

A MODEL FOR THE PERFORMANCE OF SURFACE-SCANNING INSPECTION  
SYSTEMS

by

Benjamin Duaine Buckner

A Thesis Presented in Partial Fulfillment  
of the Requirements for the Degree  
Master of Science

ARIZONA STATE UNIVERSITY

May 1996

A MODEL FOR THE PERFORMANCE OF SURFACE-SCANNING INSPECTION  
SYSTEMS

by

Benjamin Duaine Buckner

has been approved

January 1996

APPROVED:

\_\_\_\_\_, Chair

\_\_\_\_\_

\_\_\_\_\_  
Supervisory Committee

ACCEPTED:

\_\_\_\_\_  
Department Chair

\_\_\_\_\_  
Dean, Graduate College

## ABSTRACT

A model for surface-scanning inspection systems, concentrating on particle detection and the particular features of the Arizona State University instrument, is developed from an analysis of the operation of such systems. Software implementing some of its features was written and tested and results for different instrument configurations and particle signals compared. The model accounts for the statistical properties of particle signal randomness, optical noise, electrooptical quantization noise, and electrical noise using a probabilistic analysis of the relevant physical phenomena. Its results are presented in terms of detection and false-count probability, receiver operating characteristics, and lower detection limits, illustrating the relations between several important parameters and instrument performance.

To my parents.

## ACKNOWLEDGEMENTS

I would of course like to thank Dr. Hirleman for giving me the opportunity to complete this thesis research and for keeping it moving in the right direction. I also appreciate the efforts of my Laser Diagnostics Laboratory colleagues Brent Nebeker and Greg Starr in helping to get the data for many of the simulations together.

Also, I would like to extend thanks to Dr. Stan Stokowski of Tencor Instruments and Randy Goodall of SEMATECH for some informative conversations and suggestions about which instrument features were of particular interest.

Finally, I am grateful to SEMATECH and its Particle Counting and Microroughness Task Force which provided the funding for this research.

## TABLE OF CONTENTS

	Page
LIST OF TABLES .....	viii
LIST OF FIGURES.....	ix
1. INTRODUCTION .....	1
2. MODELING STRATEGIES .....	3
3. COMPUTING THE STATISTICS.....	7
A. Physical Model.....	7
B. Detection Criteria.....	12
C. Finding The Probabilities.....	14
1. Single Detector, Single Sample Criteria.....	14
2. Multiple Detector Criteria.....	17
3. Multiple Sample Criteria .....	17
D. Particle-Scatter Signal Statistics .....	18
E. Particle-Scatter Signals with Multiple-Sample Criteria .....	25
F. Optical Noise .....	31
G. Independent Optical Noise Sums.....	35
H. Dependant Variable Sums.....	36
I. Electrical Noise Sources.....	38
	Page
4. RESULTS AND ANALYSIS OF PERFORMANCE .....	40
A. Receiver Operating Characteristic Curves .....	40

B. Results.....	44
C. Conclusion .....	48
REFERENCES.....	51
APPENDIX I: MODEL CODE .....	54
APPENDIX II: INTEGRATION OVER THE SCATTERING SPHERE.....	76
APPENDIX III: CALCULATING $P_D$ FOR COINCIDENCE OF ADJACENT SAMPLES.....	80

## LIST OF TABLES

	Page
Table 1. Symbols Used.....	57

## LIST OF FIGURES

	Page
Figure 1. ASU Surface Scanning Inspection System Schematic.....	8
Figure 2. Obtaining $P_M$ and $P_F$ from H0 and H1 PDFs. ....	15
Figure 3. Obtaining $P_N$ and $P_D$ from H0 and H1 PDFs. ....	15
Figure 4. Example particle signal distributions for the one-dimensional (solid) and two dimensional (dashed) cases. ....	24
Figure 5. Analytical framework for multiple-scan detection with two-dimensional randomness. ....	26
Figure 6. Multiple samples of a single particle. $I_p$ is the irradiance striking the particle in each case. ....	27
Figure 7. Oblique beams scatter Gaussian at smaller widths because they subtend a greater number of effective scatterers. ....	33
Figure 8. Spatial noise irradiance distribution according to region on the beam width vs. detector aperture plain. ....	34
Figure 9. Left half of ROC plot for multiple scan criterion specifying detection when at least one illuminated scan region produces a signal over threshold. ....	42
	Page
Figure 10. ROCs for same case as in Figure 8 but using a criterion that registers detection only if at least one pair of adjacent scans produced threshold-crossing signals on both scans in the pair. ....	43
Figure 11. Analytically calculated ROC curves for various scan pitches. ....	45

Figure 12. Monte-Carlo (dashed) and analytically calculated (solid) ROCs for a multiple scan case using a criterion of at least one threshold crossing to indicate detection.....	45
Figure 13. ROC plots for total integrated particle scattering cross-sections of PSL spheres of various diameters.....	46
Figure 14. Lower detection limits for three different $P_D$ - $P_F$ criteria with respect to RMS surface roughness at various particle sizes. ....	47

## 1. INTRODUCTION

Reduction of the dimensions of integrated circuits has proceeded at a frantic pace for many years and shows few signs of stopping in the near future. With this reduction, the sensitivity of IC manufacturing processes to particle contamination has correspondingly increased, as small particles that may have once been mere specks on the surface structures of several years ago are boulder-like obstructions at the scale of today's diminutive devices. It is generally reckoned that a particle of  $1/5$  to  $1/3$  of the minimum line width can constitute a "killer" defect, and with the production line widths beginning to enter the  $.35\ \mu\text{m}$  realm, a particle of a diameter as little as  $70\ \text{nm}$ , far below the resolution limit of any optical microscope, can be of interest to manufacturers.

Since the minimum particle size of interest is shrinking, the instruments used to detect these particles before expensive processing is wasted on a contaminated surface must become correspondingly more sensitive. One of the more popular types of such instruments are laser-scattering surface-scanning inspection systems (SSISs), which scan wafer surfaces with a tightly focused laser beam and analyze the scattered light for telltale signs of particle contamination. With smaller particles, this analysis becomes increasingly difficult, with the particle-scattering cross-section rapidly receding into the instrument noise with decreasing particle size. Thus, these instruments must be operated in regimes where the particle signal is difficult to distinguish from noise.

The chief consequence of noise in this context is the generation of false counts by the instrument. As the instruments are made more sensitive in order to detect smaller particles, they are also more likely to misinterpret noise peaks as resulting from contaminants, and this misinterpretation can lead to the mistaken rejection of a wafer as contaminated. This is

a particularly expensive mistake if the wafer has already been subject to extensive processing.

Thus the mistakes of either missing contaminants and not rejecting a wafer or being fooled into rejecting a wafer on account of noise can be costly and both can easily happen in today's sensitive IC processes. Therefore, the goal of this project is to create a mathematical model for the physical and statistical properties of laser-scattering SSISs in order to predict the performance of these instruments in detecting contaminant particles. The particular figures of most interest are the probability of detection given the presence of a particle,  $P_D$ , and the probability of false count given the absence of a particle,  $P_F$ , for some particular instrument configuration and operating regime. Knowledge of these figures also allows characterization of the lower size detection limit for various types of particles for an instrument, which is of great interest in gauging its capacity for detecting the ever-smaller killer particle.

## **2. MODELING STRATEGIES**

There are a number of possible approaches to this model, many of which have been used in the past for similar tasks. In general, these fall into two large categories, which will here be termed “simulation” models and “analytical” models (Baghzouz and Tan drew a similar though less general dichotomy<sup>1</sup>).

Simulation models simulate possible outcomes for the system outputs to be tested in proportion to the likelihoods of these outcomes, which is usually achieved through the use of pseudorandom number generators with appropriate distributions. These outcomes are then tested with the detection criteria and binned accordingly. By counting the number of outcomes in each bin, the probability of each outcome is determined to some accuracy. In some types of systems it is possible to do this through a complete enumeration of possible outcomes or an enumeration to a certain resolution, but this is not usually practical in a SSIS apparatus.

The simulation method most appropriate to such surface scanners is the well known Monte Carlo method. Random numbers are generated representing the various random quantities affecting the system and these are combined according to the physical model and the detection criteria are then applied. We simply then count the outcomes. There are a great number of subtle variations to this scheme designed to improve its performance and speed. It has several inherent drawbacks, one being that it has no guarantee of convergence due to its random nature. Additionally, its accuracy tends to be rather poor. To its advantage, it has a certain ease of implementation, particularly in that it requires no lengthy statistical analysis of the detection criteria (at least in its simplest forms), needing only a software implementation of the detection algorithm to make it work.

Wheeler and Ikeuchi used a simulation-based sensor modeling process in their work on object recognition. They simulate recognition processes for a large number of possible images in order to characterize the distribution of possible outcomes<sup>2</sup>. Their work also has potential relevance to the more complicated problem of defect characterization by scattering.

The analytical methods are more varied. Their general approach is to use the probability distributions of the stochastic inputs to the system to determine probability distributions associated with the various outcomes. There is a number of such approaches, and we begin by discussing a few common approximation methods.

One of the easiest analytical methods to use is the Gaussian approximation, which is often implicit in casual analyses of stochastic systems. This is merely the assumption that all random variables involved have a Gaussian (or normal) probability density function (PDF) as specified by their individual means and variances. This is essentially what the common noise figure, the signal-to-noise ratio (SNR), specifies, being a ratio of the standard deviation to the mean.

Noise characteristics of systems are frequently estimated using SNRs, combining the standard deviations as the familiar square root of the sum of squares and adding the means as simple sums. Though it is not always stated, this obviously requires the assumption that the distributions follow these particular summation rules. The Gaussian, being the most common example of such a distribution, seems to be the usual presumed model. This assumption often works fairly well, particularly where a large number of independent random variables are involved (by the central limit theorem), but it has a tendency to perform poorly in the tail regions of the distribution<sup>3</sup>. Unfortunately the upper tail is the

region that usually contributes most to false count probability since most instruments are designed and operated to minimize this, as will become clear later.

The advantage of the Gaussian approximation lies chiefly in the combining properties of the Gaussian distribution, as mentioned above, as well as the in fact that it saves having to specify the sometimes elusive nature of the distributions of the relevant random variables. It should be mentioned, though, that those combining properties only apply to statistically independent random variables, so that the suitability of the approximation becomes highly questionable when statistically dependent variables are involved.

Another approximation method that has been used is known as the Chernoff-bounds method. It uses moment generating functions for distributions of the signal to determine upper and lower bounds on probabilities such as  $P_F$  and  $P_D$ <sup>4</sup>. It could be fairly useful in cases where the moment-generating functions (MGFs) could be calculated more easily than the PDFs, but, in general, obtaining the MGFs is just as difficult as obtaining PDFs, so it is hard to see many good reasons for not simply integrating the PDFs to obtain an actual value as opposed to a bound.

Having examined a few common analytical approximation methods, we naturally come to the method which is theoretically exact, to which they strive to approach. This is to use the real PDFs for the input variables and go through the appropriate manipulations to obtain the desired probabilities. Of course, this method is always limited to a certain accuracy when computed numerically and its performance depends on how well the input parameters are really known. The drawback to it is of course that it can sometimes be extremely hard to analyze and can require a great deal of computer time, but if it is feasible to do it should produce very good answers numerically and can even yield closed-form expressions in

certain special cases. It is this difficult but exact method that has been chosen for the main thrust of this research in an effort to give it a certain generality and avoid vagaries of approximation methods which can be easily misapplied and are rather unlikely to give usable answers in many cases of interest. The Monte Carlo method is also used here to some extent to provide independent confirmation of these analytical results.

### **3. COMPUTING THE STATISTICS**

As explained, the chosen method of computing the probabilities of interest attempts to derive theoretically exact expressions and implement them numerically, although the software implementation will neglect less-dominant noise sources which will have little comparative influence on the statistics. The first steps of this process tend to apply to most instrument configurations, but we must first identify the various random processes involved and consider how they interact physically in order to model them. In many cases, it is this process which proves to be the biggest obstacle to successful probabilistic modeling, and heuristic approaches to characterizing the distributions due to random processes have recently been explored by Woodbury, *et al.*<sup>5</sup> The current work takes a more analytical approach to the problem, though the Minimum Relative Entropy method from their work has considerable possibilities for the verification and fine-tuning of this instrument model.

#### **A. Physical Model**

The typical wafer scanner, the ASU instrument included (Figure 1), collects light and directs it to a photo-electric transducer element or detector, usually a photodiode or photomultiplier tube (PMT). Essentially, this integrates optical irradiance over a given solid angle to give a certain input power at the detector or set of detectors. This input power is randomized from many sources and it provides a key point at which the analysis should start, being a sort of boundary between the optical and electrical signals.

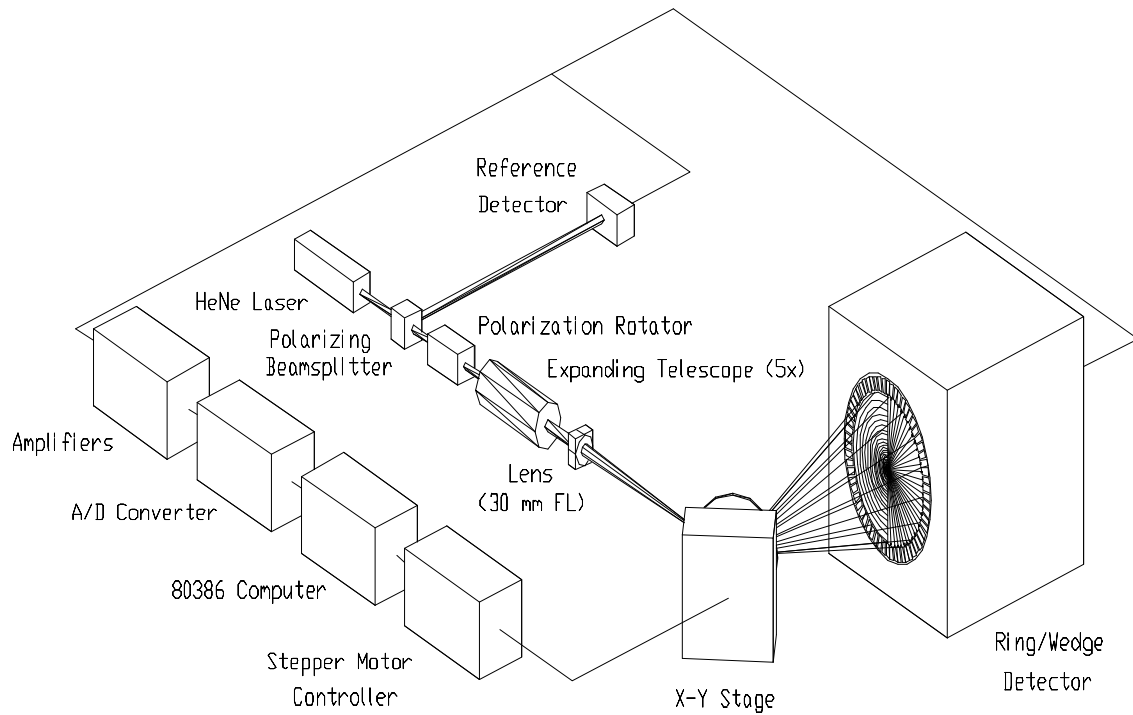


Figure 1. ASU Surface Scanning Inspection System Schematic.

In some instruments, the receiving optics can collect scattered light with non-uniform efficiency as a function of angular position on the scattering sphere, but this consideration is not likely to be significant and is not particularly difficult to deal with through the inclusion of the usual pupil function used frequently in the analysis of optical processing systems.

The light collected can come from several distinct sources of greater and lesser importance:

- Particle or defect scatter
- Surface scatter
- Specular reflection of the illuminating beam

- Medium scattering (by air molecules or suspended particles)
- Ambient light

In particle detection, the first source is the signal of interest. All others are part of the noise that degrades the performance of the instrument. The particle scatter and surface scatter are the most important and the ones that tend to be random within the scope of the measurements. Specular reflection may contribute somewhat to the signal, but it has very little variation, and most systems are designed to eliminate it as much as possible. Ambient light and medium scattering are both usually of less influence and not appreciably random, although scattering by dust particles may involve particle numbers small enough that the variance could become non-negligible.

