CIVIL ENGINEERING STUDIES
Illinois Center for Transportation Series No. 07-013
UILU-ENG-2007-2029
ISSN: 0197-9191

EVALUATION OF RISK IN END-RESULT SPECIFICATIONS FOR ASPHALT PAVEMENT CONSTRUCTION

By

William Buttlar
University of Illinois at Urbana-Champaign

and

Anshu Manik
University of Illinois at Urbana-Champaign

Research Report FHWA-ICT-07-013

A report of the findings of

ICT R27-23
Performance Related Specifications

Illinois Center for Transportation
December 2007

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-ICT-07-013		

4. Title and Subtitle Evaluation of Risk in End-Result Specifications for Asphalt Pavement Construction	5. Report Date October 18, 2007
	6. Performing Organization Code
7. Author(s) W. G. Buttlar and A. Manik	8. Performing Organization Report No. FHWA-ICT-07-013 UILU-ENG-2007-2029
9. Performing Organization Name and Address Illinois Center for Transportation Department of Civil and Environmental Engineering University of Illinois 205 North Mathews – MC-250 Urbana, IL 61801	10. Work Unit (TRAIS)
	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address Illinois Department of Transportation Bureau of Materials and Physical Research 126 East Ash Street Springfield, IL 62704-4766	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

End-Result Specifications (ERS) for asphalt pavement construction offer potential benefits over method-related specifications. They can be used in conjunction with or replacement of traditional QC/QA specifications as a means to enhance contractor innovation, reduce agency testing burden, and enhance overall pavement quality. Unlike other manufacturing sectors, the measure of pavement quality is not as simple as detecting and quantifying defective items. The quality of pavements is assessed with imperfect measuring tools operated by humans, who may inadvertently or intentionally introduce measurement variability or bias. As a result, the ability to measure quality and assign appropriate payment bonuses and penalties is an imperfect system.

This report details the development of a simulation tool which can be used to analyze specification risk and to develop ERS systems with user-managed risk levels. The program, called Simulated Risk Analysis (SRA), computes the risk of overpayment (agency risk) or underpayment (contractor risk) as a function of many factors, including: number of tests, production and measurement variability, bias, pay formula and pay caps, and specification limits. It also considers the quality assurance and third-party testing scheme used. In general, SRA can be used to develop a better understanding of how changes in individual ERS specification parameters can affect the payment risk for the contractor and agency. This knowledge can be used to explore the possibility of developing desirable changes in an existing ERS, such as reducing sample size, reducing risk, optimizing tolerance limits, changing pay factor equations, and the pros and cons of pay factor equations with payment caps.

17. Key Words	18. Distribution Statement
End-Result, Specifications, Construction, Asphalt, Pavement, Risk, Quality, QC, QA, Simulation	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Acknowledgement and Disclaimer

*This publication is based on the results of IHR-R23****, Performance Related Specifications****. IHR-R23 was conducted in cooperation with the Illinois Center for Transportation; the Illinois Department of Transportation, Division of Highways; and the U.S. Department of Transportation, Federal Highway Administration. The authors wish to thank Mrs. Laura Shanley, Mrs. LaDonna Rowden, and Mr. Jim Trepanier for their valuable guidance and considerable efforts in gathering and sharing field data and data analysis results during the course of this study.*

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AASHTO	American Association of State Highway and Transportation Officials
AC	Asphalt Content
Eq.	Equations
ERS	End Result Specifications
FHWA	Federal Highway Administration
Gmm	Maximum Theoretical Specific Gravity
High CI	Upper Limit of Confidence Interval
IAPA	Illinois Asphalt Paving Association
ICHRP	Illinois Cooperative Highway Research Program
IDOT	Illinois Department of Transportation
Low CI	Lower Limit of Confidence Interval
MS	Microsoft
NRB	Narrow Risk Band
PF	Pay Factor
PRS	Performance Related Specifications
PWL	Percent within Limits
QC/QA	Quality Control/Quality Assurance
Spec Lmits	Specification Limits
SRA	Simulated Risk Analysis
Std. Dev.	Standard Deviations

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

A cooperative investigation to develop and refine End-Result Specifications (ERS) for asphalt pavement construction was conducted under the Illinois Cooperative Highway Research Program (ICHRP), Project R-23. Specifications for asphalt pavement construction in Illinois have evolved from highly prescriptive material and method specifications, to quality control, quality assurance (QC/QA) specifications, and more recently to ERS on a demonstration basis. ERS demonstration projects have been led by the Illinois Department of Transportation's (IDOT) Bureau of Materials and Physical Research (BMPR) and undertaken by the IDOT districts on a voluntary basis since the year 2000. These comprehensive projects were focused on larger hot-mix asphalt (HMA) projects (8,000 tons and higher), and often involved interstate or state highway resurfacing projects.

One of the benefits of ERS is the introduction of true incentive/disincentive clauses for the control of material parameters that are believed to be linked to pavement quality. In Illinois the pavement qualities which determine payment are in-situ density, asphalt content, and plant air voids of plant-produced HMA. The use of an end-result approach gives the contractor more freedom in the attainment of those end-results, i.e., equipment choices, plant and field operations, etc., and therefore promotes contractor innovation and creates avenues for lower bid costs, while assuring material quality. The introduction of payment incentive/disincentives requires regular material sampling and testing. But more importantly, because ERS shifts some of the responsibility from the agency to the contractor, it is important to understand the relative risks assumed by each party in such a specification. As will be demonstrated in this report, the existence of significant measurement device variability and measurement device bias increases the overall risk in an ERS system.

Specification Risk

Most of the choices made in the development of an ERS have an effect on specification risk. Risks are undertaken by both the contractor and agency. The introduction of new specification criteria and/or the adjustment of certain specification attributes can shift the risk from the contractor to the agency and vice-versa. In some cases, specification changes can widen or narrow the range of risk.

Some of the key contributors to risk in ERS are:

- Contractor testing versus agency testing
- Frequency of testing and/or number of samples
- Variability and/or bias of test device and/or test procedure
- Specification parameters, including:
    - Specification limits
    - Pay factor equation
    - Pay "caps"

- Acceptance test frequency and acceptance tolerance
- Third-party testing provisions

Contractor risk must be controlled in order to arrive at fair payment, which would lead to lower and more consistent bid estimates over time, and would minimize disputes. Agency risk must be controlled in order to ensure that high quality pavement is produced, so that desired serviceability and safety levels are maintained over the design life.

In the past, researchers have attempted to develop statistical or simulation tools to help understand and balance risks in asphalt construction specifications. A computer simulation program called OCPLOT, developed in FHWA Demonstration Project 89 by Weed (1996), is available for generating OC curves. OCPLOT was found to be user-friendly and very useful for initial assessment of relative risks, allowing the user to vary the following factors: sample size, pay factor equation, specification limits, and retest provisions. The program allows the user to assess the probability of acceptable material being rejected (defined as contractor risk) and the probability of rejectable quality material being accepted (defined as agency risk) over the long run. However, a number of the factors that appear to be related to risk, including measurement device variability and testing bias, are not considered in OCPLOT. In addition, it can be argued that the most tangible measurement of risk should be linked to the financial impact on the project, i.e., how *risk* affects *what is actually paid versus what should have been paid.* Thus, one of the tasks in the ICHRP R-23 project involved the development of a simulation program that could be used to quantify and balance fiscally quantified risks, or *payment risk,* for the purpose of developing a rational and equitable end-result specification for asphalt pavement construction in Illinois.

One of the necessary steps in the assessment of payment risk is to clearly define the risk metric. The one used in this project is very straight forward:

Payment Risk = Payment made to the contractor – "Correct" payment

Ideally, tests performed by different parties on the same material should give very similar results. However, in practice even split samples will show different results when the tests are carried out by two different agencies, or in two different labs. Because of these uncertainties there is a risk of accepting rejectable quality material and vice-versa.

In the ERS approach, a percentage of acceptable quality (Percent Within Limits, or PWL) is determined, rather than pass/fail criteria used in typical QC/QA approaches (Figure 1.1). Payment is then made based on this percent within limits value (Patel, 1996). Because of the uncertainties involved with the test results the payment made also may be more or less than what it would be if the actual quality of the construction would have been exactly determined (Weed, 1996; Willenbrock, 1976; NCHRP, 1976; Barros et al., 1983; Puangchit et al., 1983; Afferton et al., 1992; AASHTO, 1995). Overpayment of the contractor is often referred to as "agency risk" while underpayment is often termed "contractor risk." Throughout this document, *positive values of risk refer to the instance where the agency paid more than required (agency risk) and negative values of risk indicate that the agency paid less than what the contractor deserved (contractor risk).*

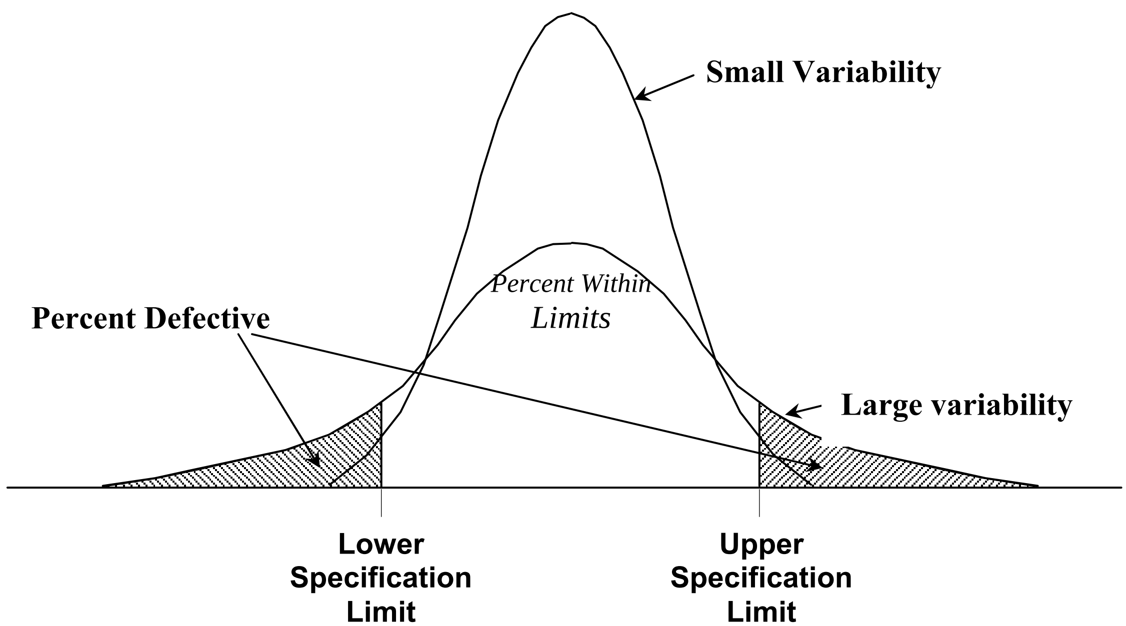


Figure 1.1: Concept of Percent Within Limits (PWL)

Outline of ICHRP R-23 Risk Analysis Research

To address the need for estimating and analyzing payment risks in an ERS, researchers at the University of Illinois at Urbana-Champaign have developed a series of risk simulation models that provide the user a virtual environment to quickly generate and analyze thousands of realistic ERS data sets. The first simulation model developed was ILLISIM (Buttlar and Hausman, 2000). This was followed by PaySim and BiasSim (Aurilio et al., 2002) which used different models and catered to different aspects of risk analysis and simulation. The latest model developed is called Simulated Risk Analysis, or SRA. SRA combines the capabilities of each of the earlier programs into a single program, with added features to simplify the process of conducting sensitivity analyses.

Chapter 2 describes the development of these simulation models, starting with ILLISIM. ILLISIM takes input parameters such as measurement variability, production variability, mean production, sample size, and sampling technique, and generates data which simulate test results that are collected in the field or at the plant to assess construction quality for ERS. The generated data is based on the assumption that data collected in a typical construction project would be normally distributed (Hall and Williams, 2002). Analysis is then performed on simulated data to estimate the contractor and agency payment risks, along with statistically determined confidence intervals. ILLISIM was developed with a Microsoft Excel interface, with analysis code written in Visual Basic for Applications (VBA). Chapter 2 also describes the improvements in ILLISIM which resulted in a new program, called PaySim. PaySim used a different simulation engine based upon the chi-square distribution rather than normal distribution, which permitted simulations to be less time consuming, which was a concern with ILLISIM. Furthermore, PaySim was coded in C++ and made into executable program, which also enhanced the processing speed considerably. The program, however, continued to utilize an MS Excel user interface.

The development of a third version of the simulation program, called BiasSim, was necessitated with the recognition of the fact that bias in field measurements played a significant role in the risk involved with pay factors (Buttlar et al., 2001). BiasSim was exclusively dedicated to analysis of effects of bias on risk as described in Chapter 2. Finally, a unified simulation program was developed, which combined the capabilities of ILLISIM/ Paysim and BiasSim, called Simulated Risk Analysis (SRA), which is also presented in detail in Chapter 2.

SRA has been extensively used to analyze Illinois Department of Transportation end-result specifications. The analysis and results are presented in Chapter 3. Chapter 4 summarizes project activities and findings, presents study conclusions, and presents recommendations for futher study. Additional details of the earlier models, namely ILLISIM, PaySim and BiasSim, have been provided in the Appendicies.

CHAPTER 2

Development of Simulation Models

Analysis of data obtained from actual construction projects corresponding to various quality characteristics like in-situ density, air-voids content, and asphalt content, have shown the data to be largely normally distributed (Hall and Williams, 2002). Although the target for a particular quality characteristic is generally a fixed value because of certain uncontrollable factors, it is almost never possible to produce exactly at that level.

Variability observed in the field includes both production variability and measurement variability. Production variability includes all variability introduced due to field compaction variables, variability in the quality and physical characteristics of source materials, changes in the relative proportions of ingredients in the plant-produced asphalt mixture, changes in plant operational characteristics, changes in equipment operators, etc. Measurement variability is the variability which is introduced by measuring devices, test procedures, and operator techniques and human error. In addition to variability around the actual value, a measurement bias may be introduced as well. Bias refers to a consistent shift in data and can be introduced by device calibration errors, human error, or by the intentional biasing of measurements and/or recorded data. Two common examples of device calibration bias relevant to the IDOT ERS program are: the use of an incorrect ignition oven calibration factor, or an improper angle calibration in a Superpave gyratory compactor.

Estimating and Expressing Variability

Setting bias aside for the moment, air voids, density, and asphalt content data collected in a typical ERS project can be assumed to have normally distributed fluctuations. This can be mathematically modeled as:

$$(d / AV / AC) = \mu + \sigma_{prod.} + \sigma_{meas.}$$

Where,

$(d / AV / AC)$	represents density, or air voids, or asphalt content data
μ	represents mean
$\sigma_{prod.}$	represents the production variability and
$\sigma_{meas.}$	represents the measurement variability

It should be noted that contractor, agency, and third-party data are expected to follow this model. Third party, as referred to here, is an independent testing entity employed by the agency for resolving disputes in test measurement results. Since all the parties test the same material using split samples, the mean and production variability is the same for all parties. The difference observed in the test data from the contractor and the agency, for example, can be attributed to the measurement variability. Therefore, the model when applied to the contractor data would be:

$$(d / AV / AC)_{Contractor} = \mu + \sigma_{prod.} + \sigma_{meas,Contractor}$$

and when applied to the agency, would be:

$$(d / AV / AC)_{Agency} = \mu + \sigma_{prod.} + \sigma_{meas,Agency}$$

Since each measurement is performed on the spilt samples of the same material, the two values modeled above would form paired data. Subtracting the second from the first would eliminate the mean and production variability terms.

$$(d / AV / AC)_{Contractor} - (d / AV / AC)_{Agency} = \sigma_{meas,Contractor} - \sigma_{meas,Agency}$$

Both terms on the right side of the equation come from a normal population. Therefore, their difference also would be normally distributed. Therefore,

$$\sigma_{meas,Contractor} - \sigma_{meas,Agency} = N(\mu, \sigma_{Comb})$$

Where,

$N(\mu, \sigma_{Comb})$ represents a normally distributed population with μ as mean and σ_{Comb} as combined standard deviation where:

$$\mu = 0$$

$$\sigma_{Comb} = \sqrt{\sigma^2_{meas,Contractor} + \sigma^2_{meas,Agency}}$$

Further, it can be assumed that the measurement variability for one type of test, like core density or asphalt content, would be fairly similar. Then, field data could be pooled in order to obtain a typical value for measurement variability, which assumes:

$$\sigma_{meas,Contractor} = \sigma_{meas,Agency} = \sigma_{meas}$$

Therefore,

$$\sigma_{meas} = \frac{\sigma_{Comb}}{\sqrt{2}}$$

Simulation

The motivation for using simulation to quantify specification risk can be summarized as follows:

- Risk in a construction specification arises from the fact that the process produces material with significantly varying properties, but the measurement of such fluctuations is relatively expensive and therefore a limited number of measurements can be taken. The problem is further complicated since, unlike some other manufacturing processes, measurement variability and bias also exists due to the use of imperfect measuring devices.
- In order to quantify payment risk in a specific end-result specification, one must first statistically describe how the aforementioned uncertainties in calculated pay would, in the long run, fluctuate from the ideal pay.
- In order to statistically describe payment error, one must either use a closed-form analytical solution or a simulation tool. Except for the simplest of specifications, closed-form solutions are not possible to formulate. Simulation approaches generally require the computation and analysis of thousands of simulated production runs in order to arrive at model convergence. With modern computing power, tens of thousands of construction scenarios can be simulated in tens of minutes. Furthermore, the amount of simulation time required is not of critical concern, since the simulations are used in the creation or adjustment of a specification. Once the specification is developed, implementation of the resulting ERS does not require the simulation to be run.
- In order to simulate variability in asphalt pavements material properties, one must be able to sequentially simulate: 1) production and/or construction variability; 2) effects of random sampling of the variable material; 3) effects of measurement variability and/or bias, and finally; 4) the effects of tester bias on the final reported test measurement values.
- In order to estimate risk in terms of effects on pay, the software must also simulate the formulas and decision tree logic contained in the construction specification.
- Finally, a useful simulation tool would provide a convenient user interface, facilitating the rapid generation of input files, executing the analysis engine, and providing a statistically-oriented analysis of data (post-processing). An advanced tool would also be able to create a database of results, and a post-processor for evaluating the database at a later time (to save computational time).

One of the main challenges in the development of a risk simulation program is the ability to generate tens or hundreds of thousands of field measurements from thousands of simulated construction projects. Based upon the assumption of normality discussed earlier, this relies on generating a normally distributed random number sequence with mean and standard deviation values or ranges to be studied as input by the user. In Chapter 3 we report the extensively studied values estimated from observations of actual field project standard deviations in Illinois.

The following sections describe the development of a series of ERS simulation models, developed in this project, of progressively increasing sophistication and increased user options and flexibility, including:

- ILLISIM – The Original Simulation Model
- PaySim – A Second-Generation Simulation Model

- BiasSim – A Simulation Tool for Analyzing Bias
- Simulated Risk Analysis (SRA): The Latest Model

The most current program, Simulated Risk Analysis (SRA), is a culmination of its predecessors and represents a highly functional, user-friendly ERS development and analysis tool.

The reader who is interested in the historical development of ERS and simulation tools in Illinois should read on. Readers interested in learning about the most recent specification and simulation tools used in Illinois should skip to the section entitled "Simulated Risk Analysis (SRA): The Latest Model."

ILLISIM: The Original Simulation Model

The first computer program developed to analyze payment risks was called ILLISIM. Details are provided in Appendix A1 and in Buttlar and Hausman (2000) and Buttlar et al. (2001). ILLISIM was used to model the Illinois ERS demonstration projects in 2000.

ILLISIM randomly generates simulated values for the quality characteristics within given SUBLOTS and LOTS of material on a paving job. It should be noted that capitalization is used for the terms 'LOTS' and 'SUBLOTS' in this report to be consistent with the nomenclature used in previous IDOT reports. The user has the ability to determine how ILLISIM evaluates the source(s) of variability depending on how easily individual sources of error can be identified. If a given characteristic has separable, measurable sources of variability, the user can determine how each source independently affects the determination of quality. Standard deviation is considered as an estimate of the variability of construction. Using density as an example, ILLISIM can consider three individual elements of variability (longitudinal, transverse, and measurement device). However, if the user wishes to analyze a database of historical measurements from which individual sources of variability cannot be deduced, the total standard deviation from the data set can be used.

ILLISIM uses the simulated measurements to compute a mean, standard deviation, percent within limits, and pay factor for each LOT of material considered. A minimum of 1000 LOTS were typically simulated for each unique group of input parameters considered. ILLISIM keeps track of a large number of runs, so that a statistical distribution of correct pay, versus actual pay for individual LOTS and complete JOBS, can be plotted.

The sampling schemes that can be simulated and analyzed using ILLISIM for as-constructed pavement density can be described as follows:

- **Dual-Stratified Random Sampling Method** — A length of pavement, or LOT, can be divided into equal SUBLOTS, which can be further subdivided by the number of transverse measurements desired per SUBLOT. Sampling locations are based upon a conventional stratified random layout in the longitudinal direction. In the transverse direction, samples are to be taken at the 2-, 4-, 6-, 8-, and 10-ft offsets, in random order. Means and standard deviations are then computed using all measurements.

- **Stratified-Average Sampling Method** — This method utilizes an identical sampling layout as the dual-stratified method. However, the mean and standard deviation are computed in a different manner. First, an average density is obtained for each of the SUBLOTS within a LOT. Then, a LOT average and standard deviations are computed using all the SUBLOT averages.

The average of properties measured within the LOT is the same between the two methods, but the stratified-average approach decreases the standard deviation and masks the variability that may occur across the mat.

The motivation for investigating the stratified-average method was to stabilize PWL-predictions on a per-LOT basis in an attempt to minimize the possibility of frequent disputes, particularly when marginal quality levels arise.

Inputs for ILLISIM

The user supplies the following inputs to ILLISIM:

(1) Mean value of as-produced or as-constructed quality characteristic (e.g. density, asphalt content, etc.) to be considered, or, more commonly, a range of such mean values

(2) Standard deviation(s) of the quality characteristic(s) associated with production and construction

(3) Standard deviation of the measurement device
(4) Number of measurements
(5) Sampling arrangement (e.g., completely random, dual-stratified random, stratified-averaging method, etc., described in more detail in a later section)
(6) Specification limits
(7) Pay factor equation
(8) Pay limits or "caps" (per lot and per job)

ILLISIM Computations and Output

First, simulated density measurements are used to obtain averages and standard deviations. Next, PWL values and pay factors are determined. A separate program called "Baseline" determines the "correct pay" for the input values given, based upon a very large number of simulations. Pay factor differences per LOT and per JOB are computed using ILLISIM, which are then compared to the correct pay value. Pay factor differences arise since a discrete number of measurements will not typically lead to an exact measure of mean and standard deviation for any given LOT.

Plots are generated to assess payment differences, or payment errors, that can be expected for a given set of inputs. These results are generally shown across a range of mean desity, asphalt content (AC), or air voids to illustrate the increased risk of payment error for LOT averages that happen to be near the specification limits (e.g., when marginal quality levels arise). Maximum and minimum payment errors (risks) per LOT (based upon 1000 LOTS) and per JOB (100 JOBS) are given. Also plotted are the 95% confidence intervals for pay differences relative to mean pay, which allow the analyst to identify typical risk envelopes, independent of possible extreme values for maximum or minimum pay difference. Finally, by defining the 95% confidence intervals on payment error as a "risk index," risk levels are also compared between different sampling methods, number of measurements, and such.

ILLISIM can be used to determine possible operating ranges where a given level of payment can be obtained, under various levels of process and device standard deviation. ILLISIM was used to assist IDOT in developing sampling schemes, adjusting specification limits and sample sizes for their asphalt ERS specification. More details are provided in Appendix A1.

PaySim: A Second-Generation Simulation Model

While ILLISIM was useful in shaping early decisions in ERS specification development, one major drawback of the model was the amount of time required to run the software. The factors behind the long computations times were:

(a) ILLISIM uses a reverse Monte Carlo Simulation algorithm. It relies on generating thousands of random numbers, processing them and repeating this procedure many, many times.

(b) ILLISIM was encoded completely in Microsoft Excel and Visual Basic for Applications, which are not optimized for large numerical problems.

To overcome these limitations, another simulation model called PaySim was developed. PaySim used an entirely new mathematical model to generate simulated data. This model does not require generation of thousands of normally distributed random numbers for each iteration, as was the case with ILLISIM. Instead, PaySim generates random numbers following a Chi-squared distribution. Ultimately, generated values are identical in nature to those generated by ILLISIM, but they are arrived at more efficiently. Also, the coding of the main simulation engine was done in the C programming language. These enhancements brought about appreciable improvement in the speed of the simulation process. In addition, PaySim was made to be more user-friendly and versatile. Details of the new mathematical model used in PaySim can be found in Appendix A2.

Inputs for PaySim

(1) Device variability
(2) Production variability
(3) Number of samples
(4) Number of sublots
(5) Analysis range for the quality characteristic being analyzed
(6) Specification limits
(7) Pay cap option (cap before averaging or after averaging)
(8) Precision in simulation required (four levels available)
(9) Confidence interval desired

Outputs from PaySim

The simulation is fully automated to complete all the tasks and produce risk plots for the quality characteristic being analyzed and in the range as defined in the inputs. The list of inputs also gives an idea of the versatility of the simulation, because practically any combination of input parameters can be chosen and analyzed. The output is in the form of risk plots showing the risk to the agency in terms of pay factor.

BiasSim: A Simulation Tool for Analyzing Bias

Besides variability, measurements of quality characteristics are prone to bias, or consistent shift in the measurements. ILLISIM and PaySim primarily dealt with issues related to production and measurement variability, number of specimens, sampling schemes, and tolerance limits. However, Buttlar et al. (2001) demonstrated that bias can significant effect payment risk. Furthermore, in order to accurately estimate production and measurement from field data, especially when data are to be pooled between multiple projects, bias must be first subtracted from the data to avoid arriving at highly inflated estimates of variability. BiasSim primarily focuses on the effects that such bias can have on the measurements and therefore on the pay factors and specification risk (Aurilio et al. 2002). Details about the BiasSim program can be found in Appendix A3. A brief summary follows.

Determining Bias Magnitude

Table 2.1 provides an example of the calculation of bias from field data. In this example, 10 split samples (adjacent cores, longitudinally aligned on the pavement and closely spaced) were taken to determine the as-constructed density of a pavement. The difference in the contractor and agency test results can be used to estimate the magnitude of bias. It is assumed that split samples have identical properties. Since a large number of samples have been obtained during IDOT ERS demonstration projects, reliable estimates of measurement bias have been obtained. More estimates of bias are given in Appendix A3 and have been reported in Buttlar et al., 2002.

Table 2.1: Example of bias calculation from in-place density data

Job	Contractor	Agency	Difference	Mean of Diff (Bias)
District 8	92.60	91.81	0.79	0.11
	93.67	93.84	-0.17	
	93.92	93.92	0.00	
	92.77	93.39	-0.62	
	93.80	93.88	-0.08	
	93.96	94.21	-0.25	
	92.85	91.81	1.04	
	95.45	95.12	0.33	
	94.17	94.46	-0.29	
	95.04	94.67	0.37	

Inputs for BiasSim

Microsoft Excel with visual basic programming is used as the interface for the user to enter the following inputs:

(1) Quality Characteristic to be analyzed
(2) Production variability
(3) Device variability for contractor (multiple inputs possible)
(4) Device variability for agency (multiple inputs possible)
(5) Sample size per job
(6) Number of cases to be analyzed (for batch processing)
(7) Range of quality characteristic values for analysis
(8) Specification limits
(9) Comparison tolerances
(10) Precision desired in simulation
(11) Confidence interval

Output from BiasSim

The outputs from the simulation are plots displaying risk in pay factor (%PF) for a given set of parameters and in the range of analysis desired. In the case of batch processing of simulation runs, all the cases are first computed, stored to a database, and then plotted. The plots also provide the lower limit and upper limit of confidence interval respectively. The level of significance for the confidence limits can be chosen by the user. In general, the existence of bias in measurements creates a skew in the risk plots, as presented in the Appendix (and later, in Chapter 3). Obviously, for the case of a contractor result biasing towards the middle of the specification limits, a positive risk (agency risk) exists.

The BiasSim program can be used to set comparison limits for quality assurance, number of QA samples, and specification decision tree logic for agency, contract, and third party test comparisons. Because of test variability, bias is not estimated accurately in QA comparison limits. This inaccuracy has been studied both in terms of number of invalid comparisons for a given job size over long runs, and in terms of its effect on payment risk for both parties. Invalid comparisons are the ones which are incorrectly assessed by the QA portion of the specification.

Simulated Risk Analysis (SRA): The Latest Model

ILLISIM, PaySim and BiasSim have been found to be useful tools for the analysis of risks in ERS systems in Illinois. Ultimately, it was necessary to combine the capabilities of these simulation tools into a single, combined risk analysis program because of the concurrent presence of production variability, measurement variability and bias. This has now been accomplished, in the SRA program.

How Does SRA Work?

In terms of the simulation engine, input, and output, SRA is truly a combined form of ILLISIM and BiasSim with additional features added for enhanced analysis capabilities and flexibility. Unlike the previous programs, SRA generates measurements to simulate the third-party tests, when needed. Third-party tests are used when the contractor and agency measurements do not match and the contractor chooses the option of having a third party resolve the dispute. Therefore, the full decision tree used in the present IDOT ERS is implemented.

SRA generates three sets of simulated data, with N measurement values in each set. Detailed flow charts describing the algorithms used in SRA are provided in Figures 2.1 through 2.3. The three sets correspond to the contractor, agency (District), and third-party measurements. N is the sample size for the job. The contractor and district measurements are then compared according to the DOT specifications. First, one out of every five measurements from the contractor and the district, corresponding to one split sample, is randomly selected and compared. If the two measurements are found to be within the tolerance limits specified for N=1 comparison, the contractor measurements are accepted for calculating payment. If the N=1 comparison fails, the contractor has the choice of accepting the district measurements or invoking the N=3 comparison with the third party.

The simulation engine has logic enforcing the rule that if the district measurements are closer to the middle of the acceptance limits, the contractor would choose to defer to the District measurements. This would normally increase the percent within limits, since the benefit of moving towards the middle of the specification range would normally outweigh the negative impact of an increased standard deviation. If the contractor chooses to invoke third-party testing, then three split sample measurements, chosen according to the comparison specifications, are averaged and compared. The tolerance limit for this N=3 comparison is generally stricter than that used in the N=1 comparison. If the average of the three measurements is within this tolerance limit the contractor measurements are accepted for pay calculation. Otherwise the third-party measurements are accepted. Thus a separate list of accepted measurements is generated. These measurements are then used to determine the percent within limits (PWL) according to the specifications. PWL is used to calculate pay factor for the contractor. The pay factor equation from the 2004 and 2005 IDOT ERS specification is taken by default, as shown below.

