NEXT TIME:

* Specify that if you u se 1/*e* decorrelation defn, you must compare to 1/*e* for the theoretical ACF
* Specify all angles to be plotted in degrees
* Provide info on interpreting/labeling ACF axis

GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL ENGINEERING

ECE 6272 FALL 2010

COMPUTER PROJECT #1

Assigned: Tuesday, August 31, 2010

**Due Date for On-Campus Students: Tuesday, September 14 @ 9:35 AM Eastern Time**

**Due Date for Video Students: Tuesday, September 21 @ 4:00 PM Eastern Time**

This project is to be done *individually.* Each student must develop his or her own computer code in its entirety. Students are not to discuss the theory or approaches to coding the theory with one another, nor are they to assist in debugging each other’s work. You may ask Dr. Richards questions regarding theory and implementation of the project, including asking them at the beginning or end of class or during office hours, when others can benefit as well. MATLAB is the preferred language, but others are acceptable; the point is to try the experiments, not to improve your MATLAB skills.

Reports will be graded on completeness in addressing the assignment and quality of results. They will not be graded on programming style or efficiency or on writing quality, except that the programming and the writing should be clear enough to be reasonably understandable. Questions or clarifications about the assignment should be directed to Dr. Richards.[[1]](#footnote-1) Errata, revisions and hints (if any) will be made available via the class T-Square site or during class.

# PROBLEM

In Section 2.2.5 of the text, it was demonstrated that the radar cross section (RCS) of a target modeled as a set of “many” individual point scatterers was a sufficiently complex function of aspect angle (or frequency) that it could be modeled as a random variable; and further that the law of large numbers suggests that one assume an exponential probability density function (PDF) for the RCS (equivalently, a Rayleigh PDF for the square root of the RCS, which I will refer to as the “voltage”). Figures 2-8 through 2-10 were an example of this model. In this project, you will repeat this example, noting the effect of varying the number of scatterers. You will also test the validity of the estimated decorrelation interval in angle and in frequency that was derived (Eqns. 2.62 – 2.64) based on a line array of scatterers.

# REPORT FORMAT

To aid in the grading, please observe the following format constraints. All plots should be adequately labeled and identified so I can tell which plot goes with which part of the assignment.

* The first page should be a cover page with the class number, your name, and the project title.
* The second page or two of your report should contain the following specific items, with supporting explanation (parameters used, derivations, observations on the results, *etc.*), and your answers to any questions asked about this portion of the assignment:
  + the plot of composite RCS *vs.* aspect angle for the single 50 scatterer target (¶3.1.4); and
  + the plot of the single-target data histogram and its overlaid exponential PDF (¶3.2.1 and 3.2.2).
* The next page or two of your report should contain the following specific items, with supporting explanation and your answer to related questions:
  + the plot of the averaged 10-target data histogram and its overlaid exponential PDF (¶3.3).
* The next page or two of your report should contain the following specific items, with supporting explanation (parameters used, derivations, observations on the results, *etc.*):
  + the plot of the single-target RCS *vs.* frequency data histogram and its overlaid exponential PDF for the fixed aspect angle, frequency-stepped case (¶3.4).
  + the plot of the RCS *vs.* frequency data 10-target averaged histogram and its overlaid exponential PDF for the fixed aspect angle, frequency-stepped case (¶3.4).
* The next page or two of your report should contain the following specific items, with supporting explanation (parameters used, derivations, observations on the results, *etc.*):
  + a plot of the magnitude of the RCS autocorrelation function over a ±3º interval for a fixed frequency (¶3.5.3) for both the single target case, and the 10-target averaged case; and
  + a plot of the magnitude of the RCS autocorrelation function over a ±30 MHz interval for a fixed aspect angle (¶3.6) for both the single target case, and the 10-target averaged case; and
  + an explicit statement of the decorrelation criterion you used, your resulting measurement of the decorrelation intervals in angle and frequency, and a specific statement of how well it agrees (or doesn’t) with the theoretical prediction.
* All code should be kept together at the end.

Your report must be typed, not handwritten, and should have figures included in the report, not in a separate document. On-campus students should submit a hard copy. If you will be absent from class on the day the report is due, turn it in in advance of the due date and time, either at my office or in my mailbox in the ECE mail room. If you deliver an advance hard copy, you need to get another ECE faculty or staff member to initial it and record the date and time so I have a way to know when it was submitted. If you cannot deliver a hard copy in advance, e-mail me your report by the due date and time.

Distance learning students should e-mail an electronic copy to me. PDF format is preferred, but I can also accept Microsoft Word. Do not send TeX or postscript documents. All text, figures, and code should be in a single PDF (or Word) file.

If you anticipate a problem in submitting your work on time, contact me before it is due, not when it is due or afterwards.