The powers of the first four listed sources are directly proportional to the power of the illuminating beam and consequently may have some additional temporal noise due to the power fluctuations of the laser.

Thus for most purposes, the optical power to the detector can be viewed as the sum of two random powers,  $P_S$ , the signal from particle scatter, and  $P_N$ , the noise, along with a constant power from other sources,  $p_c$ , or

$$P_{\text{opt}} = P_N + P_S + p_c \quad \text{Eqn. 1}$$

This last power is merely an offset to the mean of the total power and does not represent a stochastic quantity.

The mean values of most of these quantities may be calculated to a certain degree of accuracy using published theoretical techniques or even be measured if a suitable apparatus is available. The calculation of particle scatter,  $P_s$ , is the most complicated and generally requires a detailed numerical model to obtain results of useful accuracy.

A number of such particle scattering models exist and the topic has been the focus of major efforts at ASU<sup>6,7</sup>, as well as efforts by many other groups.

$P_N$ , which comprises surface scatter noise primarily, can be more or less complicated to compute, depending on the nature of the surface. For random rough surfaces, the problem has been treated extensively in terms of getting surface scatter as a deterministic quantity and in terms of the statistics of random surface scatter, but there are few treatments which combine the two types of analyses. Statistical treatments generally assume that the deterministically computed quantities correspond to mean values, and more physically oriented treatments seldom discuss the statistical nature of surface scatter. Beckmann and Spizzichino's classic treatment<sup>8</sup> is one notable exception which covers both, though it predates many important recent developments in both areas. In the present work, the approach of treating the results of deterministic theories as mean values is used. Since the current models are intended as more of a generic treatment of systems with a wide range of possible operating parameters, mean values of a reasonable order of magnitude are considered sufficient.

The main distinctions among classes of surface scattering models are between the scalar scattering models and the vector scattering models, distinguished by whether polarization is treated (the vector models), and between the two major height scale regimes for the surface roughness, with the wavelength of the illumination being the point of demarcation between

them. There is also a distinction between total integrated scatter (TIS) models and angle resolved scatter (ARS) models, but TIS is really just ARS integrated, so there is not much point in discussing the TIS case separately.

Using Stover's dichotomy based on mathematical approaches<sup>9</sup>, the more commonly encountered model in SSIS contexts is the Rayleigh-Rice vector perturbation theory. It is only valid for surface height deviations much less than a wavelength and for surface variation slopes less than 1. The other model, the Kirchoff diffraction theory, has the advantage of being easier to formulate mathematically and of holding valid for much larger height variations than Rayleigh-Rice. It occurs in both scalar and vector formulations, though the scalar formulation is not very good in predicting ARS<sup>10</sup>. However, the Rayleigh-Rice method tends to be more exact than Kirchoff within its range of validity<sup>11</sup>.

The constant power,  $p_c$ , is probably best determined empirically, since it is largely due to stray reflections and optical "leaks" in the instrument. This term can also include detector dark current in the guise of noise-equivalent power (NEP). Its medium scattering component from the gas in the instrument should not be too difficult to compute, but this is not likely to be the dominant source. For low scattering angles (near the specular beam), it has been noted that scattering from air-suspended particles can be somewhat troublesome,<sup>12</sup> and presumably Rayleigh scattering from the gas molecules themselves would appear in this case, if in any. The molecular Rayleigh scattering component is of indeterminate significance in general, though it could be expected to rapidly increase with decreasing wavelength of the illuminating beam.

Having now derived an expression for the optical power reaching the detector, we consider the conversion of the optical signal to an electrical signal. This is expressed as a function

$S(P_T)$ , where  $S$  has been chosen to represent the output signal which could be voltage or current. This function can usually be considered to have a fairly simple linear form within the operating regime of the detector, or at least its mean can be. Statistically, it subsumes a potentially very significant noise process due to the quantum nature of photons and the carriers they produce in the detector. This is the familiar though mathematically rather complicated shot noise. This will be dealt with in some detail later.

Once the signal has become electrical, it is usually amplified by associated electronics, which often introduces still more noise sources. Certain types of detectors have a built-in carrier-avalanche amplification process. These avalanche processes cause a deviation from classical shot noise statistics in the detector. There are also various additive noise sources, which increase the mean of the electrical signal such as stray electromagnetic signals and the detector dark current. One of the most important electrical noise sources, thermal or Johnson noise, adds to the variance but not the mean. Thus, we see a final output of a general form

$$S_o = S(P_N + P_S + p_c) + S_N, \quad \text{Eqn. 2}$$

where  $S_N$  subsumes the various post-detection additive electrical sources.

## **B. Detection Criteria**

Unfortunately, this simple expression given above belies the complexity of the statistics underlying its components and the statistics of their combinations. Computing their effect on detection statistics requires a look at exactly what criteria are used to determine whether a given signal or set of signals is interpreted as a detection or not.

The first and simplest criterion is the single-detector, single-scan threshold criterion. If the magnitude of one signal exceeds a threshold, that is registered as a detection. Not surprisingly, computing the statistics for this case is not excessively difficult.

A second group of criteria involves multiple simultaneous detectors positioned at different points around the scattering sphere.

There are numerous ways these could be combined, but the present project restricts itself to two methods, namely the threshold for a sum of signals (really very similar to the simple case) and multiple detector coincidence, in which a detection is registered when each detector exceeds a threshold. This second group is also relatively easy to analyze, since the sum case is mostly equivalent to the first type with a larger area of integration. Its second case requires only the multiplication of the results from the first type, to a good approximation.

The third and most complex group of criteria considered here involves multiple scans or multiple samples. In these schemes, a detection is registered when some pre-detection criterion is met in some combination of multiple scan/sample instances, normally with the scans or samples of two or more neighboring regions being considered. Here we will consider only simple threshold tests for the pre-detection criteria, but, even for this case, the analytical computation of the statistics requires a radically different approach from that which suffices for the first two types.

It is hoped that an analysis of these three types of cases will provide an outline for analyzing the statistics for the many possible types of binary detection schemes not covered, though these represent a good coverage of those currently in use.

Having sketched out the broad physical picture of how the signal gets from the wafer to the detection/non-detection result, we consider details of these computations and noise sources, beginning with an overview of how the statistics for each case are calculated.

### **C. Finding The Probabilities**

#### *1. Single Detector, Single Sample Criteria*

For the first detection criterion, the simple threshold test of a single detector,  $P_D$  is the probability that, when a particle is present, the output signal exceeds a certain threshold.  $P_F$  is the probability that the output signal does so when no particle is present, simply through noise fluctuations. The complements of these probabilities are sometimes specified, namely  $P_M=1-P_D$ , the miss probability, and  $P_N=1-P_F$ , the no-count probability.  $P_M$  and  $P_F$  are the error probabilities for the particle-present and no-particle conditions. Clearly, the probability density function (PDF) for the particle-present signal and PDF for the non-particle signal will tell us these probabilities if we simply integrate them above the threshold value (Figures 2 and 3).

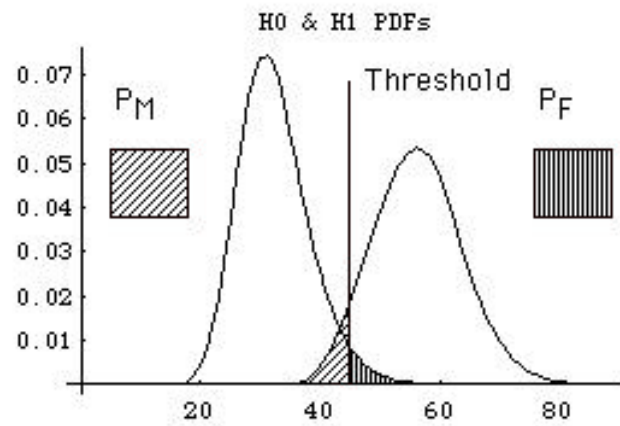


Figure 2. Obtaining  $P_M$  and  $P_F$  from  $H_0$  and  $H_1$  PDFs.

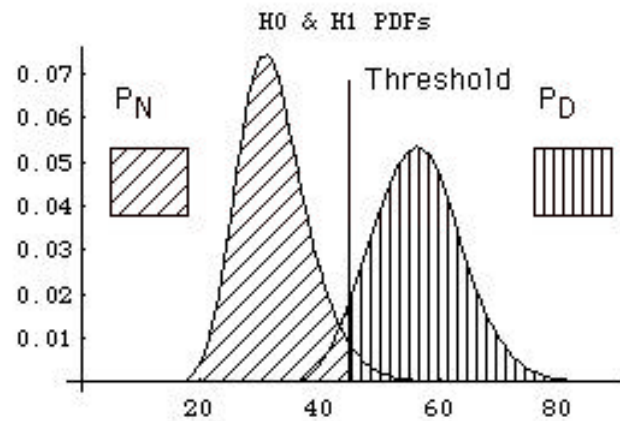


Figure 3. Obtaining  $P_N$  and  $P_D$  from  $H_0$  and  $H_1$  PDFs.

In detection statistics, this is referred to as the “binary detection problem.” It is expressed in terms of hypothesis testing, with the so-called “experimental” hypothesis ( $H_1$ ) being that

there is a particle under the beam; the “null” hypothesis (H0), the complement, is that there is no particle under the beam. So we write the respective conditional signal PDFs as  $f_s(s|H1)$  and  $f_s(s|H0)$  and can write the expressions for the probabilities as

$$P_F = \int_{thresh}^{\infty} f_s(s|H0)ds \quad Eqn. 3$$

$$P_D = \int_{thresh}^{\infty} f_s(s|H1)ds \quad Eqn. 4$$

In theory, these hypotheses are best tested using the likelihood ratio test (LRT), which takes the received signal and compares the PDFs for the H1 and H0 cases. If the probability density for H1 is higher for that value of the signal, the hypothesis is accepted. If H0 PDF is higher at that point, it is rejected (meaning we take it that there is no particle). The ratio of the PDFs is aptly called the likelihood ratio,  $\Lambda(s)$ , and the above test is equivalent to testing whether this ratio is greater or less than some value. A little consideration will confirm the well-known result that this test is the optimum test. Obviously, if we pick the most likely hypothesis, we will optimize our chances of being right. The LRT can be equivalent to our simple threshold test with the appropriate threshold provided that the likelihood ratio is monotonic. This always allows us to separate the domain into two continuous regions where one or the other hypothesis should be chosen. Otherwise we would have several separated regions in which we should choose a certain hypothesis, a condition which cannot be met by a single simple threshold test. Fortunately, in this type of problem the LRT is normally monotonic throughout the range of threshold values of practical interest, so threshold tests of the signal normally correspond to the LRT.

Thus we need the output PDFs for two cases, with a particle and without a particle. These are calculated using the theorem for determining the PDFs of sums of independent random variables.

### *2. Multiple Detector Criteria*

For the second type, multiple detector criteria, we compute PDFs for each detector as above. For the first sub-case of summed detectors, we convolve the with-particle PDFs into a total with-particle PDF and then convolve the without-particle PDFs into a total without-particle PDF. Then the upper-tail integration from the first type yields the desired  $P_D$  and  $P_F$ . The second sub-case where a detection is registered if all detectors pass the threshold simply requires the multiplication of the  $P_{D,s}$ s, and the total  $P_F$  is similarly just the product of  $P_{F,s}$ s.

### *3. Multiple Sample Criteria*

Multiple sample criteria, the third type, usually require a much different approach because the particle signals between scans or samples are not statistically independent. This is due to the fact that the same particle may produce scattering for several different scans or samples, and, since the particle presumably does not move between these, there will clearly be some sort of dependence between the signals. This complicates many types of multiple-sample detection schemes because joint probabilities of various outcomes of separate scans are no longer simply multiplicative. For example, in the commonly used multiple scan coincidence case, if the probabilities were independent we could simply multiply two  $P_{D,s}$  together to get the probability of detection on both of the corresponding samples. This multiplicative property does not hold true if the scans are statistically dependent.

Fortunately, there is a solution to this problem which is not overly difficult to implement. This will be presented later after the discussion of the particle-scatter signal statistics, the understanding of which is necessary to the solution.

#### **D. Particle-Scatter Signal Statistics**

One of the most significant peculiarities of this problem is that the particle scatter signal itself is usually random. This is a little hard to see at first, but it is primarily the result of the non-flat (usually Gaussian, and assumed so here) irradiance profile or shape of the beam. Since the particle position is random, the part of the beam that strikes it is also random, and since the irradiance is not constant throughout the beam, different particle positions result in varying amounts of scatter for the same particle. If the beam center, the maximum irradiance point, swept every point on the surface (as with infinitesimal scan pitch), this would not be a problem, since the beam would eventually hit the particle at its maximum point. However, such a scan is clearly not practical. Instead the beam sweeps only along selected portions of the surface, usually in parallel rows, the distance between these rows being known as the scan pitch. Therefore, though the particle may be swept by the illuminating beam many times, the maximum irradiance that it encounters under the beam could be from as low as the beam irradiance at a distance from its center equal to the scan pitch up to as high as the maximum irradiance at the beam center. In addition, the scattering may be sampled at intervals along the row. This allows for a further randomization in that direction, characterized by the sampling time interval, which, multiplied by the scan speed, yields the spatial sample interval along that direction. Some instruments (though not the ASU instrument) eliminate much of this latter randomness with analog electronics which detect signal peaks in an essentially continuous manner. This

usually ensures that the particle will only be caught when the beam is centered over the particle at its peak brightness along the scan row, though it is possible that random fluctuations would cause the particle peak to be mislocated, thus introducing some randomness in that dimension. An exact analysis of the likelihood of a noise peak mislocating the beam peak has not yet been undertaken, but it would require a detailed knowledge of the analog processing which tends to be unique to every model of instrument.

An additional, though usually less significant, source of randomness comes from “jitter,” or random fluctuations of the beam position. This has been neglected in the present analysis, but it could be accounted for in the beam position distribution if it was well characterized.

To deal with the randomness of the particle signal for the simple and multiple-detector detection criteria (but not the class of multiple scan criteria, to be discussed later), we need to determine the nature of the deviance of the irradiance encountered by the particle, best expressed by its PDF. This PDF will also describe the deviance of particle-scatter signal, which is approximately linearly related to the irradiance of the incident beam (inasmuch as the detector has a linear response and the scattering process is linear.)

Developing a probability density function for the beam irradiance striking the particle and consequently the PDF for the particle signal can be fairly involved. Therefore we begin with the two most general cases in which the wafer and detection region (as explained in detail later) are assumed to be effectively infinite in extent. The first of these cases can be described as one-dimensional randomness.

For the one-dimensional randomness case, in which the particle signal is always detected at its maximum along the row, the derivation is less difficult. The Gaussian beam is often described by its maximum irradiance,  $I_{\max}$ , and its beam radius,  $w_0$ , the half-width distance

between the  $1/e^2$  points. This of course is a beam that illuminates the surface with radial symmetry. For Gaussian beams that illuminate the surface elliptically, such as seen in oblique incidence, the parameters  $w_a$  and  $w_b$ , for the lengths of the semimajor and semiminor  $1/e^2$  beam axes can be used. Alternatively the incidence angle and beam radius  $w_0$  will suffice. However, in the one-dimensional case, a single parameter  $w$  for the half-width perpendicular to the scan row suffices for this discussion.