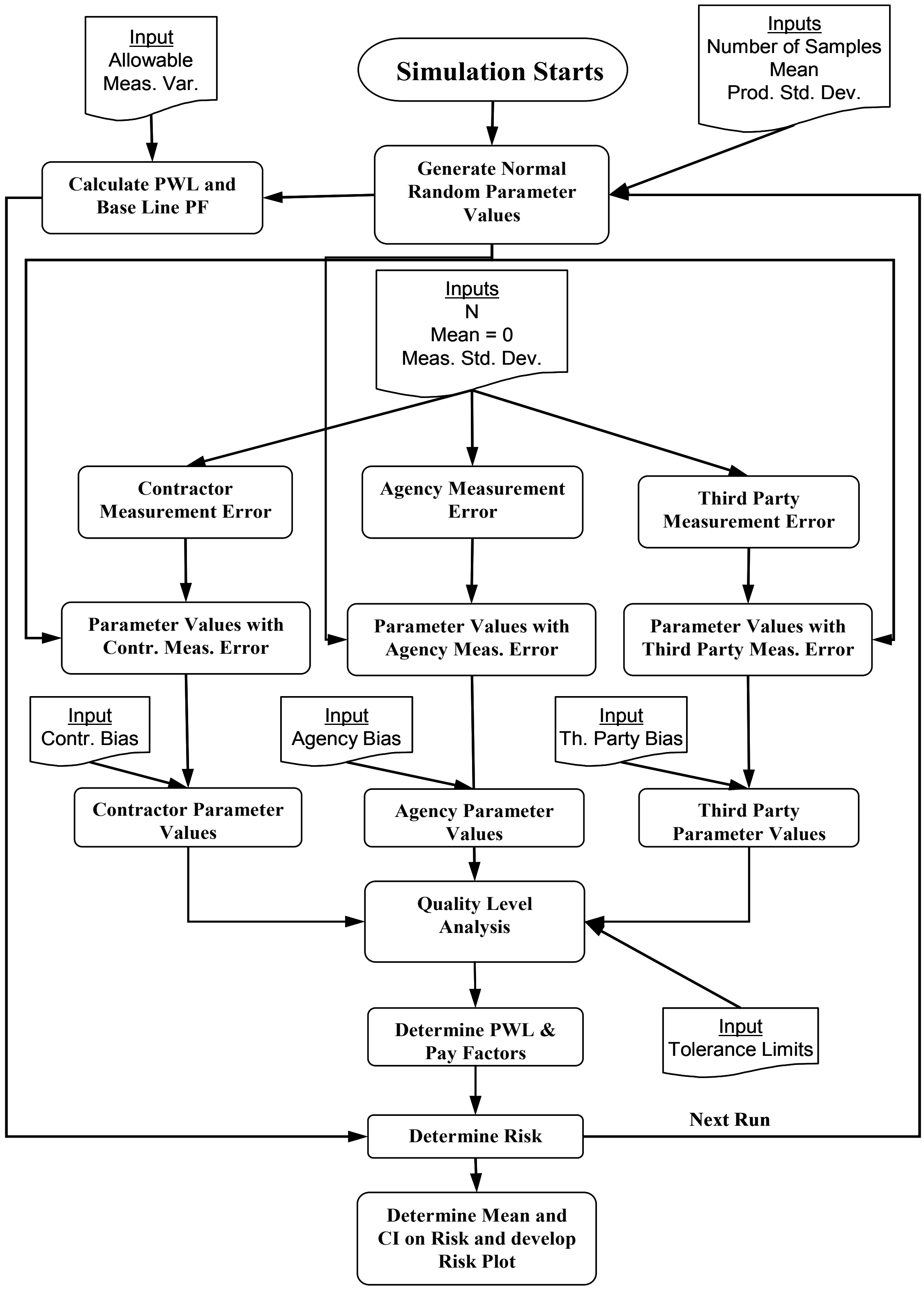
$$PF = 0.53 + 0.5 * PWL$$

Where,

PF = Pay Factor (%)
PWL = Percent Within Limits

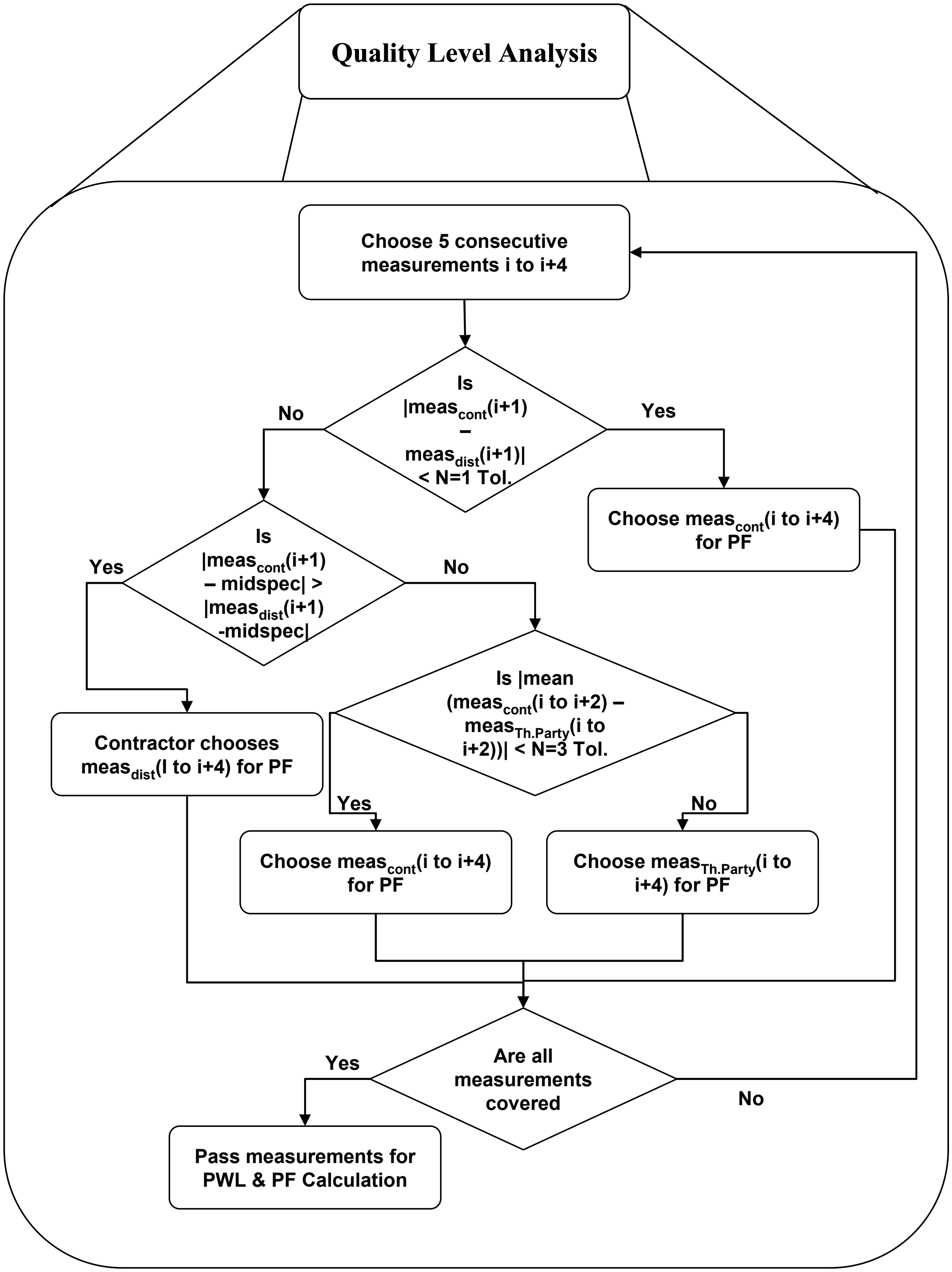
Depending on the accuracy desired, a certain number (generally between 1,000 and 10,000) of such sets of data are generated and PF calculated. The mean of all pay factors is then calculated and confidence intervals (90 percent by default) on the pay are determined. This set of pay factors is generally not normally distributed, because of the nonlinear nature of the acceptance logic, and the effects of maximum pay and pay caps. Therefore, mean and standard deviation of pay factor results should not be used to determine the confidence limits. Rather, a numerical assessment of the cutoff points, which provide the prescribed number of values inside the confidence intervals, is used. These calculations are performed at a particular mean value and repeated across the range of mean values specified by the user at a user-specified interval. Please refer to the brief user's guide in the next section for guidelines for specifying the interval. The output is in the form of plots for mean risk, ideal pay, and actual pay, along with confidence interval limits as described earlier.

As previously mentioned, the contractor, agency, and/or third party can introduce bias. While there is no perfect method available to estimate individual bias values, paired data from split samples from any two of the three parties can be used to investigate the possibility of bias. While the possibility of bias canceling (bias from both parties, in same direction) or bias compounding (bias from both parties, in opposite directions) exists, and would complicate bias analysis, it is expected that significant bias would most often occur in the data of a single party. While all of the aforementioned cases can be modeled without difficulty, up to this point the SRA program has only been used to study the effects of bias introduced by a single party.



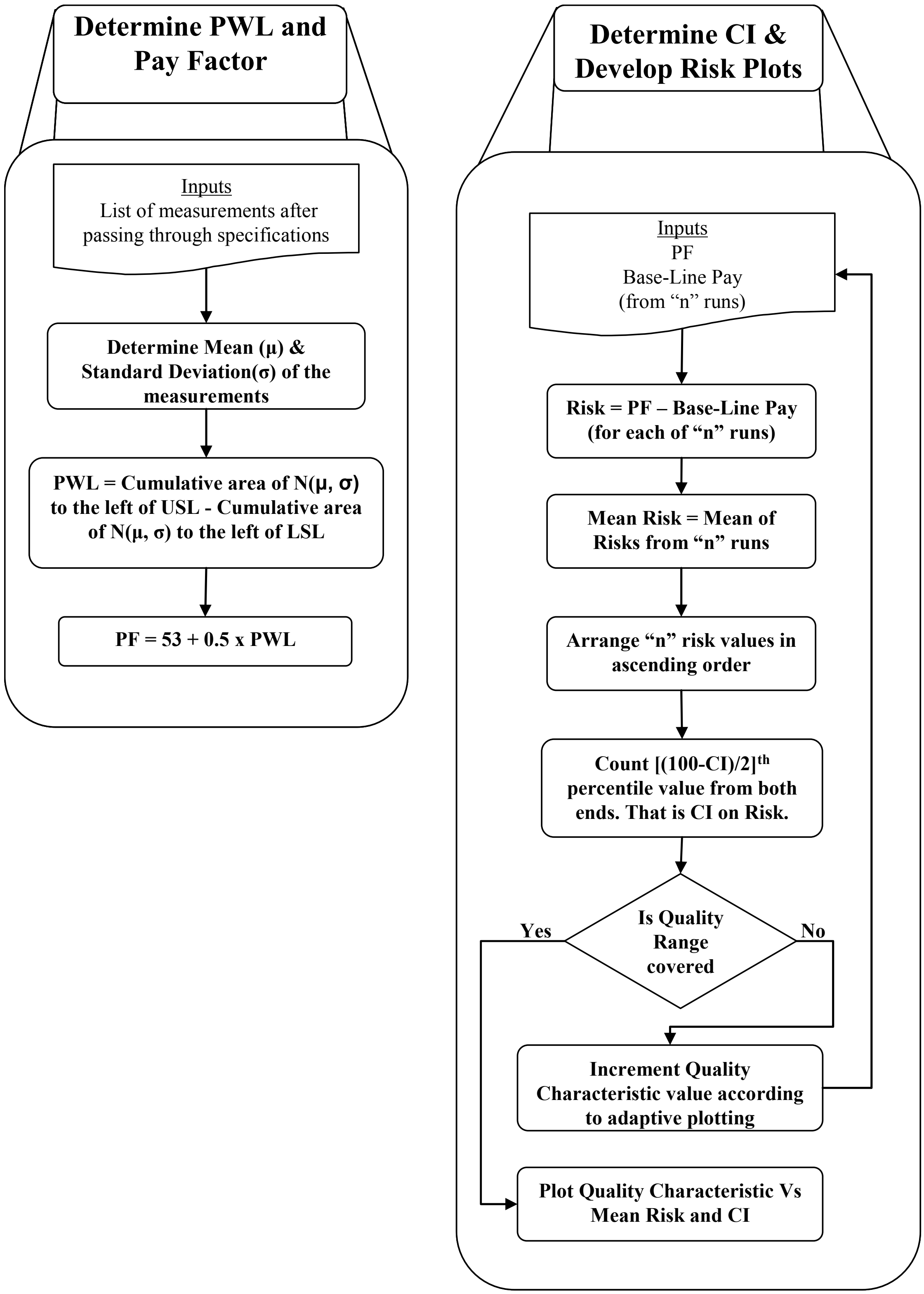
Key: Meas.–Measurement; Prod. – Production; Std. dev. – Standard Deviation; Contr – Contractor; PF – Pay Factor; BMPR – Bureau of Materials and Physical Research (same as Third Party)

Figure 2.1: Risk simulation and analysis procedure



Key: meas. – Measurement; midspec – middle of specification limits; PF – Pay Factor;

Figure 2.2: Quality assurance checks performed by SRA



Key: CI – Confidence Interval; PF – Pay Factor; PWL – Percent Within Limit;
N(μ, σ) – Normal Probability Distribution With Mean = μ and Standard Deviation = σ

Figure. 2.3: Determination of PWL and PF & Determination of CI and development of Risk Plots

SRA presents analysis results for two main classes of problems:

(1) *Analysis with original simulated data*: In this case specification limits are applied on the original simulated data. Then percent within limits (PWL) and pay factors (PF) are calculated.

(2) *Analysis with data from which relative bias has been removed*: Although individual absolute values of bias are not known, it is possible to remove any relative bias detected. An important point is that tolerance specifications are applied on the difference in measurement values of two parties rather than their individual values. Therefore, it is possible that before applying the tolerance specifications, relative bias is subtracted from the differences. Then, ideally the differences should represent measurement variability only. This procedure, if applied, would tend to reduce payment risk, if the data were appreciably biased, as compared to an analysis where bias was left in the data and used to arrive at a higher variability estimate.

Table 2.2 presents an example to illustrate the calculation and removal of relative bias. It should be noted that a very small number of measurements are shown in this example. In reality each party would have 100 to 250 or more measurements in a typical ERS project in Illinois.

Table 2.2: Example calculation for relative bias calculation and removal

Job	Contractor	Agency	Difference	Mean of Diff (Bias)	Normalized Difference
District 8	92.60	91.81	0.79	0.11	0.68
	93.67	93.84	-0.17		-0.28
	93.92	93.92	0.00		-0.11
	92.77	93.39	-0.62		-0.73
	93.80	93.88	-0.08		-0.19
	93.96	94.21	-0.25		-0.36
	92.85	91.81	1.04		0.93
	95.45	95.12	0.33		0.22
	94.17	94.46	-0.29		-0.40
	95.04	94.67	0.37		0.26
Mean	93.82	93.71	0.11		0.00

The columns "Contractor" and "Agency" represent density measurements in $\%G_{mm}$ corresponding to the named party. Each row shows the values for split sample measurements. Differences between the paired values are then calculated, and the mean of the differences provides an estimate of the relative bias. The last column shows the differences when relative bias is removed from the differences.

Brief Users' Guide for SRA

The main simulation engine for SRA has been developed in the program Matlab. However, Microsoft Excel with Visual Basic programming is used as the interface. The simulation is run in two stages, as follows.

(1) Microsoft Excel is used as the interface (SRA.xls). Figure 2.4 illustrates a portion of the user interface. The following points may be helpful in using the SRA Excel interface.

- **Working Directory**: The first task is to specify the working directory. A backslash ("\") should not be used at the end of the directory name. This is the directory where the input and output files will be stored by the program. This can be different from where SRA.xls is actually stored.
- **Output File Name**: The name of the output file is then specified, without a file extension. The program automatically saves results to an Excel file with a .csv file extension.
- **Case**: Since the SRA program is set up for batch processing up to 30 runs, the user must provide analysis parameters for a corresponding number of rows, as explained below. For convenience, any analysis case can be skipped by un-checking the case (check box is located one row above the Case number).
- **Precision Required**: Even with recent improvements to algorithms and the computing environment, the SRA simulation places significant demands on processing time. However, if one only desired to observe general trends for quick reference it is possible to choose a lower precision level in order to reduce analysis time. There are four standard levels of precision available, "High", "Medium", "Low" and "Crude". There are default numbers of runs that are sent to the simulation engine depending on the precision input by the user. But the user can also change the number of runs corresponding to these levels of precision. Table 2.3 gives the default number of runs associated with the precision levels. These defaults can be changed by editing cells "F39" to "F42" in the Excel Worksheet entitled "Home."

Table 2.3: Number of runs associated with precision levels in SRA

Desired Precision	No. of Runs
Crude	50
Low	1000
Medium	5000
High	10000

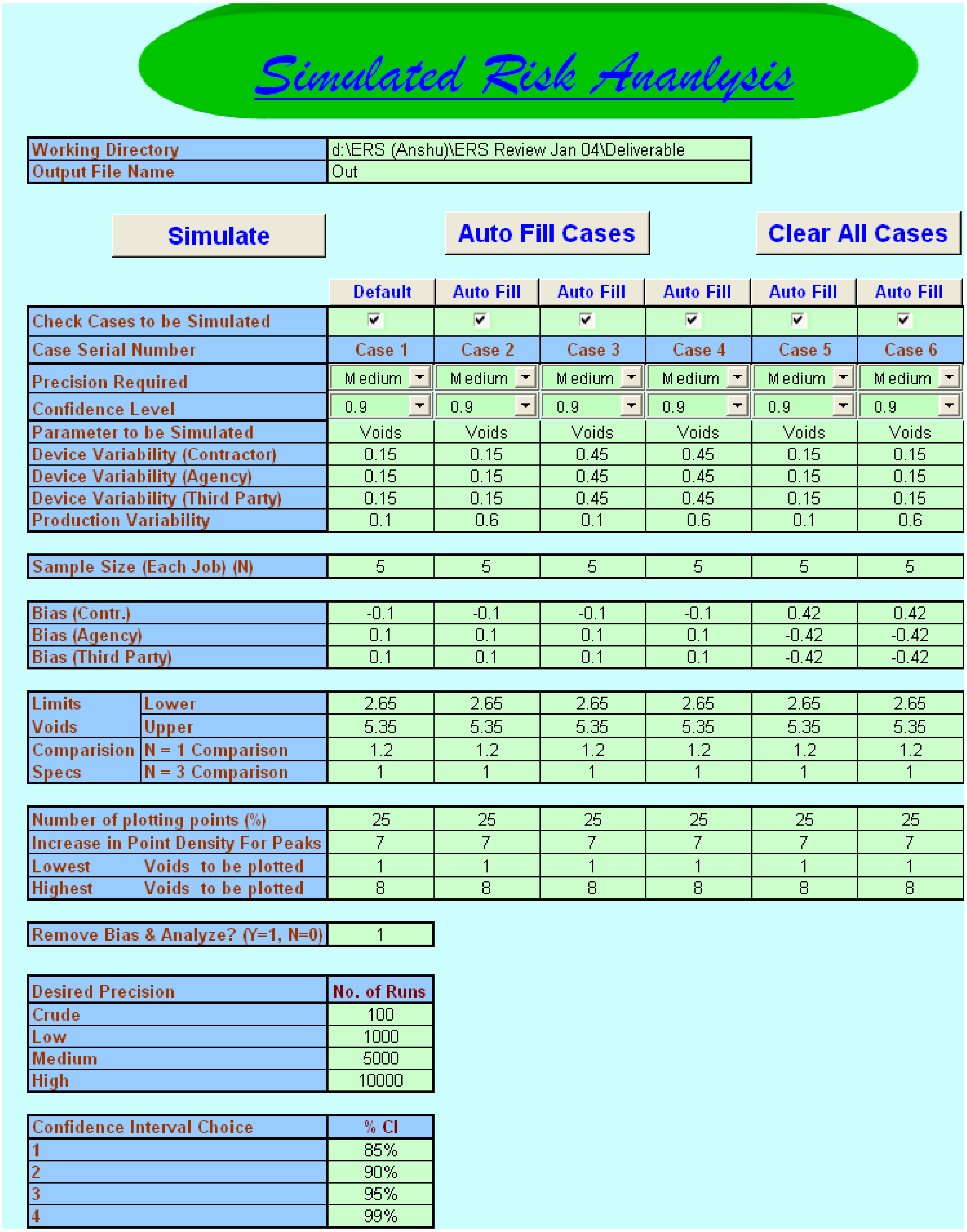
- **Confidence Level**: Four standard confidence limits can be chosen from the drop-down list. Custom confidence limits can be typed directly into cell "F45." The new entry will automatically appear in the drop-down list, which can then be selected. Each case can have a different confidence level associated with it.

- **Measurement Variability (Contractor, Agency and Third Party)**: Measurement variability is also sometimes referred to as device variability, and is a user-defined variable. Estimates of measurement variability can be computed from split sample test data, as described earlier in this report. This parameter has a strong influence on risk.
- **Production Variability**: This refers to the variability of selected physical properties of the as-produced mixture or as-constructed pavement, independent of measurement variability. As noted in the earlier analysis, this does appreciably affect payment risks. Although it is generally not possible to obtain direct measure of this parameter, it can be estimated by mathematically extracting the measurement variability from the total variability, using the equations provided earlier in this chapter.
- **Sample Size**: This is the number of specimens to be tested in the project by the contractor.
- **Bias (Contractor, Agency and Third Party)**: These are the estimates of individual bias in the measurements of the contractor, agency and the third party.
- **Limits (Upper and Lower)**: These are the upper and lower specification limits for acceptance of the product for the quality characteristic being analyzed.
- **Comparison Limits (Tolerance Limits)**: Tolerance limits define the maximum acceptable difference in test measurements between the comparing parties (e.g. Contractor and Agency). If the two readings are considered different this can be resolved by either the contractor accepting the agency readings or by using third-party measurements (in IDOT specifications). Also, according to the current IDOT specifications one out of five samples are compared first (N=1 comparison) and if the difference is outside the allowable limit, the average based on three samples are compared (N=3 comparison), this time using the tighter tolerance limits.
- **Number of Plotting Points:** This is the minimum number of plotting points desired in the risk plot to be generated by the simulation. The simulation uses adaptive plotting to put in more points if there are abrupt changes in the plot.
- **Increase in Point Density for Peaks:** This is the maximum increase in plotting density for adaptive plotting. More density would mean more processing time. If three is input in this field there will be at maximum three times as many points in the region of the peak as there are in regions of the plot where the curve is flat. It has been observed that having a density higher than 7 does not lead to a further gain in accuracy.
- **Lowest and Highest Points to be Plotted:** This defines the range of the quality characteristic chosen for analysis. All the risk would be calculated in this range. Practically, values far from the specification limits do not have risks, because everything would be rejectable quality. So, a good range could be from lower specification limit minus 4 times the standard deviation to the upper specification limit plus 4 standard deviations. This range, however, should be widened by the amount of bias on both sides.

There is no harm in taking a wider range except for the fact that the density of plotting points would be less or the number of points would need to be increased, leading to higher processing time.

- **Remove Bias and Analyze**: SRA simulates the field measurements with bias, if provided by the user. The user can opt for a second analysis to be done after removal of relative bias in addition to the analysis with biased data by keying in 1 (one) in this field. If the field has 0 (zero), SRA would not produce risk analysis results for the case when relative bias is removed.
- **Default:** If the user does not have a preference, the Default button can be used to restore typical default values.
- **Auto Fill:** This button can be used to copy the parameter values from the previous case. This is quite handy when the user changes only a few values from one case to the next.
- **All Clear:** This button can be used to clear all the parameter values before entering new values.
- **Simulate:** After all the values have been entered and precision levels, confidence intervals and cases have been chosen, the user can use this button to generate an input file named "input.txt". This input file is placed in the working directory, which can be edited by the user with a text editor. However, the layout of the file should not be altered. Multiple spaces are considered as single space by the simulation program. The sequential position of the entries in rows and columns are important, but not the spacing (free form input file). After this button is pressed, the program will ask if input.txt should be overwritten, etc.

(2) The main simulation engine was coded in Matlab. The user should set the working directory for SRA as the current directory in Matlab. It is assumed that the input file has been prepared in a previous step, as described above. Now the user simply clicks on the "Simulate" button. While the program will stop only after all the cases have been run, the command window will constantly show the case number being simulated and the percentage progress for that run. The percentage of progress is only for that case run and not for the entire batch run.



Simulated Risk Ananlysis

Working Directory	d:\ERS (Anshu)\ERS Review Jan 04\Deliverable
Output File Name	Out

Simulate | Auto Fill Cases | Clear All Cases

	Default	Auto Fill	Auto Fill	Auto Fill	Auto Fill	Auto Fill
Check Cases to be Simulated	☑	☑	☑	☑	☑	☑
Case Serial Number	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Precision Required	Medium	Medium	Medium	Medium	Medium	Medium
Confidence Level	0.9	0.9	0.9	0.9	0.9	0.9
Parameter to be Simulated	Voids	Voids	Voids	Voids	Voids	Voids
Device Variability (Contractor)	0.15	0.15	0.45	0.45	0.15	0.15
Device Variability (Agency)	0.15	0.15	0.45	0.45	0.15	0.15
Device Variability (Third Party)	0.15	0.15	0.45	0.45	0.15	0.15
Production Variability	0.1	0.6	0.1	0.6	0.1	0.6
Sample Size (Each Job) (N)	5	5	5	5	5	5
Bias (Contr.)	-0.1	-0.1	-0.1	-0.1	0.42	0.42
Bias (Agency)	0.1	0.1	0.1	0.1	-0.42	-0.42
Bias (Third Party)	0.1	0.1	0.1	0.1	-0.42	-0.42
Limits Lower	2.65	2.65	2.65	2.65	2.65	2.65
Voids Upper	5.35	5.35	5.35	5.35	5.35	5.35
Comparision N = 1 Comparison	1.2	1.2	1.2	1.2	1.2	1.2
Specs N = 3 Comparison	1	1	1	1	1	1
Number of plotting points (%)	25	25	25	25	25	25
Increase in Point Density For Peaks	7	7	7	7	7	7
Lowest Voids to be plotted	1	1	1	1	1	1
Highest Voids to be plotted	8	8	8	8	8	8
Remove Bias & Analyze? (Y=1, N=0)	1					

Desired Precision	No. of Runs
Crude	100
Low	1000
Medium	5000
High	10000

Confidence Interval Choice	% CI
1	85%
2	90%
3	95%
4	99%

Figure 2.4: Snapshot from Excel interface for SRA

(3) When the run is over, a file named "PF.csv" will be generated and placed in the working directory. The user can open this file in MS Excel. The output file "PF.csv" has the following three parts:

(a) The values of all the input parameters used in the simulation.

(b) The Main output of risk estimates. This section has output in 26 columns. The first 14 columns correspond to the analysis using data with relative bias, and the next 14 columns correspond to the data from which relative bias has been removed. This second set of output columns would not be generated if the user chooses not to analyze the data after removal of relative bias. In each set of output the first four columns are mean quality characteristic values and corresponding mean, lower confidence limit and upper confidence limit for payment risk determined. The 5^{th} and 10^{th} columns are the same as the first column and have been repeated to facilitate plot generation. The next four columns (6^{th} to 9^{th} columns) give the percentage of cases when: the N=1 comparison passes; the contractor accepts the district results; the N=3 comparison passes, and; the N=3 comparison fails, respectively. The 11^{th} to 14^{th} columns give the ideal pay, mean actual pay and upper and lower confidence limits on actual pay respectively. These two sets of values would give an idea of how much risk can be reduced if the relative bias is removed from the data before actually applying the specifications for pay factor calculations.

Outputs (a) and (b) are repeated for each case presented for analysis by the user.

(c) The third part of the output comes at the end of the PF.csv file. This gives the summary results from all the runs. This has two sets of results. The first set corresponds to the data simulated with bias. The second set corresponds to the data from which relative bias has been removed. The summary results in each set include maximum positive risk, maximum negative risk and Narrow Risk Band. The Narrow Risk Band concept will be explained in the next chapter.

CHAPTER 3

Analysis of IDOT End Result Specifications

This chapter presents detailed results from risk analyses performed using the SRA program on data collected by IDOT on ERS projects, primarily between 2000 and 2004. SRA has functionality at two distinct levels: 1) as an analysis tool at the standard user's level, and 2) as a research tool for more in-depth analysis and development of end result specifications. This chapter presents results of risk analyses which describe and explore:

- Typical results and plots from SRA and recommendations for interpretation of results;
- Estimated payment risks associated with IDOT's 2004 ERS specification;
- Capped versus uncapped pay factor equations, and;
- Two possible strategies for reducing contractor in-situ density testing without significantly increasing risk.

Use of SRA at the Standard User's Level

The following sensitivity analysis was carried out to determine how production variability, measurement variability, bias, and sample size affect risk. The analysis was extended to determine how these factors affect each other's effect on risk, i.e., interaction effects.

Sensitivity Analysis

The factors that were included in this analysis are

(1) Production Variability
(2) Measurement/ Device Variability
(3) Bias
(4) N (Number of samples in the job)

A set of simulations were run with the values of the factors given in Table 3.1. Two levels were chosen for each factor. Rather extreme values were intentionally chosen to clearly demonstrate the direction in which risk changes with the change in that factor. Also, the factors have been chosen in such a way that with further analysis the interaction of effect of one factor with another factor, or factors, can be determined using experimental design principles.

Table 3.1: Parameter values used in the sensitivity analysis with SRA

Run Number	σ_{prod}	σ_{meas}	Bias			n
			Contractor	Agency	Third Party	
1	0.3	0.2	0	0	0	15
2	0.8	0.2	0	0	0	15
3	0.3	0.5	0	0	0	15
4	0.8	0.5	0	0	0	15
5	0.3	0.2	0.4	-0.4	-0.4	15
6	0.8	0.2	0.4	-0.4	-0.4	15
7	0.3	0.5	0.4	-0.4	-0.4	15
8	0.8	0.5	0.4	-0.4	-0.4	15
9	0.3	0.2	0	0	0	40
10	0.8	0.2	0	0	0	40
11	0.3	0.5	0	0	0	40
12	0.8	0.5	0	0	0	40
13	0.3	0.2	0.4	-0.4	-0.4	40
14	0.8	0.2	0.4	-0.4	-0.4	40
15	0.3	0.5	0.4	-0.4	-0.4	40
16	0.8	0.5	0.4	-0.4	-0.4	40

Figures 3.1-3.4 show the output from these runs. These plots give the mean payment risk and confidence interval. The plots have been arranged in groups of four. Each group of plots has only two parameters varying and the other two parameters are held constant. Between left and right plots in each set production variability has been varied, and likewise between the top and bottom plots measurement variability has been varied. Bias has been increased in sets of plots in Figures 3.2 and 3.4 as compared to those in Figures 3.1 and 3.3 respectively. Between the first two sets of plots and last two sets, the number of measurements used per job has been increased from 15 to 40.

Since the SRA program provides new results that have been previously unavailable to the pavement engineering community, the analysis of risk plots deserves introductory comments and suggestions. So before a detailed discussion of the aforementioned simulation runs are presented, below are some suggestions and concepts to bear in mind when interpreting risk plots:

- The results have been plotted along a range of values for the quality characteristics, i.e., percent density (% G_{mm}). The way to interpret the results is to think of each discrete location along the plot as an ERS job which ended with a mean value for that quality characteristic of that particular value. For that given mean of production, along with the other parameters used in the simulation run, one can then see the expected average pay (over the long run, if one could evaluate similar jobs with a similar mean) and the confidence interval, which helps describe the range of variability that can be expected from job-to-job within those particular characteristics. *It is easy to mistakenly get an overly pessimistic view of ERS risk when analyzing risk plots for individual factors.* In reality, several quality characteristics are included in the pay factor equation (three currently in Illinois), and usually most of these quality characteristics will be produced within the Narrow Risk Band (see the following section), thereby reducing the overall payment risk for the job.

- Measurement variability tends to create large "vertical spikes" in the risk plot in the vicinity of the USL and LSL (cf. Figure 3.5). In general, these spikes are symmetrical with respect to the middle of the specification limit. So focusing attention on the LSL, for instance, as the average production decreases from the middle of the specification range towards the LSL, the contractor risk begins to increase very suddenly (risk becomes more negative due to increased chance of under pay). The risk reaches a maximum absolute value (local minimum on the plot), then recovers and crosses the zero point at the LSL (except when bias is present) and reverses in magnitude towards positive (agency risk) as the average production moves below the LSL. A simple way to think about this is that measurement variability creates pure error, which greatly increases risk when the average production is near the specification limits. This is because the measurement variability artificially inflates the overall measured standard deviation (as compared to actual production variability), which puts the contractor in more jeopardy when production is just within the specification limits, and puts the agency in more jeopardy when production is just outside the specification limits. The normal distribution curve has the most area in the middle of the curve, and hence, measurement variability has the largest impact on risk when the normal distribution is nearly centered on the specification limit. For example, the normal curve may be slightly to the left of the LSL due to random measurement fluctuations when it should have been slightly to the right, creating a relatively large pay factor error or risk.
- The level of production variability can affect the importance of measurement variability. In general, it is best if measurement variability is small as compared with production variability. When this occurs, the effect of measurement variability on risk near the specification limits is greatly minimized.
- Simple bias creates non-symmetrical risk plots and increases overall risk. The non-symmetries result from the fact that a simple, one-way bias (such as calibration error) will tend to decrease the risk for a given party on one side of the risk plot (pushing values towards the target), but tend to increase that party's risk on the other side of the plot (pushing values away from the target). Similar to measurement variability, bias is not an indicator of the real pavement's quality, so it leads to appreciable increase in risk.
- It should be noted that a second type of bias, not directly addressed in this study, can exist. This would be the case where a party intentionally biases all measurements towards the target (middle of spec limits). In this case, the risk for the other party (non-biasing party) would be increased at both specification limits. QA comparisons and QA policies would help minimize this problem.