# SIMULATION REQUIREMENTS

## Simulation of a 50-Scatterer Target Response

For your first step, you must perform an RCS computation simulation similar to the example of Figures 2-8 and 2-9 in the textbook. This should include the following steps:

1. Generate a set of 50 point scatterer (*x,y*) coordinate pairs. The scatterers are to be randomly distributed in a box that is ±5 meters in the *x* dimension (along the initial radar boresight) and ±2.5 meters in the *y* dimension, centered at (*x*,*y*) = (0,0). By “randomly distributed” I mean that the *x* coordinate should be drawn from a suitable uniform random PDF, and the *y* coordinate from an independent uniform random PDF. In MATLAB, the function rand will be helpful. Once you have defined a set of target scatterers, they remain fixed; you do not generate new scatterer distributions as the radar moves.
2. Assume that the RCS *i* of each individual scatterer is the same, namely *i* = 1 m2. Assume a monostatic (transmitter and receiver share the same antenna) radar with a transmit frequency of 10 GHz (X band), located initially at coordinates (*x* = +10 km, *y* = 0 km). You will compute the relative RCS of the composite echo from all 50 scatterers as the radar moves around the target at constant range of 10 km from the center of the target box. Define the aspect angle between the radar and the target as the angle between the +*x* axis and the line connecting the target box center to the radar, with positive angle corresponding to counter-clockwise rotation. For example, the radar is at an aspect angle of zero degrees when at its initial position above, and is at +90 degrees when the radar coordinates are (*x* = 0 km, *y* = +10 km).
3. Compute the maximum aspect angle change required to decorrelate the RCS regardless of look direction using Eqn. (2.63). We would expect RCS values computed at angles differing by this amount or more to be approximately independent of one another.
4. Use Eqn. (2.51) to compute the composite RCS as a function of aspect angle. Compute this every ½° in aspect angle. Is this angle step size consistent with the decorrelation angle computed in the previous step? Plot the resulting RCS *vs.* aspect angle on a decibel scale, similar to Figure 2-9. Be careful as to whether you use the magnitude or magnitude-squared and 10log10(·) or 20log10(·) to convert to decibels. Compute the mean of the linear scale RCS data. How is this related to the number of scatterers and their individual RCS values?

## Probabilistic Model

Once you have a series of RCS values *vs.* aspect angle, the plot should show random-looking behavior similar to Figure 2-9. As described in class, the “many scatterers” assumption suggests that if we model the RCS as a random variable, then the exponential PDF is a reasonable assumption. In this portion of the problem, we will visually test this assumption through the following steps:

1. Form a histogram of the RCS data. In MATLAB, the function hist is convenient for this. I suggest 100 angle bins in the histogram. You will need to normalize the histogram so that its area, when plotted, matches the theoretical PDF; that is, the area has to equal 1. One way to appropriately normalize and plot a histogram in MATLAB uses the following code (the linear scale RCS data is in the vector variable RCS):

[count,centers]=hist(RCS,100);

bin\_size = centers(2)-centers(1);

area = sum(count)\*bin\_size;

density = count/area;

bar(centers,density,1);

1. Plot the exponential PDF of Eqn. (2.52) in the text as an overlay on the histogram and compare to the histogram. The expression for this PDF is



Note that you will need the mean of the RCS data, , as a parameter in the exponential PDF.  is the mean of the RCS data on a linear, not decibel, scale! If you correctly normalized the histogram, it should be unnecessary to further normalize the PDF; you should be able to use the equation above “as is”.

In MATLAB, you can cause the plot of the exponential PDF to appear on the same plot as your histogram by using the hold command. For example, one could overlay the exponential formula above using code similar to this:

p = exp(-centers/sigmean)/sigmean;

hold on;

plot(centers,p);

hold off;

If all goes well, you should obtain a plot similar to Figure 2-10 of the textbook. (Note: I used a different normalization in Fig. 2.10 in the text, so the vertical scale you get won’t match exactly. The normalization we are using here is better.)

Comment on the match between the histogram and the theoretical PDF. Is it perfect? If not, describe briefly the differences.

## Expected Value Over “Many” Random Targets

Because the scatterer positions are random, the histogram is itself a random function. The expected value of the histogram function is what should actually match the theoretical PDF. Thus, if we were to repeat the 50-scatterer trial many times with different sets of 50 random scatterers and then average the histograms, we would expect a better result.

Once you are able to generate good data and the corresponding histogram for a single random target, modify your code to do this for 10 different random targets, all of the same size and number of scatterers, but with a different set of random scatterer locations in each target. The code should generate the data and then the histogram for each individual target, exactly as above, but should then average the ten histograms to estimate the expected value of the RCS histogram. In terms of the code suggestions above, generate a histogram for each random target. In order to average these correctly, they must all be computed using the same set of RCS bins. This can be done with code something like this:

(generate data for target #1 in variable RCS1 ….)