It can be found without much difficulty that the beam irradiance along a line perpendicular to the scan row and passing through the beam center is

$$I = i_{\max} \times \exp\left(-\frac{2R^2}{w_0^2}\right) \quad \text{Eqn. 5}$$

This is the expression for the one-dimensional case. Since  $R$ , the distance of the particle from the center of the beam, is a random variable, the sampled irradiance  $I$ , of which  $R$  is a function, is also a random variable. The scattering from a small particle essentially samples this function  $I(R)$ , so we may apply a well known theorem concerning the transformation of random variables to find the PDF of  $I$ , namely,

$$f_I(i) = \frac{dR(i)}{di} \cdot p_R \quad \text{Eqn. 6}$$

which is simplified considerably by the fact that  $p_R$ , the PDF of  $R$  is uniform. This holds as long as the transformation is monotonic, which is the case here<sup>13</sup>.

The derivative is not difficult, but the resulting functional form is rather difficult to use in the integrals and integral transformations which must be applied to it. Finding this derivative,

$$\frac{dr(i)}{di} = \frac{w_0}{\sqrt{2 \ln \frac{i}{i_{\max}}}},$$

Eqn. 7

giving the PDF

$$f_1(i) = \frac{w_0}{i \sqrt{2 \ln \frac{i_{\max}}{i}}} p R$$

Eqn. 8

A property of this distribution, which I call the “bathtub” from its shape, should be noted: It goes to infinity at  $I=0$  and  $I= i_{\max}$ . This is exactly what it should do, if the particle is an infinitesimal point and the wafer is of infinite extent. This point assumption is often quite adequate if the particle is less than 1/10 of the beam radius and it is difficult to distinguish the results when it is 1/100 of the beam radius.

For a non-point-like particle, the distribution must obviously be computed numerically from the consideration that the particle is integrating over part a Gaussian curve. Additionally, the non-point-like particle experiences a potentially significant change in irradiance over its extent, which can affect its scattering signature in a non-linear manner. This effect clearly would have to be determined by computing an  $i-r$  curve with a particle scattering model and then differentiating it numerically.

The assumption of an infinite wafer works very well in most realistic cases, but it is often in our interest to limit the size of the area considered for the purposes of the statistical model. This will be dealt with in some detail later.

The two-dimensional case in which there is some appreciable randomness of particle position along the scan row due to sampling constraints is more difficult to deal with conceptually, though the actual computation is not so lengthy for the infinite wafer.

We begin by recognizing that we are now dealing with ellipses of equal irradiance on the surface and that the probability of finding a particle within a region on the surface is proportional to the area of that region. This being the case it, would seem that we need a relationship between the differential area of an ellipse and the differential irradiance at an irradiance corresponding to that ellipse. If we multiply this expression  $da(i)/di$  by the uniform areal PDF  $p_A$  we have obtained the PDF of the irradiance. Another way to look at this is that the cumulative probability of the particle falling within a given ellipse is proportional to the area within it (with proportionality constant  $p_A$ ), or

$$F_E(a) = ap_A \quad \text{Eqn. 9}$$

This probability is equal to the upper-tail cumulative probability of the sampled irradiance exceeding the value corresponding to that ellipse and its characteristic area,

$$F_I^u(i) = F_E(a) = ap_A. \quad \text{Eqn. 10}$$

Since the PDF is the derivative of the cumulative distribution function (or more precisely the negative derivative, since we are dealing with the upper tail),

$$f_I(i) = -d(F_I^u(i))/di = -da(i)/di p_A \quad \text{Eqn. 11}$$

which, not coincidentally, is almost identical to the familiar PDF transformation law.

It is fairly easy to see that the irradiance function is

$$I(X, Y) = i_{\max} \times \exp \left[ -2 \left( \frac{X^2}{w_a^2} + \frac{Y^2}{w_b^2} \right) \right]$$

Eqn. 12

which with proper manipulation becomes

$$1 = \frac{2}{\ln \frac{i_{\max}}{I}} \left( \frac{X^2}{w_a^2} + \frac{Y^2}{w_b^2} \right),$$

Eqn. 13

obviously the equation for the equal-irradiance ellipses,

$$1 = \left( \frac{X^2}{a^2} + \frac{Y^2}{a^2} \right), a = \sqrt{\frac{\ln \frac{i_{\max}}{I}}{2}} \times w_a, a = \sqrt{\frac{\ln \frac{i_{\max}}{I}}{2}} \times w_b,$$

Eqn. 14

with semimajor and semiminor axes lengths a and b. The area of this ellipse is

$$A = \pi ab = \frac{\pi}{2} \ln \frac{i_{\max}}{I} w_a w_b, I < i_{\max}.$$

Eqn. 15

Differentiating this with respect to I, negating, and multiplying by  $p_A$  then gives us the PDF for the infinite wafer and two-dimensional randomness,

$$f_I(i) = \frac{\pi w_a w_b}{2I} p_A.$$

Eqn. 16

It is immediately apparent that this distribution is very different from that in the one dimensional case, which peaks strongly at  $I = i_{\min}$  rather than dying off monotonically to the value at  $i_{\max}$  as in the 2-D case (Figure 4).

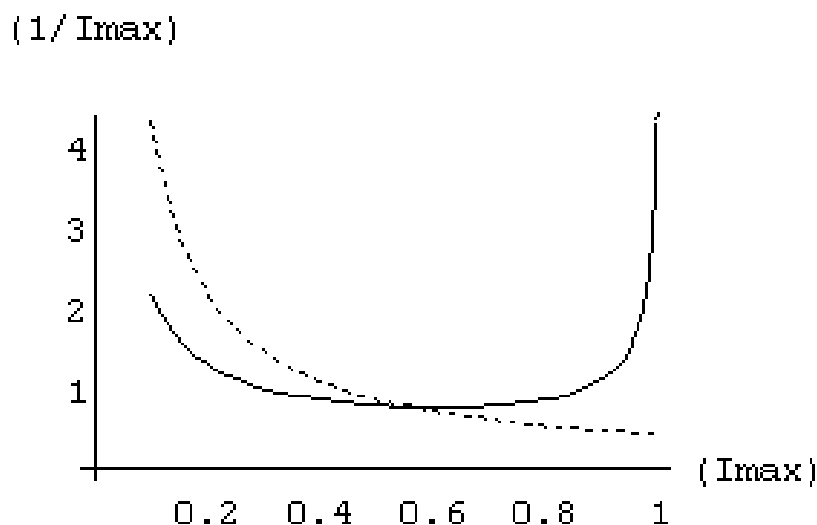


Figure 4. Example particle signal distributions for the one-dimensional (solid) and two dimensional (dashed) cases.

These two distributions are the basic limiting forms for most signal distributions encountered in variations of this problem. However there is a major complication which we need to address.

If we formulate the problem as a binary hypothesis test over one data acquisition with the form “there is no particle” versus “there is a particle”, we face a subtle question: “Where is there no particle?” By this simple statement of the problem, if we consider the whole wafer at once, we find ourselves in the undesirable position of considering any fulfillment of the detection criteria anywhere on the wafer to be a valid detection so long as there is at least one particle on the wafer, even if it lies 100 beam radii from the center of the beam. This is an absurd approach. Thus, we find that a binary hypothesis of the form “there is no particle within some region” versus “there is a particle within some region” may serve better.

The approach used here is to pick an arbitrary region such that the irradiance from the beam can be considered negligible outside the region. With the single-scan detection criterion in the one-dimensional case, this region can simply be defined as  $I_{\min} < I < I_{\max}$ ,  $I_{\min}$  being the irradiance at the outer boundary set to some small fraction of  $I_{\max}$ . 1% is a reasonable choice for this. In the two-dimensional case it is convenient to pick the region within the  $I_{\min}$  ellipse. If this is done, we end up with the same forms as two general PDFs but where the probability density is zero for  $I < I_{\min}$ . Normalization is accounted for in the proper choice of  $p_r$  or  $p_A$ , which differ for the decreased sample regions.

### **E. Particle-Scatter Signals with Multiple-Sample Criteria**

As alluded to earlier, we do not calculate the final PDFs to deal with the particle-scatter signal statistics with multiple-sample detection criteria. This is due to the fact that the particle scattering signals are not statistically independent, so detection probability must be calculated in a much different manner. This becomes clear when we realize that, if a detection is registered during one sample, the likelihood of registering a detection on an adjacent sample is greater due to possible scattering by the putative particle in the outlying parts of the beam.

As with the previous cases, there is a need to restrict the consideration to certain regions, but now we consider separate sample regions as well, limiting ourselves to regions of non-negligible irradiance. For the two-dimensional randomness case, sample regions should be defined in such a way as to tile the surface, generally with rectangles if the scan pattern is rectangular, though other patterns are at least conceivable (Figure 5). The reason for this is that, given a particle on the surface somewhere, there will be one particular sample (ignoring the insignificant case of “fence-sitters” that lie between regions) in which the beam manages

to hit the particle with a higher irradiance than any other scan. The samples taken in the region around this sample will all illuminate the particle with decreasing irradiance as their distance from it increases. The one-dimensional randomness case is analogous but requires dividing a trail of peaks perpendicular to the scan trace direction into segments with length equal to the scan pitch. This simpler case is what we examine. It is relatively easy to expand the examination to two dimensions, but the concepts are somewhat clearer in one.

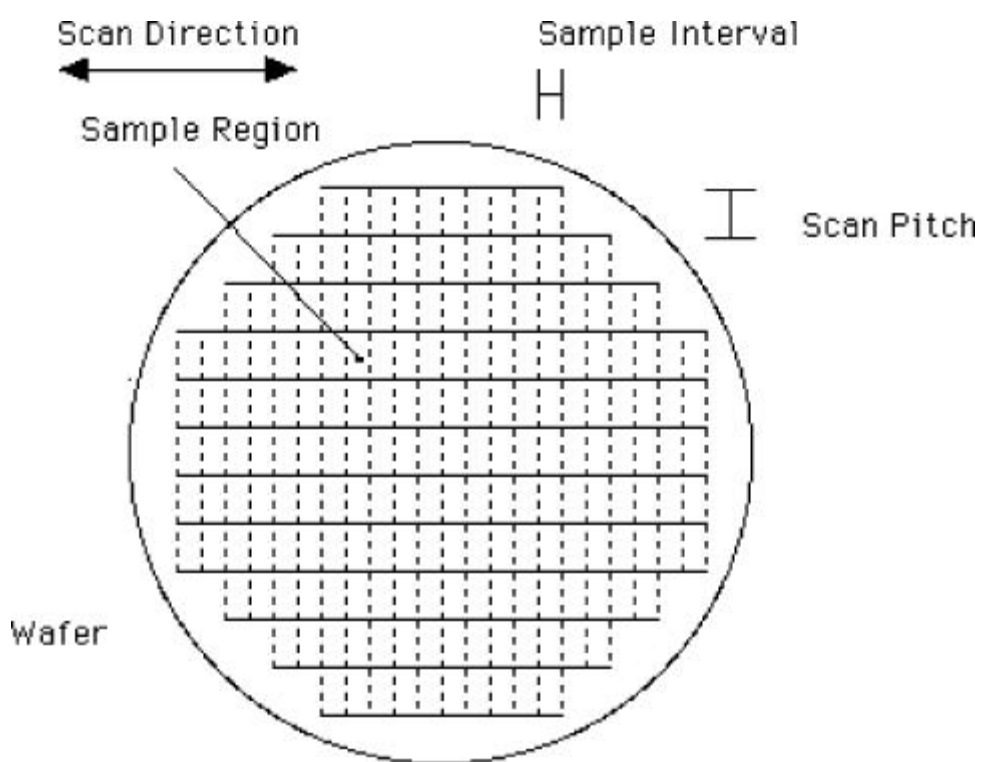


Figure 5. Analytical framework for multiple-scan detection with two-dimensional randomness.

The method for computing  $P_D$  and  $P_F$  in such cases relies on expressing the problem in terms of the position random variable or variables rather than in terms of several dependent

particle scattering random variables. To compute the probability of getting at least one detection of a particle in a set of scans, we limit our consideration to those scans where the particle illumination is non-negligible. Considering the one-dimensional case with hypothetical perfect peak detection and no pitch-wise jitter, we have a series of equally-spaced scans which encounter the particle scattering peak at various powers according to the position of the particle relative to the beam center. For the Gaussian beam considered in the present work, this gives us a series of equally-spaced samples of the Gaussian irradiance curve (Figure 6).

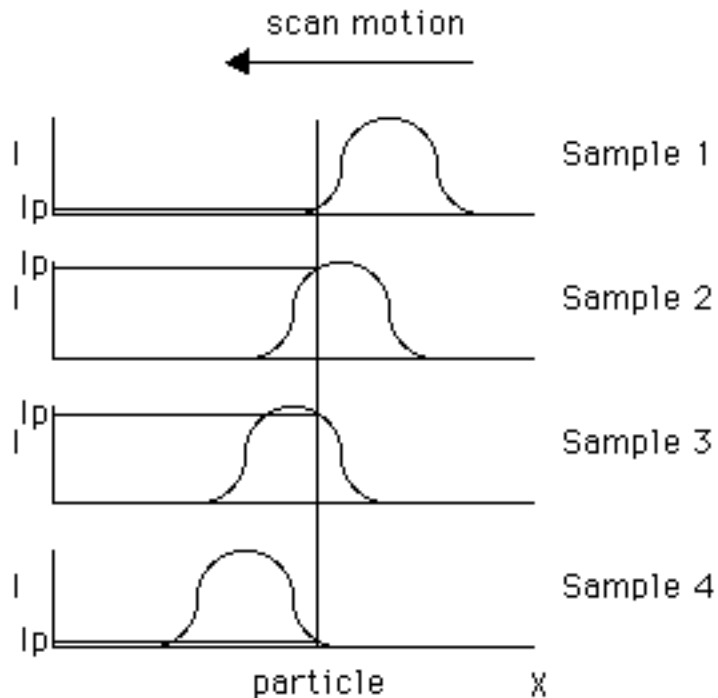


Figure 6. Multiple samples of a single particle.  $I_p$  is the irradiance striking the particle in each case.

We can divide the irradiance curve into segments along the position axis with the same spacing as the scan pitch such that the particle will certainly be encountered in each of these segments exactly once, but, since the particle position is random, the exact point where the particle is encountered in each segment is unknown. A crucial observation is that the encounter position in each segment is going to be some distance  $\phi_s$  from the left boundary of the segment and that this distance will be the same for every segment. This distance will be called “scan-phase” since this situation is quite analogous to periodic phenomena. Further, we observe that  $\phi_s$  is the random position variable of interest. This arises from the fact that the absolute segment in which the particle exists is unimportant. Neglecting the case of particles near the edge of the surface, no matter where the particle is, there will always be a series of such segments, so that the series of scattering powers encountered is a function only of the scan phase. This assumes that all statistics are stationary with respect to translation along the surface, but that is generally the case with a bare wafer and is assumed throughout this and most works on the subject.

Thus we can express the irradiance encountered in each segment  $k$ , numbered from  $-n$  to  $+n$ , as

$$i_{\text{particle}}(\mathbf{f}_s) = i_{\text{max}} \times \exp\left[-\frac{2(\mathbf{f}_s + kx_s)^2}{w_0^2}\right]. \quad \text{Eqn. 17}$$

$x_s$  is the scan pitch, and  $w_0$  is the beam radius, as seen earlier.

The next step is to recognize that every possible value of  $\phi_s = \mathbf{f}_s$  corresponds to a value of  $P_D(\mathbf{f}_s)$ . Because  $\phi_s$  is independent of the noise random variables, the expected value of  $P_D$  is just

$$P_D = \int_0^{x_s} f_{\mathbf{f}_s}(\mathbf{f}_s) P_D(\mathbf{f}_s) d\mathbf{f}_s \quad \text{Eqn. 18}$$

where  $f_{\Phi_s}(\phi_s)$  is the PDF of  $\Phi_s$ , uniform except if taking scan jitter into account.

The functional form of  $P_{D(\Phi_s)}$  depends on the particular type of multiple-scan test in question. If we want to know the probability of getting at least one detection of the particle in the  $2n+1$  non-negligible scans, we use

$$P_D(\mathbf{f}_s) = 1 - \prod_{i=-n}^n P_{M_i}(\mathbf{f}_s) \quad \text{Eqn. 19}$$

because the probability of detecting on at least one relevant scan is the complement of the probability of missing on all  $2n$  relevant scans.