Returning to the results of the simulation study, as presented in Figures 3.1-3.5, some of the salient points observed from the simulation results are as follows:

- As production variability is increased the risk becomes more distributed over the range. In other words a "Narrow Risk Band," or the extent of production averages in which the confidence interval of risk is very small or zero, is reduced with increasing production variability. For lower production variability the risk plot has sharper peaks near the acceptance limits and confidence limits are generally narrower in the middle.
- As measurement variability is increased confidence limits on payment risk, in general, become wider. But this widening of the confidence limits also depends on the production variability present. The increase is very significant when production variability is low and nearly insignificant when production variability is very high.
- Increase in measurement variability has a more profound effect on the mean risk than the confidence interval. If, in a particular case with low measurement variability the mean risk is close to zero, there is a significant increase in the mean risk with increased measurement variability. This signifies that the probability of contractor underpayment (in the case of negative mean risk) or agency overpayment to the contractor (in case of positive mean risk) is very high.
- Increase in bias induces significant increase in risk. This can change the magnitude as well as sign of risks involved, as compared to the data without bias. Interestingly the effect of bias is more significant when production variability is low than the case when production variability is high. One explanation of this phenomenon could be that increase in production variability induces variability in the data used for calculating base line pay as well. Therefore, the larger differences due to bias get overshadowed by the large production variability, leading to a smaller difference between ideal and actual pay.
- An increase in the number of measurements in the job always narrows the confidence interval width in the risk plot significantly. The magnitude of reduction in risk is higher when the data have higher production and measurement variability.

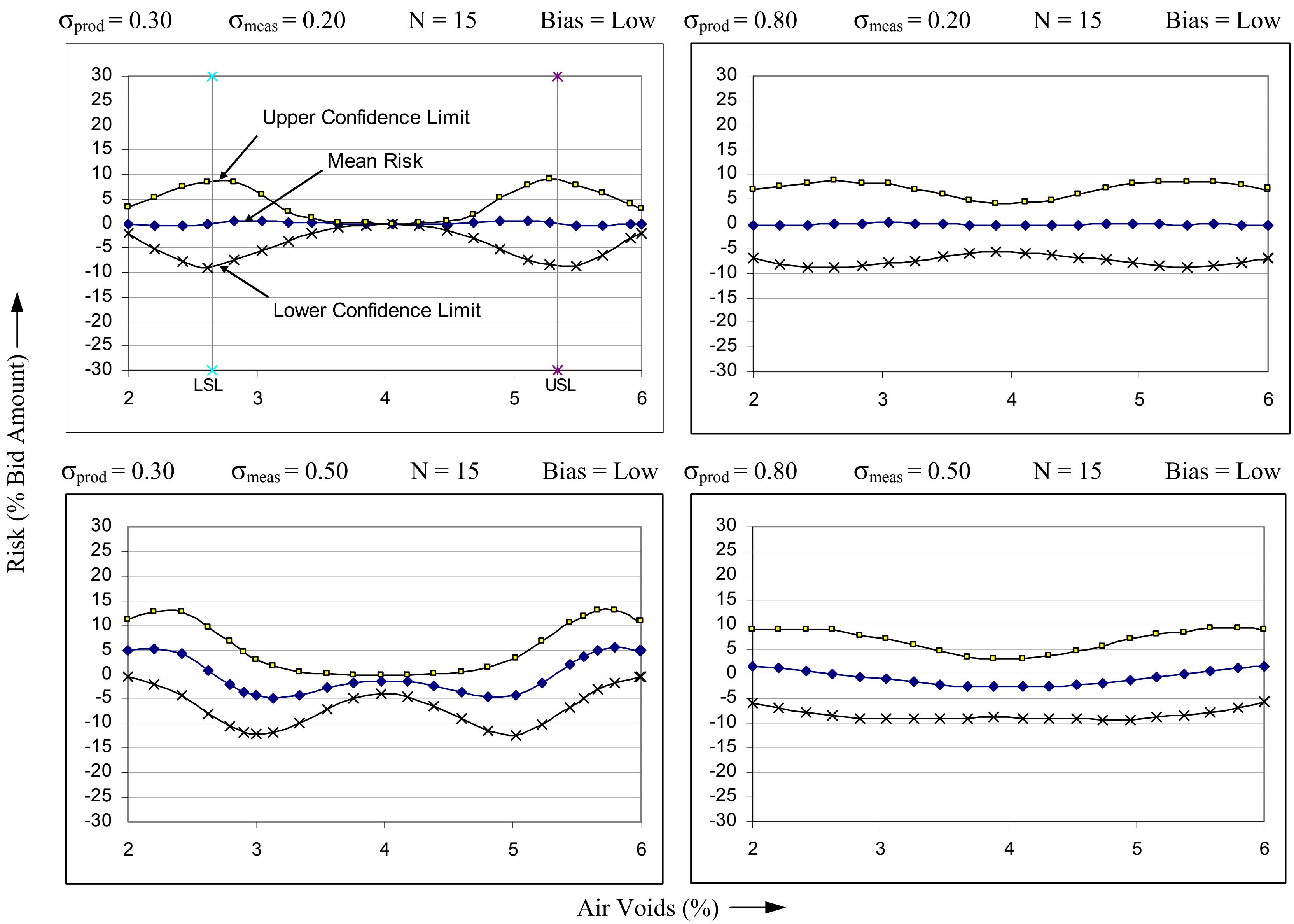


Figure 3.1: Sensitivity analysis simulation run plots from SRA (LSL, USL: Lower and Upper Specification Limits)

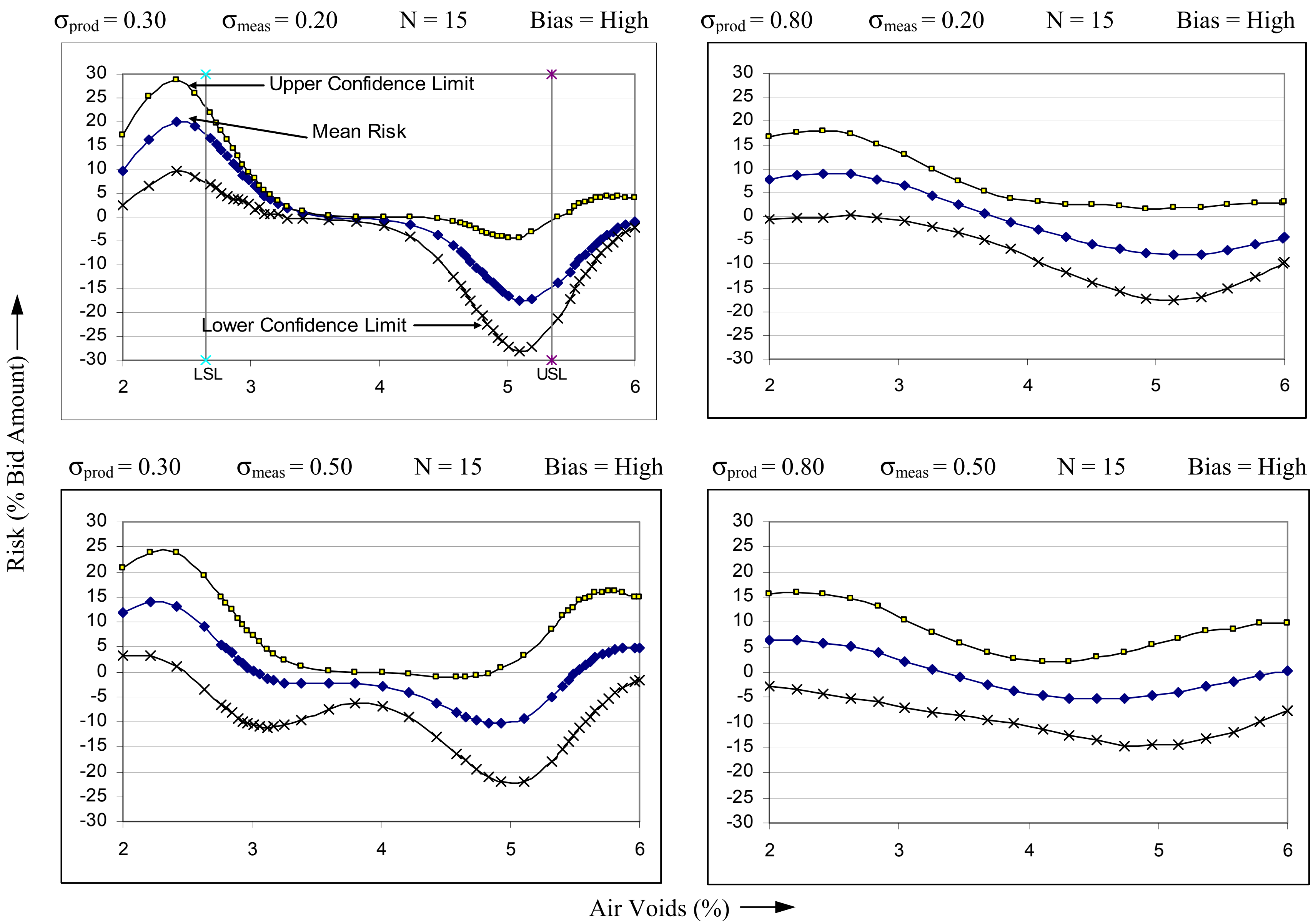


Figure 3.2: Sensitivity analysis simulation run plots from SRA (LSL, USL: Lower and Upper Specification Limits)

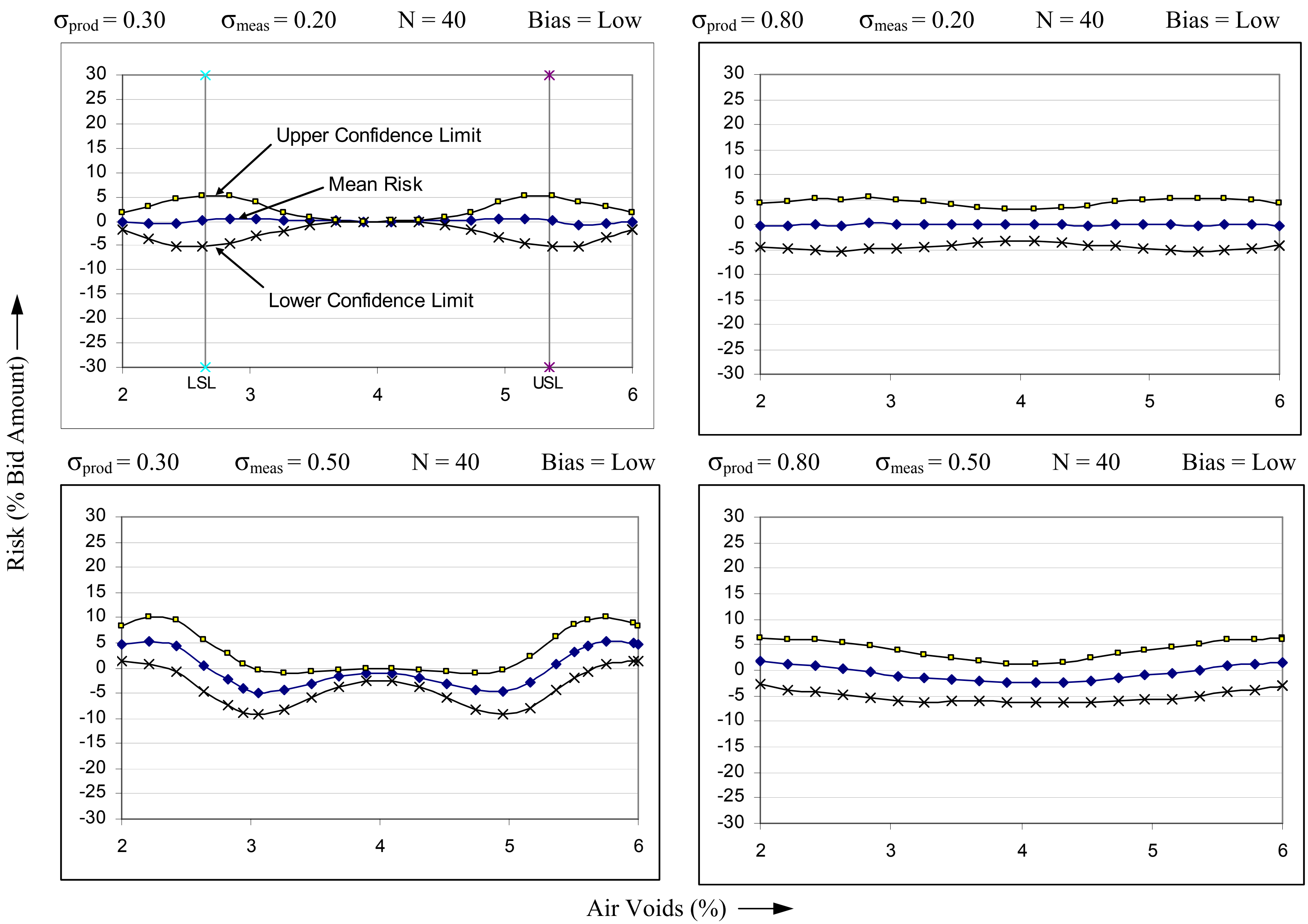


Fig 3.3: Sensitivity analysis simulation run plots from SRA (LSL, USL: Lower and Upper Specification Limits)

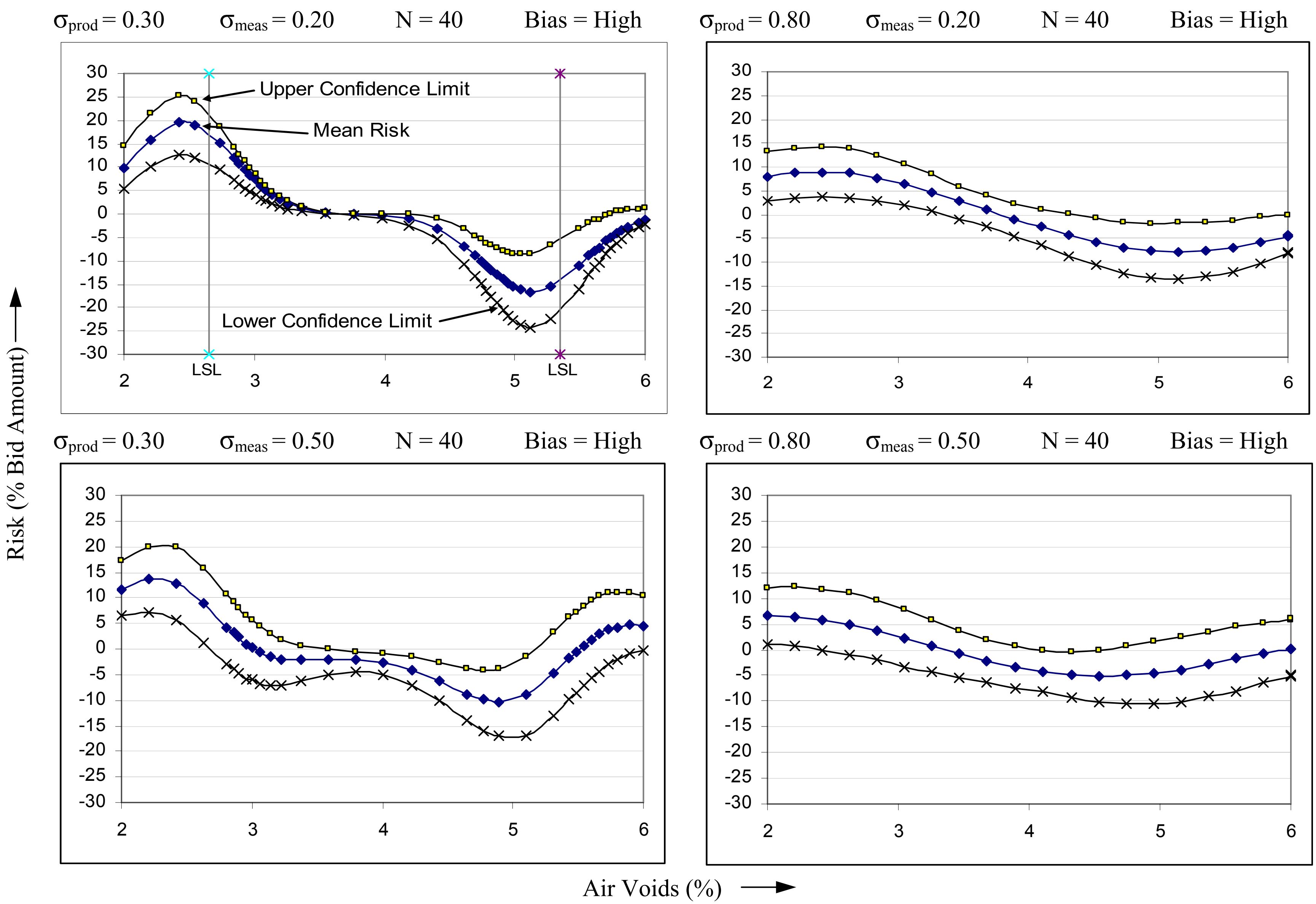
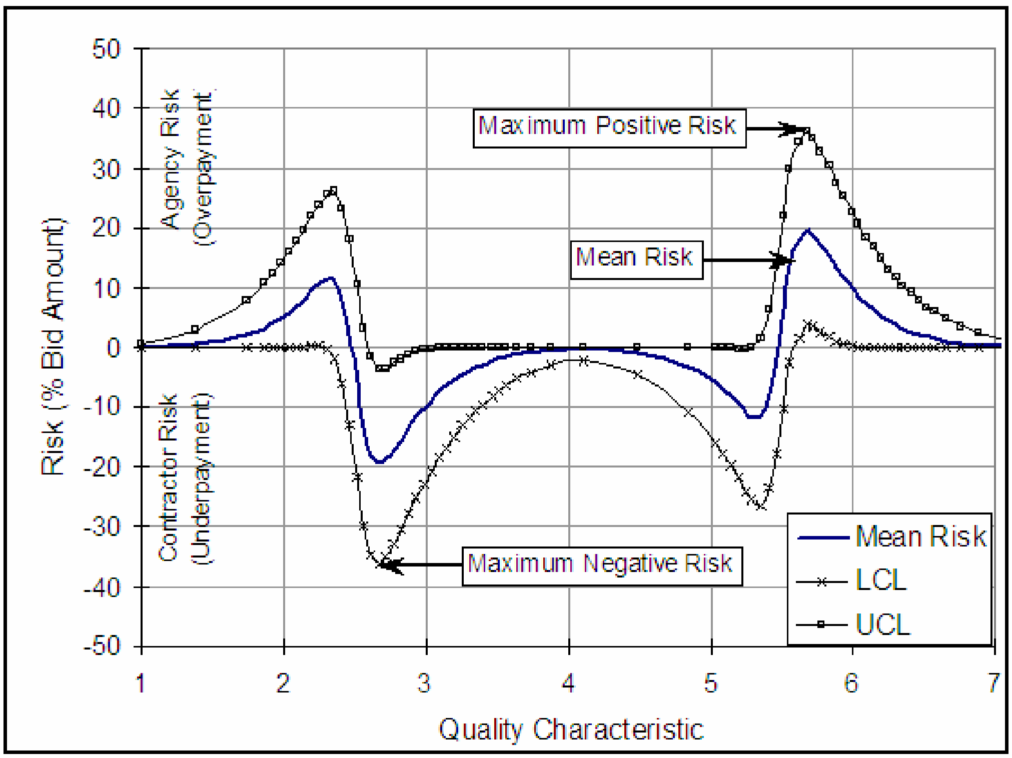


Fig 3.4: Sensitivity analysis simulation run plots from SRA (LSL, USL: Lower and Upper Specification Limits)

The previous section demonstrated the typical risks involved in ERS, as predicted by SRA, and how typical ERS parameters like sample size and bias affect this risk. It is desirable to use SRA in the design and adjustment of an ERS such that targeted risk levels are achieved while minimizing testing burden and limiting the potential for disputes and the need for third-party testing. To accomplish this, one approach would be to repeatedly run the simulation program and manually determine the parameter values giving lower risk. This would, however, require appreciable expertise on the part of the user. Therefore, it is desirable to have a computational tool which does this for the user. This in turn necessitates that the SRA results be further analyzed to produce objective quantities rather than just trends. Three ways were identified in which SRA risk results can be quantitatively characterized. They are:

(1)Maximum risk: This represents the peaks in the plots and gives the maximum amount of risk for the given set of parameter values within specification limits. It is notable that it would be more appropriate to use the peaks of confidence intervals rather than those of the mean risks. This is because there is a 50% possibility that risk would be greater than mean risk. *Maximum positive risk* (MPR) would represent the risk for the agency and *maximum negative risk* (MNR) would represent the risk for the contractor.

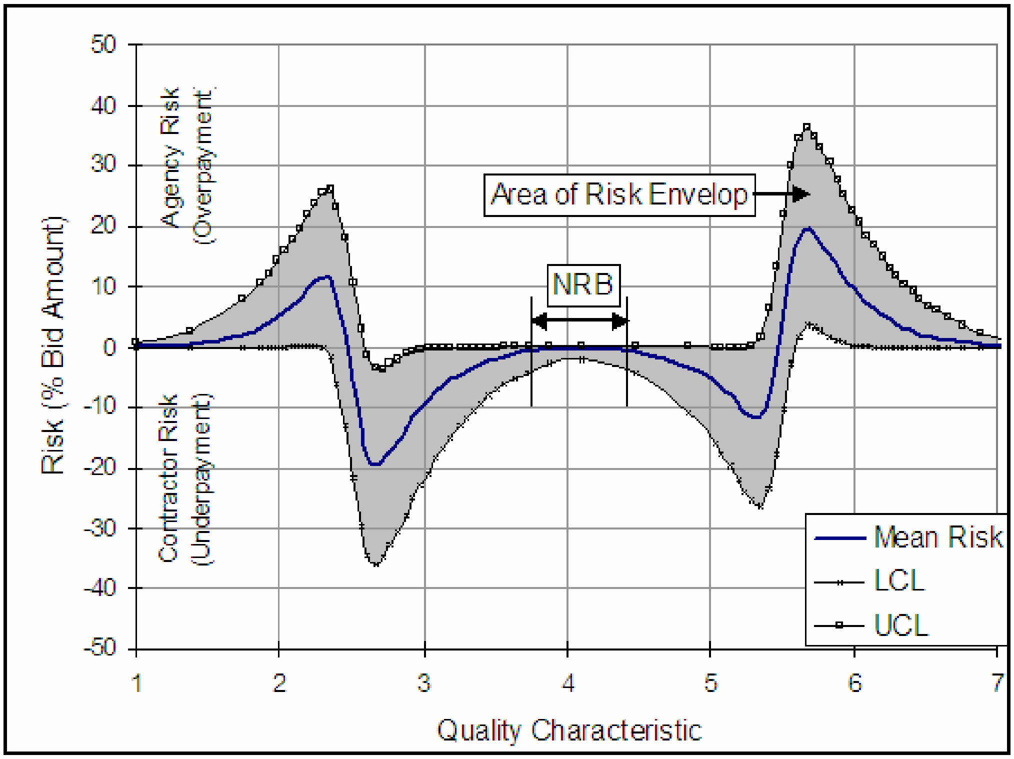


Key: LCL: Lower Confidence Limit; UCL: Upper Confidence Limit

Figure 3.5a: Concept of Maximum Positive and Negative Risk

(2) Narrow risk band (NRB): It is defined as the range of the quality characteristic (i.e. voids, AC or Density) within specification limits where width of 90%

confidence interval for risk is less than 5%. Figure 3.5 illustrates this characteristic.



Key: LCL: Lower Confidence Limit; UCL: Upper Confidence Limit

Figure 3.5b: Concept of Narrow Risk Band (NRB) and Area of Risk Envelope

(3) Composite Risk Index (CRI): Visual observation of the risk plot provides vital information about its characteristics. But comparing different risk plots is somewhat subjective. A *Composite Risk Index* (CRI) was developed which can characterize a full risk plot and could be used for objectively comparing risk plots from different scenarios of variabilities as well as different types of specifications. CRI is calculated in two steps. In the first step 200 equidistant quality characteristic values are identified within the specification limits. At each of these points 100 representative evaluations of risk (as done in a Monte Carlo based simulation) are picked and their moment is taken about the zero risk line. Taking moments ensures that higher risk values contribute more to overall risk. A reasonable amount of risk at any point is somewhat acceptable. But higher risks certainly throw a red flag because that may lead to more disputes and strained relationships between the contractor and the agency. Finally all the moment values are averaged. This represents the point risk index. In the second step all the point risk indices are averaged across the specification limit window, giving higher weights to those towards the middle of the window. Varying weights penalize a specification or scenario in which higher payment risks are expected, even when the contractor is quite close to

the target or the middle of the window. The weights were determined by extensive analysis with the goal of keeping CRI sensitive to spatial location of the data while balancing its effect on CRI because of other factors.

The set of 16 experimental runs were repeated 4 times to obtain replicate values for calculation of standard errors. Table 3.2 gives the mean values of the abovementioned parameters for each experiment. The change that is observed in the risk level as a result of the parameter value going from the lower to upper level can be quantified using the principles of experimental design. This change is technically termed as "effect of parameter." Table 3.3 shows the calculated effects for each of the parameters. Effects related to a single parameter (i.e., X1, X2, etc.) are referred to as a *main effect* of that parameter. When two or more parameters are involved then it is referred to as an interaction effect.

A negative effect for maximum positive risk means that the maximum risk for the agency will decrease with an increase in that parameter value. Conversely, a negative effect for maximum negative risk would mean that maximum risk for the contractor would increase. Some of the conclusions that can be derived from the table of effects are:

- Effect of increase in production variability on the *maximum* positive risk is negative, and on the maximum negative risk the effect is positive. Therefore, the increase in production variability actually brings the risk down for the agency as well as the contractor. Because of higher variability in production the contractor will rightfully receive lower payment. So, although the payment will be lower it is justified and hence the risk of incorrect payment is low. Another way to look at it would be that increased production variability tends to desensitize the effect of measurement variability, as mentioned previously.
- The extent of the narrow risk band decreases with increase in production variability. Recall that the narrow risk band represents the area where maximum pay is deserved (100 PWL is achieved) and where the probability of an estimated PWL below 100 is negligible. In other words, if production variability is high, then even for production averaging in the middle of the specification limits there is possibility of payment risk. When the production variability is low, then maximum pay will almost certainly be awarded when it is deserved.
- Looking at the main effect of measurement variability on maximum positive risk may indicate that MPR goes down as measurement variability becomes higher. But there is another factor that must be considered here. Bias has the most significant effect on MPR and bias has significant negative interaction with effect of measurement variability. Also, the main effect of bias is positive and much greater in magnitude than that of measurement variability. Therefore, the interaction effect coming from bias can overshadow the effect of measurement variability on MPR. When plots with zero (or low) bias are observed it is clear that an increase in measurement variability increases MPR, especially when production variability is low. The same logic holds for MNR as well.
- Bias has significantly higher effect on the maximum positive and negative risk, both of which increase with increase in bias.

- Increase in bias also reduces the width of narrow risk band. However the effect is relatively much smaller than that caused by increases in measurement variability or production variability.
- Increase in number of samples in the job, N, appreciably narrows the 90% confidence interval. This would also translate into a slight increase in NRB if the envelope is near the zero risk line. The decrease in the magnitude of risk generally decreases with the square root of N, as intuitively expected.
- The interaction of production and measurement variability does not have appreciable effect on maximum positive or negative risk.
- Production variability and bias have an appreciable effect on maximum positive and negative risk, but a much smaller effect on the extent of the narrow risk band.

During the replicate runs it was observed that the risk values, as well as all the risk plot characteristic values, had absolutely no variation up to the second decimal place. This provided confidence that the results obtained were very repeatable, i.e., effect estimates are very precise and would not vary from one run to another. This is not a trivial matter, since simulation programs can sometimes be affected by imperfections in random number generation and the number of simulations required for a stable result will vary from problem to problem.

Table 3.2: Mean values of risk characteristics for data with bias

Run Number	σ_{prod}	σ_{meas}	Bias			n	NRB	Max Neg Risk	Max Pos Risk	CRI	Area
			Contractor	District	Thparty						
1	0.3	0.2	0	0	0	15	1.38	-9.02	8.84	2.18	12.88
2	0.8	0.2	0	0	0	15	0.00	-8.99	8.88	6.10	21.73
3	0.3	0.5	0	0	0	15	0.50	-12.48	9.27	5.72	17.99
4	0.8	0.5	0	0	0	15	0.00	-9.40	8.70	7.33	22.63
5	0.3	0.2	0.4	-0.4	-0.4	15	1.23	-27.60	23.25	9.88	16.66
6	0.8	0.2	0.4	-0.4	-0.4	15	0.00	-17.32	17.12	10.15	22.60
7	0.3	0.5	0.4	-0.4	-0.4	15	0.00	-22.60	18.54	9.72	25.65
8	0.8	0.5	0.4	-0.4	-0.4	15	0.00	-14.69	14.38	9.56	26.14
9	0.3	0.2	0	0	0	40	1.64	-5.24	5.60	1.30	7.88
10	0.8	0.2	0	0	0	40	0.00	-5.33	5.33	3.90	13.22
11	0.3	0.5	0	0	0	40	1.01	-9.41	5.96	4.79	11.03
12	0.8	0.5	0	0	0	40	0.00	-6.40	5.44	5.16	13.74
13	0.3	0.2	0.4	-0.4	-0.4	40	1.46	-24.18	21.55	9.50	10.69
14	0.8	0.2	0.4	-0.4	-0.4	40	0.00	-13.78	13.70	8.98	13.93
15	0.3	0.5	0.4	-0.4	-0.4	40	0.58	-17.41	14.92	8.33	15.91
16	0.8	0.5	0.4	-0.4	-0.4	40	0.00	-10.76	10.82	7.57	15.97

Key: District – Agency; NRB – Negative risk band; CRI – Composite Risk Index.

Table 3.3: Effects of the parameters on payment risk for biased data

Parameter		NRB	Maximum Risk		CRI	Area
			Positive	Negative		
X1	**Sig-Prod**	-1.28	-2.95	5.16	0.92	3.91
X2	**Sig-Dev**	-0.68	-2.03	1.04	0.77	3.69
X3	**Bias**	-0.05	9.53	-10.26	4.65	3.30
X4	**n**	0.37	-3.21	3.70	-1.39	-7.99
X1 X2		0.44	0.61	0.00	-0.65	-1.93
X1 X3		0.11	-2.62	3.65	-1.21	-1.47
X1 X4		-0.13	-0.24	-0.17	-0.49	-1.07
X2 X3		-0.05	-2.21	3.32	-1.61	1.27
X2 X4		-0.04	-0.23	0.10	-0.23	-0.95
X3 X4		0.02	0.13	0.32	0.16	-0.65
X1 X2 X3		-0.01	0.82	-1.53	0.48	-0.22
X1 X2 X4		-0.20	0.27	-0.17	0.04	0.48
X1 X3 X4		0.04	-0.18	-0.12	0.15	0.29
X2 X3 X4		0.03	-0.29	0.44	-0.23	-0.37

Key: NRB – Negative risk band; CRI – Composite Risk Index; Sig-Prod – Standard Deviation of Produced Material; Sig-Dev – Standard Deviation of Measurement Device.

Use of SRA as a Research Tool

At the standard user's level, sensitivity analyses can easily be conducted with SRA to examine effecs of a number of user-modifiable parameters on specification risk, as described in the previous section. Parameters which are not user-modifiable from the standard program interface are fixed at those used in IDOT's current ERS specification. For example the SRA program does not allow changes to the pay factor equation from the standard interface. But the advanced user of SRA can modify the code as required for the analysis, thereby enabling almost unlimited specification approaches to be analyzed. Thus, the analyst would be able to compare the risk characteristics of entire specification systems. The following is an example of a research-level risk analysis conducted using SRA.

Advanced Risk Analysis with SRA

Example #1: Pay Factor Equation

In 2004 IDOT modified its pay factor equation from (3) to (4).

$$PF = 0.55 + 0.5 * PWL \quad \text{(But, if } PF>1.03\text{, then } PF = 1.03\text{)} \tag{3}$$

$$PF = 0.53 + 0.5 * PWL \tag{4}$$

Where:

PF = Pay Factor (%)
PWL = Percent Within Limits

Thus, starting in 2003 the pay factor equation was modified to be more stringent (lower pay for a given PWL level). The payment cap was no longer used, since the maximum pay given directly by the formula was 103%.