[count,centers]=hist(RCS1,100);

bin\_size = centers(2)-centers(1);

area = sum(count)\*bin\_size;

density1 = count/area;

(generate data for target #2 in variable RCS2 ….)

[count,centers]=hist(RCS2,centers);

area = sum(count)\*bin\_size;

density2 = count/area;

(repeat process for RCS3 through RCS10)

density = (density1 + density2 + ... + density 10)/10;

Note the difference in the hist call for the 2nd through 10th histograms as compared to the first call. Once the average density is computed, plot it with an overlaid exponential PDF. (Use the mean of ALL the data from all 10 targets to scale the PDF.) Does this averaged histogram match the theoretical PDF better than the histogram of the single target done earlier?

The code to do all of this in MATLAB may be simpler if you define RCS to be a matrix with one column for the data from each target. In this case count (and therefore area and density) will be matrices also, with one column for each target. However, any coding technique that works is fine. The sum function can then be used to add the density variables as part of the averaging.

## Variation with Frequency

Repeat the steps in sections and 3.3 for 10 50-scatterer targets (you can use the same targets or a new set of targets). This time, fix the aspect angle of the radar at zero degrees, and vary the frequency of the radar. Specifically, vary the frequency in 15 MHz steps starting from 10 GHz and going to 14.5 GHz; this is a total of 300 frequencies. (Note that the 15 MHz frequency step is the frequency decorrelation interval predicted by Eqn. (2.64), adapted to the 2D case as described in paragraph 3.6 of this assignment). Form the normalized histogram, averaged over 10 random targets, and overlay a plot of the exponential PDF. Also plot the histogram and PDF for a single random target. Describe how well the histogram matches the theoretical PDF for the single- and 10-target cases.

## Decorrelation in Angle

In the textbook, we derived the deterministic autocorrelation function of the complex echo signal voltage (not the RCS) from a line array target, and used this to estimate decorrelation intervals in angle and frequency. For decorrelation in angle, we obtained the estimate (Eqn. 2.63)



Even though this was derived for a simple line array target, we can try to apply it to our more complex target, interpreting *L* as the width of the target normal to the line of sight from the radar. In the simulations in this exercise, we have *L* = 5 m (assuming we use data concentrated around an aspect angle of zero degrees) and *F* = 10 GHz = 1010 Hz. The result is ** = 3 mrad = 0.172°. In this section of the problem, we will compute the central portion of the autocorrelation of our simulated complex echo data and determine whether it is consistent with the formula’s estimate of decorrelation angle.

1. Because we want to estimate a decorrelation interval that is expected to be a little less than 0.2°, we need complex echo data at angle increments significantly less than this, say about 0.01 or 0.02°. On the other hand, we don’t require echo data for a full circle; a few degrees around ** = 0° will be adequate. Modify your 10-target simulation to compute the complex echo at an increment of no more than 0.02° (finer is better) over a range of ±3°.
2. Compute the autocorrelation function of the complex echo data for one random target. In MATLAB, the xcorr function will be helpful. Note that, unlike the line array case in the textbook, your target does not have spatial symmetry and the resulting autocorrelation function will not, in general, be real valued.
3. Plot and examine the central portion of the magnitude of the complex autocorrelation function and estimate the amount of angle change that decorrelates the complex echo data. The definition of “decorrelating” is subject to interpretation. Reasonable criteria include the first zero or near-zero in the magnitude, or the point at which the function falls to 1/*e* (about 37%) of its peak value, or the first local minimum, *etc.*[[2]](#footnote-2) State your criterion. How well does the observed decorrelation interval agree with the estimate in the equation above?
4. Now extend this to compute the autocorrelation functions for 10 random targets. Average the complex autocorrelation functions and plot the magnitude of the result, and compare the decorrelation estimate that results with the theoretical value. Do you get better agreement?

## Decorrelation in Frequency

Repeat the steps in ¶3.5 for a fixed aspect angle of zero degrees and a varying frequency to get the RCS correlation function in frequency. The predicted decorrelation interval in frequency, from Eqn. 2.64, is



where *L*sin** was the length of the line array target projected along the radar LOS. Adapting this to our 2D target, we can use



where *LLOS* is the length of target box along the radar line of sight. At 0° aspect angle, this is 10 m and the result is *F*= 15 MHz. Evaluate the magnitude of the autocorrelation function for *F* = 10 GHz ± 30 MHz using frequency steps of 1 MHz (less if the run time is not too slow). Plot and evaluate the results for a single target and for the average of 10 targets in a manner similar to ¶3.5.3.