For the often used case in which the signal must cross threshold for two adjacent scans to constitute a detection<sup>14</sup>, the expression is considerably more tedious because of the large number of cross-terms between the probabilities for the various segments. The best method found during the course of this research is to take the probabilities for each possible outcome that would constitute a detection and add those probabilities. This requires identifying the detection outcomes and no efficient approach to this has yet been found, so that each the possible combinations of possible events must be tested individually. This is implemented in the model code in the file DBLCOINC.C in the function `coincprob(...)` (see Appendix I), but it executes extremely slowly in this general form. When the code is specifically designed for one particular number of sample regions, it can be reasonably efficient (as shown also in `coincprob(...)`), but the Monte-Carlo simulation method is generally much faster for this case than the general analytical form. Appendix III

gives a description of how to get the  $P_D$  formula for particular numbers of sample regions, and this might be coded as a general-case algorithm with sufficient motivation.

Unfortunately, the Monte Carlo method can be difficult to implement well in this particular case because the commonly used linear congruential pseudorandom number generators tend to have significant correlation between subsequent calls<sup>15</sup>. This is not much of a problem in some cases, but this one requires a series of independent random values for the noise and then subjects them to a coincidence test. Coincidence tests are in fact used because of their sensitivity to correlation, so this correlation may substantially alter Monte Carlo results.

The calculation of  $P_D(\Phi_s)$  (or its complement) for each individual segment is quite similar to the integration method used in simpler cases. The optical noise PDF is integrated but is shifted to account for the particle scatter signal at that particular  $\phi_s$  value. In terms of

$i_{\text{particle}}(\mathbf{f}_s)$ ,

$$P_{D_i}(\mathbf{f}_s) = \int_{\text{thresh}}^{\infty} f_{\text{noise}}(i - i_{\text{particle}}(\mathbf{f}_s)) di . \quad \text{Eqn. 20}$$

For post-optical noise sources, the shifted PDF must be processed using methods discussed later for the other cases.

Obviously, putting all these computations together yields a total procedure that is not trivial and that can be even more complicated with more involved detection criteria, so Monte Carlo computation might be preferred here, keeping in mind the necessity of having a pseudorandom number generator with a low degree of correlation.

## **F. Optical Noise**

Unlike most common types of electrical noise, many optical noise sources in this problem tend to actually add to the mean of the signal rather than to simply randomize it. They also are mostly statistically independent, two facts that will later allow the computation of their effects. The most significant of these tends to be scattering from surface roughness, which is very well researched and understood in most aspects. With the coherent illumination most common in particle detection, this surface scattering appears as laser speckle, which is similarly well researched due to the importance of the general speckle phenomenon in numerous technologies. Its statistics can be quite complicated however.

The analysis of speckle statistics usually begins with the development of the irradiance (sometimes also called intensity, power per area) statistics. In the most basic form, the surface is assumed sufficiently rough that the phase of scattered waves is completely randomized and that a “large” number of independently scattered waves interact. This assumption allows the invocation of the central limit theorem, revealing the scattering statistics to show a tractable near-Gaussian distribution.

It is found that the irradiance statistics may be approximated by the negative exponential distribution, provided the above assumptions hold, but in wafer scanners the point irradiance is not measured but rather the light is integrated over some solid angle.

In the commonly used “boxcar” approximation method, we consider the speckle pattern to be composed of highly correlated regions called cells, a composition which will be familiar to anyone who has viewed a laser speckle pattern. Across these cells the irradiance is roughly constant, having a value which follows the point irradiance distribution. The characteristic or “typical” area of these cells is usually considered to be roughly the

correlation area of the speckle pattern<sup>16</sup>. Thus, the solid angle of integration will contain some “typical” number of speckle cells, each cell with an independent negative exponential irradiance distribution. These assumptions lead clearly to the detector power being a sum of negative exponential random variables, which consequently follows the gamma distribution. Numerous experiments have shown that this gamma distribution models the observed statistics quite well.

An exact form for the integrated speckle distribution has been derived using the Karhunen-Loève expansion<sup>17</sup>, but this exact form is considerably more difficult to use and derive since it requires the solution of an integral equation using the aperture shape.

It is important to note that there will be a characteristic detector aperture solid angle under which a distribution closely following the negative exponential form will be observed whereas for greater solid angles the approximate gamma form will be observed, with values for the “shape” parameter increasing with solid angle. Ultimately, the gamma distribution approaches the normal distribution as the detector aperture becomes much larger than the speckle cell, as would be expected from the central limit theorem.

As noted above, this analysis is sufficient when the given assumptions are met, and they usually are for wafer scanner cases, except when the size of the illuminating beam is particularly small.

Rough surfaces have a typical correlation length, which expresses the lateral dimension of the random variation in surface height. When the laser beam radius is on the same order as this surface correlation length, there will clearly be only a small number of independent effective scatterers within the illumination. This tends to invalidate the assumption of Gaussian statistics based on the central limit theorem, which holds as the number of

variables summed becomes large. Normal-incidence beams are more prone to this than oblique-incidence beams, due to the fact that the oblique beam tends to subsume more effective scatterers than a normal beam of the same width (Figure 7).

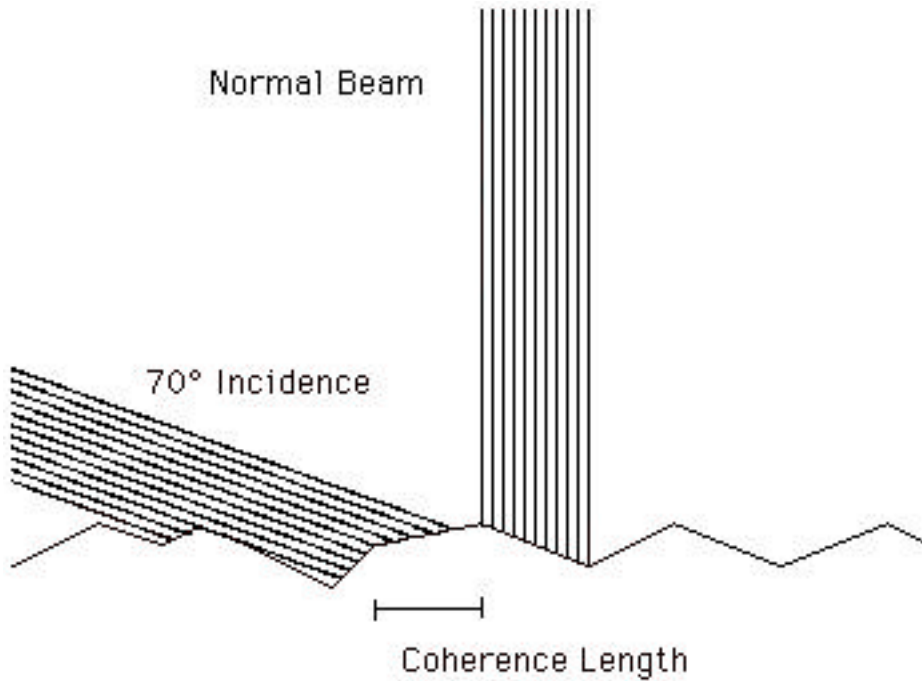


Figure 7. Oblique beams scatter Gaussian at smaller widths because they subtend a greater number of effective scatterers.

Jakeman and Pusey<sup>18</sup> have studied this problem extensively and have found that the statistics for analogous scattering in turbulent media often can be modeled well by modified Bessel function or K distributions of the form (altering their notation somewhat)

$$p(I) = \frac{2b}{\Gamma(\mathbf{a})} \left( \frac{bI}{2} \right)^{\mathbf{a}} K_{\mathbf{a}-1}(bI), \quad \text{Eqn. 21}$$

where  $I$  is the irradiance and  $\mathbf{a}$  and  $b$  are parameters of the distribution.

Indeed, the statistics of surface-roughness scattering and turbulent-medium scattering have a great deal in common, and this has led to considerable theoretical cross-pollination between the two areas.

The application of these distributions to scattering by surface roughness has been studied by numerous authors, recently by Sanchez-Gil *et al.*, who found good agreement with experiment<sup>19</sup>.

The distributions for integration of multiple speckle cells with K-distributed irradiances cannot generally be represented in terms of simple functions, but can be computed numerically as following Lure and Yang<sup>20</sup>.

Figure 8 shows the distributions applicable to speckle statistics in various regions in continuum of the beam width vs. detector aperture.

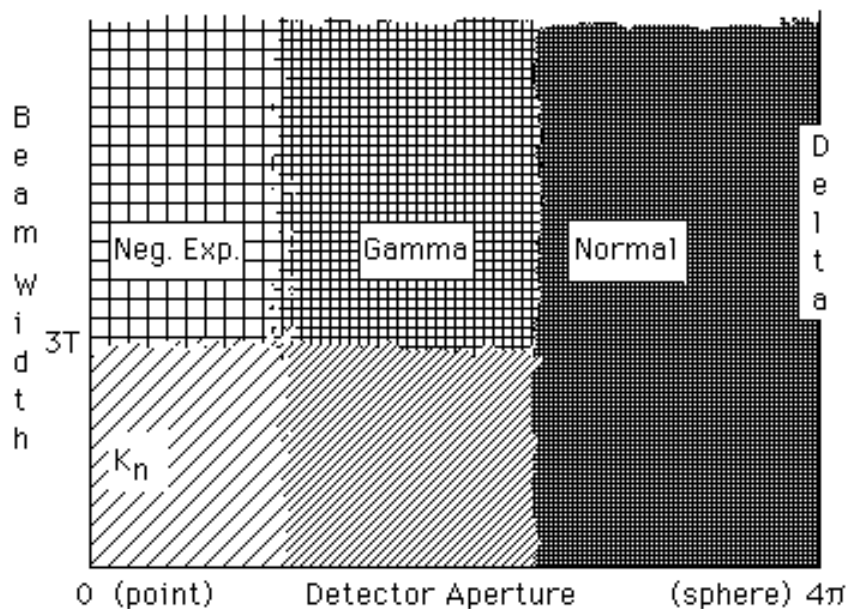


Figure 8. Spatial noise irradiance distribution according to region on the beam width vs. detector aperture plain.  $T$  is roughly the correlation length of the surface and  $3T$  is the

approximate transition point to non-Gaussian statistics for normal or near normal incidence.

In addition to the speckle effect, surface scatter also depends on the presence or absence of a particle on the illuminated region, since much of the light scattered by the particle will not be involved in surface scattering. This effect will tend to be small for small particles (compared to the beam width), but it would make the surface-scatter noise slightly dependent on the particle signal. This has been neglected since the assumption of a particle relatively small compared to the beam width has already been introduced to eliminate consideration of the irradiance profile significantly varying over the particle's extent.

### **G. Independent Optical Noise Sums**

Composing the surface scatter noise and whatever other optical noise into noise sources into one effective noise source ( $P_N$  above) is fairly easy due to their statistical independence. Calculation of PDFs for sums of independent random variables is easy to do, a simple convolution of the PDFs of the addends. In many cases, it might be worthwhile to perform this operation in the Fourier domain, where the Fourier transform (FT) of the sum-PDF is the product of the FT of the addend PDFs. In fact this can lead to easily obtained analytical expressions since the FT of the PDF, the characteristic function, is tabulated for most common distributions and frequently turns out to have an exponential form, one which is amenable to multiplication and subsequent inverse transformation to return to the power domain.

For the first two detection criteria, this approach also works for combining the particle signal power with the optical noise to get the total signal power, at least if the possible

dependencies of the spatial noise between multiple detectors are neglected, as they usually can be, assuming the detectors are spaced by more than a few speckle correlation lengths.

The complicated log-root-reciprocal form of the particle signal PDF prevents the application of this method to analytically determining particle signal+noise PDFs. Despite this though, the particle signal PDF may be convolved with a noise PDF of known form without too much difficulty by numerical methods, as is done here.

The third type of detection criterion, with multiple scans, requires an entirely different combination of the optical noise with the particle signal, since the particle signals between scans are highly correlated. It is in fact the presence of the correlation that makes this criterion particularly advantageous. This will be outlined later.

#### **H. Dependant Variable Sums**

At the optical-electrical interface, things are not quite so simple. Shot noise is not a statistically independent source. For the example of simple Poisson-distributed shot noise, the parameter  $l$  depends on the detector current, or more precisely on the expected value or mean of the current. This really represents the effect of granularity or quantization of the medium carrying the signal, which is first light, composed of discrete photons, and then electrical current, composed of discrete charge carriers.

The mean current represents the current to be expected as the granularity vanishes or as the current is observed over a long period, and the Poisson distribution indicates the probability that the observed number of discrete carriers arriving in a given time will have a certain value given a mean current.

To account for this, we must first recognize that shot noise has no effect on the mean of the signal (that is, it neither sources nor sinks power on average) but only increases its variance. To calculate the signal PDF resulting from this, we must examine the problem from a fundamental level.

At each point on the abscissa of the resultant PDF, the signal value axis, there is a contribution to its probability from each member of an entire set of possible signals because the signal input to the shot-noise process is random itself. Each of these members has a different random mean power  $P_m$ , the distribution of which is the PDF of the signal before the shot noise,  $f_{P_M}(p_M)$ . Each possible mean  $P_m = p_m$  is subjected to random shot noise producing a discrete random output signal  $P_N = Nq/t$  which has a Poisson distribution of mean,  $I = p_m/q$ , which is the mean number of photons arriving in the sample interval, explicitly

$$f_{P_N}(k) = \frac{I^k e^{-I}}{k!}, \quad I t = p_m t/q, \quad \text{Eqn. 22}$$

where  $q$  is the quantum, the photon energy (or carrier charge if the analysis is approached from the other side) and  $t$  is the sample time. The probability of obtaining this value for a given  $p_N$  may then be seen to be

$$P_{P_N} = \int_0^{\infty} \frac{\exp\left(\frac{-p_M t}{q}\right) \left(\frac{p_M t}{q}\right)^{P_N t/q}}{\left(\frac{P_N t}{q}\right)!} f_{P_M}(p_M) dp_M, \quad \text{Eqn. 23}$$

the probability of the intersection of these events, integrated over the whole ensemble of possible means. Similar logic leads to the above convolution theorem concerning the PDF of sums of independent random variables.

Optical detection systems always suffer from this type of noise to some extent, particularly at low signal strength, because the total energy detected over an integration time will approach the photon energy. Electrical systems tend to display it chiefly in devices like p-n junction diodes and photomultiplier tubes (PMTs) where the carriers must cross some barrier, such as a space-charge region with p-n junctions or an evacuated region with PMTs. Of course, many wafer scanners use photodiodes or PMTs as optical transducer elements so they might be said to have multiple sources of shot noise. However, since carrier generation events correspond to photon incidence events, the noise sources are strongly correlated and hence behave effectively as a single shot-noise source, at least for the initial generation. However, in devices like PMTs and avalanche photodiodes (APDs) which produce multiple carriers per photon incidence, this holds true only for the first generated carrier from a photon. Each subsequent carrier generation event in the amplification process is also random, so that this further randomizes the signal. For these avalanche amplification processes, the Poisson-distribution is not always a good model, but often the avalanched carriers propagate coherently enough to be approximated as single Poisson-distributed pulses.

### **I. Electrical Noise Sources**

There are numerous types of electrical noise apart from the electrooptical shot noise. These include thermal or Johnson noise,  $1/f$  or flicker noise, and various types of stray random signals that are introduced into the circuit. Of course, which source is dominant depends greatly on the particular electronics found in the instrument. It is however suspected that electronic noise in the ASU instrument is primarily thermal in origin.

This thermal noise is not difficult to deal with statistically. Like shot noise its contribution to the signal is Poisson-distributed but can usually be approximated with a Gaussian distribution. Unlike shot noise, its distribution is not dependent on the mean signal, and its contribution can be treated as a random variable with a zero mean and some variance which depends on temperature. Thus, its effect on the signal distribution may be computed using the familiar convolution method with a response function that is a Gaussian centered at the origin.