A series of simulation runs were carried out using SRA to investigate the impact that this change would have on IDOT ERS pay factors and the associated payment risk. The example presented here uses combined and measurement variability, estimated from several ERS demonstration projects. First, a brief description of the method for estimating combined and measurement variability is presented, with the help of an example. In Table 3.4 the columns labeled as "Measured" are the field core densities measured in ERS projects. For each job the density values are averaged to arrive at a mean. Then the difference between the first mean (contractor mean for job 1 in this case) and the mean for each job is subtracted from the measured density to obtain a normalized density. By normalizing the means from each job, it is then possible to combine or 'pool' the data sets in order to obtain a robust estimate of the variability in measured pavement density (combined variability of measurement, production, and laydown). In this example the combined variability was found to be 0.85. In the full analysis, hundreds of such measurements have been taken from ERS demonstration projects in Illinois over the past five years, as will be summarized below.

Table 3.4: Example calculations for estimating combined variability

Job	Contractor Density			District Density		
	Measured	Mean	Normalized	Measured	Mean	Normalized
1	93.6	93.4	93.6	93.8	93.6	93.5
	93.1		93.1	92.6		92.3
	94.7		94.7	94.7		94.4
	93.3		93.3	93.8		93.5
	92.7		92.7	92.8		92.6
	93.3		93.3	93.9		93.6
	93.0		93.0	93.5		93.2
	94.4		94.4	94.7		94.4
	92.6		92.6	93.1		92.8
	92.9		92.9	93.6		93.3
2	91.5	92.9	92.0	90.9	92.6	91.7
	91.5		91.9	91.9		92.6
	93.6		94.0	94.0		94.7
	94.3		94.7	94.1		94.9
	93.3		93.8	92.5		93.3
	92.9		93.4	91.5		92.2
	93.0		93.4	92.6		93.3
	93.2		93.6	93.6		94.3
	Combined Standard Deviation = 0.85					

Table 3.5 presents an example for calculation of the variability associated with the measurement of density. In this case, only a small number of pairs of data has been presented to quickly illustrate the concept. In reality mostly the number of measurements would be anywhere between 100 and 250. The columns labeled "Contractor" and "District" (the agency) report the densities measured by each of these two parties. The next column is the difference of these two densities. It should be noted that each pair of densities is measured on split samples. Therefore the difference between them is expected to be predominantly composed of measurement variability and bias present in the data. The mean value of the differences for each job gives an estimate of the bias present. To remove this relative bias, the mean of differences is subtracted from the individual differences found to produce normalized differences. The normalized differences are then sorted to identify outliers. Outliers are generally identified as those normalized difference values which lie farther than three times the standard deviation of the normalized differences from their mean. Since the density measurements are expected to be normally distributed, the differences also should be normally distributed and hence this definition of outliers is retained. The standard deviation of the normalized differences from which outliers have been removed provide an estimate of the measured variability as discussed in the preceding section. In the example presented here the standard deviation of the normalized differences was found to be 0.50. Therefore, measurement variability for this set of paired tests can be found as follows.

$$\sigma_{comb} = 0.50$$

$$\sigma_{meas} = \frac{\sigma_{comb}}{\sqrt{2}} = \frac{0.50}{\sqrt{2}} = 0.35$$

As stated earlier actual measurement variability is estimated using hundreds of such data points.

Table 3.5: Example calculations for measurement variability

Job	Contractor	District	Difference	Mean of Diff.	Normalized Diff.	Norm. Diff. In Asc. Ord.	Std. Dev of NDAO
1	93.6	93.8	-0.2	**-0.3**	0.1	-0.7	**0.50**
	93.1	92.6	0.5		0.8	-0.7	
	94.7	94.7	0.1		0.4	-0.7	
	93.3	93.8	-0.5		-0.2	-0.4	
	92.7	92.8	-0.2		0.1	-0.3	
	93.3	93.9	-0.6		-0.3	-0.3	
	93.0	93.5	-0.5		-0.2	-0.2	
	94.4	94.7	-0.3		0.0	-0.2	
	92.6	93.1	-0.5		-0.3	-0.1	
	92.9	93.6	-0.7		-0.4	0.0	
2	91.5	90.9	0.6	**0.3**	0.3	0.1	
	91.5	91.9	-0.4		-0.7	0.1	
	93.6	94.0	-0.4		-0.7	0.1	
	94.3	94.1	0.2		-0.1	0.3	
	93.3	92.5	0.8		0.5	0.4	
	92.9	91.5	1.4		1.2	0.5	
	93.0	92.6	0.4		0.1	0.8	
	93.2	93.6	-0.4		-0.7	1.2	

Key: NDAO – Normalized Difference in Ascending Order.

Using data from several ERS demonstration projects from the years 2000 to 2002, the following measures of in situ density variation were obtained:

- Combined variability = 1.15%, where '%' refers to percentage of max theoretical specific gravity, G_{mm}
- Measurement variability = 0.56%

For combined variability, the highest value for a given project was found to be 1.45% and the lowest was 0.85 %.

The total number of samples (N) cored for measuring density in ERS demonstration projects have varied between 100 and 250. Therefore, different simulations were run with N = 50, 100 and 200. Therefore, nine simulations were run using the old pay factor equation and then repeated using the new PF equation. Table 3.6 presents all the combinations of parameters used in the analysis.

Table 3.6: Parameter values used in simulation runs for the analysis

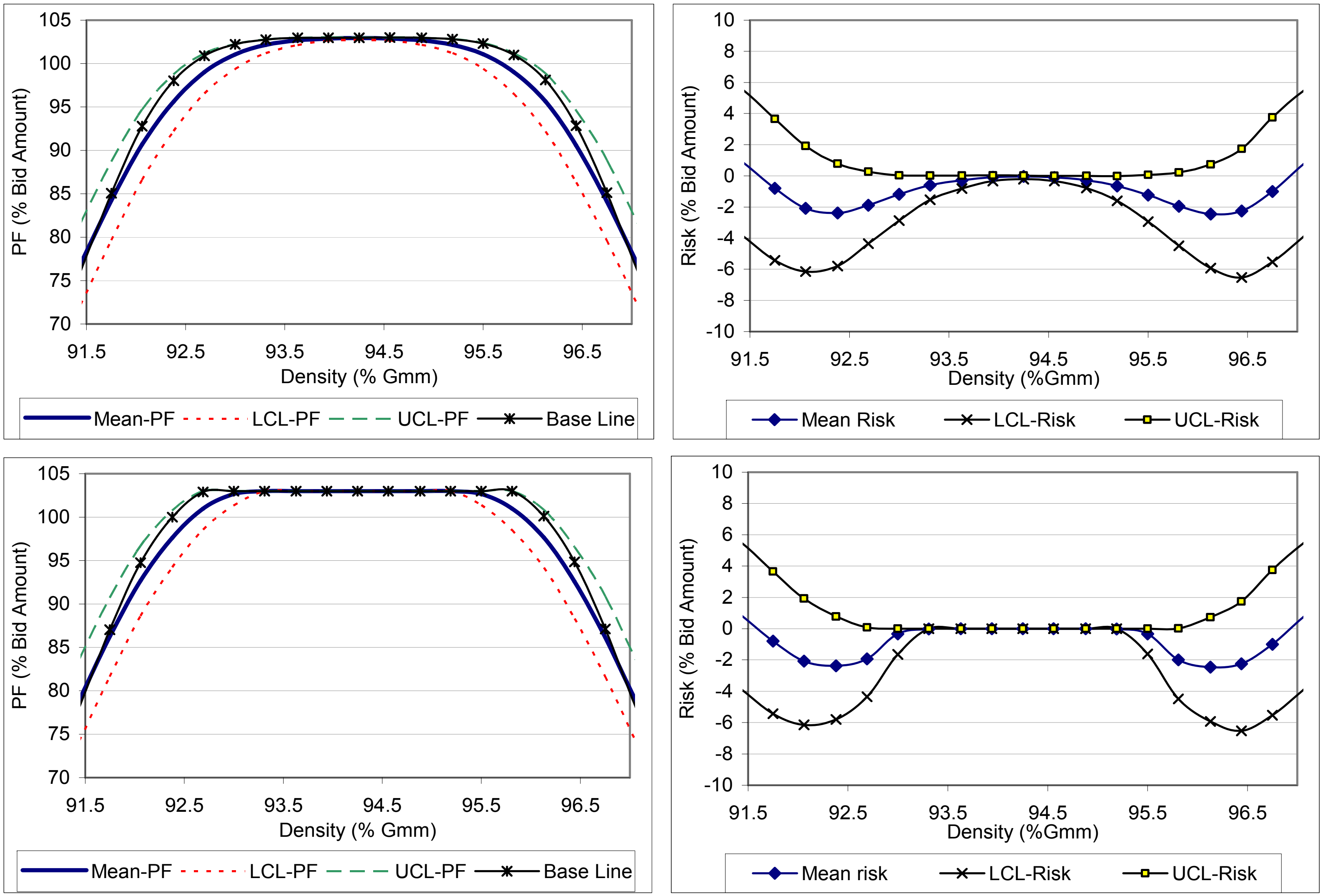
Sim. Run #	Combined Variability	Production Variability (σ-p)	Measurement Variability (σ-m)	Total # Samples (N)
1	0.85	0.64	0.56	50
2	1.15	1.00	0.56	50
3	1.45	1.34	0.56	50
4	0.85	0.64	0.56	100
5	1.15	1.00	0.56	100
6	1.45	1.34	0.56	100
7	0.85	0.64	0.56	200
8	1.15	1.00	0.56	200
9	1.45	1.34	0.56	200

Results of Pay Factor Analysis

Figures 3.6-3.14 present the risk plots, from the simulation runs evaluating IDOT's Superpave asphalt ERS, under the older and newer pay factor equation. One challenge of simulation modeling is developing a systematic scheme for the analysis of the enormous number of results that are generated. The present analysis is further complicated by the fact that some of the simulation outputs, such as the risk factor, are new concepts and therefore not familiar to most readers. To assist the reader, a detailed summary is presented to outline and describe the scheme used herein to present standard "sets" of simulation results:

- The plots on the left half of the page show the actual (simulated) mean pay determined by the simulation for all mean densities, along with its 90% confidence interval (CI). They also show the base line pay or ideal pay, in a thick solid line. With this plot it can be readily observed how the average and +/- 90% CI of pay levels would vary for different projects (with different variabilities) as a function of the mean density. A baseline pay below the mean pay line represents agency risk, as the agency would be paying above the correct pay in this instance. The parts of the plot where these lines either cross or merge with each other represents balanced risk for both parties. However, if the ideal pay line goes above the mean pay line there is higher probability that the contractor would be underpaid relative to the correct pay, i.e., contractor risk. The probability of risk in either case can be observed based on the distance of the ideal pay line from the confidence intervals for the mean pay.
- The plots on the left half of the page are the risk plots (as described earlier in the report), which result after replotting the curves on the left after normalizing relative to the baseline (or correct) pay.
- The first row of plots in each set of plots (top two) represents the analysis using the new pay factor equation and the second row of plots (bottom two) correspond to the old pay factor equation.

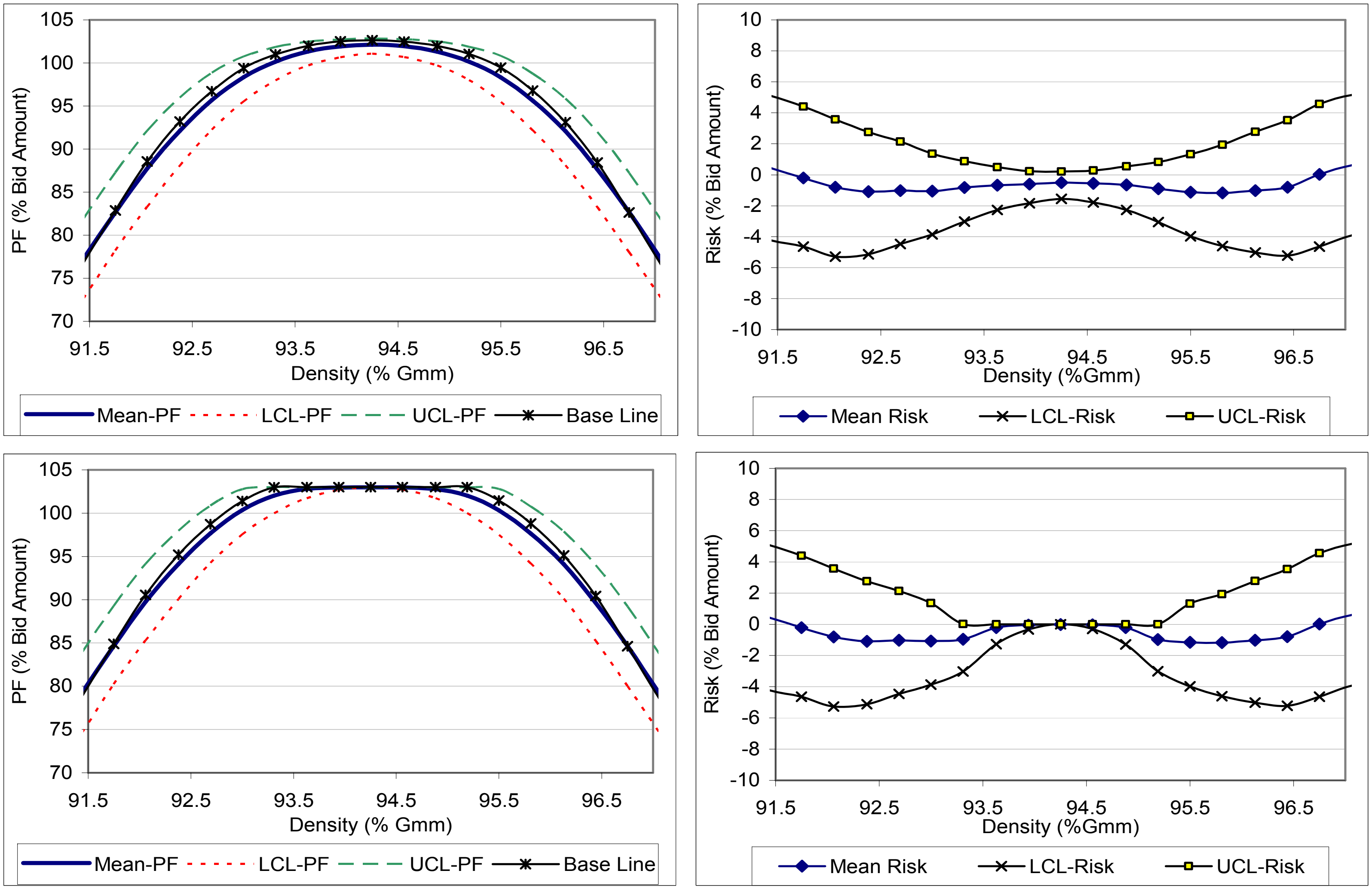
New PF Equation Measurement Variability = 0.56 Production Variability = 0.64 N = 50



Old PF Equation Measurement Variability = 0.56 Production Variability = 0.64 N = 50

Figure. 3.6: Analysis results for low production variability and low number of specimens: old vs. new PF equation

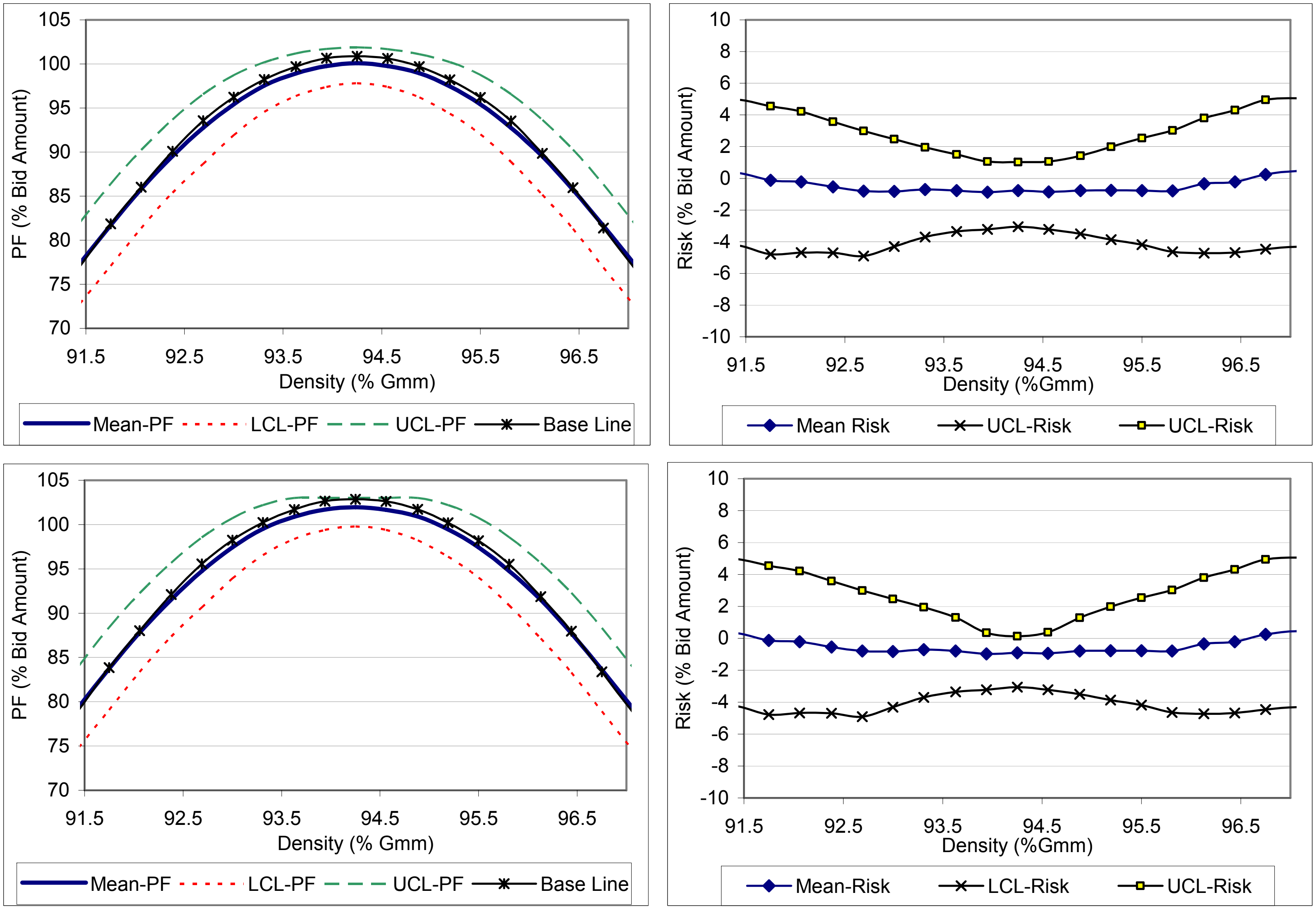
New PF Equation Measurement Variability = 0.56 Production Variability = 1.00 N = 50



Old PF Equation Measurement Variability = 0.56 Production Variability = 1.00 N = 50

Figure. 3.7: Analysis results for medium production variability and low number of specimens: old vs. new PF equation

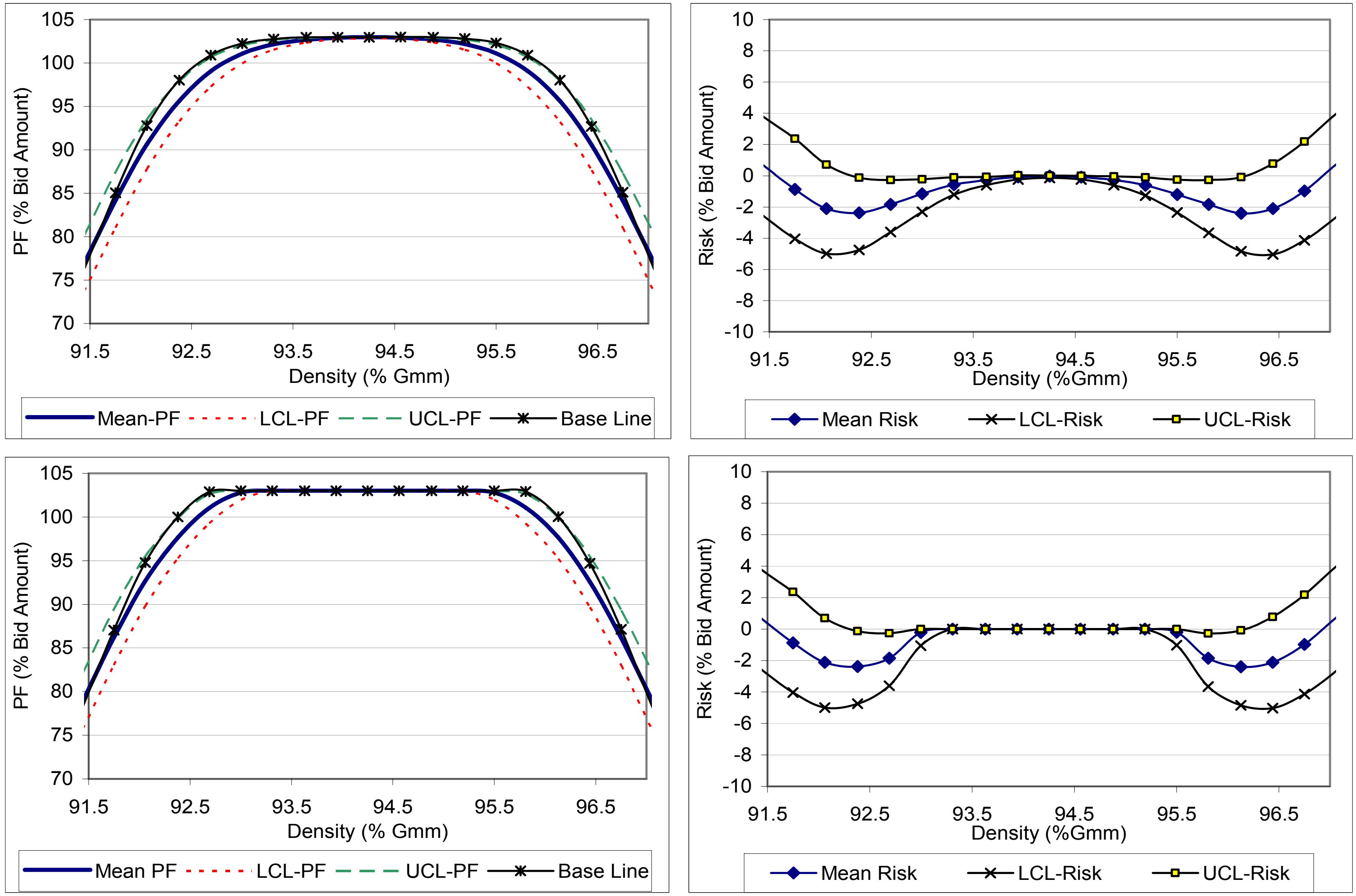
New PF Equation Measurement Variability = 0.56 Production Variability = 1.34 N = 50



Old PF Equation Measurement Variability = 0.56 Production Variability = 1.34 N = 50

Figure. 3.8: Analysis results for high production variability and low number of specimens: old vs. new PF equation

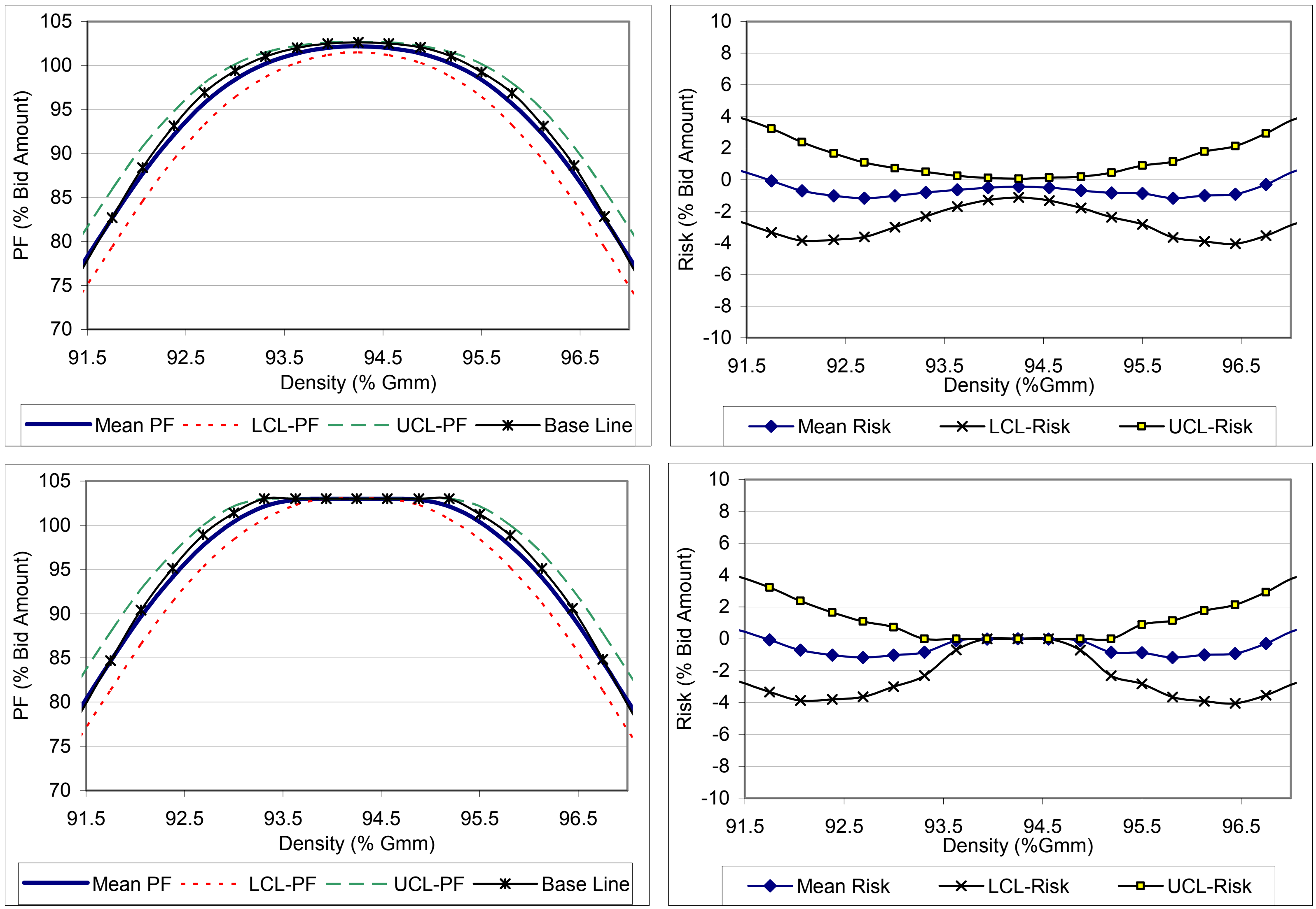
New PF Equation Measurement Variability = 0.56 Production Variability = 0.64 N = 100



Old PF Equation Measurement Variability = 0.56 Production Variability = 0.64 N = 100

Figure. 3.9: Analysis results for low production variability and medium number of specimens: old vs. new PF equation

New PF Equation Measurement Variability = 0.56 Production Variability = 1.00 N = 100



Old PF Equation Measurement Variability = 0.56 Production Variability = 1.00 N = 100

Figure. 3.10: Analysis results for medium production variability and medium number of specimens: old vs. new PF equation

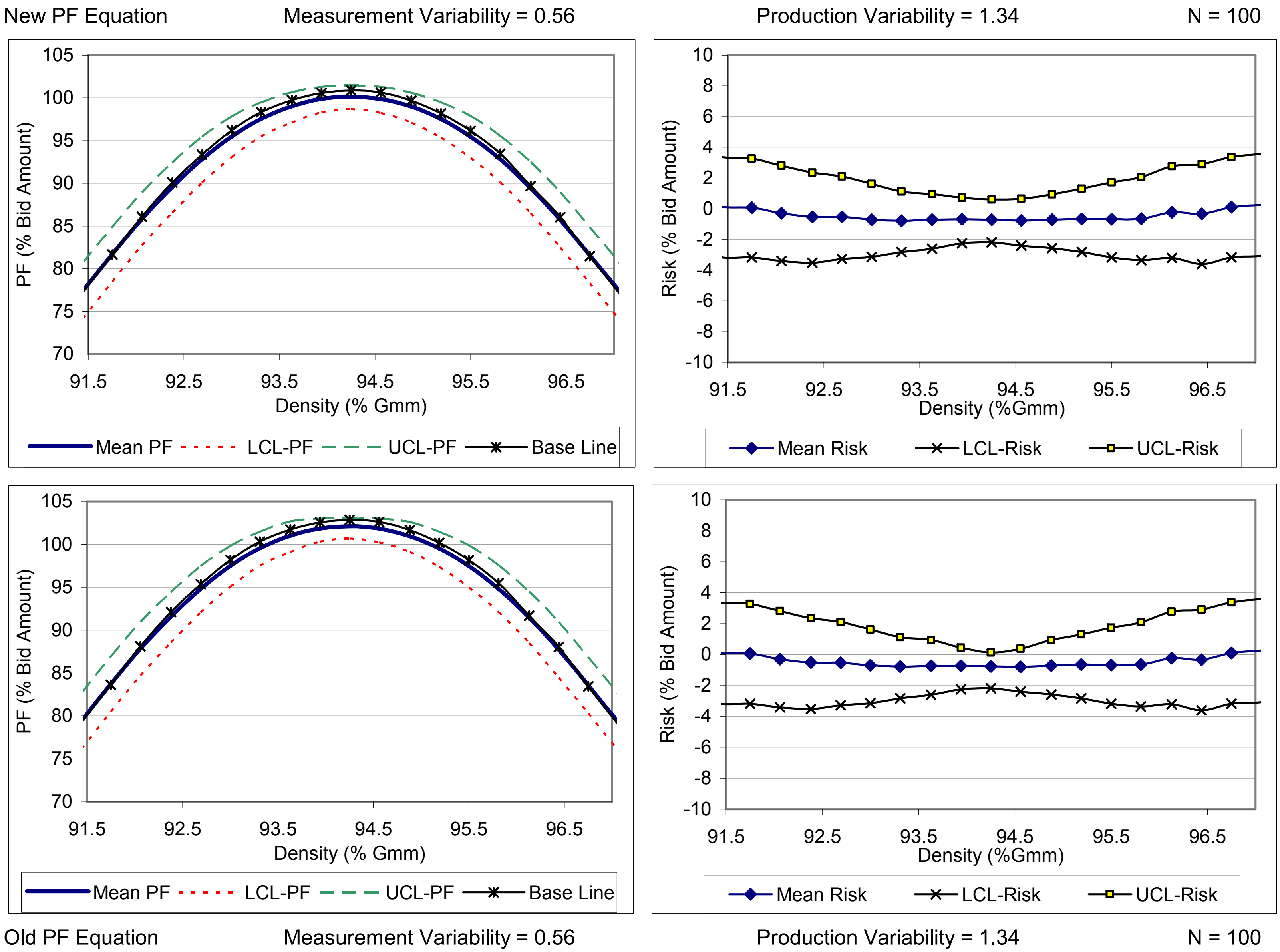
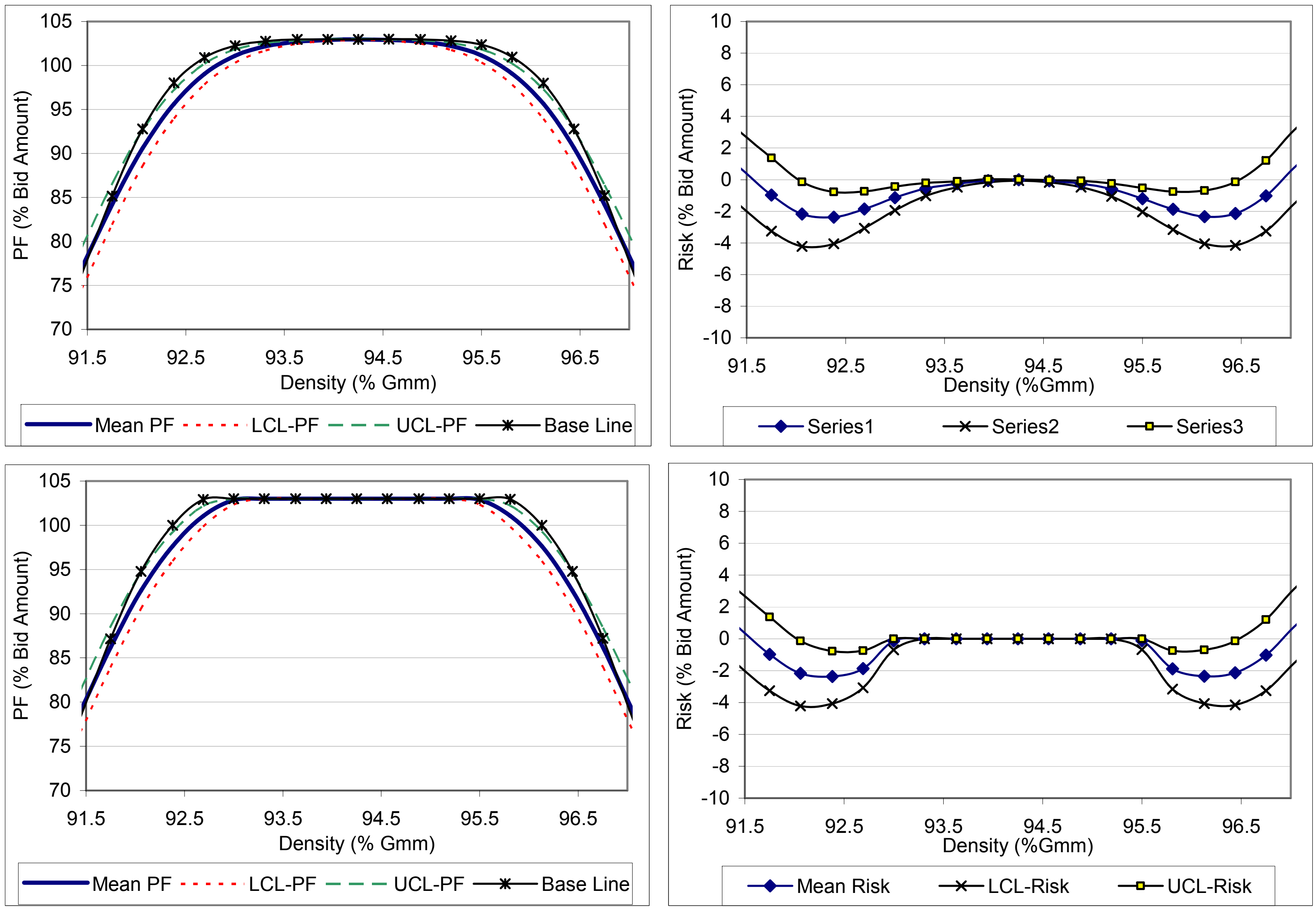