# ADDITIONAL DATA AND PROCESSING ISSUES

## MATLAB Coding Style

If you are using MATLAB, and are new to it, you should know that it is can be helpful to write your code to take advantage of the matrix-vector approach of MATLAB, and to avoid for loops if you want to enjoy reasonable run times. You should also take advantage of MATLAB’s assumption that variables are complex. As an example, suppose I have a scalar constant alpha, a vector x, and another vector y (where x and y are the same length). I can multiply *each* element of x by alpha with the simple command

z = alpha\*x;

I can compute the complex exponential  for each element of x, placing the result in a new vector z, with the single command

z = exp(j\*alpha\*x);

Note that MATLAB predefines j as the square root of –1 unless you override it (ditto for i). Finally, you can perform element-by-element operations using the so-called “dot” notation of MATLAB. To form a new vector in which each element is the product of the corresponding elements of x and y, write

z = x.\*y;

*Note that no* for *or* while *loops are needed in any of these cases, nor was it necessary to declare anything* REAL *or* COMPLEX. MATLAB is very efficient in executing these vector statements; it is much less efficient in carrying out the same computations using a standard C or FORTRAN-like for loop approach. One cannot always avoid using loops, nor should you go to extraordinary lengths to do so; but you should look for opportunities to use the matrix-vector notation of MATLAB.

While you might or might not find it useful in this project, it is worthwhile to mention MATLAB’s handling of transpose here. For a vector x, the operation

x’

gives the conjugate transpose (also called Hermitian transpose), not just the simple transpose. That is, the vector is not only transposed, its elements are also conjugated at the same time. The operation

x.’

gives just the transpose without conjugating the elements. The MATLAB functions conj and transpose can also be used. Unexpected complex conjugations due to the transpose operator is one of the more common bugs in MATLAB code using complex data.

You may find it convenient to have more than one plot available at a time; that is, to not have each plot overwrite the previous one. MATLAB supports multiple plot windows. If you wish to have a plot occupy a different window than the previous plot, use the figure command. For example, suppose I have vectors x, y, and z; and I want to plot y vs. x in one window, and z vs. x in another. The following commands will do this:

figure(1)

plot(x,y)

figure(2)

plot(x,z)

or just

figure

plot(x,y)

figure

plot(x,z)

Plots can be copied and pasted from the MATLAB figure window into other programs. Note, however, that to copy the plot, you need to use the Copy Figure pick list item on the Edit menu of the figure window; CTRL-C does not work.

The class T-Square site has a link to MATLAB help resources in the “Resources” tab; this may be useful to new MATLAB users.

## Range Equation Considerations

The radar range equation is used to estimate the amplitude of a received echo, given the transmitted echo amplitude, target range and RCS, and several system parameters. All of these parameters are the same for each target in our simulation except for the individual ranges. Ideally we would weight each of the individual echo amplitudes by *R*-4. However, if the nominal radar range is large compared to the target size, the variation in individual ranges, and thus in the *R*4 factor, is negligible. In this exercise, we have a target maximum dimension of 10 m, and a nominal range of 10 km. Thus the maximum variation in the *R*4 term is about (10,010/10,000)4 = 1.004, which is not significant for our purposes. We can therefore ignore the differences in echo amplitude due to range variations. For higher precision work, we should include the *R*-4 term.

However, the range differences *are* significant in terms of wavelengths; a range change of 10 m at 10 GHz (** = 3 cm) is 333 wavelengths. This is why we *do* take the individual ranges into account in computing the phase of the individual echoes.

## Single and Multiple Bounce

All of the energy impinging on a scatterer is not reflected back toward the radar; most of it scatters in other directions. Some can scatter forward or to the side, hit other scatterers, and then reflect back toward the radar. This is called “multiple bounce” scattering. The direct reflection of energy without bouncing off of additional scatterers is called “single bounce” scattering. The technique used in this exercise accounts only for single bounce scattering. Proper accounting for multiple bounce scattering would require more detailed modeling of the scattering characteristics of the individual scatterers. In most cases, significant energy is lost on each bounce, so that the single bounce echoes are most significant.

## RCS GUI

You can download a set of MATLAB tutorial resources, along with a powerpoint presentation giving some info on how to use each one, from the textbook web site [www.radarsp.com](http://www.radarsp.com). One of these is a GUI-based demo called RCS which implements the single-target version of much of what you have done in this project. You can use it to experiment with the many-scatterers-plus-one-dominant model and the Rician and chi-square fits to that data.

1. Office: Klaus 3354, 404-894-2714, <mark.richards@ece.gatech.edu>. Office hours TBD, but drop-ins and appointments welcome. [↑](#footnote-ref-1)
2. A cynical person might observe that this ambiguity about defining decorrelation amounts to a “fudge factor” for making things work out well. [↑](#footnote-ref-2)