## **4. RESULTS AND ANALYSIS OF PERFORMANCE**

### **A. Receiver Operating Characteristic Curves**

Once all the noise sources have been analyzed and the detection and false count probabilities have been computed, it remains to analyze them. The performance of such detection systems as wafer scanners is typically expressed in terms of the receiver operating characteristic (ROC). This is a plot of the  $P_D$  versus  $P_F$ .

One advantage of this representation is that it demonstrates the range of possible probabilities for all possible thresholds. The threshold of the instrument is typically easily altered and in fact by altering this threshold either  $P_D$  or  $P_F$  (but not both) may be set to any possible value. Thus in such detection systems it is common to set the threshold to fix one of these to some appropriate value and to compare system performances by the value of the other probability. Normally,  $P_F$  is fixed in this manner and often to a value of 5%, indicating that 5% of tests when no particle is present will result in a false detection.

The ROC is also useful for determining the threshold value for the likelihood ratio test, which was discussed earlier. When the  $P_F$  is fixed at some value, the slope of the ROC curve at this point is in fact the likelihood ratio, and, with this particular method of setting the threshold, the test is known as the Neyman-Pearson criterion. The fact that the slope is the likelihood ratio easily follows from the recognition that the slope is the ratio of the derivatives of  $P_D$  and  $P_F$  and that, since  $P_D$  and  $P_F$  are the complements of two cumulative distribution functions, their derivatives are simply the negatives of the corresponding PDFs, and the likelihood ratio is just the ratio of these PDFs.

Additionally, the ROC illustrates a concept which is of particular interest in this application, the lower detection limit. This is the limit at which we consider the particle signal to be too weak to be reliably detected.

It can be expressed in terms of the value of the peak of the particle signal or alternatively in terms of some parameter upon which this monotonically depends, such as scattering cross-section, which is the parameter ultimately of greatest interest. However, the definition of what probabilities represent "reliable detection" is not entirely clear or universally agreed upon, but a common definition, and one which will at least be sufficient for this discussion, is that the particle can be detected reliably if, with the threshold set for a  $P_F$  of 5%, the probability of detection  $P_D$  is at least 95%, from now on called the "95-5" criterion. To see this on an ROC plot, one must draw a vertical line at  $P_F = 0.05$  and a horizontal line at  $P_D = 0.95$ . An ROC curve which crosses the  $P_F$  line at  $P_D \geq 0.95$  corresponds to a particle which can be reliably detected.

Figures 9 and 10 illustrate this with a family of simulated ROC curves. The most widely dashed curve corresponds to a given particle signal and the subsequent lines with more closely spaced dashing correspond to integer multiples of that particle signal. Only the solid line crosses the  $P_F = 5\%$  line where  $P_D$  exceeds 95% and consequently we can see that the minimum detectable particle signal (and its corresponding size) would be between the solid curve which passes the criterion and the first dashed curve which does not.

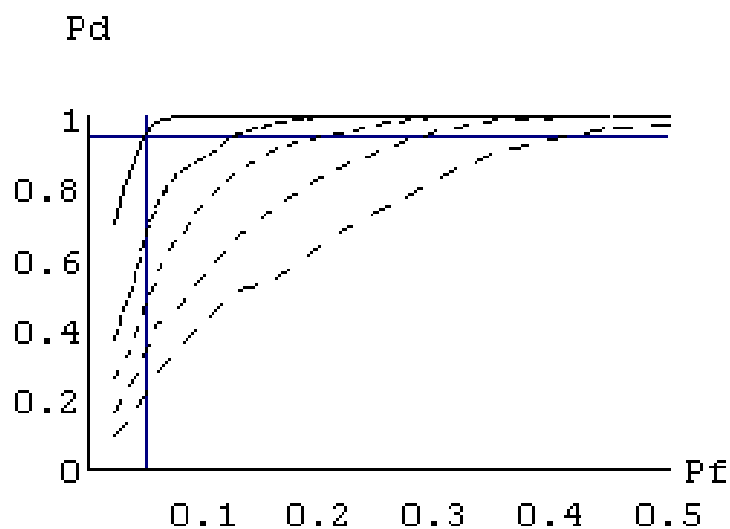


Figure 9. Left half of ROC plot for multiple scan criterion specifying detection when at least one illuminated scan region produces a signal over threshold. Simulation based on irradiances observed on ring 21 of the ASU instrument (a single detector) using one-dimensional randomness. The far-right trace (most widely spaced dashing) indicates a signal for a  $.204 \mu\text{m Si}_3\text{N}_4$  particle. Curves with successively closer dashing represent signals with irradiances that are integer multiples of that of the rightmost signal. Thus the solid leftmost curve is for a signal that has 5 times the irradiance of first signal and is the only one in the series to meet or surpass the 95%-5% lower detection limit criterion, so as to be considered detectible. Calculated using the analytical method.

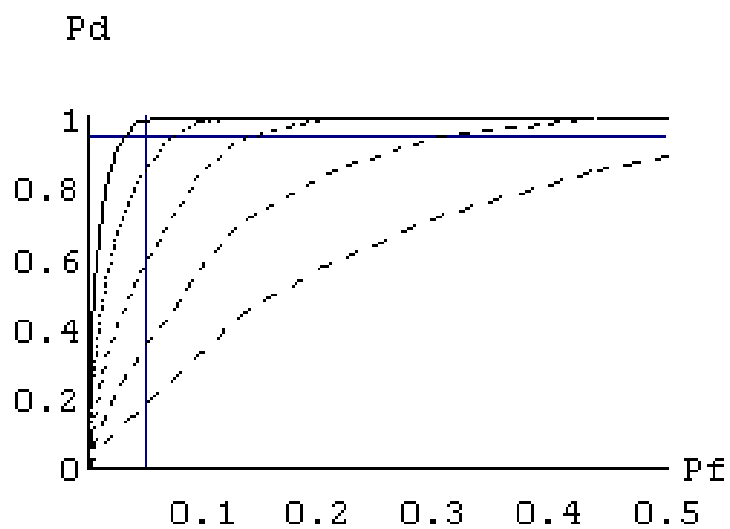


Figure 10. ROCs for same case as in Figure 9 but using a criterion that registers detection only if at least one pair of adjacent scans produced threshold-crossing signals on both scans in the pair. Note the superior performance over the criterion used in Figure 8. Calculated using the Monte Carlo method.

## B. Results

Most of the results of this project are expressed in ROC curves. Figs. 9 and 10, as discussed above, show some of those results and illustrate their interpretation, clearly showing the effects of changing particle scattering cross-section and different detection algorithms.

Another interesting ROC plot computed with the same code is a family of curves for different scan pitches with one-dimensional randomness shown in Figure 11. This once again illustrates the efficiency of the system improving with a parameter, in this case the scan pitch.

Figure 12 shows ROC plots generated by the analytical method and Monte Carlo method for the same case.

Next, Figure 13 shows a set of ROC curves for simulated scattering from a series of PSL spheres of different sizes. The drastic difference between curves on either side of the 95-5 point suggests that the transition of the family of curves through it can be expected to be extremely abrupt with respect to particle size.

Finally, Figure 14 is a plot of lower detection limits in terms of surface roughness (spatial noise) vs. particle size. It was developed with an iterative search for  $P_D, P_F$  corresponding to three different lower detection limit criteria though the threshold/RMS surface roughness space for different particle sizes. This shows how differing definitions of lower detection limit can significantly alter the value of that limit, as well as the way the limit in terms of particle size can change with surface scattering noise.

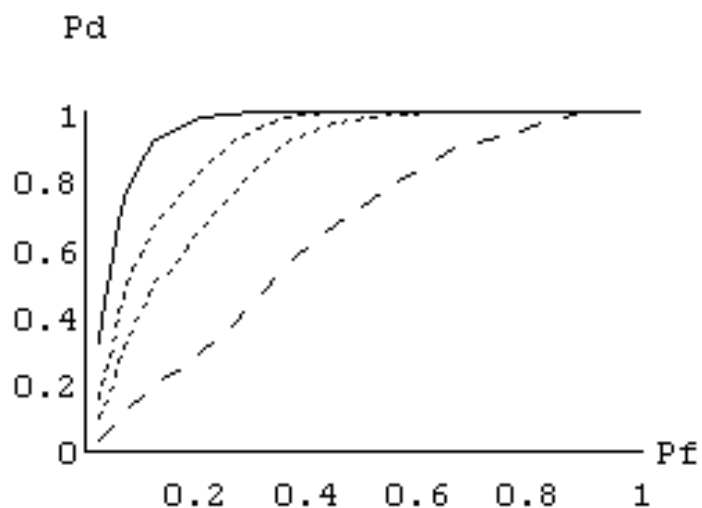


Figure 11. Analytically calculated ROC curves for scan pitches (from left to right) of 2.5, 5, 10 and 20  $\mu\text{m}$  and beam width of 20  $\mu\text{m}$ . Particle signal of  $1.7\text{E-}7$  W, gamma dist. noise with parameter 2 and  $3.7\text{E-}7$  W. At-least-one-detection criterion.

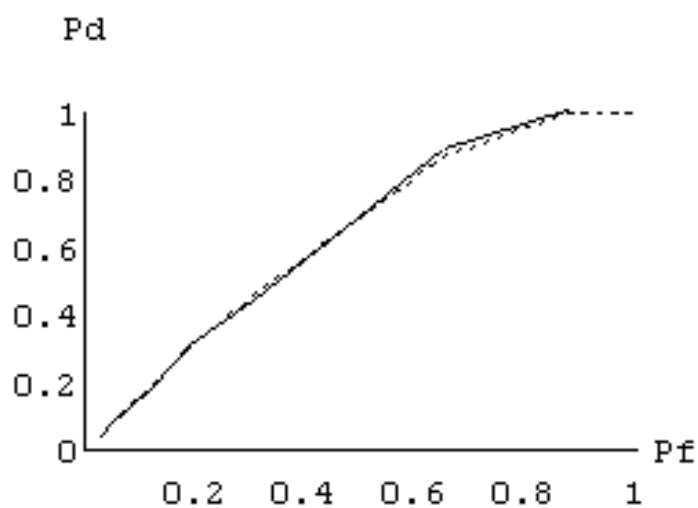


Figure 12. Monte-Carlo (dashed) and analytically calculated (solid) ROCs for a multiple scan case using a criterion of at least one threshold crossing to indicate detection.

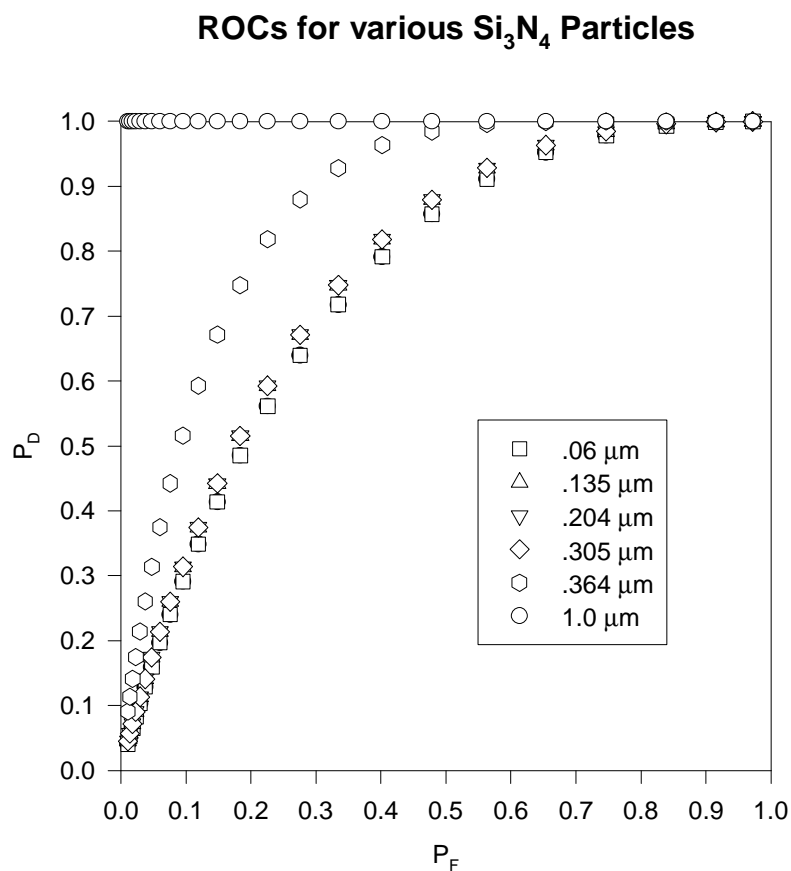


Figure 13. ROC plots for total integrated particle scattering cross-sections of PSL spheres of various diameters. The multiple-sample case was used with one-dimensional coincidence of adjacent samples. Noise assumed as gamma-distributed (shape parameter 2) with mean equal to roughly 1/3 the power from the 1.0 μm particle and 3 times that from the .364 μm. Cross sections simulated by Brent Nebeker using DDSURF. The illumination was 632.8 nm laser light with a 20 μm, 10 mW beam. The spatial sample interval (scan pitch) was 10 μm.

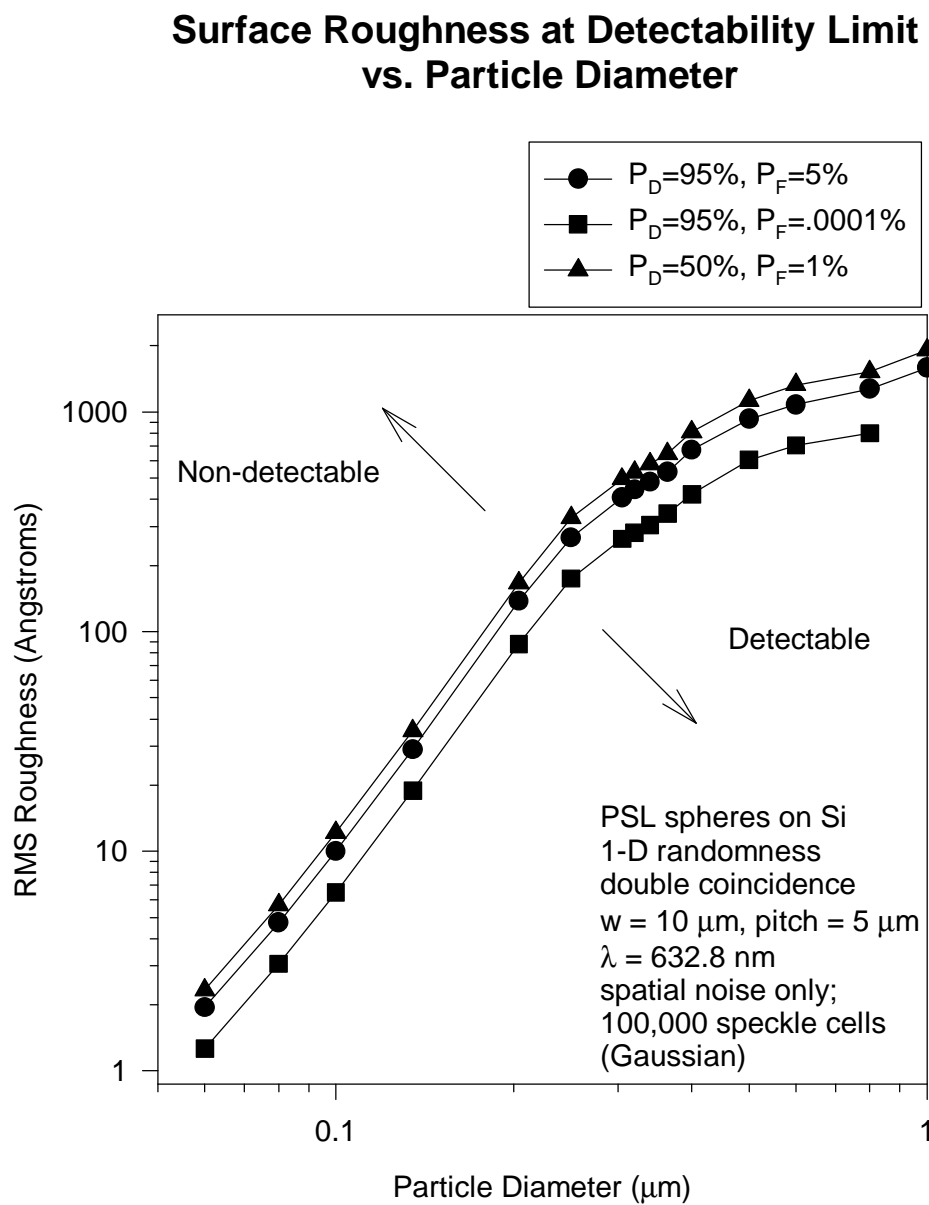


Figure 14. Lower detection limits for three different  $P_D$ - $P_F$  criteria with respect to RMS surface roughness at various particle sizes. Particle scattering cross-sections courtesy of Brent Nebeker.