Figure. 3.11: Analysis results for high production variability and medium number of specimens: old vs. new PF equation

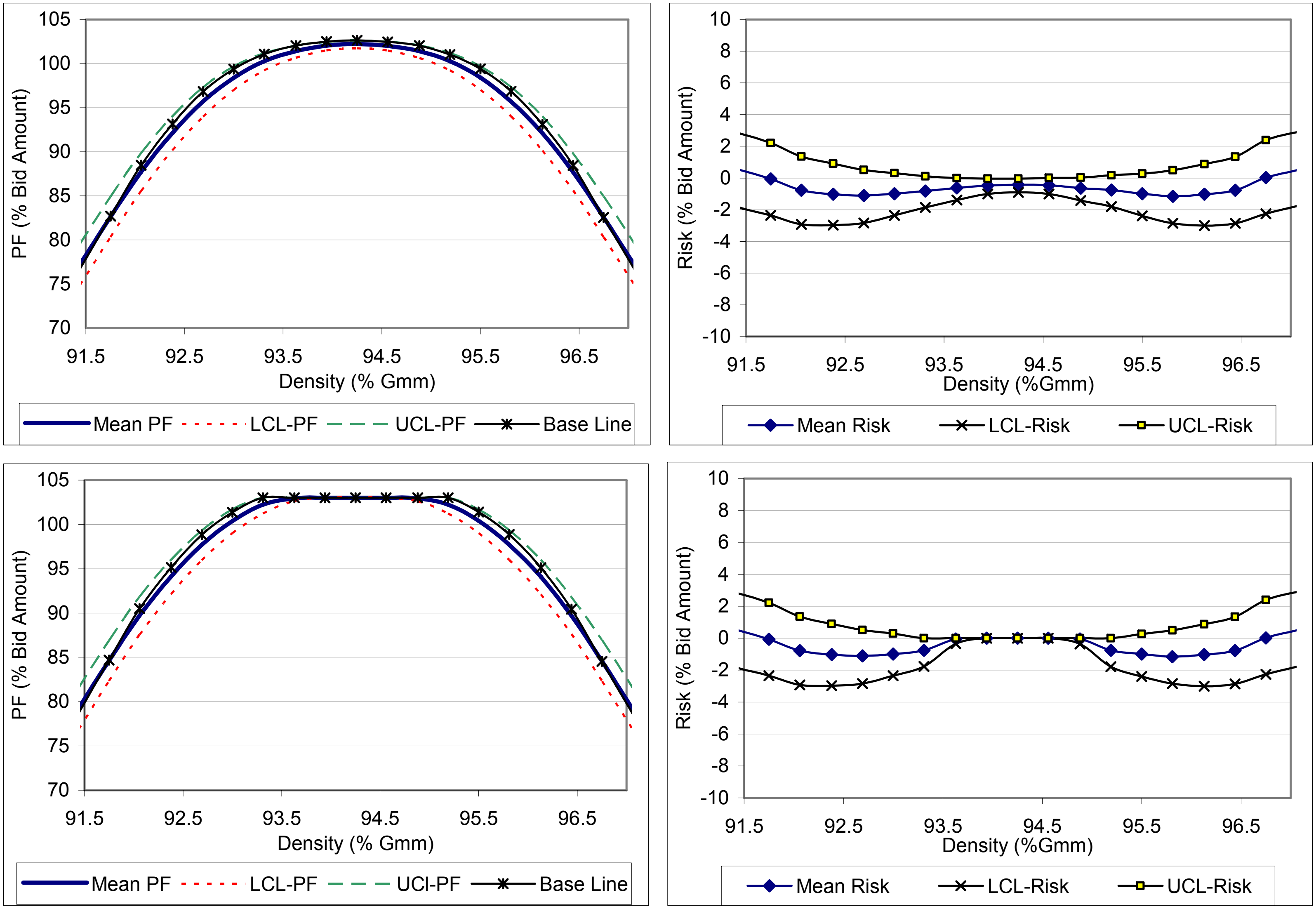
New PF Equation Measurement Variability = 0.56 Production Variability = 0.64 N = 200



Old PF Equation Measurement Variability = 0.56 Production Variability = 0.64 N = 200

Figure. 3.12: Analysis results for low production variability and large number of specimens: old vs. new PF equation

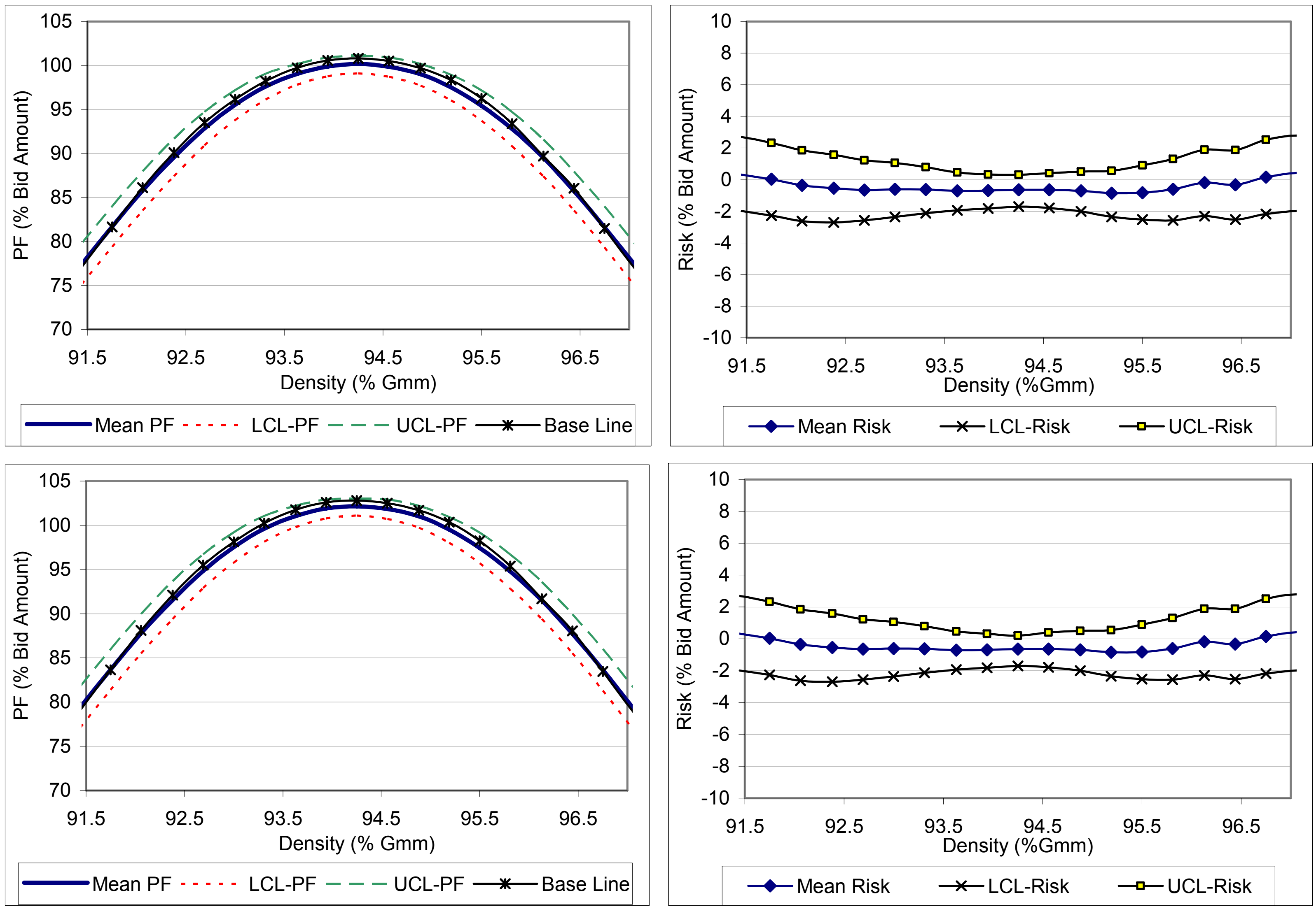
New PF Equation Measurement Variability = 0.56 Production Variability = 1.00 N = 200



Old PF Equation Measurement Variability = 0.56 Production Variability = 1.00 N = 200

Figure. 3.13: Analysis results for medium production variability and large number of specimens: old vs. new PF equation

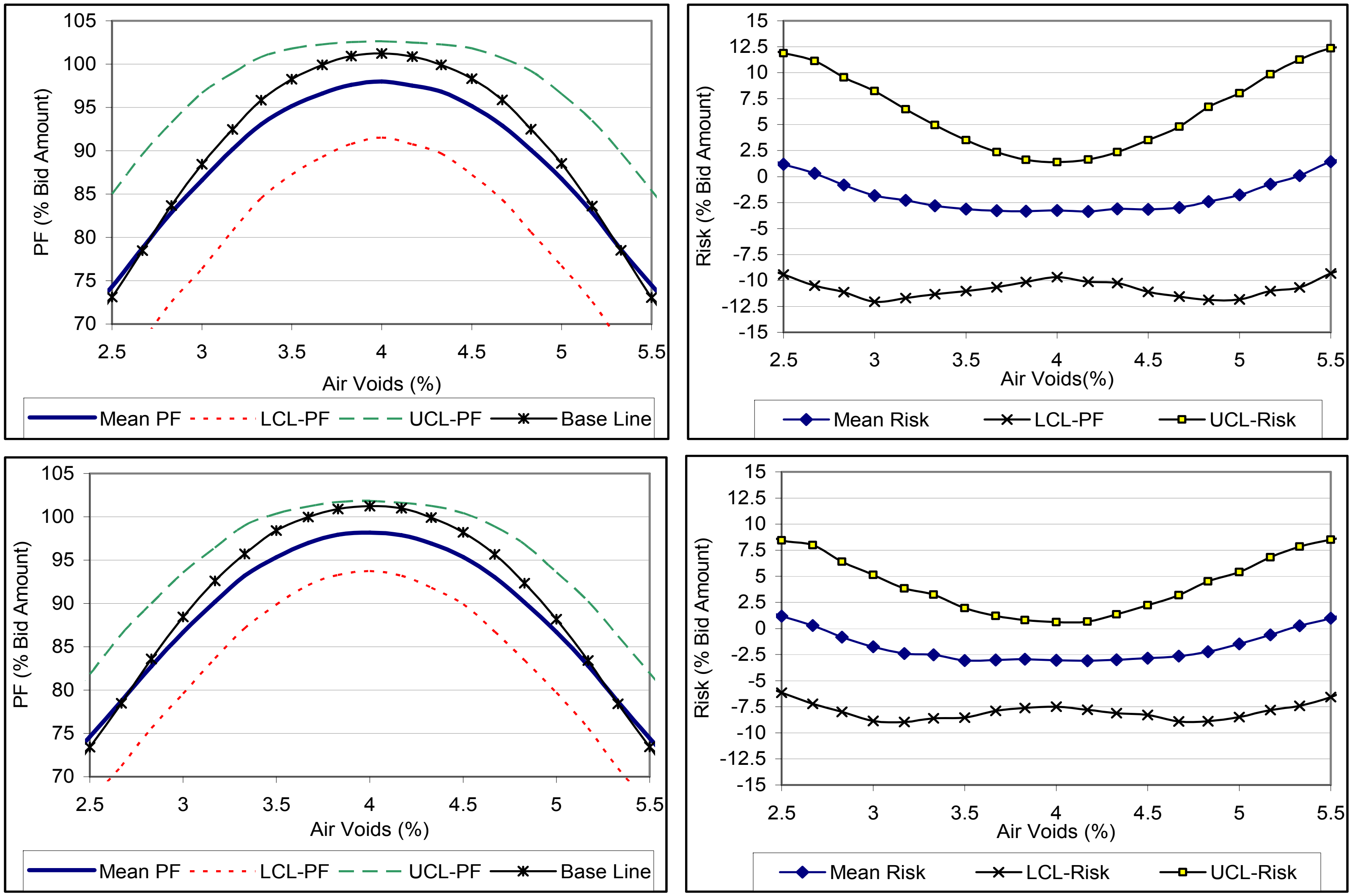
New PF Equation Measurement Variability = 0.56 Production Variability = 1.34 N = 200



Old PF Equation Measurement Variability = 0.56 Production Variability = 1.34 N = 200

Figure. 3.14: Analysis results for high production variability and large number of specimens: old vs. new PF equation

Measurement Variability = 0.53 Production Variability = 0.60 N = 10



Measurement Variability = 0.53 Production Variability = 0.60 N = 20

Figure 3.15: Analysis results for air voids: N=10 vs. N=20

Discussion of Pay Factor Analysis Results

The simulation results presented in Figures 3.6-3.15 can be summarized as follows:

- As expected, the plots show a clear downward shift in the actual payment, baseline payment, and upper and lower confidence limits for the new pay factor equation relative to the old PF equation. This can be observed by comparing either the vertically aligned plots on the left-hand side of a given page, or the vertically aligned plots on the right-hand side of a given page.
- There is an appreciable difference in the width of the narrow risk band. The narrow risk band was reduced in all the cases. As apparent in the left-hand plots, in the old pay factor equation the final pay was capped at 103%, although the maximum pay calculated using the equation was 105%. Therefore, once the ideal pay or baseline (represented by the thick solid line in the middle plots) reached 103%, any further increase in density towards the middle of the spec limits (and thus improved the PWL), did not improve the PF above 103%. Therefore, in these areas there is no difference between the baseline pay and the actual pay leading to low or no risk, which constitutes the narrow risk band. But with the new pay factor equation, pay cap is no longer in effect. Therefore, the difference between baseline pay and actual pay, if any, would continue until maximum pay is reached.
- In the case of either of the pay factor equations, as the number of specimen tested (N) is increased, the confidence interval for pay as well as for risk becomes narrower around the corresponding mean values. Therefore, the advantage of using more samples to reduce probability of risk remains intact with the modification of the pay factor equation.
- It should be noted that in the case of analysis performed on voids (Figure 3.15), the simulation results are presented in a different arrangement compared to that used for the density simulations (Figures 3.6-3.14). The two rows of vertically aligned plots on the page for Figure 3.15 allow a comparison of sample size, corresponding to cases with N=10 and N=20, respectively. In this case all the plots correspond to the new PF equation. Also the measurement and production variabilities for these simulation runs were calculated using the voids data from ERS demonstration projects (as described in the previous section) and found to be 0.53 and 0.60, respectively.
- The most prominent feature in the plots of voids in Figure 3.15 are, that although the mean risk is similar in magnitude, the confidence intervals are much wider and the peak of confidence interval goes as high as 12.5% pay. This can also be deduced from the left-hand side plots which show that the confidence interval on the actual pay is comparatively much wider. Therefore, the difference between the ideal pay line and the confidence interval lines is also greater. Although the tolerance limits for density and voids comparison are the same, the number of samples tested in the case of voids is generally much smaller, leading to higher risk levels.
- The higher ratio of measurement variability relative to production variability in the case of voids also creates a larger confidence interval on payment risk.

The key conclusions that can be deduced from this analysis are:

(1) *The absence of a payment cap in the new PF equation has the effect of reducing the narrow risk band for the specification.* This may tend to increase the likelihood of a

dispute between parties, since the contractor risk is higher. The main motivation for changing the pay factor equation was to bring the actual project pay more in line with pay levels deemed appropriate, based upon an IDOT internal review of demonstration projects. To this end, other methods of arriving at a lower pay factor can also be explored, which may not have the same effect of reducing the narrow risk band. One example would be to alter the way in which pay factors are combined, such as using the lower of the two plant test pay factors instead of using the average. Another possibility would be to use a multiplicative approach to combining some or all of the pay factors, which would avoid the "averaging out" of a low pay factor with one or two higher pay factors. Currently IDOT uses 30% of the asphalt content PF, 30% of the PF on voids from gyratory compacted specimens in the plant, and 40% of the PF for field density as measured from cores.

(2) Payment risks are higher for air void measurements as compared to field density. This is due to two factors: (i) the reduced number of samples tested, and (ii) the higher ratio of measurement variability as compared to production variability. Since the production variability levels seem reasonable, the primary area of concern is the measurement variability for this parameter. Efforts to reduce the measurement variability associated with plant voids will pay large dividends in terms of reducing payment risks and therefore reducing the possibility for disputes.

Example #2: Reducing Sample Size of In-Situ Density Testing

Larger sample sizes provide higher confidence in computed averages and standard deviations and thus better estimates of pay factor. This means that payment risk also would be lower, as has been demonstrated through sensitivity analysis. But larger sample size means more sampling and testing which translates into higher personnel needs and higher costs. In addition, for the case of in-situ density measurement, the larger sample produces destructive core holes in the pavement, which even though filled with patching material, will not perform as well as undisturbed pavement. IDOT as well as the highway contractors have shown keen interest in exploring the possibility of reducing the sample size in the case of in-situ density testing, if possible, without significantly increasing risk.

Since payment risk is affected by several factors, e.g, sample size, tolerance limits, and quality level analysis procedure, there could be several ways to adjust the other factors so that the risk can be maintained within a certain acceptable level. Following are two such cases which demonstrate that it is possible to reduce sample size, although to a limited degree, without appreciably increasing payment risk.

Exploratory Analysis for Reduced Sample Size

This section presents the results of two analyses aimed at exploring ways to reduce sample size in IDOT's asphalt ERS. The two approaches are:

(1) Reduction of contractor sample size, similar QA testing amount

(2) Reduction of number of cores across mat from 5 to 3.

Approach 1: Reduction of contractor sample size and similar QA testing amount

Figure 3.16 (a) shows a typical risk plot when analysis is done for in-situ density. The results correspond to the 2003 ERS used by IDOT. Selected typical values that remain unchanged in the analysis are as follows.

- Production variability = 0.40 %G_{mm}
- Measurement Variability = 0.75 %G_{mm}
- Bias = 0.0 for all the three parties

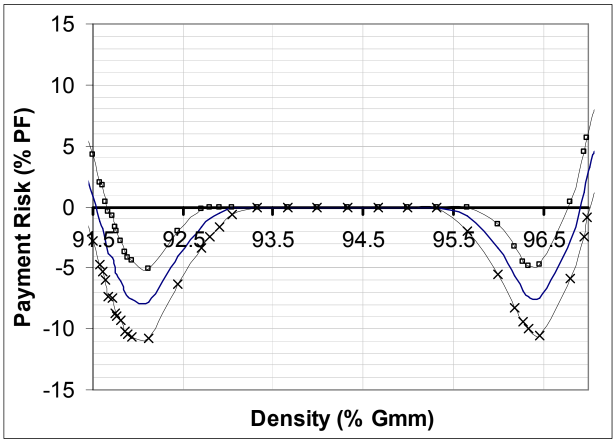
The plot shows risk corresponding to density between 91.5% and 97% G_{mm}. It should be noted that within this range of density most risk is assumed by the contractor, while agency risk is higher for production outside of the specification limits. The regions of higher agency risk are outside of limits selected for plots. Here the focus is on the areas where the vast majority of most production has been found to occur on the ERS projects in Illinois; that is, between the upper and lower specification limits. The plot in Figure 3.16 (a) corresponds to a sample size of 90. The plot shown to the right (Figure 3.16 (b)) shows the risk when the following two significant changes are made:

- The number of samples have been reduced to one fifth i.e. N = 18
- Minimum QA testing has been increased to 100% as compared to 20% in the 2003 end result specifications of IDOT.

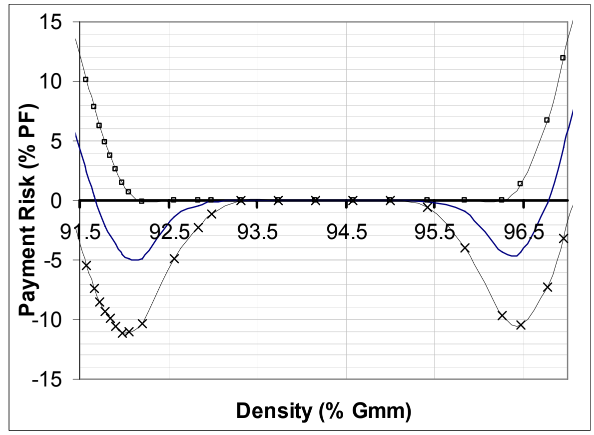
Following are the changes in the risk characteristics apparent from Figures 3.16 (b) as a result of the two aforementioned changes in the 2003 ERS:

- Maximum negative risk (contractor risk) for the 90% confidence interval remains almost the same
- The upper limit of the confidence interval has moved up from -5% to zero in the middle while some portion was shifted into the positive (agency) risk region
- The maximum negative mean risk has decreased from -8% to -5%

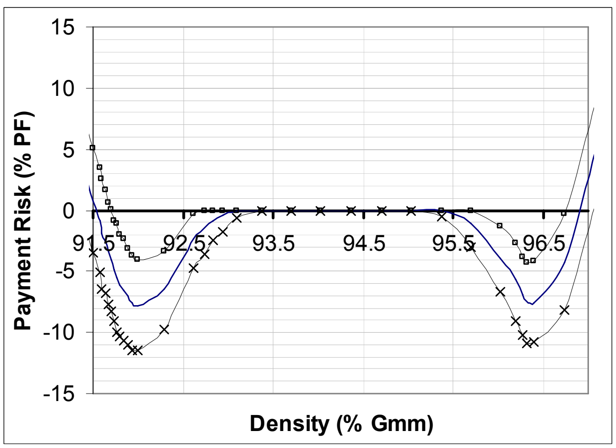
In summary, in the original 2003 specifications for this particular case, the worst-case scenario for a production average within the specification limits would lead to 5 to 11% contractor underpayment of the bid amount, with 90% certainty. After the abovementioned changes (reduction in sample size and narrowing of comparison limit), the underpayment in the worst-case scenario would range between zero and 11% of the bid amount, with 90% certainty. The overall payment risk is appreciably reduced for the contractor, while the agency risk is slightly increased for jobs produced with overall average density within the specification limits. This demonstrates that a much smaller contractor sample size can be used with minimal change in risk for production averaging at a level between the specification limits. Also, although the minimum QA test frequency has been increased by five times, the total number of QA tests required by the state remains unchanged since the sample size was reduced by a factor of five.



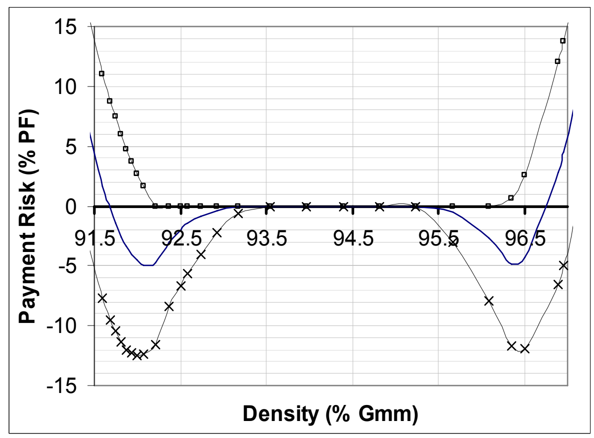
(a) N=90, n=1 Tol. = 1.20 Min. QA = 20%



(b) N=18, n=1 Tol. = 0.80, Min. QA = 100%



(c) N=60, n=1 Tol. = 1.20 Min. QA = 20%



(d) N=12, n=1 Tol. = 0.80, Min. QA = 100%

Figure 3.16: Effect of Reducing Contractor Sample Size and Narrowing Comparison Limits

A similar attempt was made to explore the possibility of reducing the sample size by the same proportion for a smaller job where N = 60. The plots shown in Figures 3.16 (c) and (d) correspond to this. They show that although the negative mean risk has been reduced, the maximum negative risk increased from -11.5% to -12.5%. The increased agency risk at the USL and LSL to a rather high level of 15% makes a reduced sample size of 12 questionable. This trend was even more pronounced for an analysis conducted with N = 30 and a reduced N of 6, indicating a limit beyond which further reduction in sample size leads to unacceptable risk levels. Nevertheless, the above analysis clearly suggests that the SRA tool can be used by the agency to optimize ERS sampling sizes, balancing the relative tradeoffs between reduced sampling and testing and increased agency.

Approach 2: Reduction of number of cores across mat from five to three

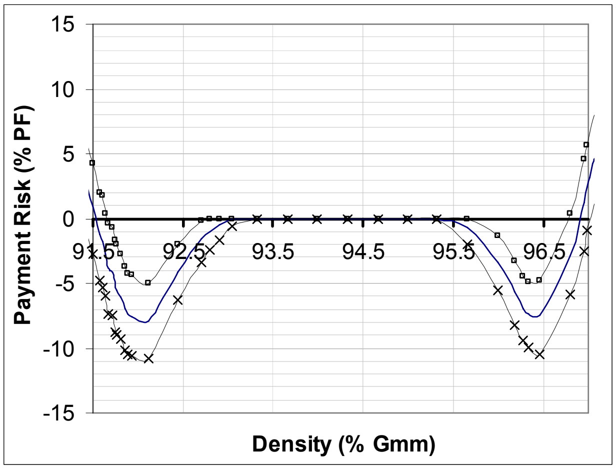
The second strategy for reducing the contractor sample size consists of changing the number of cores to be sampled across the mat for each sublot from 5 to 3 while still performing QA tests on at least one of the three. The other parameter values are the same as those used in the preceding example. Figure 3.17 (a) shows the risk plot expected when adhering to the 2003 ERS, while Figure 3.17 (b) shows the case when only three cores are taken across the mat. For the reduced testing scheme, the maximum negative (contractor) risk has increased only slightly, by -0.5%, as a result of this change. At the same time the upper limit of the 90% confidence interval increased by 1%, thus widening the confidence interval slightly, similar to the results found in approach #1. Once again, while the total number of cores would be greatly reduced (54 as compared to 90), the contractor risk is decreased on average and the agency risk is only slightly increased. Figures 3.17 (c) and (d) show the case when sample size is reduced from 60 to 36, also showing a favorable reduction in sample size with relatively little change in risk.

Discussion of Reducing Sample Size Results

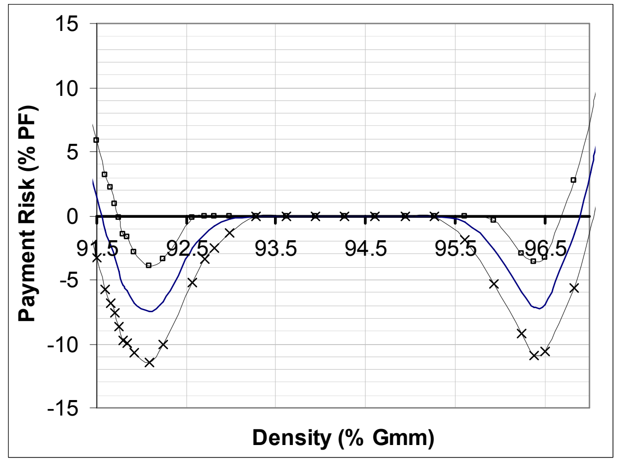
It should be noted that in all of the cases above, some additional risk is assumed by the agency for the case of production averages outside of specification limits. This does not commonly occur in practice since production outside of specification limits represents a PWL below 50 and thus a pay factor below 80%, which would be highly undesirable for the contractor. Nevertheless, the additional risk assumed by the agency must be considered when evaluating the pros and cons of adopting the test reduction strategies outlined above. On the other hand, these possible specification changes have the dual benefit of 1) reducing testing burden (and pavement damage in the case of field density), and 2) slightly reducing contractor average risk for production between the specification limits. Based upon this analysis, it seems that the advantages outweigh the disadvantages and it would be advantageous for both parties involved if the sample reduction strategies were employed.

Summary

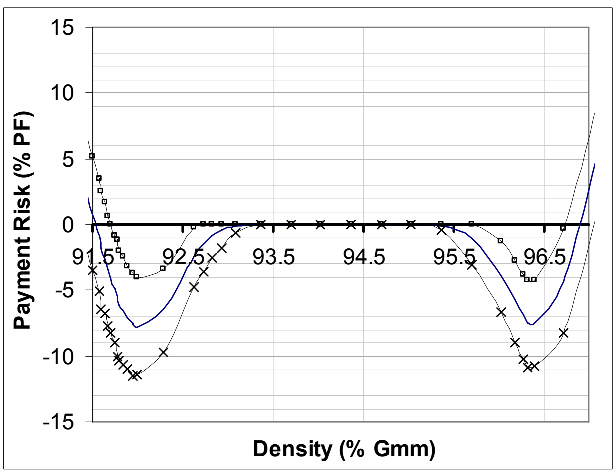
This chapter presented example strategies to demonstrate research-oriented uses of SRA. In general, SRA can be used to develop a better understanding of how changes in individual ERS specification parameters can affect the payment risk for the contractor (seller) and agency (buyer). This knowledge can be used to explore the possibility of developing desirable changes in an existing ERS, such as reducing sample size, reducing risk, optimizing tolerance limits, changing pay factor equations, and the pros and cons of pay factor equations with payment caps. An analysis of the old and new IDOT pay factor equation for Superpave asphalt ERS was conducted, which highlighted the pros and cons of the new pay factor formula. The absence of a pay cap in the pay factor formula appears to create a slight increase in the risk levels in the IDOT ERS system at the higher pay levels. A small, residual contractor risk throughout much of the



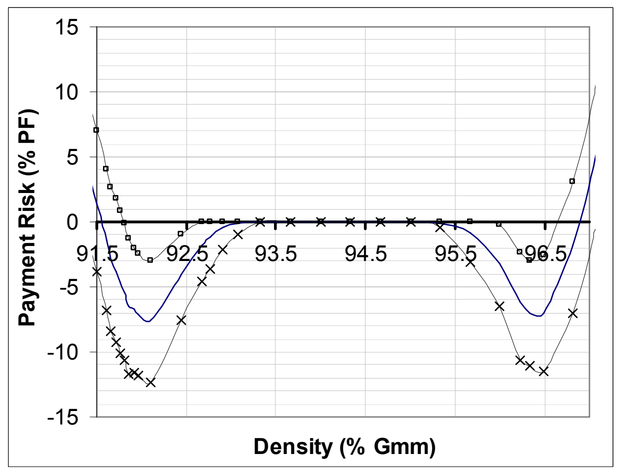
(a) N=90, No. of cores per sublot = 5



(b) N=54, No. of cores per sublot = 3



(c) N=60, No. of cores per sublot = 5



(d) N=36, No. of cores per sublot = 3

Figure 3.17: Another possible strategy for reducing sample size

narrow risk band is present. Specific examples were presented of how IDOT's existing ERS for Superpave HMA could be modified to reduce contractor and agency testing. Two strategies were presented which appear to be promising methods for reducing the number of field cores required, while tending to balance risks between parties more equitably. This would also have the benefit of reducing the amount of pavement damage caused by coring and patching of the new pavement, resulting in enhanced pavement life and possibly even enhanced safety over the pavement's life.

