required.

17. **Fig77new** - computes Figure 7.7 in "Fundamentals of Statistical Signal Processing: Detection Theory", S. Kay.

18. **gendata** - generates a complex or real AR, MA, or ARMA time series given the filter parameters and excitation noise variance.

19. **kalman** - implementation of the vector state-scalar observation linear Kalman filter. See (13.50)-(13.54) of "Fundamentals of Statistical Signal Processing: Estimation Theory" by S. Kay for more details.

20. **kalmanexample** - uses the linear Kalman filter to estimate the tap weights for a random TDL channel. It generates Figures 13.16-13.18 in "Fundamentals of Statistical Signal Processing: Estimation Theory", S. Kay. It requires the function subprogram kalman.m.

21. **laplaciannoise** - uses a memoryless transformation of a uniform random variable to generate a set of independent Laplacian noise samples.

22. **linearmodel** - computes the optimal estimator of the parameters of a real or complex linear model. Alternatively, it is just the least squares estimator.

23. **linearmodelexample** - implements a line fit to a noise corrupted line. The linear model or least squares estimator is used. The function subprogram linearmodel.m is required.

24. **MAexample** – plots out the PDF of an MA process

25. **mlevar** - computes the mean, variance, PDF of the MLE for the power of a WGN process and compares it to the CRLB.

26. **montecarloroc** - uses a Monte Carlo approach to determine the detection performance of a Neyman-Pearson detector for a DC level in WGN. The true

performance is shown in "Fundamentals of Statistical Signal Processing: Detection Theory", S. Kay, in Figure 3.9 for d^2=1. The function subprogram roccurve.m is required.

27. **pcar** - estimates the frequencies of real sinusoids by using the principal component AR approach. Futher details can be found in "Modern Spectral Estimation: Theory and Application", by S. Kay.

28. **pdf** - computes and plots the PDF of a set of data.

29. **per** - computes the periodogram spectral estimator. Futher details can be found in "Modern Spectral Estimation: Theory and Application", by S. Kay.

30. **perdetectexample** - illustrates the detection performance of a periodogram, which is an incoherent matched filter.

31. **perexamples** - illustrates the capability of the periodogram for resolving spectral lines.

32. **plot1** – plots a sinusoid

33. **psk** - implements a matched filter receiver for the detection of a PSK signal. The data are assumed real.

34. **pskexample** - illustrates the optimal detection/decoding of a PSK encoded digital sequence. The bits are decoded and the probability of error is computed and compared to the number of actual errors. The external function subprogram psk.m is required.

35. **Q** - computes the right-tail probability (complementary cumulative distribution function) for a N(0,1) random variable.

36. **Qinv** - computes the inverse Q function or the value which is exceeded by a N(0,1) random variable with a probability of x.

37. **repcorr** - implements a replica correlator for either real or complex data.

38. **repcorrexample** - illustrates the replica-correlator. It requires the subprogram repcorr.m.

39. **roccurve** - determines the ROCs for a given set of detector outputs under H0 and H1.

40. **sampling** – plots out an analog sinusoid and the samples taken

41. **seqls** - implements a sequential least squares estimator for a DC level

in WGN of constant variance.

42. **shift** - shifts the given sequence by a specified number of samples. Zeros are shifted in either from the left or right.

43. **signdetexample** - implements a sign detector for a DC level in Gaussian-mixture noise. A comparison is made to a replica correlator, which is just the sample mean.

44. **sinusoid** - generates a sinusoid

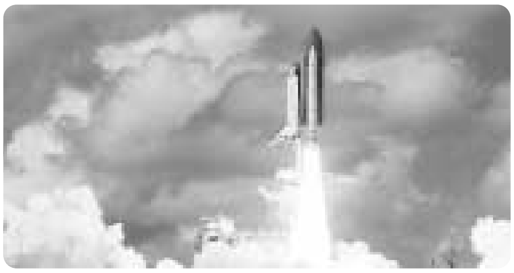
45. **stepdown** - implements the step-down procedure to find the coefficients and prediction error powers for all the lower order predictors given the filter parameters and white noise variance of a pth order AR model. See (6.51) and (6.52). This program has been translated from Fortran into Matlab. See "Modern Spectral Estimation" by S. Kay for further details.

46. **timedelaybfr** - implements a time delay beamformer for a line array of 3 sensors. The emitted signal is sinusoidal and is assumed to be at broadside or at 90 degrees (perpendicular to line array).

47. **wgn** - generates and plots the time series, histogram, and estimated PDF for real white Gaussian noise.

48. **wiener** - implements a Wiener smoother for extracting an AR(1) signal in white Gaussian noise and also for predicting an AR(1) signal for no observation noise present.

Boost Your Skills with On-Site Courses Tailored to Your Needs



The Applied Technology Institute specializes in training programs for technical professionals. Our courses keep you current in the state-of-the-art technology that is essential to keep your company on the cutting edge in today's highly competitive marketplace. For 20 years, we have earned the trust of training departments nationwide, and have presented on-site training at the major Navy, Air Force and NASA centers, and for a large number of contractors. Our training increases effectiveness and productivity. Learn from the proven best.

ATI's on-site courses offer these cost-effective advantages:

- You design, control, and schedule the course.
- Since the program involves only your personnel, confidentiality is maintained. You can freely discuss company issues and programs. Classified programs can also be arranged.
- Your employees may attend all or only the most relevant part of the course.
- Our instructors are the best in the business, averaging 25 to 35 years of practical, real-world experience. Carefully selected for both technical expertise and teaching ability, they provide information that is practical and ready to use immediately.
- Our on-site programs can save your facility 30% to 50%, plus additional savings by eliminating employee travel time and expenses.
- The ATI Satisfaction Guarantee: You must be completely satisfied with our program.

We suggest you look at ATI course descriptions in this catalog and on the ATI website. Visit and bookmark ATI's website at http://www.ATIcourses.com for descriptions of all of our courses in these areas:

- Communications & Computer Programming
- Radar/EW/Combat Systems
- Signal Processing & Information Technology
- Sonar & Acoustic Engineering
- Spacecraft & Satellite Engineering

I suggest that you read through these course descriptions and then call me personally, Jim Jenkins, at (410) 531-6034, and I'll explain what we can do for you, what it will cost, and what you can expect in results and future capabilities.

Our training helps you and your organization remain competitive in this changing world.