### **C. Conclusion**

This instrument modeling task has succeeded in identifying major factors affecting SSIS performance, in determining how to account for these factors to compute performance statistics, and in implementing simple modeling code to perform these computations.

Furthermore, it has demonstrated how these performance statistics may be expressed to compare different SSIS designs and operating parameters and how the notion of lower detection limit relates to them.

There is considerable further work that may be performed in this area. In particular, the model should be refined to reflect existing instruments more closely. Also of interest would be the design and execution of statistical experiments using an instrument with well-known and well-characterized operating parameters and data processing algorithms in order to gauge the accuracy with which the model succeeds in combining diverse and separately well-understood models to model the entire system. Current work in upgrading the control software of the ASU instrument should make this goal attainable. Models for more detection criteria might also be of some interest.

Table 1. Symbols Used

(Note the use of the common convention that random variables are in upper case while deterministic constants are in lower. Similarly, probabilities use upper-case “P” and probability densities functions use lower case “f”).

$I$	Irradiance (random)
$i_{\max}$	Maximum irradiance (beam center)
$f_A$	Areal probability density
$f_s(s)$	Electronic signal PDF
$P_D$	Detection probability
$P_F$	False-count probability
$P_M$	Miss probability
$P_N$	No-count probability
$R$	Sample position phase (random)
$S(P_T)$	Optical-electronic function
$S_N$	Additive electronic noise
$w_0$	$1/e^2$ beam radius
$S_o$	Electronic output signal
$p_C$	Constant optical noise power
$P_N$	Stochastic optical noise power
$P_{\text{opt}}$	Total optical power

$P_s$ 

Particle scattering power

 $\Lambda$ 

Likelihood ratio

## REFERENCES

- <sup>1</sup> Y. Baghzouz and O.T. Tan, "Probabilistic Modeling of Power System Harmonics," IEEE Transactions of Industry Applications **IA-23**, 173-180, (1987).
- <sup>2</sup> M.D. Wheeler and K. Ikeuchi, "Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence **17**, 252-265, (1995).
- <sup>3</sup> S.D. Personick, P. Balaban, J.H. Bobsin, and P.R. Kumar, "A Detailed Comparison of Four Approaches to The Calculation of the Sensitivity of Optical Fiber Receivers," IEEE Transactions on Communications, 541-548 (1977).
- <sup>4</sup> *ibid.*
- <sup>5</sup> A. Woodbury, F. Render, and T. Ulrych, "Practical Probabilistic Ground-Water Modeling," Ground Water **33**, 532-538, (1995).
- <sup>6</sup> M. Liswith, *Numerical Modeling of Light Scattering by Individual Submicron Spherical Particles on Optically Smooth Semiconductor Surfaces*, M.S. Thesis, Dept. of Mechanical and Aerospace Engineering, Arizona State University, (1994).
- <sup>7</sup> E.J. Bawolek, D.K. Vaughan. E.D. Hirleman, "Computational and Experimental Study of Light Scattering by 0.364  $\mu\text{m}$  Spheres on Si and SiO<sub>2</sub>," in *Proceedings of the First International Symposium on Ultra Clean Processing of Silicon Surfaces*, University of Leuven, Belgium, 33-34, (1992).
- <sup>8</sup> P. Beckmann and A. Spizzichino, *The Scattering of Electromagnetic Waves from Rough Surfaces*, MacMillan, New York, (1963).

- <sup>9</sup> J.C. Stover, *Optical Scattering: Measurement and Analysis*, McGraw-Hill, New York, 59-61, (1990).
- <sup>10</sup> J.M. Elson and J.M. Bennett, *Vector Scattering Theory*, *Optical Engineering* **18**, 116-124, (1979).
- <sup>11</sup> P. Beckmann and A. Spizzichino, *The Scattering of Electromagnetic Waves from Rough Surfaces*, MacMillan, New York, 107, (1963).
- <sup>12</sup> C. Amra, D. Torricini, and P. Roche, "Multiwavelength (0.45-10.6  $\mu\text{m}$ ) angle-resolved scatterometer or how to extend the optical window," *Applied Optics* **32**, 5462-5474 (1993).
- <sup>13</sup> M. Barkat, *Signal Detection and Estimation*, Artech House, Boston, 41, (1991).
- <sup>14</sup> J. Pecen, A. Neukermans, G. Kren, and L Galbraith, "Counting Errors in Particulate Detection on Unpatterned Wafers," *Solid State Technology*, 149-154, (1987).
- <sup>15</sup> W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, New York, 277 (1994).
- <sup>16</sup> J.W. Goodman, "Statistical Properties of Laser Speckle Patterns" in *Laser Speckle and Related Phenomena*, J.C. Dainty ed., Springer, New York, 49, (1975).
- <sup>17</sup> M.A. Condie, *An Experimental Investigation of the Statistics of Diffusely Reflected Coherent Light*, Ph.D. Thesis, Dept. of Electrical Engineering, Stanford University, (1966).
- <sup>18</sup> E. Jakeman and P.N. Pusey, "Significance of K Distributions in Scattering Experiments," *Phys. Rev. Lett.* **40**, 546-550 (1978).

<sup>19</sup> J.A. Sanchez-Gil, M. Nieto-Vesperinas, and F. Moreno, "Speckle Statistics of EM Waves Scattered from Perfectly Conducting Random Rough Surfaces," *J. of the Optical Society of America A* **10**, 2628-2636, (1993).

<sup>20</sup> Y.M. Lure and C.C. Yang, "Probability Distributions for the Integrated Intensity and Photocount Associated with a K-Distribution for Laser Intensity," *Applied Optics* **28**, 5250-5258, (1989).

<sup>21</sup> F. Pearson II, *Map Projections: Theory and Applications*, CRC Press, Boca Raton, FL, 129-133 (1990).

**APPENDIX I**  
**MODEL CODE**

The model code was written in C++ using Borland C++ 4.51. The target environment was Win32s, a 32-bit API that overlays Windows 3.1. This allowed the use of full 32-bit code with the attendant execution speed gains over 16-bit code. The IMSL C Numerical Libraries version 2.0 for Windows NT (which uses the Win32 API) were used for most standard computations.

The code avoids using console input and output streams since these are not supported in the Win32s subset of Win32 and instead output is written to files. It was possible to use the GUI version of the API, but this would have entailed more time spent developing a GUI interface than was deemed appropriate. The current design should however build and execute with little or no alteration under any Win32 environment, including Windows NT, Windows '95, and 32-bit DPML.

It should be noted, however, that exception handling under Win32s was generally poor and often exceptions raised by the library functions were not trapped by the C++ exception handling mechanism. This made debugging the code quite difficult in some cases because the path of execution could only be traced by writing strings to the log files. In a few cases, the program failed to flush the file buffers after exceptions so it was then even necessary to flush the file stream buffer after writing every flag string. There is a 32-bit debugger provided with the compiler, but it tends to be unstable and was generally found to be useless in most cases, with the exception of when debugging 32-bit DPML code when launching the debugger from DOS (and not a DOS window).

ROCGEN.CPP contains the main function, WinMain(...). (A console-mode version would use the more familiar main(...) function.) This function is executed when the program is launched. In this version, it calls the functions DProb1DAtLeast2Of5(...) and

DProb1DAtLeast2Of5MC(...) to compute analytical and Monte Carlo results for  $P_F$  and  $P_D$  in the multiple-scan coincidence case. The functions DProb1DAtLeast1Of5(...) and DProb1DAtLeast1Of5MC(...) can replace them to get results for the probability of getting at least one threshold-crossing event in a sequence of samples. These are computed for a sequence of threshold values and stored in arrays. When the values have been calculated, the function prints the arrays in the form of  $(P_F, P_D)$  pairs to the file ROCDATA.OUT.

### ROCGEN.CPP

```

/* The main function for the ROC-plot generation code. All physical quantities
   are given in MKS units.
   Physical parameters are intended to roughly reflect those of the ASU instrument.
*/

```

```

#include <stdlib.h>
#include <stdio.h>
#include <windows.h>
#include <math.h>
#include <malloc.h>
#include "modlfns.h" // Model function prototypes
#include "imsl.h"    // IMSL header file
#include "imsls.h"  // IMSL stat. functions header file
#include "data.h"   // Model data type definitions

DWORD WhichException(DWORD code, FILE *efile);
int fmthout(float x, FILE *stream);

#define PLANCK 6.6256E-34 // Planck's constant
#define CLIGHT 3.0E8 // Speed of light

int PASCAL WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpszCmdLine,
                  int nCmdShow)
{
    struct SURFACE sfSurf; // Surface data
    struct BEAM bmBeam; // Beam data
    struct DETECTOR dr; // Detector data

```

```

FILE *file,*datafile;
float quantum,IntSurfScatt,max,thresh;

float MaxScatter;

int NumThresh=3;
float *NoisePDF;

// Physical parameters
bmBeam.fWavelength=.635E-6; /* HeNe */
bmBeam.fPower=.01; /* 10 mW */
bmBeam.fWidth=.000020; // 20 um
bmBeam.fPitch=.000010; // 10 um
sfSurf.fRMSRough=10.E-9; /* 10 A roughness -- not used currently */
dr.fQEfficiency;

float EPhoton=PLANCK*CLIGHT/bmBeam.fWavelength;

// DATEL ADC-EH8B2 -- estimated
float fSampTime=.000050;

int Dim=100; // PDF dimensions

float step;
int count;

// Start an exception block -- any exception in this block should
// trigger the following exception handler. Doesn't always work.
try
{
    // Open an output file for the ROC data points
    datafile=fopen("rocddata.out","w");
    // Open a log file to record progress
    file=fopen("rocgen.log","w");
    if(file== NULL || datafile==NULL)
    {
        fputs("Could not open all output files.\n",file);
        return 0;
    }

    // Used by the shot noise computations
    quantum=EPhoton/fSampTime;

    IntSurfScatt=3.75E-7; // Integrated surface scatter power, in W
                        // ring 21, Si3N4 .204 um
    MaxScatter=1.07E-7*1.; // Integrated particle scatter power, ring 21
    fprintf(file,"Max Scatter photon count = %f\n",MaxScatter/quantum);

```

```

max=IntSurfScatt*8.; // Maximum domain value considered
step=max/Dim; // increment size in the power domain

NoisePDF=(float *)malloc(Dim*sizeof (float));
if(NoisePDF==NULL)
{
    fputs("Malloc failed!\n",file);
    return -1;
}

// Use gamma distribute noise with mean = IntSurfScatt. Variance is
// determined by the shape parameter, which is just a guess in this case.
for(count=0;count<Dim;count++)
{
    NoisePDF[count]=GammaPDF(step*count,IntSurfScatt,2.);
}
}
except(1)
{
    fputs("Exception raised during initialization\n",file);
    WhichException(GetExceptionCode(),file);
}

try
{
    int CountThresh;
    int lim=2;
    float *pf;
    float *pdm,*pda;
    float ThreshStep=IntSurfScatt*2./NumThresh;

    // allocate storage for result arrays
    pf=(float *)malloc(NumThresh*sizeof(float));
    pda=(float *)malloc(NumThresh*sizeof(float));
    pdm=(float *)malloc(NumThresh*sizeof(float));

    // Find limit such that illumination out to .01 IMax is considered
    // significant
    // lim=(int)ceil(abs(sqrt(log(100.)/2.)*bmBeam.fWidth/bmBeam.fPitch-.5));

    fputs("Beginning threshold iteration:\n",file);
    fprintf(file,"Illumination limit = %d\n",lim);
    for(CountThresh=0,thresh=0.;CountThresh<NumThresh;
        CountThresh++,thresh+=ThreshStep)
    {
        // Find false count probability

```

```

pf[CountThresh]=DProb1DAtLeast2Of5(Dim, quantum, step, NoisePDF,
bmBeam.fPitch, bmBeam.fPitch, bmBeam.fWidth, 0., thresh, 0,file);

// Do Analytical for getting thresholds on two adjacent scans
pda[CountThresh]=DProb1DAtLeast2Of5(Dim, quantum, step,
NoisePDF, bmBeam.fPitch, bmBeam.fPitch, bmBeam.fWidth,
0., thresh, lim,file);

// Do Monte Carlo for getting thresholds on two adjacent scans
pdm[CountThresh]=DProb1DAtLeast2Of5MC(quantum,IntSurfScatt,2.0,
bmBeam.fPitch, bmBeam.fWidth, 0.*MaxScatter, thresh, lim,file);
fprintf(file,"%d.",CountThresh);
}

// Output results to output file. Use Mathematica compatible
// format for easy cut-and-paste.

fputc('\n',datafile);
// Output analytical results
fputc('{',datafile);
for(CountThresh=NumThresh-1;CountThresh>=0;CountThresh--)
{
    fprintf(datafile,"%c\n{",(CountThresh==(NumThresh-1))?' ':'');
    fmthout(pf[CountThresh], datafile);
    fputc(',',datafile);
    fmthout(pda[CountThresh], datafile);
    fputc('}',datafile);
}
fputc('}',datafile);

// Output Monte Carlo results
fputc('\n',datafile);
fputc('{',datafile);
for(CountThresh=NumThresh-1;CountThresh>=0;CountThresh--)
{
    fprintf(datafile,"%c\n{",(CountThresh==(NumThresh-1))?' ':'');
    fmthout(pf[CountThresh], datafile);
    fputc(',',datafile);
    fmthout(pdm[CountThresh], datafile);
    fputc('}',datafile);
}
fputc('}',datafile);
}
except(1)
{
    fputs("Exception raised while finding ROC\n",file);
    WhichException(GetExceptionCode(),file);
}

```

```

        fprintf(file, "\nIMSL ERR: %d\n", imsl_error_code());
    }

    return 0;
}

// return square of x
float sqr(float x)
{
    return x*x;
}

// Identify exception code and write description to the stream efile
DWORD WhichException(DWORD code, FILE *efile)
{
    switch(code)
    {
        case EXCEPTION_ACCESS_VIOLATION:
            fputs("The thread tried to read from or write to a virtual address for which it
does not have the appropriate access.",
                efile);
            break;
        case EXCEPTION_BREAKPOINT:
            fputs("A breakpoint was encountered.", efile);
            break;

        case EXCEPTION_DATATYPE_MISALIGNMENT:
            fputs("The thread tried to read or write data that is misaligned on hardware that
does not provide alignment. For example, 16-bit values must be aligned on 2-byte
boundaries; 32-bit values on 4-byte boundaries, and so on.",
                efile);
            break;
        case EXCEPTION_SINGLE_STEP:
            fputs("A trace trap or other single-instruction mechanism signaled that one
instruction has been executed.",
                efile);
            break;
        case EXCEPTION_ARRAY_BOUNDS_EXCEEDED:
            fputs("The thread tried to access an array element that is out of bounds and the
underlying hardware supports bounds checking.",
                efile);
            break;
        case EXCEPTION_FLT_DENORMAL_OPERAND:
            fputs("One of the operands in a floating-point operation is denormal. A
denormal value is one that is too small to represent as a standard floating-point value.",
                efile);
            break;
    }
}

```

```
    case EXCEPTION_FLT_DIVIDE_BY_ZERO:
        fputs("The thread tried to divide a floating-point value by a floating-point
divisor of zero.",
            efile);
        break;

    case EXCEPTION_FLT_INEXACT_RESULT:
        fputs("The result of a floating-point operation cannot be represented exactly as
a decimal fraction.",
            efile);
        break;

    case EXCEPTION_FLT_INVALID_OPERATION:
        fputs("This exception represents any floating-point exception not included in
this list.",
            efile);
        break;
    case EXCEPTION_FLT_OVERFLOW:
        fputs("The exponent of a floating-point operation is greater than the magnitude
allowed by the corresponding type.",
            efile);
        break;

    case EXCEPTION_FLT_STACK_CHECK:
        fputs("The stack overflowed or underflowed as the result of a floating-point
operation.",
            efile);
        break;
    case EXCEPTION_FLT_UNDERFLOW:
        fputs("The exponent of a floating-point operation is less than the magnitude
allowed by the corresponding type.",
            efile);
        break;
    case EXCEPTION_INT_DIVIDE_BY_ZERO:
        fputs("The thread tried to divide an integer value by an integer divisor of zero.",
            efile);
        break;
    case EXCEPTION_INT_OVERFLOW:
        fputs("The result of an integer operation caused a carry out of the most
significant bit of the result.",
            efile);
        break;

    case EXCEPTION_PRIV_INSTRUCTION:
        fputs("The thread tried to execute an instruction whose operation is not allowed
in the current machine mode.",
```

```

        efile);
        break;

    case EXCEPTION_NONCONTINUABLE_EXCEPTION:
        fputs("The thread tried to continue execution after a noncontinuable exception
occurred.",
            efile);
        break;
    }
    fputc('\n',efile);

    return code;
}

```

```

// Special function to write a float in Mathematica-style scientific
// notation to stream.