Chapter 4

Summary, Conclusions, and Recommendations

Summary

End-Result Specifications (ERS) for asphalt pavement construction offer potential benefits over method-related specifications. They can be used in conjunction with or replacement of traditional QC/QA specifications as a means to enhance contractor innovation, reduce agency testing burden, and enhance overall pavement quality. Unlike other manufacturing sectors, the measure of pavement quality is not as simple as detecting and quantifying defective items. The quality of pavements is assessed with imperfect measuring tools operated by humans, who may inadvertently or intentionally introduce measurement variability or bias. As a result, the ability to measure quality and assign appropriate payment bonuses and penalties is an imperfect system.

Risk is a natural entity in just about every business enterprise, but in order to properly utilize and administer a contractual process involving risk, one must first have an accurate measure of that risk. In the past, existing methods for balancing risks in pavement construction-related ERS contracts, including the AASHTO approach, did not properly consider all of the factors affecting risk. Therefore, the administration of such contracts has carried the heavy burden of loosely defined specification risk levels.

This report detailed the development of a simulation tool which can be used to analyze specification risk and to develop ERS systems with user-managed risk levels. The program, called Simulated Risk Analysis (SRA), computes the risk of overpayment (agency risk) or underpayment (contractor risk) as a function of many factors, including: number of tests, production and measurement variability, bias, pay formula and pay caps, and specification limits. It also considers the quality assurance and third-party testing scheme used. In general, SRA can be used to develop a better understanding of how changes in individual ERS specification parameters can affect the payment risk for the contractor and agency. This knowledge can be used to explore the possibility of developing desirable changes in an existing ERS, such as reducing sample size, reducing risk, optimizing tolerance limits, changing pay factor equations, and the pros and cons of pay factor equations with payment caps.

An analysis of the old and new IDOT pay factor equation for Superpave asphalt ERS was conducted, which highlighted the pros and cons of the new pay factor formula. Specific examples of how IDOT's existing ERS for Superpave HMA could be modified to reduce contractor and agency testing were presented. Two strategies were presented which appear to be promising methods for reducing the number of field cores required, while tending to balance risks between parties more equitably. This would also have the benefit of reducing the amount of pavement damage caused by coring and patching of the new pavement, resulting in enhanced pavement life and possibly even enhanced safety over the pavement's life.

Limitations of Current Simulation Programs

While the simulation models presented in this report were shown to provide significant insight into the phenomena of payment risks involved with ERS systems, there are clearly limitations to the models. Some of the key limitations are now discussed.

(1) Human factors in decision making: Any highway construction project, or testing program, is subject to human decision making. An example of simple human decision making that is considered in the SRA model is the logic of whether a contractor will accept or reject district data depending on which result is closer to the specification target. At the very best it can be accepted only as a simplified model as compared to how complicated and dynamic human thinking can be in these situations. For example, complex human decision making could arise when a material ERS is used in conjunction with a lane-rental incentive/disincentive contract clause. Quality could possibly be exchanged for expediency towards the end of a project if the magnitude of the lane rental bonuses and penalties greatly exceed the ERS bonuses and penalties when a simple additive or weighted averaging scheme is used for the combined pay factors. It is impossible to develop computer logic to perfectly predict human decision making.

(2) Non-standard practices: All such construction projects are ultimately monitored and run by human beings. Although efforts are made to standardize professional practices like construction, testing, analysis and reporting, slight or appreciable deviations from such practices are not unprecedented. Only the simplest forms of human error and bias can be considered and/or detected, and therefore, it must be acknowledged that risk assessment provides a useful estimate but not an exact value of specification risk. In extreme cases the data may be willingly shaped in a particular way. It is almost impossible for a mathematical or algorithmic model to simulate such phenomena. Rather than attempt to model these complex situations, it is assumed that their occurrence is infrequent and can be minimized through programs of quality compliance, inspection, and auditing.

(3) Normality assumption: Literature indicates that quality characteristics, e.g., in-situ density, air void content, etc., in highway construction projects are generally normally distributed. But this is an empirical observation and not a rigorously proven fact. Therefore, simulations like ILLISIM or SRA, which are based on this assumption, may not provide reliable results in cases where the data are actually not normally distributed. For example, pavement density may be suddenly shifted during construction due to many variables, including weather change, equipment or operator changes, or a change in the rolling pattern. This would create a bimodal rather than a normal distribution. However, this particular case has been studied and reported by Buttlar et al. [2001] to have a relatively low impact on project pay factors.

In summary, it should be recognized that simulation results are indicative rather than predictive. This means that these results should be used to guide decisions in general. But they cannot be used to provide direct predictions for a particular project. It should also be acknowledged that the level of risk for each party that will be viewed as acceptable are not fixed values and that higher contractor risk can translate into higher dispute rates and higher bid estimates. Risk levels must be selected by the agency based upon many factors, such as agency staffing limitations and expertise level, material and overall project costs, user-delay costs, local contractor expertise, and level of competition between local contractors.

Conclusions

Based upon the results of this research, the following conclusions have been drawn:

1. End-Result Specifications for asphalt pavement construction involve non-negligible risk to the agency and contractor due to the presence of measurement variability and testing biases.
2. The SRA program provides realistic, repeatable measures of ERS risk, which can be used to develop, analyze, and adjust ERS systems.
3. Based upon the analyses conducted in this study, it apprears to be feasible to significantly reduce the number of pavement cores taken on higher tonnage Illinois ERS projects without a significant impact on payment risk.
4. The absence of a pay cap in the ERS pay factor formula introduces a small, residual contractor risk for production near the center of the specification limits based upon the definition of risk adopted in this study.

Recommendations

Based upon the findings of this study, the following recommendations are made:

1. Since it appears that the number of pavement cores taken on Illinois ERS projects can be reduced considerably without a significant impact on payment risk, it is recommended to specify reduced number of cores and density measurements for future IDOT ERS projects. Additional SRA modeling runs could be performed to fine-tune the number of tests required.
2. As future changes are made to the ERS specification, the reinstatement of the pay factor cap should be considered as a means to reduce risk levels at higher pay factors.

Recommended Areas for Future Work

1. Characterizing risk plots: Risk plots are plots of payment risk against mean value of a certain quality characteristic. The range of the quality characteristic is generally the full range of values that may occur in an actual project. Depending on the value of other parameters such as production and measurement variability, and number of samples the shape of the risk plot can change considerably. This change can be easily observed visually. But to be able to subject this to rigorous mathematical analysis it is necessary to characterize these plots in a quantifiable way. Then multiple sets of plots can be easily compared. In this report such an attempt was made by identifying the maximum positive

and negative risks across the range. But more such characteristics may be required for more complete analysis, like area enclosed by the confidence limits within a certain width of quality characteristic, etc.

2. Modeling human decision elements: If more of human decision elements involved in actual highway construction projects can be included in the model, the results obtained from the simulation may be more realistic. Such modeling is possible by tapping into the experience of officials and professionals working on such projects.

3. More work can be done to determine in what situations the project data may not be normally distributed. This would help identify possible cases where the simulation results may not be applicable. Also, work can be done as to how the simulation results can be modified in those cases.

4. Smoothness and thickness of the as-built pavement are also important characteristics in determining payment to the contractor. Further study is needed to determine how best to combine these pay factors with those associated with material quality.

Other Recommended Uses for the Current Simulation Program

With slight modifications, the SRA simulation program can also be used to develop or modify ERS programs for soils and aggregates, Portland cement concrete pavements, and other transportation materials and constructed facilities.

REFERENCES

AASHTO *Quality Assurance Guide Specification.* AASHTO, Washington, D.C., 1995.

Afferton, K. C., J. Freidenrich, and R. M. Weed. Managing Quality, Time for a National Policy. In *Transportation Research Record 1340*, TRB, National Research Council, Washington, D.C., 1992, pp. 3–39.

Barros, R. T., R. M. Weed, and J. H. Willenbrock. Software Package for the Design and Analysis of Acceptance Procedures Based Upon Percent Defective. In *Transportation Research Record 924*, TRB, National Research Council, Washington, D.C., 1983, pp. 85–93.

Buttlar WG, Hausman, J. J., "ILLISIM Program for End-Result Specification Development," *Journal of the Transportation Research Board*, National Research Council, National Academy Press, Washington, D. C., 1712 125-138 (2000).

Buttlar, W. G., Shanley, L., Aref, Susanne, "End-Result Specification Development: The Illinois Projects", *Proceedings Canadian Technical Asphalt Association* 46 303-331 (2001).

Hall, Kevin D., Stacy G.Williams, "Establishing Variability for Hot-Mix Asphalt Construction in Arkansas", 81st Annual Meeting, Transportation Research Board, National Research Council, Washington D.C., 2002.

NCHRP *Synthesis of Highway Practice 38: Statistically-Oriented End-Result Specifications*. TRB, National Research Council, Washington, D.C., 1976, pp. 8–9.

Patel, A. J. *Development of Fatigue Based Pay Adjustment Factors.* Ph.D. thesis. Graduate College, University of Illinois at Urbana–Champaign, 1996, p. 45.

Puangchit, P., R. G. Hicks, J. E. Wilson, and C. A. Bell. Development of Rational Pay Adjustment Factors for Asphalt Concrete. In *Transportation Research Record 911*, TRB, National Research Council, Washington, D.C., 1983, pp. 70–79.

Vince, Aurilio, William G. Buttlar, Anshu Manik, Laura Shanley Accounting for Bias and Between-Laboratory Test Differences *Proceedings of 47th Annual Conference of the Canadian Technical Asphalt Association*, Calgary, 2002.

Weed, R. M. OCPLOT: PC Program to Generate Operating Characteristics Curves for Statistical Construction Specifications. In *Transportation Research Record 1491*, TRB, National Research Council, Washington, D.C., 1996, pp. 18–26.

Willenbrock, J. H. *Statistical Quality Control of Highway Construction,* Vol. I and II. FHWA, U.S. Department of Transportation, 1976.

APPENDIX A1

ILLISIM

University of Illinois researchers developed a computer simulation program, called ILLISIM, to analyze the risks involved with end-result specifications (ERS). The detailed assessment of agency and contractor risks possible through ILLISIM can assist in establishing sampling protocols, measurement methods, specification limits, retest provisions, pay scales, and pay caps in such a manner to balance the tradeoffs between number of samples and payment risks, and hence disputes.

A flowchart describing a typical execution sequence in ILLISIM is given in Figure A1.1a-b. Based upon the assumption that construction and measurement variability can be adequately approximated by a normal distribution curve (bell curve) [Hall (2002)], ILLISIM randomly generates quality characteristics within given SUBLOTS and LOTS of material on a paving job.

The user has the ability to determine how ILLISIM evaluates the source(s) of variability depending on how easily individual sources of error can be identified. If a given characteristic has separable, measurable sources of variability, the user can determine how each source independently affects the determination of quality. Standard deviation is considered as an estimate of the variability that is being mentioned in this report. Using density as an example, ILLISIM can consider three individual elements of variability (longitudinal, transverse, and measurement device). However, if the user wishes to analyze a database of historical measurements from which no individual source of variability can be separated, one individual standard deviation can be used to encompass all of the variability throughout the process.

Based upon the inputs, ILLISIM generates possible measurement readings that would be encountered during construction, using random numbers and an inverse normal distribution generation algorithm (Monte Carlo simulation). The inverse algorithm takes a mean value of a quality characteristic, standard deviation, and random number, and outputs a density value at the location on the bell curve associated with the random number supplied. The random number represents the cumulative area under the standard normal distribution curve. For instance, a random number of 0.025 would happen to give the quality characteristic at the lower 95% confidence interval (of a two-tailed curve), while a random number of 0.5 would render the value unchanged. This process is repeated for each independent level of variability present in the system, giving a distribution of simulated measurements within a given LOT of material akin to measurements typically obtained in the field.

ILLISIM

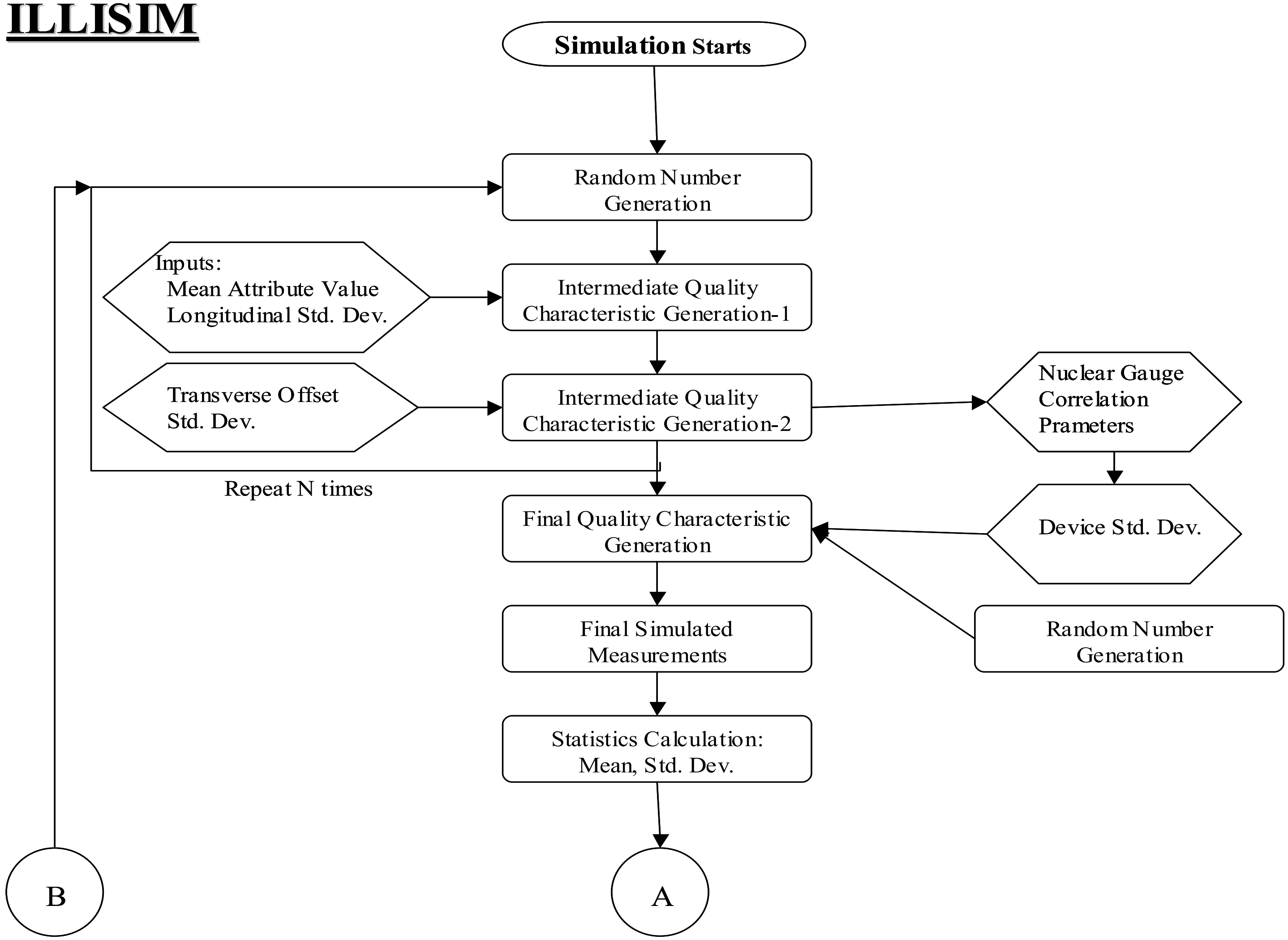


Figure A1.1a: ILLISIM program flowchart, part 1 of 2. Letters ‘A’ and ‘B’ at the bottom of the chart indicate connecting points for the ensuing figure.

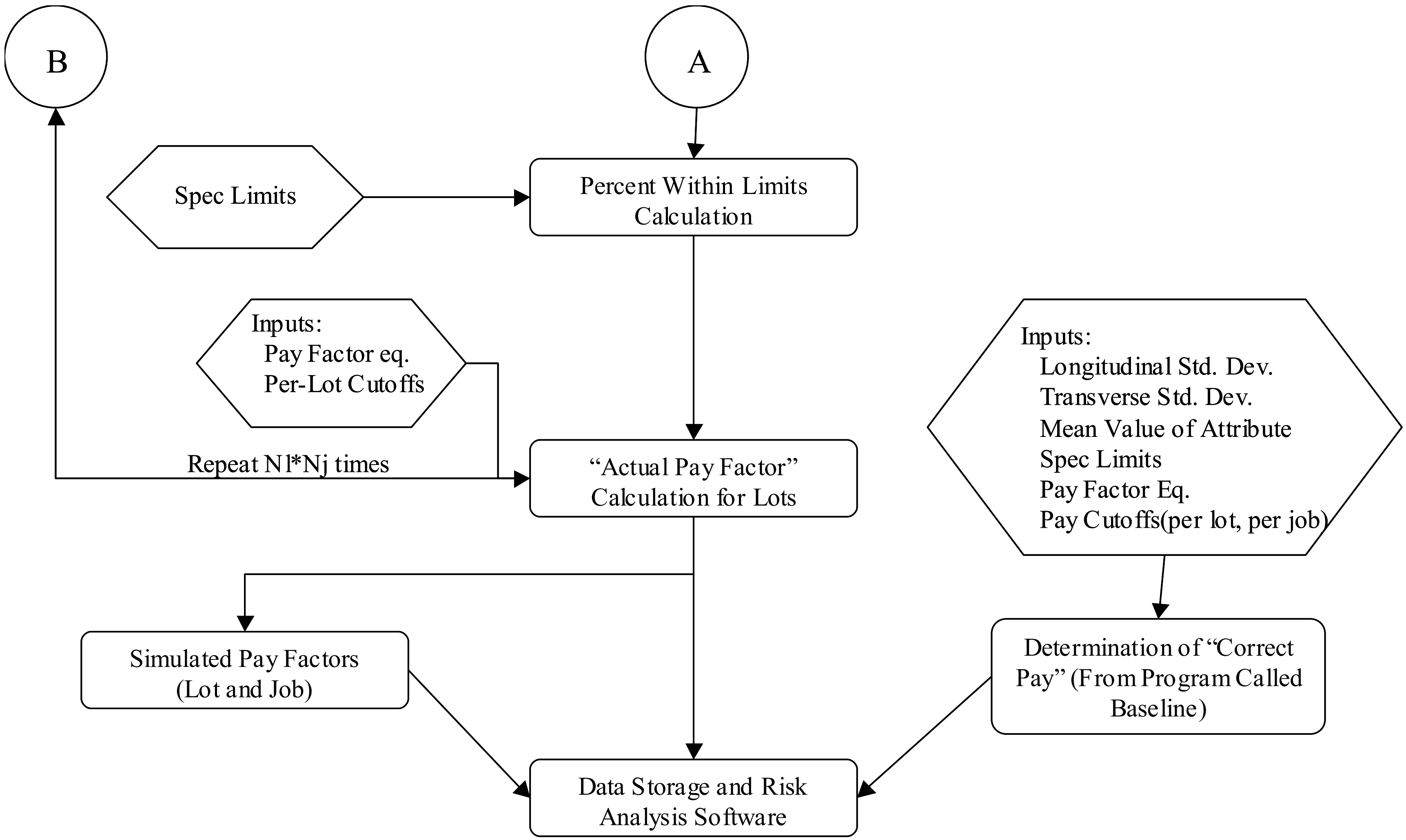


Figure A1.1b: ILLISIM program flowchart, part 2 of 2.

ILLISIM uses the simulated measurements to compute a mean, standard deviation, percent within limits, and pay factor for each LOT of material considered. For simulation modeling of processes with high variability, it is important to run a large number of simulations to adequately describe the characteristics of the system. A minimum of 1000 LOTS were typically simulated for each unique group of input parameters considered. ILLISIM keeps track of a large number of runs, so that a statistical distribution of correct pay versus actual pay for individual LOTS and complete JOBS can be plotted.

The sampling schemes considered in this study for as-constructed pavement density are summarized in Figure A1.2, which can be described as follows:

- **Dual-Stratified Random Sampling Method** — A length of pavement, or LOT, can be divided into equal SUBLOTS, which can be further subdivided by the number of transverse measurements desired per SUBLOT, as shown in Figure A1.2. Sampling locations are based upon a conventional stratified random layout in the longitudinal direction. In the transverse direction, samples are to be taken at the 2-, 4-, 6-, 8-, and 10-ft offsets, in random order. Means and standard deviations are then computed using all measurements (N=15). Similar groupings can be developed for other values of N. For instance, in a later section, a comparison is made between N=9 and N=15 measurements, where the N=9 LOT consists of three SUBLOTS with 3 measurements taken across the paving lane.

- **Stratified-Average Sampling Method** — This method utilizes an identical sampling layout as the dual-stratified method. However, the mean and standard deviation are computed in a different manner, as outlined in Figure A1.2. In summary, an average density is first obtained for each of the three SUBLOTS. Then, a LOT average and standard deviation are computed using the three SUBLOT averages.

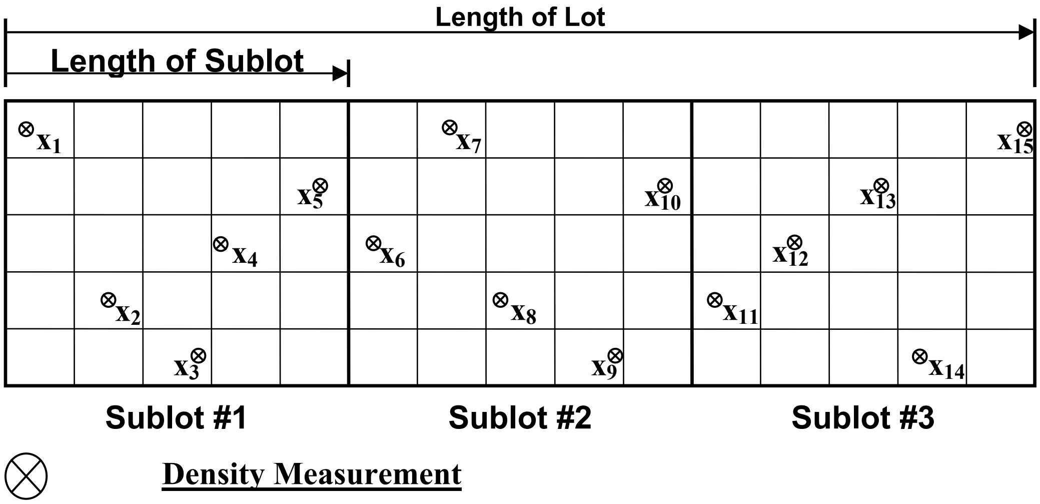
Each of the two sampling methods has distinct advantages and disadvantages. The dual-stratified method gives larger standard deviations, which reflect the combined standard deviation caused by variability in both the longitudinal and transverse directions. The stratified-average method has a smaller standard deviation, since the effect of transverse standard deviation is essentially minimized by first averaging density values in each SUBLOT. The motivation for investigating this method was to stabilize PWL-predictions on a per-LOT basis in an attempt to minimize the possibility of frequent disputes, particularly when marginal quality levels arise.

Inputs for ILLISIM

The user supplies the following inputs to ILLISIM:

(1) Mean value of as-produced or as-constructed quality characteristic (e.g. density, asphalt content, etc.) to be considered, or, more commonly, a range of such mean values.

(2) Standard deviation(s) of the quality characteristic(s) associated with production and construction



Dual − Stratified Sampling Method (N = 15) :

$$\bullet \text{Mean Density, } \overline{X}_{LOT} = \frac{\sum_{i=1}^{N} x_i}{N}$$

$$\bullet \text{Standard Deviation, } \sigma_{LOT} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{X}_{LOT})^2}{N-1}}$$

Stratified - Average Sampling Method ($N_1 = 5$ meas/sublot, $N_2 = 3$ sublots/lot)

$$\bullet \text{Mean Density per Sublot, } \overline{X}_{SUB} = \frac{\sum_{i=1}^{N_1} x_i}{N_1}$$

$$\bullet \text{Mean Density per Lot, } \overline{X}_{LOT} = \frac{\sum_{SUB=1}^{N_2} \overline{X}_{SUB}}{N_2}$$

$$\bullet \text{Standard Deviation per Lot, } \sigma_{LOT} = \sqrt{\frac{\sum_{SUB=1}^{N_2} (\overline{X}_{SUB} - \overline{X}_{LOT})^2}{N_2 - 1}}$$

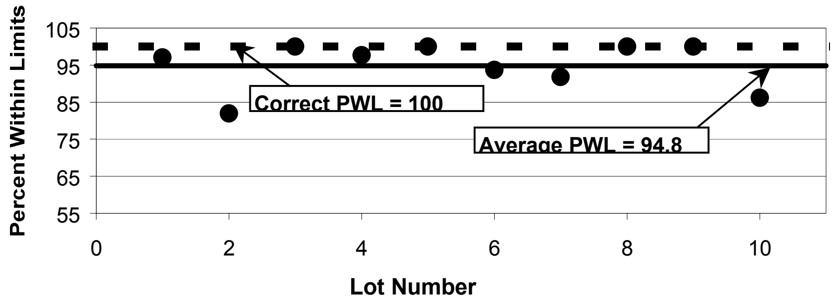
Figure A1.2: Various Schemes for Density Measurement Locations

(3) Standard deviation of the measurement device (For density measurement using a nuclear gauge, variability depends on the proximity to the mean value at which the gauge is correlated to cores, described in more detail in a later section.)
(4) Number of measurements
(5) Sampling arrangement (e.g., completely random, dual-stratified random, stratified-averaging method, etc., described in more detail in a later section)
(6) Specification limits
(7) Pay factor equation
(8) Pay limits or "caps" (per lot and per job)

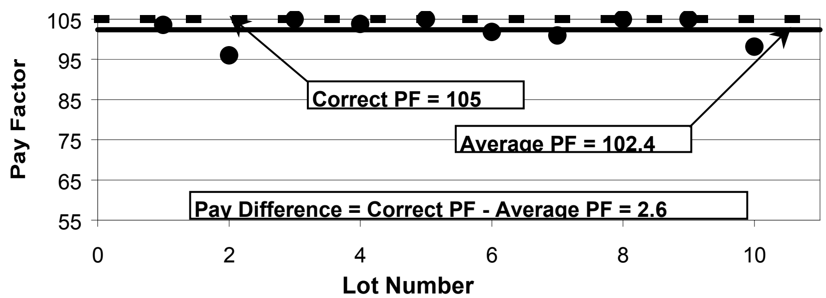
Output from ILLISIM

Figure A1.3 shows typical output from ILLISIM, and the progression of analyses that were conducted to assess relative risks for the producer and agency. First, simulated density measurements were used to obtain averages and standard deviations according to Figure A1.2. Next, PWL values and pay factors were determined (Figure A1.3 (a)). A separate program called "Baseline" was developed, which determines the "correct pay" for the input values given, based upon a very large number of simulations (uses 40,000 randomly generated density values). The definition of correct pay is somewhat arbitrary, so a definition of the approach used herein is appropriate. Correct pay was based upon the pay factor that would be determined over the long run under acceptable levels of production and measurement device variability. Pay factor differences per LOT and per JOB are computed using ILLISIM, which are then compared to the correct pay value (Figures A1.3 (a) and (b)). Pay factor differences arise since a discrete number of measurements will not typically lead to an exact measure of mean and standard deviation for any given LOT.

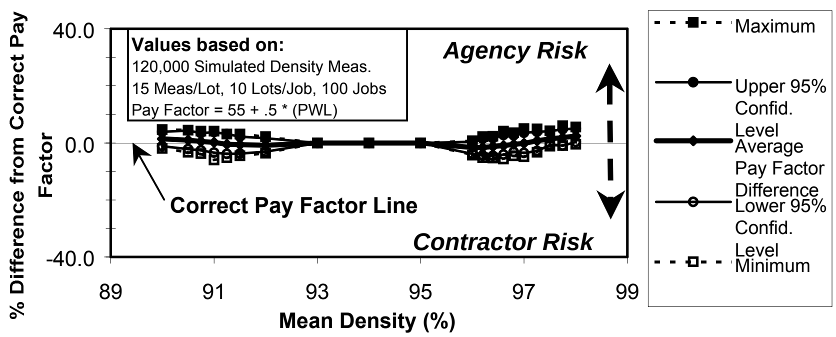
Figure A1.3 (c) illustrates a typical plot used to assess payment differences, or payment errors that can be expected for a given set of inputs. These results are generally shown across a range of mean density of construction levels, to illustrate the increased risk of payment error for LOT averages that happen to be near the specification limits (e.g., when marginal quality levels arise). Maximum and minimum payment errors (risks) per LOT (based upon 1000 LOTS) and per JOB (100 JOBS) are given. Also plotted are the 95% confidence intervals for pay differences relative to mean pay, which allow the analyst to identify typical risk envelopes, independent of possible extreme values for maximum or minimum pay difference. Finally, by defining the 95% confidence intervals on payment error as a "risk index," risk levels can be conveniently compared between different sampling methods and number of measurements, for example, as illustrated in Figure A1.3 (d).