```

```

int fmthout(float x, FILE *stream)

```

```

{
    char buf[80];

    sprintf(buf,"%g",x);
    if(strchr(buf,'e')!=NULL)
    {
        strtok(buf,"e");
        fprintf(stream,"%s 10^%s",buf,strtok(NULL,"\\0"));
    }
    else
        fputs(buf,stream);

    return 0;
}

```

```

// Compute gamma distribution PDF at x

```

```

float GammaPDF(float x, float mean, float shape)

```

```

{
    if(x<0.0) return 0.0;    // Handle negative x
    if(shape==1.0)          // Shape==1 is the exponential distn.
        return 1./exp(x/mean)/mean;
    return (pow(shape/mean,shape)*pow(x,shape-1)*exp(-shape*x/mean)/
        imsls_f_gamma(shape));
}

```

```

(END OF ROCCGEN.CPP)

```

```

CROSSSCN.CPP

```

```

/* Code for cross-scan coincidence */

```

```
#include <stdlib.h>
#include <stdio.h>
#include <alloc.h>
#include <imsl.h>
#include <imsls.h>
#include <math.h>
#include "modlfn.h"
```

```
#define TRUE -1
#define FALSE 0
```

```
/* Compute detection probability for the case in which we want at least one
   threshold event in a series of scans. One-dim. randomness.
   This version accounts for additive optical noise and shot noise.
   Later noise sources could be added though not easily.
```

N.B. The arrangement and choice of parameters reflects that used in prior functions of the same type which were used in non-multiple-scan calculations.

Parameters:

Dim -- dimension (size) of the NoisePDF array  
 quantum -- the divisor used to divide the signal power to get the number of photons received during the sample  
 step -- the size of the signal power increment determined by the size of the PDF array and the maximum value considered  
 \*NoisePDF -- pointer to the NoisePDF array, given in terms of power. This represents optical noise only.  
 ScanPitch -- the scan pitch in meters  
 res -- the desired resolution of the integral over scan phase. This is a simple finite difference method. Value should be less than 1/20 of the scan pitch to get good results  
 BeamWidth -- The  $1/e^2$  half-width of the beam in meters  
 IMax -- the maximum power scattered by the particle in watts  
 thresh -- the threshold in equivalent watts  
 nlim -- the number of scan regions out from the central region to be considered as having non-negligible illumination. Zero indicates only the central region. One indicates the three innermost regions.  
 outfile -- pointer to the FILE object for the log file stream

False count probability may be obtained by setting IMax to 0.0 (no particle scatter signal) and nlim to 0.

This routine has proven to give results very close to those from the Monte Carlo routine below for corresponding inputs.

```
*/
```

```

float DProb1DAtLeast1Of5(int Dim, float quantum,
    float step, float *NoisePDF, float ScanPitch, float res,
    float BeamWidth, float IMax, float thresh, int nlim, FILE *outfile)
{
    float pm,pd; // miss and detection probabilities
    float *xdata,*pmarray;
    float ScanPhase,ScanPos,Ioff;
    int count;
    Imsl_f_spline *spline=NULL;
    float lambda;
    int Lambda100;

    // allocate arrays
    xdata=(float *)malloc(sizeof(float)*Dim);
    pmarray=(float *)malloc(sizeof(float)*Dim);

    if(xdata==NULL)
    {
        fprintf(outfile,"Malloc failed in DProb1DAtLeast1In5\n");
        return 0.0;
    }

    pd=0;

    // loop through scan phase values
    for(ScanPhase=-ScanPitch/2.;ScanPhase<ScanPitch/2.;ScanPhase+=res)
    {
        pm=1.; // Initialize miss probability

        // loop through scan regions
        for(ScanPos=-nlim*ScanPitch;ScanPos<=(nlim*ScanPitch);
            ScanPos+=ScanPitch)
        {
            // Compute scattered power for this ScanPhase value and
            // scan region (ScanPos)
            Ioff=IMax*exp(-2.*
                abs((ScanPhase+ScanPos)*(ScanPhase+ScanPos))
                /BeamWidth/BeamWidth);

            // Compute value of Poisson lambda parameter at which to
            // start using normal approximation
            Lambda100=min(max((int)ceil((100.*quantum-Ioff)/step),1),Dim);

            xdata[0]=Ioff;
            pmarray[0]=0.;
        }
    }
}

```

```

// Get contributions with lambda < 100 using Poisson distribution
for(count=1;count<Lambda100;count++)
{
    xdata[count]=count*step+Ioff;

    // Get miss probability for this lambda at this photon count
    // Multiply by the probability of this lambda
    pmarray[count]=NoisePDF[count]*
        imsls_f_poisson_cdf(floor(thresh/quantum),
            (step*count+Ioff)/quantum);
}
// Get contributions with lambda >= 100 using Normal distribution
// mean=lambda, var=lambda
for(count=Lambda100;count<Dim;count++)
{
    xdata[count]=count*step+Ioff;

    // compute lambda
    lambda=(step*count+Ioff)/quantum;
    if(lambda==0.0) continue; // Avoid delta distn.

    // Get miss probability for this lambda at this photon count
    // Multiply by the probability of this lambda
    pmarray[count]=NoisePDF[count]*
        imsls_f_normal_cdf((floor(thresh/quantum)-lambda)/sqrt(lambda));
}

// Do a spline of the pm data
spline=imsl_f_spline_interp(Dim, xdata, pmarray, 0);
// Integrate the pm data spline, to improve the accuracy
// of integration. Probably not necessary.
// Multiply by pm accumulator -- this ANDs the probabilities
// for each scan region.
pm*=imsl_f_spline_integral(Ioff,(Dim-1)*step,spline);
}
// Add up (as an approx. to integration) the pds for each
// ScanPhase value.
pd+=1.-pm;
}
// Normalize the pd
pd/=floor(ScanPitch/res);

// Deallocate storage
free(xdata);
free(pmarray);

```

```

    return pd; // Return the result for pd
}

/* Simulate detection using the Monte Carlo method to get the
   probability for the case in which we want at least one
   threshold event in a series of scans. One-dim. randomness.
   This version accounts for additive optical noise and shot noise.
   Later noise sources could be added fairly easily.

   N.B. The arrangement and choice of parameters reflects that used in
   prior functions of the same type which were used in non-multiple-scan
   calculations.

Parameters:
  quantum -- the divisor used to divide the signal power to get the
             number of photons received during the sample
  GMean -- the mean of the gamma distributed spatial noise
  GShape -- the shape parameter of the gamma distributed spatial noise

  ScanPitch -- the scan pitch in meters
  BeamWidth -- The  $1/e^2$  half-width of the beam in meters
  IMax -- the maximum power scattered by the particle in watts
  thresh -- the threshold in equivalent watts
  nlim -- the number of scan regions out from the central region to
          be considered as having non-negligible illumination. Zero
          indicates only the central region. One indicates the three
          innermost regions.
  outfile -- pointer to the FILE object for the log file stream

False count probability may be obtained by setting IMax to 0.0
(no particle scatter signal) and nlim to 0.

For some unknown reason, it has been found that my particular IMSL
library will crash if the simulation loop in this function executes
more than roughly 500,000 times total during the execution of the
program. This limit seems to be independent of the number
of times this function is called, hence the suspicion of one of the
IMSL functions in the simulation loop. All attempts to circumvent this
limitation have failed.
*/

float DProb1DAAtLeast1Of5MC(float quantum, float GMean, float GShape,
float ScanPitch, float BeamWidth, float IMax, float thresh, int nlim,
FILE *outfile)
{
    long itr, MissCount=0, TotItr=10000;
    int isHit, *IPProc;

```

```

float *ScanPhase,ScanPos,Ioff,*GNoise,pd,signal,*fPProc,lambda;

imsls_random_seed_set(123457);
imsls_random_option(5);

fprintf(outfile,"Doing Monte Carlo\n");
fflush(outfile);

// Loop through simulations. TotItr sets the number
for (itr=0;itr<TotItr;itr++)
{
    // get random scan phase
    ScanPhase=imsls_f_random_uniform(1,0);
    ScanPhase[0]*=ScanPitch;
    ScanPhase[0]-=ScanPitch/2.;

    isHit=FALSE;

    // loop through scan regions
    for(ScanPos=-nlim*ScanPitch;ScanPos<=(nlim*ScanPitch);
        ScanPos+=ScanPitch)
    {
        Ioff=IMax*exp(-2.*
            abs((ScanPhase[0]+ScanPos)*(ScanPhase[0]+ScanPos))
            /BeamWidth/BeamWidth);

        // Simulate spatial noise as gamma distributed with specd. params.
        GNoise=imsls_f_random_gamma(1,GShape,0);
        // Multiply by scale param. to get 2-prm. gamma
        GNoise[0]*=GMean/GShape;

        // Do shot noise using Poisson distn. up to lambda<100
        if((lambda=(GNoise[0]+Ioff)/quantum)<100.)
        {
            iPProc=imsls_random_poisson(1,lambda,0);

            signal=quantum*(float) iPProc[0];
        }
        else // Use normal for large mean -- shot noise is probably not
            // very noticeable when the signal is this strong, however.
        {
            fPProc=imsls_f_random_normal(1,IMSLS_MEAN,lambda,
                IMSLS_VARIANCE,lambda,0);

            signal=quantum*fPProc[0];
        }
    }
}

```

```

        // Do threshold test, set hit flag if passed
        if(signal>thresh)
            isHit=TRUE;
    }
    // If we have not set the hit flag at least once, count as a miss
    if(!isHit)
        MissCount++;
}

// Caculate estimated pd from miss count
pd=(TotItr-MissCount)/(float)TotItr;

return pd;
}

```

(END OF CROSSCN.CPP)

DBLCOINC.CPP

```

/* Code for cross-scan coincidence */
#include <stdlib.h>
#include <stdio.h>
#include <alloc.h>
#include <imsl.h>
#include <imsls.h>
#include <math.h>
#include "modfns.h" // Prototypes

#define TRUE -1
#define FALSE 0

/* Compute detection probability for the case in which we want at least two
consecutive threshold events in a series of scans. One-dim. randomness.
This version accounts for additive optical noise and shot noise.
Later noise sources could be added though not easily.

```

N.B. The arrangement and choice of parameters reflects that used in prior functions of the same type which were used in non-multiple-scan calculations.

Parameters:

Dim -- dimension (size) of the NoisePDF array  
quantum -- the divisor used to divide the signal power to get the number of photons received during the sample  
step -- the size of the signal power increment determined by the size of the PDF array and the maximum value considered

\*NoisePDF -- pointer to the NoisePDF array, given in terms of power.

This represents optical noise only.

ScanPitch -- the scan pitch in meters

res -- the desired resolution of the integral over scan phase. This is a simple finite difference method. Value should be less than  $1/20$  of the scan pitch to get good results

BeamWidth -- The  $1/e^2$  half-width of the beam in meters

IMax -- the maximum power scattered by the particle in watts

thresh -- the threshold in equivalent watts

nlim -- the number of scan regions out from the central region to be considered as having non-negligible illumination. Zero indicates only the central region. One indicates the three innermost regions.

outfile -- pointer to the FILE object for the log file stream

False count probability may be obtained by setting IMax to 0.0 (no particle scatter signal) and nlim to 0.

This routine has proven to give results very close to those from the Monte Carlo routine below for corresponding inputs.

\*/

```
float DProb1DAtLeast2Of5(int Dim, float quantum,
    float step, float *NoisePDF, float ScanPitch, float res,
    float BeamWidth, float IMax, float thresh, int nlim, FILE *outfile)
{
    float *pds,pd,pdtot;
    float *xdata,*pdarray;
    float ScanPhase,ScanPos,Ioff;
    int count,nscan;
    Imsl_f_spline *spline=NULL;

    float lambda;
    int Lambda100;

    // allocate arrays

    xdata=(float *)malloc(sizeof(float)*Dim);
    pdarray=(float *)malloc(sizeof(float)*Dim);
    pds=(float *)malloc(sizeof(float)*(2*nlim+1));

    if(xdata==NULL)
    {
        fprintf(outfile,"Malloc failed in DProb1DAtLeast2In5\n");
        return 0.0;
    }
    pdtot=0;
```

```

// loop though scan phase values
for(ScanPhase=-ScanPitch/2.;ScanPhase<ScanPitch/2.;ScanPhase+=res)
{
    // loop through scan regions
    for(ScanPos=-nlim*ScanPitch,nscan=0;ScanPos<=(nlim*ScanPitch);
        ScanPos+=ScanPitch,nscan++)
    {
        // Compute scattered power for this ScanPhase value and
        // scan region (ScanPos)
        Ioff=IMax*exp(2.*abs((ScanPhase+ScanPos)*(ScanPhase+ScanPos))
            /BeamWidth/BeamWidth);

        // Compute value of Poisson lambda parameter at which to
        // start using normal approximation
        Lambda100=min(max((int)ceil((100.*quantum-Ioff)/step),1),Dim);

        xdata[0]=Ioff;
        pdarray[0]=0.;

        // Get contributions with lambda < 100 using Poisson distribution
        for(count=1;count<Lambda100;count++)
        {
            xdata[count]=count*step+Ioff;

            // Get detection probability for this lambda at this photon
            // count. Multiply by the probability of this lambda
            pdarray[count]=NoisePDF[count]*(1.-
                imsls_f_poisson_cdf(floor(thresh/quantum),(step*count+Ioff)/quantum));
        }

        // Get contributions with lambda >= 100 using Normal distribution
        // mean=lambda, var=lambda
        for(count=Lambda100;count<Dim;count++)
        {
            xdata[count]=count*step+Ioff;

            // compute lambda
            lambda=(step*count+Ioff)/quantum;
            // Get miss probability for this lambda at this photon count
            // Multiply by the probability of this lambda
            if(lambda==0.0)
            {
                if(floor(thresh/quantum)==0.)
                    pdarray[count]=1.;
                else
                    pdarray[count]=0.;
                continue;
            }
        }
    }
}

```

```

    }
    else
    {
        pdarray[count]=NoisePDF[count]*(1-
            imsls_f_normal_cdf((floor(thresh/quantum)-lambda)/
                sqrt(lambda)));
    }
}
// Do a spline of the pd data
spline=imsl_f_spline_interp(Dim, xdata, pdarray, 0);
// Store pd for this sample region
pds[nscan]=imsl_f_spline_integral(Ioff,step*(Dim-1),spline);
}

// Call function to evaluate coincidence probabilities given the
// array of Pds.
pd=coincprobs(nlim*2+1,pds);
// Do Pf calculaton of nlim is 0
if(!nlim)
    pd=pds[0]*pds[0];
pdtot+=pd;
}
pdtot/=floor(ScanPitch/res);

free(xdata);
free(pdarray);
free(pds);

return pdtot;
}

```

/\* Simulate detection using the Monte Carlo method to get the probability for the case in which we want at least two consecutive threshold events in a series of scans. One-dim. randomness. This version accounts for additive optical noise and shot noise. Later noise sources could be added fairly easily.

N.B. The arrangement and choice of parameters reflects that used in prior functions of the same type which were used in non-multiple-scan calculations.

Parameters:

quantum -- the divisor used to divide the signal power to get the number of photons received during the sample  
GMean -- the mean of the gamma distributed spatial noise  
GShape -- the shape parameter of the gamma distributed spatial noise

ScanPitch -- the scan pitch in meters  
 BeamWidth -- The  $1/e^2$  half-width of the beam in meters  
 IMax -- the maximum power scattered by the particle in watts  
 thresh -- the threshold in equivalent watts  
 nlim -- the number of scan regions out from the central region to  
     be considered as having non-negligible illumination. Zero  
     indicates only the central region. One indicates the three  
     innermost regions.  
 outfile -- pointer to the FILE object for the log file stream

False count probability may be obtained by setting IMax to 0.0 (no particle scatter signal) and nlim to 0.

For some unknown reason, it has been found that my particular IMSL library will crash if the simulation loop in this function executes more than roughly 500,000 times total during the execution of the program. This limit seems to be independent of the number of times this function is called, hence the suspicion of one of the IMSL functions in the simulation loop. All attempts to circumvent this limitation have failed.

\*/

```

float DProb1DAtLeast2Of5MC(float quantum, float GMean, float GShape,
    float ScanPitch, float BeamWidth, float IMax, float thresh, int nlim,
    FILE *outfile)
{
    long itr, DetectionCount=0;
    long TotItr=10000; // No. of iterations
    int isHit, *iPProc, *hits, nscan;
    float *ScanPhase, ScanPos, Ioff, *GNoise, pd, signal, *fPProc, lambda;
    int GetFalse=FALSE;

    if(!nlim) // Flag if we are trying to get Pf
    {
        GetFalse=TRUE;
        nlim=1;
    }

    // IMSL configuration functions. Set seed and congruential generator number
    imsls_random_seed_set(345712);
    imsls_random_option(3);

    fprintf(outfile, "Doing Monte Carlo\n");
    fflush(outfile);

    hits=new int[2*nlim+1]; //allocate an array to store individual
  
```

```

// sample decisions

// Loop through simulations. TotItr sets the number
for (itr=0;itr<TotItr;itr++)
{
    // get random scan phase
    ScanPhase=imsls_f_random_uniform(1,0);
    ScanPhase[0]*=ScanPitch; // scale so that 0<ScanPhase<ScanPitch
    ScanPhase[0]-=ScanPitch/2.; // shift so |ScanPhase|<ScanPitch/2

    isHit=FALSE;

    // loop through scan regions
    for(ScanPos=-nlim*ScanPitch,nscan=0;ScanPos<=(nlim*ScanPitch);
        ScanPos+=ScanPitch,nscan++)
    {
        Ioff=IMax*exp(-2.*abs((ScanPhase[0]+ScanPos)*(ScanPhase[0]+ScanPos))
            /BeamWidth/BeamWidth);

        // Simulate spatial noise as gamma distributed with specd. params.
        GNoise=imsls_f_random_gamma(1,GShape,0);
        // Multiply by scale param. to get 2-prm. gamma
        GNoise[0]*=GMean/GShape; // Multiply by scale param. to get 2-prm. gamma

        // Do shot noise using Poisson distn. up to lambda<100
        if((lambda=(GNoise[0]+Ioff)/quantum)<100.)
        {
            iPProc=imsls_random_poisson(1,lambda,0);

            signal=quantum*(float) iPProc[0];
        }
        else // Use normal for large mean -- shot noise is probably not
            // noticeable when the signal is this strong, however.
        {
            fPProc=imsls_f_random_normal(1,IMSLS_MEAN,lambda,
                IMSLS_VARIANCE, lambda,0);

            signal=quantum*fPProc[0];
        }
        // Do threshold test, set hit flag if passed, clear if not
        if(signal>thresh)
            hits[nscan]=TRUE;
        else
            hits[nscan]=FALSE;
    }
}

```

```

// Run coincidence test -- are consecutive hit entries both TRUE?
// Set isHit if yes.
if(!GetFalse)
{
    for(nscan=1;nscan<(nlim*2+1);nscan++)
        if(hits[nscan]&&hits[nscan-1])
            isHit=TRUE;
}
// Use different algorithm for false count -- uses first two scans
else
    if(hits[1]&&hits[0])
        isHit=TRUE;

    if(isHit) // Count a hit every time we set isHit
        DetectionCount++;
}
// Find detection rate
pd=DetectionCount/(float)TotItr;

delete hits;

return pd;
}

// Extremely slow function for computing Pd for coincidence of two
// consecutive samples, with tot samples and parray containing
// individual threshold-crossing probabilities for each sample.

float coinceprobs(int tot, float *parray)
{
    float mult,add=0.;
    int acount,ccount,isHit,pair,plist;

    // Fast Pd caculation for tot==5. See appendix III. Over 10 x faster.
    if(tot==5)
    {
        float p12=parray[0]*parray[1],p45=parray[3]*parray[4];

        add=(parray[1]+parray[3])*parray[2]+p12+p45;
        add-=parray[2]*(p12+parray[1]*parray[3]+p45);
        add-=p12*p45;
        add+=p12*p45*parray[2];

        return add;
    }

    // Count up to 2^tot, giving all possible combinations of

```

```

// tot bits
for(account=0;account<(1<<tot);account++)
{
    isHit=0;
    mult=1.;
    // compare bit-pairs, looking for pairs with have both bits
    // turned on, which would indicate detection. Set isHit
    // if we find one.
    for(ccount=1,pair=0x3;ccount<tot;ccount++,pair<<=1)
        if(pair&account)
        {
            isHit=-1;
            break;
        }

    if(isHit)
    {
        // For detection outcome, multiply by Pd for the sample
        // for each on bit, and multiply by 1-Pd for each off bit
        // in order to find the probability for this outcome.
        for(plist=account,ccount=0;ccount<tot;plist>>=1)
        {
            if(0x1&plist) mult*=parray[ccount];
            else mult*=1-parray[ccount];
        }
        add+=mult;
    }
    return add;
}

```

(END OF DBLCOINC.CPP)

**APPENDIX II**  
**INTEGRATION OVER THE SCATTERING SPHERE**

Scattering problems frequently tend to involve functions of spherical coordinates, and particle scattering from surfaces is no exception. To accurately model a real scanner, we must be able to find how much light falls on the instruments detectors by way of its collection optics using scattering data which is normally expressed in that system. Clearly, this is an integration problem and almost certainly a numerical integration problem, given the often complicated variation of scattering cross section with angle. However, most widely available numerical integration packages fail to implement functions for such non-Cartesian problems and many texts on the subject fail to even mention it.

Thus, after some study, it was determined that an apt approach would be to transform the coordinates to a Cartesian plane in order to allow use of one of the many sophisticated pre-packaged integration routines available, while avoiding the sometimes questionable approximation of the sphere to a flat surface which is often used for small apertures. The type of transformation required is what is known as an equal-area or area-preserving transform. These transforms have the property that the ratio of the areas of two closed forms in the original coordinate system will be the same after the transformation, or equivalently that the differential areas in the two systems differ only by a constant. This ensures that the solid angle we want to integrate over subsumes exactly the same irradiance regions in the Cartesian plane as its corresponding area.

Surprisingly, area-preserving transforms are not especially well-covered in works on calculus and analytical geometry, but it was found that these types of problems are thoroughly discussed and analyzed in cartography, where they are used for map projections. In fact, there are quite a few such transforms that have been in use for centuries, and this wealth of possible transforms gives us the ability to choose a transform that is particularly suitable for

the collection geometry of interest. It is well known that complicated boundaries of integration are often a serious impediment to numerical integration, so any method of reducing their complexity is a significant advantage.

The ASU scanning instrument uses what is known as a ring-wedge detector, which is composed of a half circle of concentric rings and a half circle of wedges radiating out from the center of the detector, giving it a considerable amount of radial symmetry. In normal operation, this detector is placed such that its surface is normal to an arbitrary scattering ray from the illuminated point of the wafer surface. In this configuration, if we place the pole of the spherical coordinate system at the center of the detector, the wedge boundaries fall along circles of constant phi (meridians) and the ring boundaries fall along lines of constant theta (parallels). Thus, if we choose the right transformation, we might preserve this boundary simplicity and be able to integrate over constant boundaries.

Fortunately, Pearson treats a simple map projection which gives us area preservation, straight meridians, and straight parallels of latitude: the polar cylindrical projection<sup>21</sup>.

Performing the trivial conversion from Pearson's latitude-longitude representation to phi-theta spherical coordinates, the transformation is

$$x = R S \cos q \quad \text{Eqn. 24}$$

$$y = R f \quad \text{Eqn. 25}$$

where R is the radius and S a arbitrary constant scale factor. It is immediately obvious that the differential area dA for the x-y coordinate system is

$$dA = dx dy = -R^2 S \sin q dq df \quad \text{Eqn. 26}$$

which differs from the differential surface area of the phi-theta sphere only by the scale factor  $S$ , negated. This shows the desired property of area-preservation.

Of course, to get the pole at the correct point relative to the detector, a rotation of the original theta-phi coordinate may be required, which can be more difficult than the projection itself due to the problems associated with ensuring the quadrants are mapped properly.

This rotation is somewhat more difficult than the projection itself, but it can similarly be found in Pearson or similar works on cartography.

## **APPENDIX III**

### **CALCULATING $P_D$ FOR COINCIDENCE OF ADJACENT SAMPLES**

To calculate  $P_D$  for the double coincidence of adjacent samples in one dimension, we first recognize that we have a random vector  $\mathbf{C}$  of  $n$  independent random variables  $S_i$ , one for each sample region, which can take binary true-false values. Each of these indicates whether its corresponding sample had a large enough signal to cross the test threshold. Since we indicate a detection when any two adjacent samples are true, we want to find the probability of the set of possible joint outcomes that have at least one set of adjacent true outcomes, for example

$$\mathbf{C}=[S_1 S_2 \dots S_n] = [1 1 \dots 0]. \quad \text{Eqn. 27}$$

With  $n$  variables that can take one of two values, we have  $2^n$  possible outcomes. Since the  $S_i$ s are assumed independent, the probability of each outcome,  $P_c$ , is the product of the probabilities  $P_i$  for the given outcome of each component variable,

$$P_c=P(\mathbf{C}=[s_1 s_2 \dots s_n])=P(S_1=s_1)P(S_2=s_2) \dots P(S_n=s_n). \quad \text{Eqn. 28}$$

To get the total probability of coincidence, we could then add the probabilities of each  $n$ -fold outcome (which are disjoint) that meets the coincidence criterion. This is the approach used by the slow general method implemented in the code. However, there are many pairs of complementary probabilities which are uselessly computed and then added in this approach. For example, when  $n = 3$ , we have the following 8 possible outcomes, with detection cases underlined:

$$[\underline{1 1 0}], [0 1 0], [1 0 0], [0 0 0], [\underline{1 1 1}], [\underline{0 1 1}], [1 0 1], [0 0 1].$$

Adding probabilities for separate detection outcomes, we get

$$P_D=P_1P_2(1-P_3)+ P_1P_2P_3+(1-P_1)P_2P_3, \quad \text{Eqn. 29}$$

and we can easily see that the first two additive terms reduce to  $P_1P_2$ . For three samples, this is not a large reduction in multiplications, but for larger numbers it becomes quite significant. For five, the former approach requires 76 (4 times 19) floating-point multiplications for the 19 out of 32 outcomes that give detections, but the method seen in the code using eliminations and distributive properties reduces this to 10 multiplications.

A systematic method has been devised to determine the  $P_D$  formula for various  $n$ , which naturally accounts for many of these eliminations.

The procedure is to begin by writing the vector with the first two variables true and the rest indeterminate, corresponding to a set of outcomes:

$$[1 \ 1 \ x \ x \ \dots \ x].$$

The probability of having 1 and 2 true with the rest indeterminate is just  $P_{12}=P_1P_2$ .

Next we add the probability for the set of outcomes where 1 is false and 2 and 3 are true,

$$[0 \ 1 \ 1 \ x \ \dots \ x],$$

where

$$P_{23}=(1-P_1)P_2P_3. \tag{Eqn. 30}$$

Note that this set of outcomes is disjoint from the first due to differing values for 1, ensuring additivity of the probabilities.

For the next,

$$[x \ 0 \ 1 \ 1 \ x \ \dots \ x],$$

covers the cases where 3 and 4 are true but is disjoint from the first two sets so that

$$P_{34}=(1-P_2)P_3P_4.$$

Succeeding cases become more complicated. We must count all cases except those which have been counted already, which we achieve by counting only cases in which the first sample preceding the true pair is false and in which none of the cases preceding it give true pairs, because all true pairs which would occur there have been counted already.

Thus the next true pair , 4-5, counts the outcomes

$$[0 \ x \ 0 \ 1 \ 1 \ x \ \dots \ x] \ (P_{45a}=(1-P_1)(1-P_3)P_4P_5)$$

and

$$[1 \ 0 \ 0 \ 1 \ 1 \ x \ \dots \ x], \ (P_{45b}=P_1(1-P_2)(1-P_3)P_4P_5)$$

but not

$$[1 \ 1 \ 0 \ 1 \ 1 \ x \ \dots \ x],$$

which has a true pair before the currently counted true pair, thus being a previously counted outcome.

This continues until the last set of outcomes with the  $n-1$  and  $n$  cases being true are counted.

Thus with five samples we add up the probabilities for the outcomes of the first four true pairs 1-2, 2-3, 3-4, and 4-5 to get

$$P_D=P_{12}+P_{23}+P_{34}+(P_{45a}+P_{45b}) \tag{Eqn. 32}$$

or

$$P_D=P_1P_2+(1-P_1)P_2P_3+(1-P_2)P_3P_4+[(1-P_1)(1-P_3)P_4P_5]+ P_1(1-P_2)(1-P_3)P_4P_5] \tag{Eqn. 33.}$$

With some algebra, this rearranges to

$$P_D = P_1P_2 + P_2P_3 + P_3P_4 + P_4P_5 - P_1P_2P_3 - P_2P_3P_4 - P_3P_4P_5 - P_1P_2P_4P_5 + P_1P_2P_3P_4P_5 \quad \text{Eqn. 34.}$$

Storage of intermediate results and the application of distributivity as used in the code can reduce these 17 multiplications, noting that the previous form itself has only 12.

Automation of this procedure could yield a faster method for handling general  $n$  samples but the motivation for this has not so far been sufficient to warrant coding the algorithm since only a few cases have been used in the present computations.

## BIOGRAPHICAL SKETCH

Benjamin Duaine Buckner was born on the banks of the Great Miami River in Dayton, Ohio in 1968. Thereafter, he graduated with a Bachelor of Science degree in physics and mathematics from Morehead State University in the forested expanse of Eastern Kentucky at the end of 1990 and then worked for a year as a computer programmer with the Kurta Corporation in Phoenix, Arizona. In 1993, after several misadventures, he undertook graduate study at Arizona State University in pursuit of the Master of Science degree in electrical engineering.