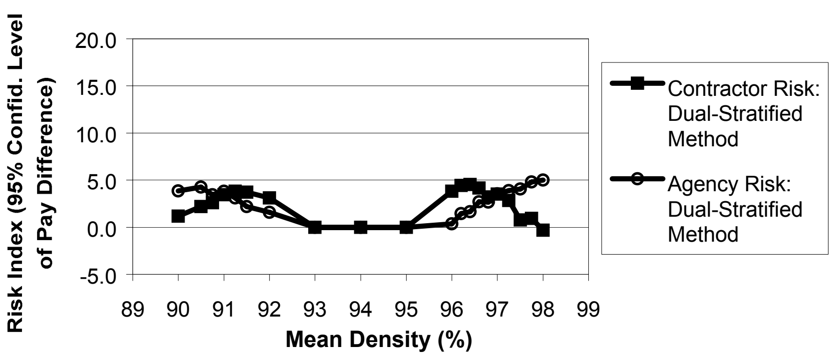
a) Percent Within Limits: Mean Density = 96%, 10 Lots



b) Pay Factor: Mean Density = 96%, 10 Lots



c) Pay Difference: Dual-Stratified Sampling Method, 15 Meas./Lot, 2000 Jobs



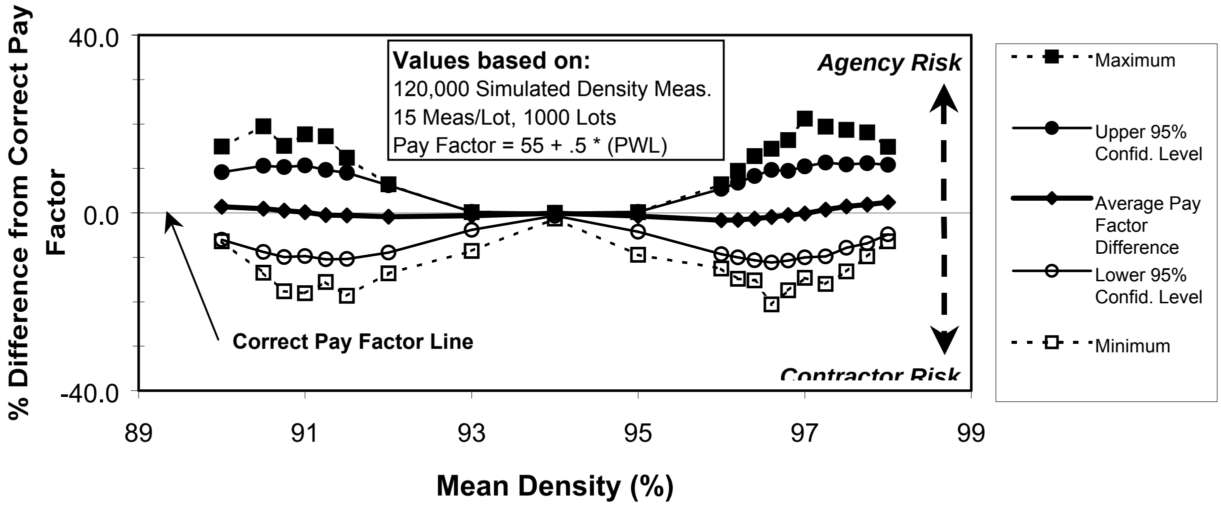
d) Risk Index: Dual-Stratified Sampling Method, 2000 Jobs

Key: PF- Pay Factor; PWL- Percent Within Limits

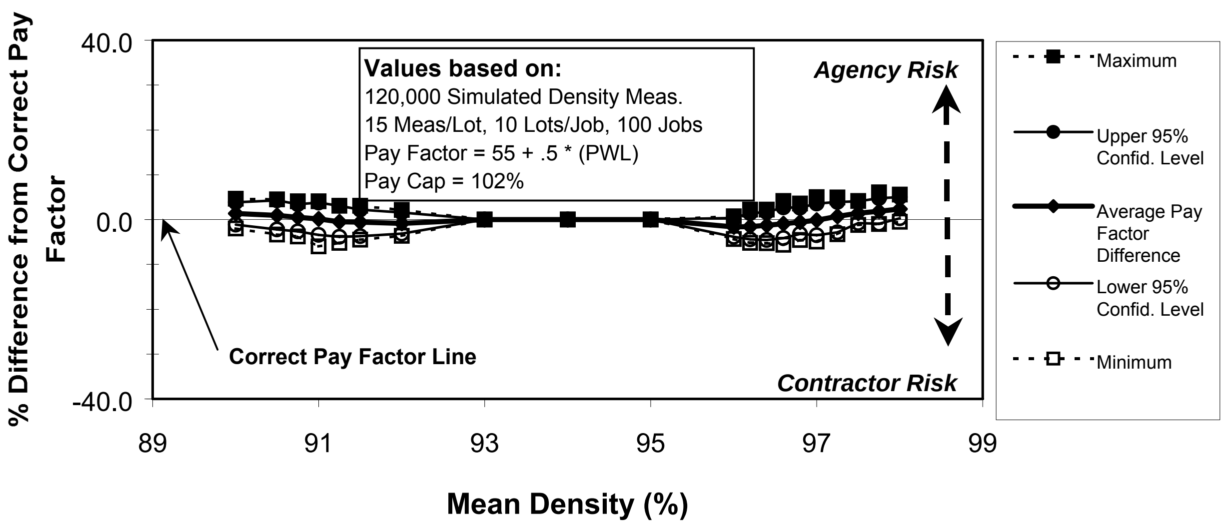
Figure A1.3: Development of a Risk Index Plot from ILLISIM Results

Figures A1.4-A1.8 summarize other outputs from the ILLISIM runs. Some general observations are as follows:

- As the number of measurements per LOT is increased, risk for both parties decrease.
- The risk of over- or under-payment is much lower when viewed on a per-JOB basis rather than a per-LOT basis (e.g., Figure A1.4 (a) versus (b)). It is important to be able to view these risks separately, since disputes can arise if the contractor risk on the per-LOT basis is too high even if the per-JOB risks are low. In general, risks tend to diminish due to the statistical tendency to arrive at the correct payment estimate as more LOTS are assessed.
- The risk level for both parties is lowest at the middle of the specification range, which, in this case, is 94% G_{mm}. This is because unless the standard deviation is exceptionally high, typical errors in estimating the mean and standard deviation in this case are not enough to cause the predicted normal distribution to shift outside the specification limits . Hence, 100 PWL is estimated almost invariably, thus eliminating risks for payment errors.
- Conversely, risks for both parties are greatest when the mean density of construction is near specification limits. This might indicate a benefit in obtaining more measurements when marginal quality is detected.
- The stratified-average sampling method (Figure A1.5) performs very well (low risks for both parties), between specification limits, but poorly in the vicinity of specification limits. This is caused by the low standard deviation resulting from the averaging method used. A low standard deviation gives a narrow bell curve, which renders the PWL prediction to be very sensitive to small errors in estimating the average density (e.g., the area under the narrow bell curve can easily shift from a very high PWL to a very low PWL with a small shift across the specification limit). So, while the lower standard deviations associated with taking the "average of the average" in the stratified-average method might intuitively be assumed to lead to the reduction of errors in PWL estimation, this is not always the case.
- The dual-stratified sampling method (Figure A1.4) does not perform as well as the stratified-average method between specification limits; however, the method results in lower risks around the specification limits.



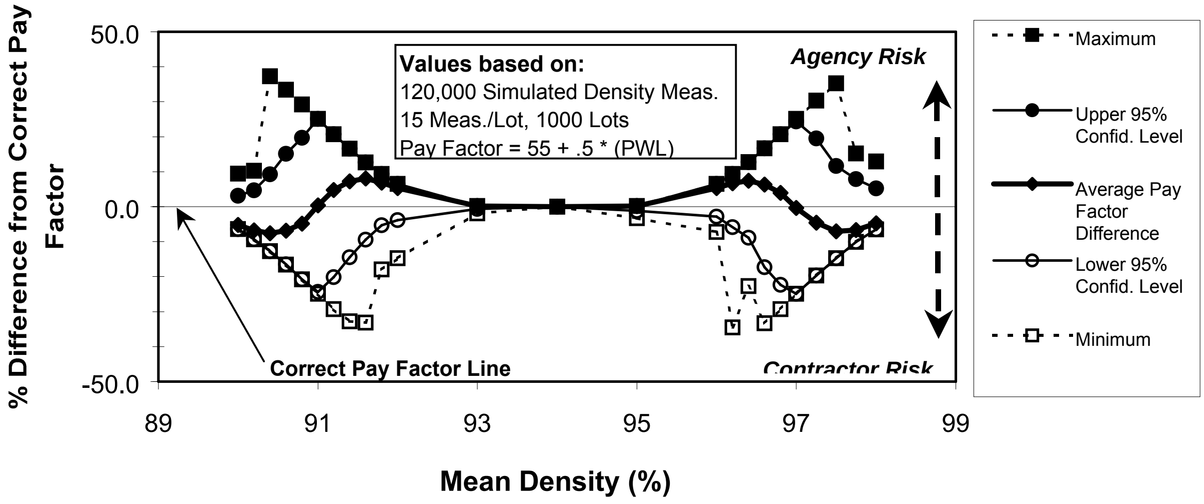
a) Per-Lot Risk Analysis



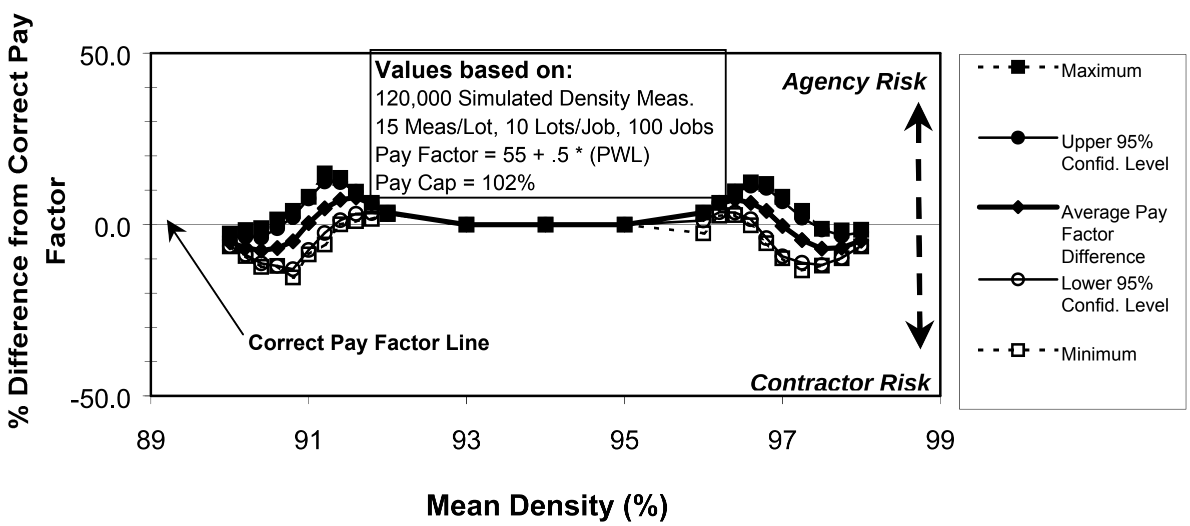
b) Per-Job Risk Analysis

Key: PF- Pay Factor; PWL- Percent Within Limits

Figure A1.4: Per-LOT Risk Analysis vs. Per-JOB Risk Analysis for Dual-Stratified Sampling Method



a) Per-Lot Risk Analysis

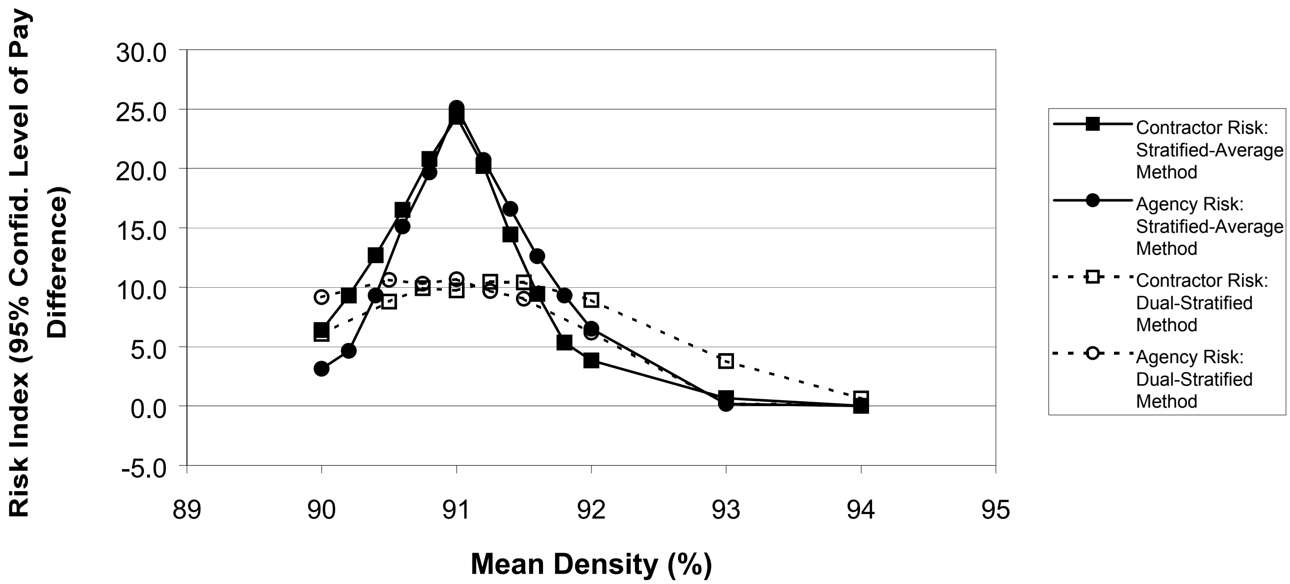


b) Per-Job Risk Analysis

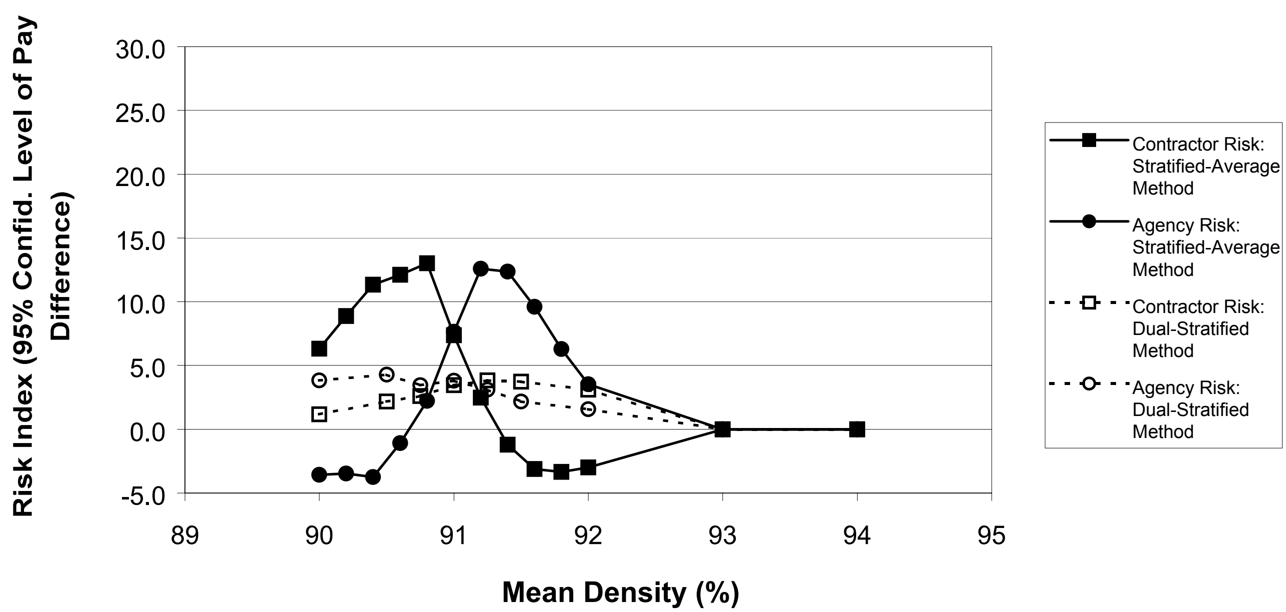
Key: PF- Pay Factor; PWL- Percent Within Limits

Figure A1.5: Per-LOT Risk Analysis vs. Per-JOB Risk Analysis for Stratified-Average Sampling Method

- Figures A1.6 and A1.7 show risk analysis plotted versus mean density. Due to the statistical nature and randomization of the Monte Carlo simulation, the results are assumed to be symmetric around the center of the specification range while any differences simulated on a per-lot basis are insignificant and will be averaged out in the per-job analysis from which pay is determined. Due to this symmetry and the fact that construction generally targets the lower end of the specification range, these figures were plotted over the lower end of the specification range with greater resolution near the lower specification limit (91%) to show trends of risk as they approach the allowable specification limits.
- Figure A1.6 is a convenient way to compare the two sampling methods considered. The risk index bound is a statistically described bound on potential payment error. In the long run, 95% of payment errors will fall within the risk index bound.
- Figure A1.7 compares nuclear gauge risks to risks associated with basing payment on density measured from pavement cores, for identical sampling methods. Although core standard deviations were modeled to be significantly lower than the nuclear gauge, the relative risks were found to be surprisingly similar.
- An additional consideration in comparing the nuclear gauge versus cores for acceptance is test bias. The aforementioned conclusion assumes that a correlation is established between density measured with the nuclear gauge and density measured on pavement cores, for which an estimate of device variability can be obtained. After correlation, the bias is assumed to be minimal. However, in practice, the accuracy of the correlation can change as a result of changes in the materials, lift thickness, properties of underlying pavement layers, and inaccuracies caused by changes in operational procedures and device operating characteristics. Periodic recalibration will obviously reduce the potential for inaccuracies due to bias; however, each recalibration requires significant coring and laboratory testing. More work is needed to assess the implications of bias on the practicality and reliability of the nuclear gauge for density acceptance.
- Bias also tends to increase or decrease the payment risks in addition to that because of other variability discussed before. To assess the risk introduced because of bias alone another simulation program, BiasSim, was developed which is discussed in later section of this report.
- Figure A1.8 illustrates the use of ILLISIM to determine possible operating ranges where a given level of payment can be obtained, under various levels of process and device standard deviation. As process standard deviation or device standard deviation increases, the mean density must be closer to the middle of the specification range to achieve 100 percent pay, and even closer to the middle of the specification range to achieve full bonus, or 102 percent pay, as illustrated in this example. Thus, if a contractor can decrease production variability and/or if the acceptance tests are run with more precision, full pay can be realized over a wider range of the mean density of construction.

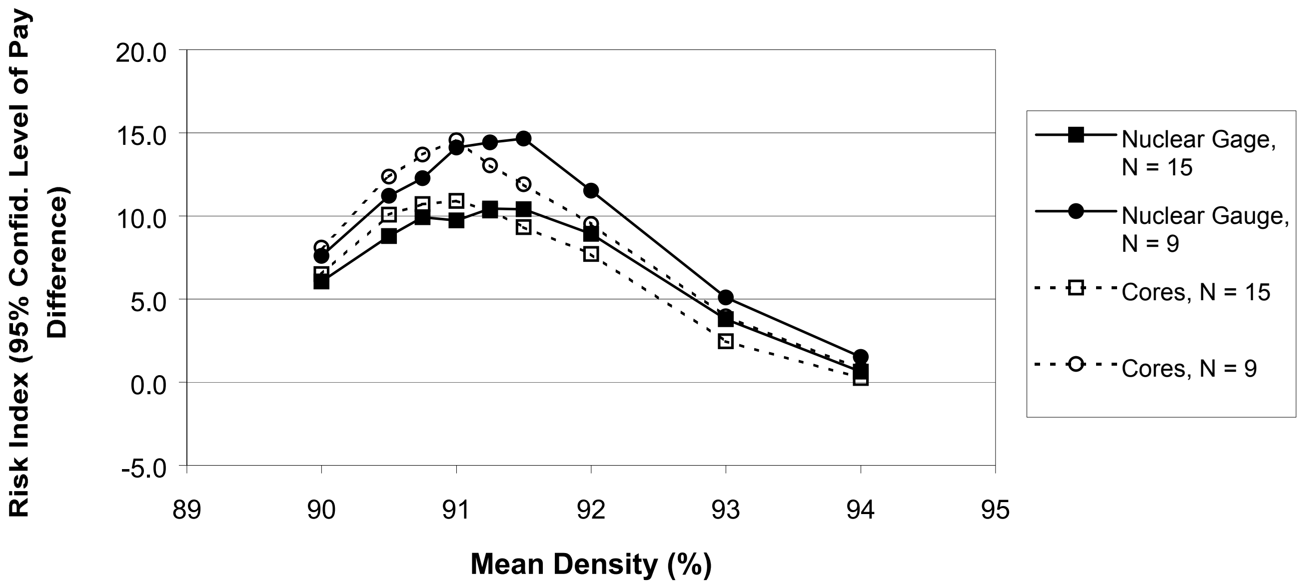


a) Per-Lot Risk Analysis

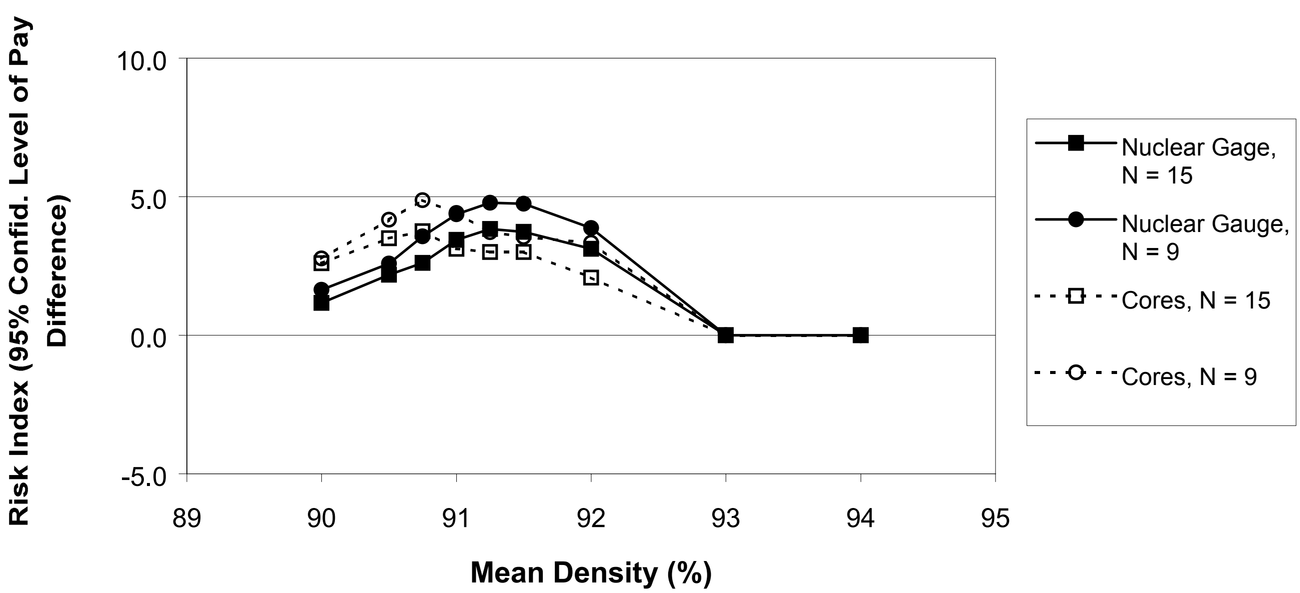


b) Per-Job Risk Analysis

Figure A1.6: Comparison of Risk Index for Contractor and Agency: Stratified-Average Method and Dual-Stratified Method vs. Mean Density (N = 15)



a) Per-Lot Risk Analysis



b) Per-Job Risk Analysis

Figure A1.7: Comparison of Risk Index for Nuclear Gauge and Core Density Measurements: Contractor Risk, Dual-Stratified Method (N = 9 and N = 15)

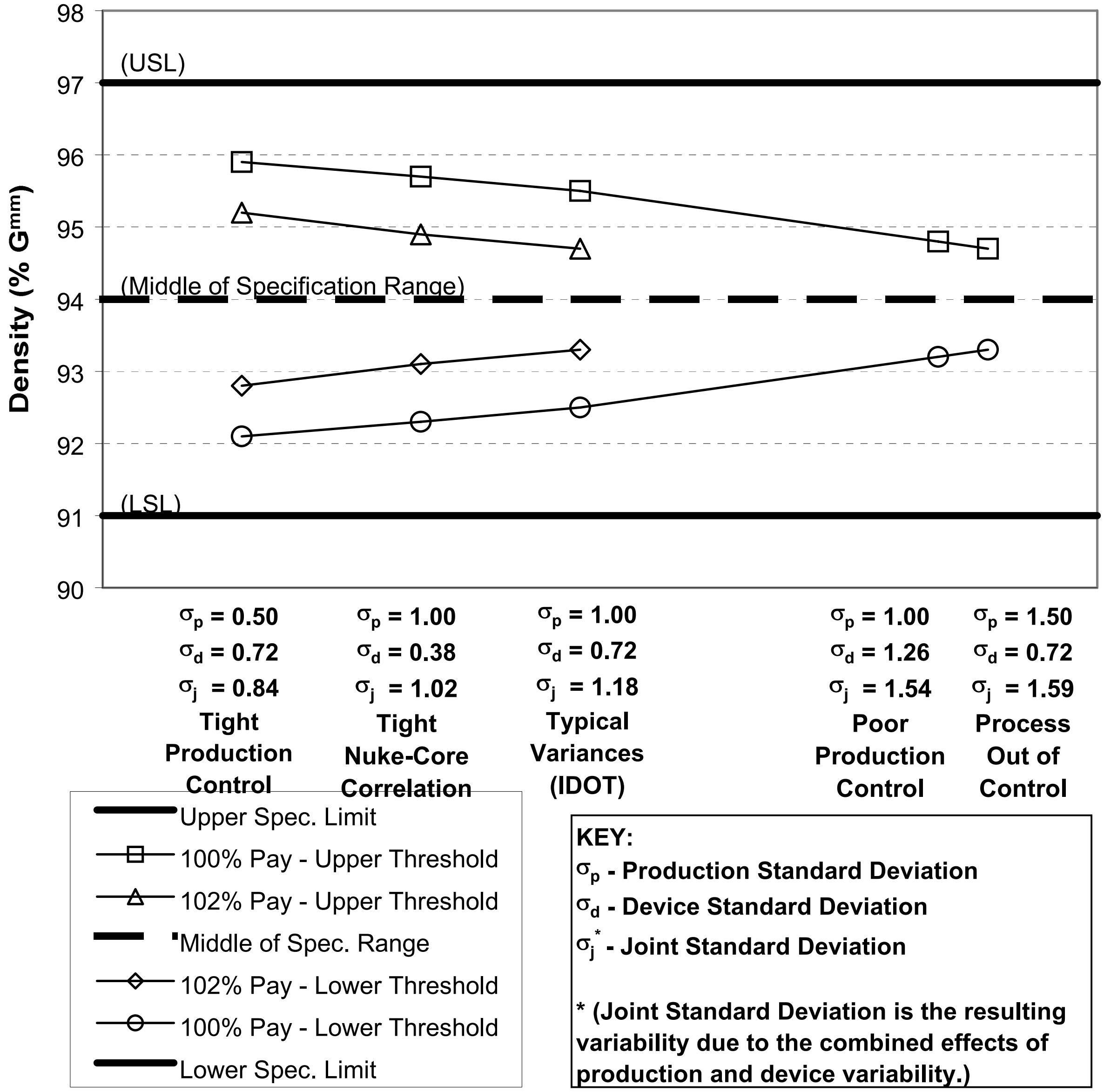
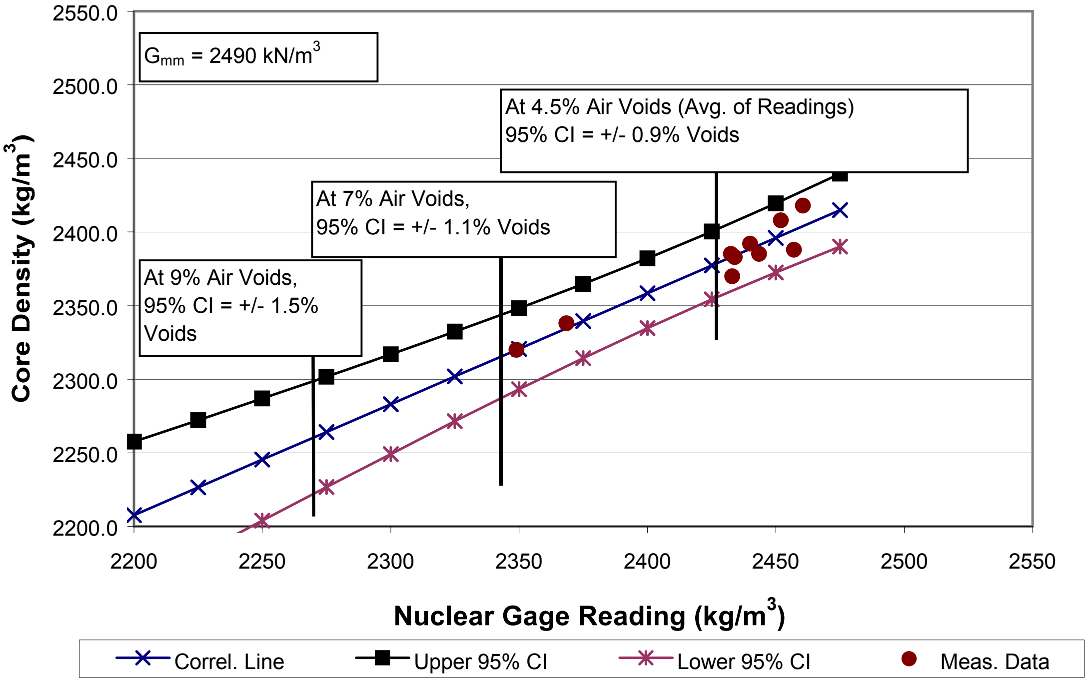


Figure A1.8: Use of ILLISIM to Determine Possible Operating Ranges as a Function of Payment Level and Combined Variability



Key: CI - Confidence Interval

Figure A1.9: Correlation between Nuclear Gauge and Cores Showing Divergent Confidence Intervals

APPENDIX A2

PaySim

The second simulation software PaySim was developed in order to:

(a) Incorporate a new simulation model
(b) Produce a more versatile simulation
(c) Reduce simulation time

The New Simulation Model

Inputs:

p – Production mean
σ_p^2 - Production variance
σ_d^2 - Device variance
n - Number of measurements taken
(p_l, p_u) - Lower and upper spec limits

Given the measurements $x_1, x_2, \ldots, x_n$ we can calculate the average $\bar{x}$ and standard deviations. The estimated percent within limits (PWL) is

$$PWL = \Phi\left(\left(p_u - \bar{x}\right)/s\right) - \Phi\left(\left(p_l - \bar{x}\right)/s\right)$$

Where $\Phi(a) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} e^{-x^2/2}$ is the cumulative density function of the standard normal distribution.

The expected PWL is equal to

$$E(PWL) = \int_0^{\infty} \int_{-\infty}^{\infty} \left\{ \Phi\left(\frac{(p_u - p)\sqrt{n} - x\sigma}{\sigma\sqrt{y}} \right) - \Phi\left(\frac{(p_l - p)\sqrt{n} - x\sigma}{\sigma\sqrt{y}} \right) \right\} \phi(x) g_{n-1}(y) dxdy$$

Where $\sigma^2 = \sigma_d^2 + \sigma_p^2$

$\phi(x)$: Probability distribution function of N(0,1)
$g_{n-1}(y)$: Probability distribution function of the chi-square distribution with n-1 degrees of freedom.

This calculation requires evaluation of double integral, but can be done using a Monte Carlo method. A quick approximation for the expected value is as follows.

Let

$$q_u = (p_u - p)/\sigma, \qquad q_l = (p_l - p)/\sigma$$

The expected PWL can be approximated by

$$E(PWL)_1 = \Phi(q_u) - \Phi(q_l)$$

A more accurate approximation (especially for larger σ) can be obtained with a second order adjustment. The second order approximation is

$$E(PWL)_2 = \Phi_2(q_u) - \Phi_2(q_l)$$

where

$$\Phi_2(a) = \Phi(a) + 0.5\left(a - 0.5a^3\right)\phi(a) I\left(|a| < \sqrt{2}\right)/(n-1)$$

The expected pay factor can be approximated given the expected PWL.

Payment Risk Distribution

The ideal PWL is

$$PWL_0 = \Phi\left((p_u - p)/\sigma_p\right) - \Phi\left((p_l - p)/\sigma_p\right)$$

The risk is due to the difference between PWL and PWL_0. The following Monte Carlo method can be used to give lower and upper limits of this risk

(1) Generate z_i (i = 1, 2, .., B) from N(0, 1)
(2) Generate y_i (i = 1, 2, .., B) from χ^2_{n-1}
(3) For each i compute

$$a_i = \Phi\left(\frac{(p_u - p)\sqrt{n} - z_i\sigma}{\sigma\sqrt{y_i}}\right) - \Phi\left(\frac{(p_l - p)\sqrt{n} - z_i\sigma}{\sigma\sqrt{y_i}}\right)$$

and

$$a_2 = \min(55 + 50a_i, c)$$

(4) Use a_2 as a random sample from the risk distribution and calculate target as follows

$$\text{Target} = \min(55 + 50\ \text{PWL}_o,c$$

Then average risk can be obtained by

$$\text{Average risk} = \text{mean}(a_2) - \text{target}$$

Although this method also uses random numbers, it is not the same as ILLISIM. It uses the same set of random numbers when production or device variances are varied. The set of random numbers used will vary only with n. B= 5000 to 10000 give pretty accurate estimates of the average risk and its lower and upper confidence limits.

This model has been incorporated in the simulation named as PaySim. The original simulation engine was written in C and converted into a standalone executable program. This executable program requires an input file to get all the input parameter values. To make it more user-friendly Microsoft Excel was used as interface. The code for the interface was written in Visual Basic. This interface allows the user to enter input values easily and makes an input file. Then it runs the main simulation engine with the input file thus generated. The simulation program puts the results (output) in an output file. The interface code takes that output file and plots it in a convenient form for the user. Figures A2.1 (a) and (b) show a flow diagram representing the overall functioning of the simulation.

<u>Inputs for PaySim</u>

(1) Device variability
(2) Production variability
(3) Number of samples
(4) Number of sublots
(5) Analysis range for the quality characteristic being analyzed
(6) Specification limits
(7) Pay cap option (cap before averaging or after averaging)
(8) Precision in simulation required (4 levels available)
(9) Confidence Interval required

<u>Outputs from PaySim</u>

The simulation is fully automated to complete all the tasks and produce risk plots for the quality characteristic being analyzed and in the range as defined by the inputs. The list of inputs also gives an idea of the versatility of the simulation because practically any combination of input parameters can be chosen and analyzed. This is very helpful in doing sensitivity analysis. The output is in the form of risk plots showing the risk to the agency (State) in pay factor depending on the magnitudes of the input parameters. Figures A2.2 (a) to (k) show a sensitivity analysis that was done using the PaySim simulation software.

Multiple simulation runs of PaySim were performed to get a risk index for pay factors for a fixed value of combined standard deviation but with varying combination of device and production standard deviations. Combined standard deviations were obtained from field and plant results from ERS demo projects in 2000 and 2001. The parameters analyzed were mix density, asphalt content and percent voids.

Further details of values used in the simulation are given in Tables A2.1 and A2.2

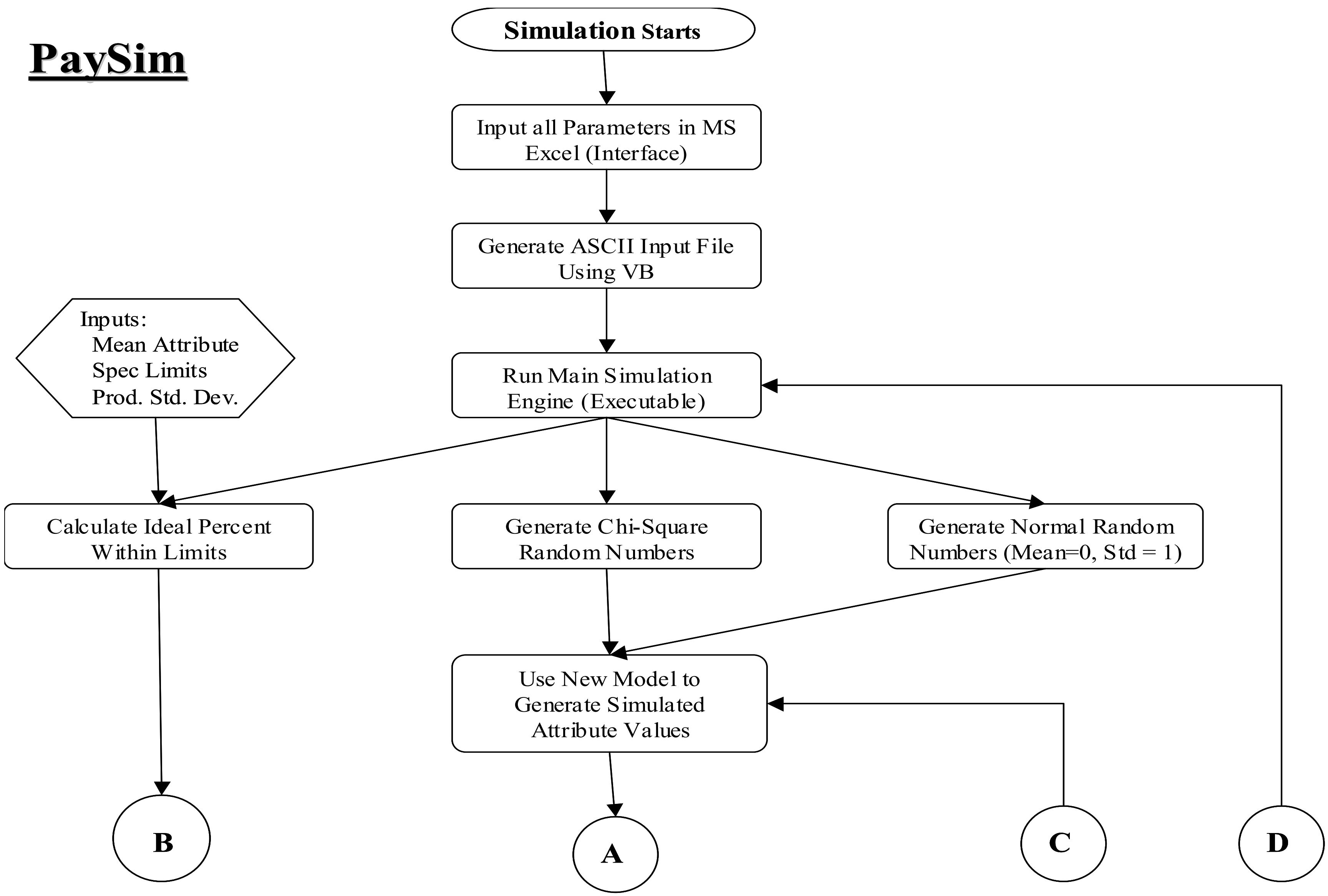
Table A2.1: Spec limits and number of samples used in the simulation with PaySim

Parameter	Target	Lower Spec Limit	Upper Spec Limit	Number of Samples
Mix Density (%Gmm)	94.25	91.5	97	50
Asphalt Content (%)	5	4.67	5.33	15
Voids (%)	4	2.65	5.35	15

Table A2.2: Device and production standard deviations used with PaySim simulation

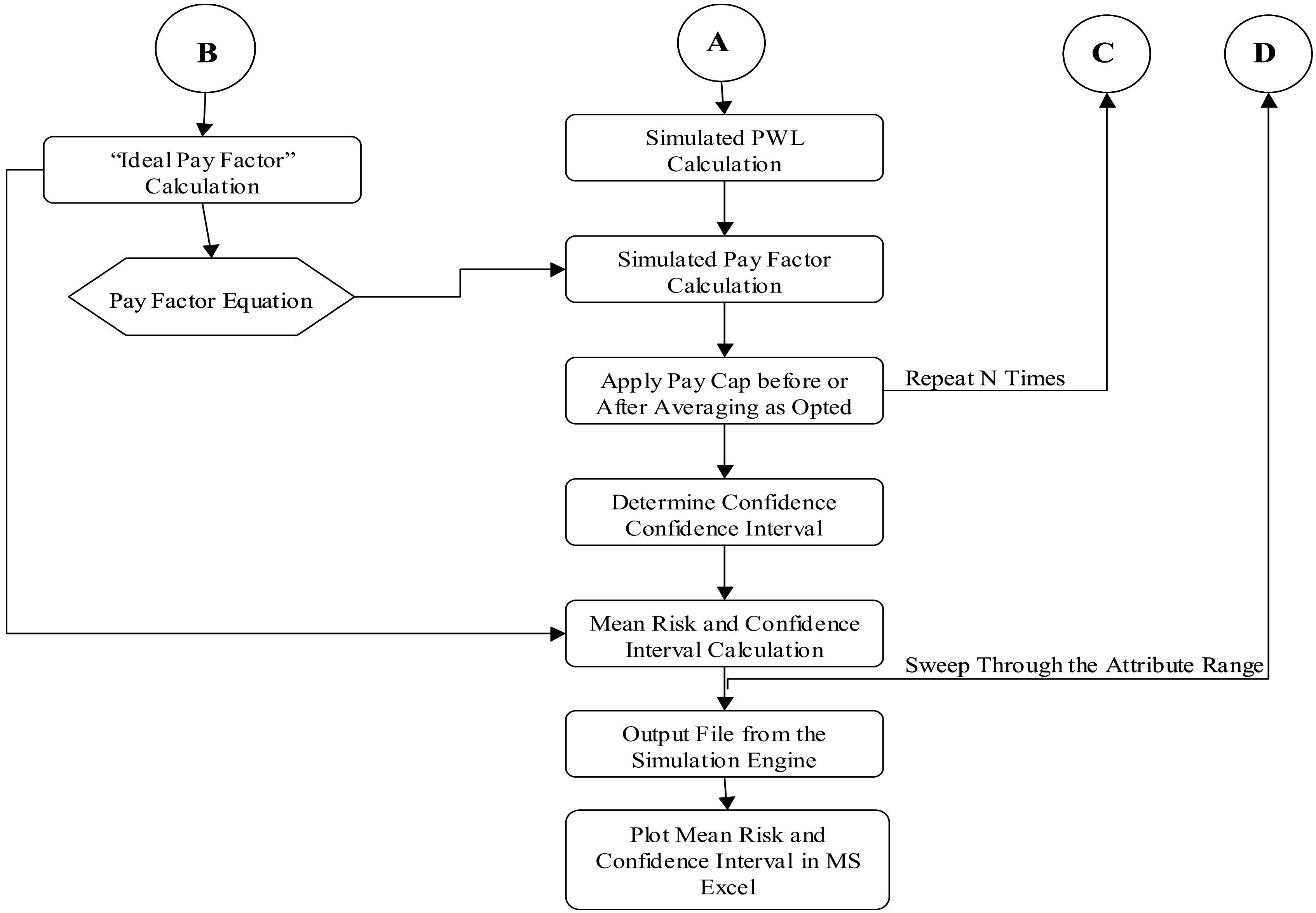
Parameter	$\sigma_{device\ 1}$	$\sigma_{production\ 1}$	$\sigma_{device\ 2}$	$\sigma_{production2}$	$\sigma_{device\ 3}$	$\sigma_{production3}$
Mix Density (%Gmm)	0.3	1.28	0.4	1.25	0.5	1.21
AC (%)	0.04	0.13	0.075	0.12	0.11	0.09
Voids (%)	0.15	0.67	0.20	0.66	0.25	0.64

Figure A2.2 shows the plots generated in the sensitivity analysis. As can be seen from the values in table A2.2, combined variability is kept constant for any particular quality characteristic but the device variability and measurement variability are being varied.



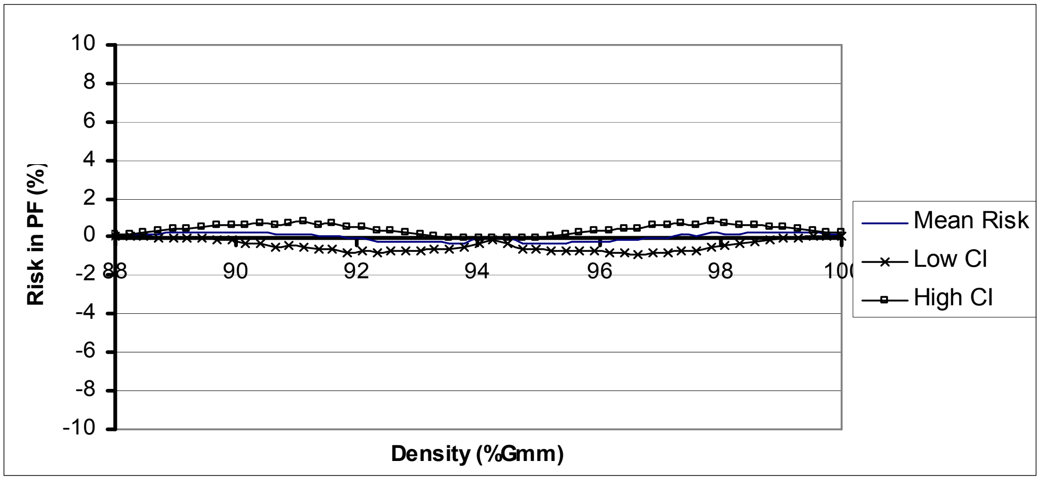
Key: MS – Microsoft; VB – Visual Basic; Std – Standard Deviation

Figure A2.1a: Outline of working of PaySim

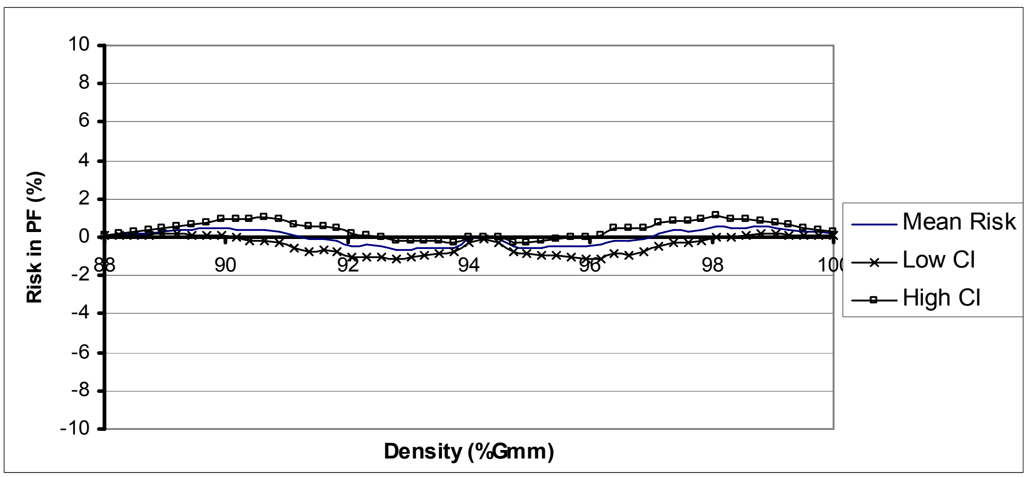


Key: PWL – Percent Within Limits; MS – Microsoft

Figure A2.1b: PaySim Program Schematic



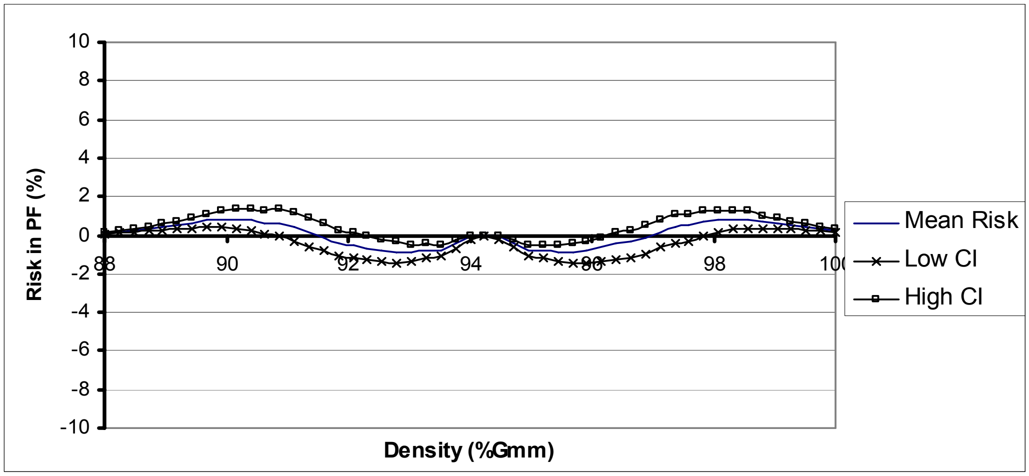
(a) : Risk for $\sigma_{combined}$ = 1.31, σ_{device} = 0.3, $\sigma_{production}$ = 1.28, N = 50, Pay Cap = 103



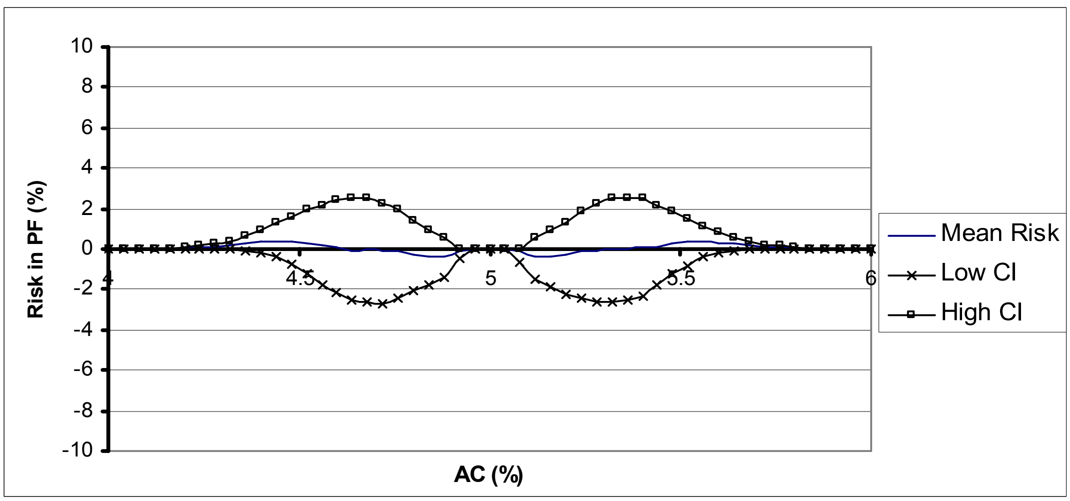
(b) : Risk for $\sigma_{combined}$ = 1.31, σ_{device} = 0.4, $\sigma_{production}$ = 1.25, N = 50, Pay Cap = 103

Key: $\sigma_{combined}$ - Combined variability; σ_{device} - Measurement or device variability;
$\sigma_{production}$ - Production variability; N- Number of samples;
Low CI- Lower limit of confidence interval; High CI- Upper limit of confidence interval ;

Figure A2.2 (a)-(b): Risk plots obtained from PaySim for different parameter levels

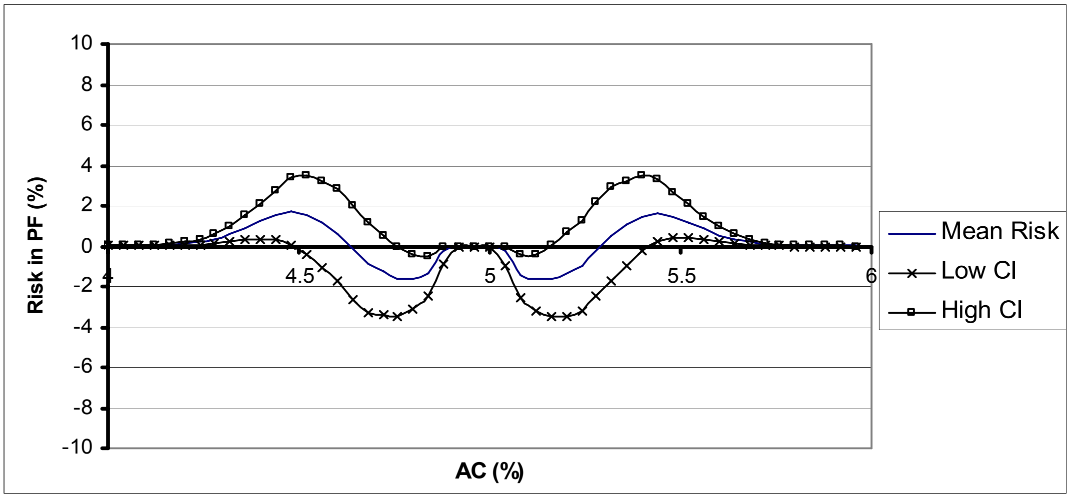


(c) : Risk for $\sigma_{combined}$ = 1.31, σ_{device} = 0.5, $\sigma_{production}$ = 1.21, N = 50, Pay Cap = 103

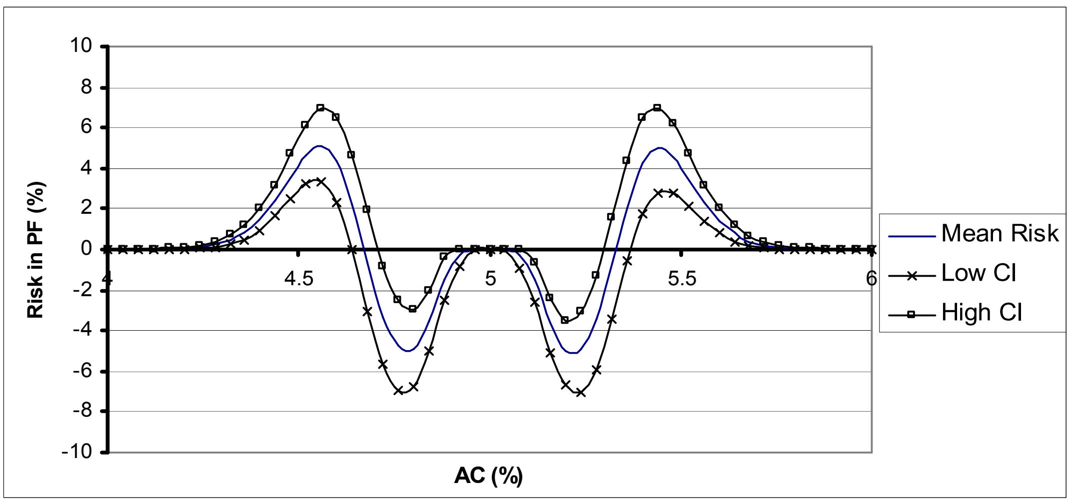


(d): Risk for $\sigma_{combined}$ = 0.14, σ_{device} = 0.04, $\sigma_{production}$ = 0.13, N = 15,Pay Cap = 103

Figure A2.2 (c)-(d): Risk plots obtained from PaySim for different parameter levels

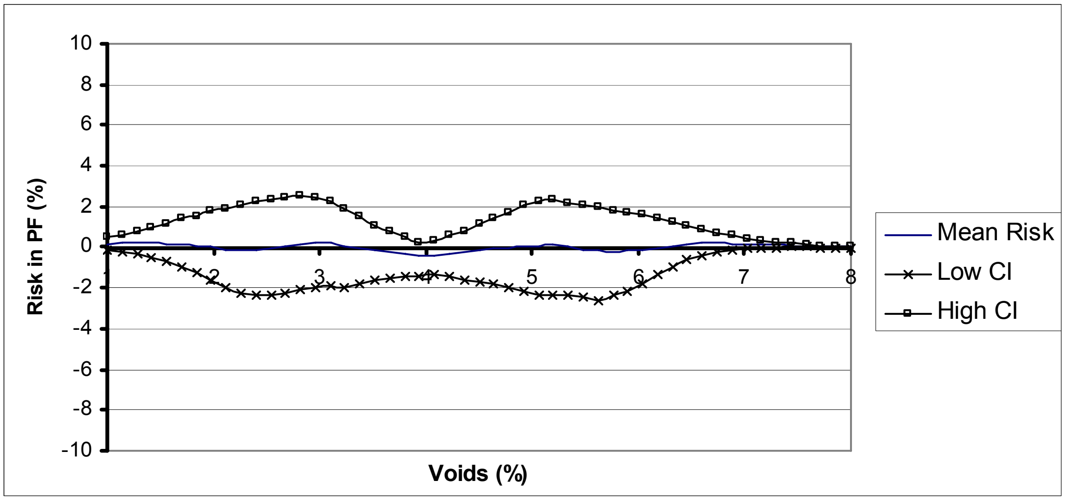


(e): Risk for $\sigma_{combined} = 0.14$, $\sigma_{device} = 0.075$, $\sigma_{production} = 0.12$, N = 15,Pay Cap = 103

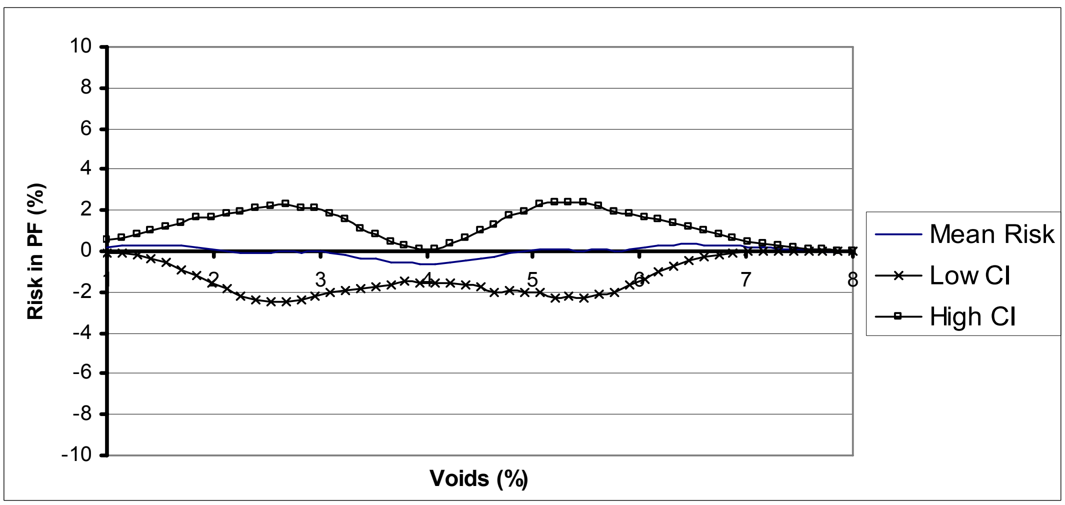


(f): Risk for $\sigma_{combined} = 0.14$, $\sigma_{device} = 0.11$, $\sigma_{production} = 0.09$, N = 15, Pay Cap = 103

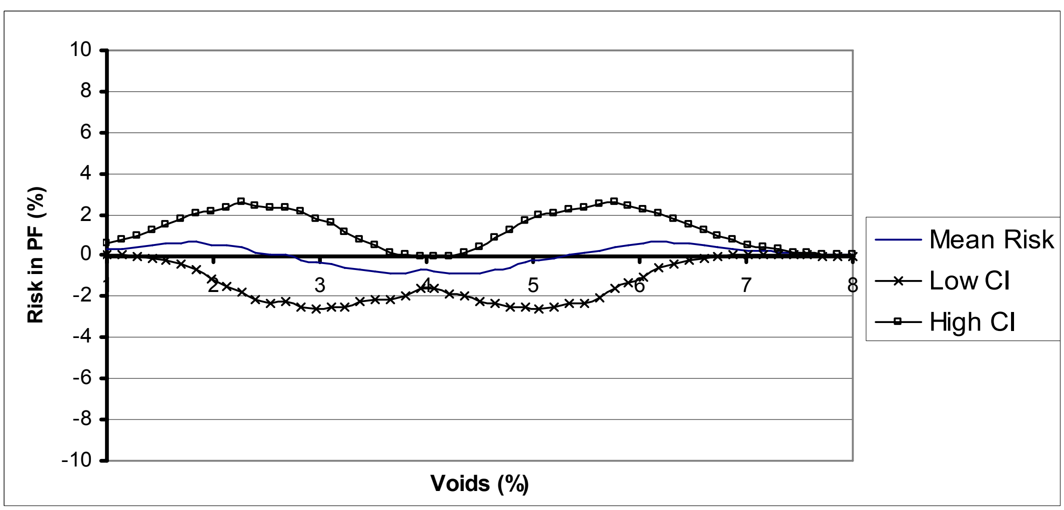
Figure A2.2 (e)-(f): Risk plots obtained from PaySim for different parameter levels



(g): Risk for $\sigma_{combined}$ = 0.69, σ_{device} = 0.15, $\sigma_{production}$ = 0.67, N = 15,Pay Cap = 103

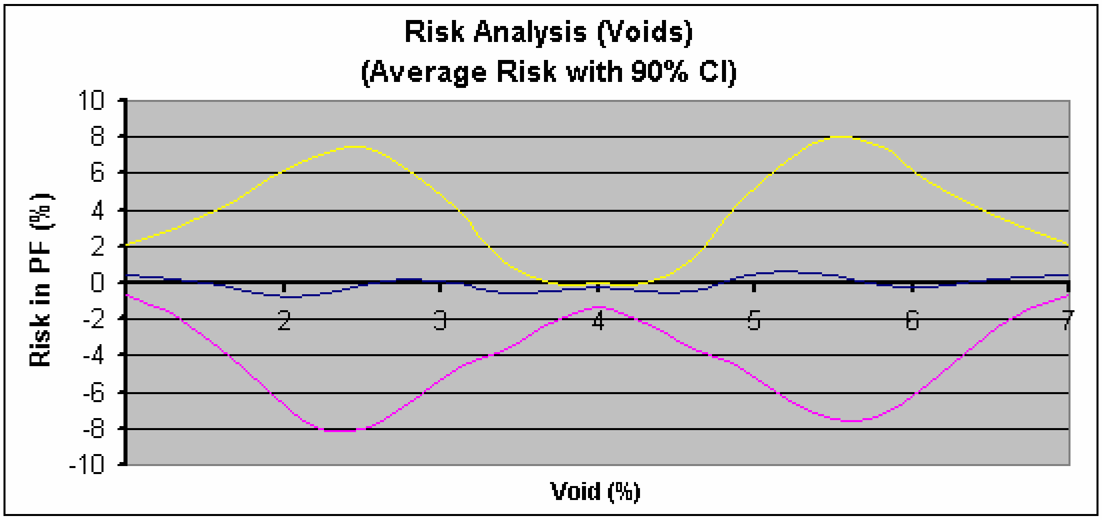


(h): Risk for $\sigma_{combined}$ = 0.69, σ_{device} = 0.20, $\sigma_{production}$ = 0.66, N = 15, Pay Cap = 103

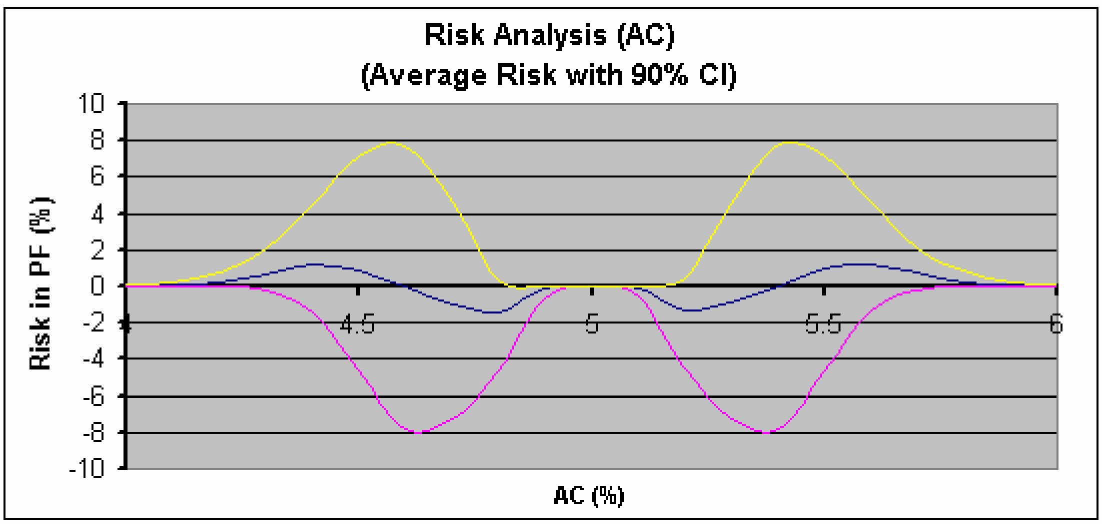


(i): Risk for $\sigma_{combined}$ = 0.69, σ_{device} = 0.25, $\sigma_{production}$ = 0.64, N = 15, Pay Cap = 103

Figure A2.2 (g)-(i): Risk plots obtained from PaySim for different parameter levels



(j): Risk for $\sigma_{combined} = 0.69$, $\sigma_{device} = 0.20$, $\sigma_{production} = 0.66$, N = 5, Pay Cap = 103



(k): Risk for $\sigma_{combined} = 0.14$, $\sigma_{device} = 0.075$, $\sigma_{production} = 0.12$, N = 5, Pay Cap = 103

Figure A2.2 (j)-(k): Risk plots obtained from PaySim for different parameter levels

There are some important points that can be noticed from the plots presented here.

(1) The magnitude of risk appears to be proportional to the ratio of device standard deviation to production standard deviation. For example, in risk plots for density (Figure A2.2) when this ratio increases from 0.23 to 0.41 maximum risk doubles from 0.8% to 1.6%. Similarly in the case of AC when the ratio goes up from 0.31 to 1.2 the risk increases from 2.4% to 6.8%. In the case of density, the risk increases from 2.3% to 2.8% with an increase in the ratio from 0.22% to 0.30%.
(2) The values of combined standard deviations used in the analysis are close to those actually observed in the field. Based on this it can be said that density seems to have

much lesser risk in pay factor, in general, than Voids. AC is shown to have the maximum risk involved.

(3) It is also noticeable that number of samples used for determining the pay factor exhibits an inverse relationship with the risks involved in payment. As the number of samples becomes smaller, the confidence interval on risk widens. A clear contrast can be seen between the plots for AC and voids with N= 15 and N=5.

APPENDIX A3

BiasSim

During highway construction, variations in the overall quality are unavoidable. Variability in as-constructed properties depend on production and measurement variability. In addition to variability around the actual value, a shift in measured properties, or bias, may also exist. The BiasSim program was developed to simulate the effects of measurements bias introduced by the contractor or agency.

The main simulation engine relies on generating a normally distributed random number sequence with mean and standard deviation as estimated from observations of actual field project overall standard deviations in Illinois (ERS demonstration projects). In the first stage standard deviation reflects only the production-realted variability. This represents the as-constructed quality of the pavement before measurement variability is introduced.

Measurement error due to variability of the instrument and/or test procedure is also expected to follow a normal distribution. Measurement error will be different for different instruments and different agencies (depending on differences in lab, operator etc.) Assuming that the mean of the error remains zero, a suitable estimate of standard deviation for measurement error was then used to generate two normally distributed error value sequences in the quality characteristic under consideration. These errors are then induced in the values generated earlier with certain mean and production variability. The resulting two sequences therefore simulate measurements taken by the contractor and that by the agency, assuming that there was no bias. Since the primary goal is to study the effect of bias, pay factor determined with these data could be considered as the reference pay factor for determining risk due to bias alone.

The final step in data generation then, is to introduce bias in the contractor and agency measurement values. Bias values have been determined for some ERS demonstration projects. These values can be used for the study here. In essence bias signifies the shift in the measurement from the actual value. A later section will show how bias can be calculated from actual field data from any project. The data with bias therefore simulate the actual measurements that one would obtain in the field for the quality characteristic concerned. Pay factor determined from these data is the pay factor that the agency will arrive at if useds actual project data.

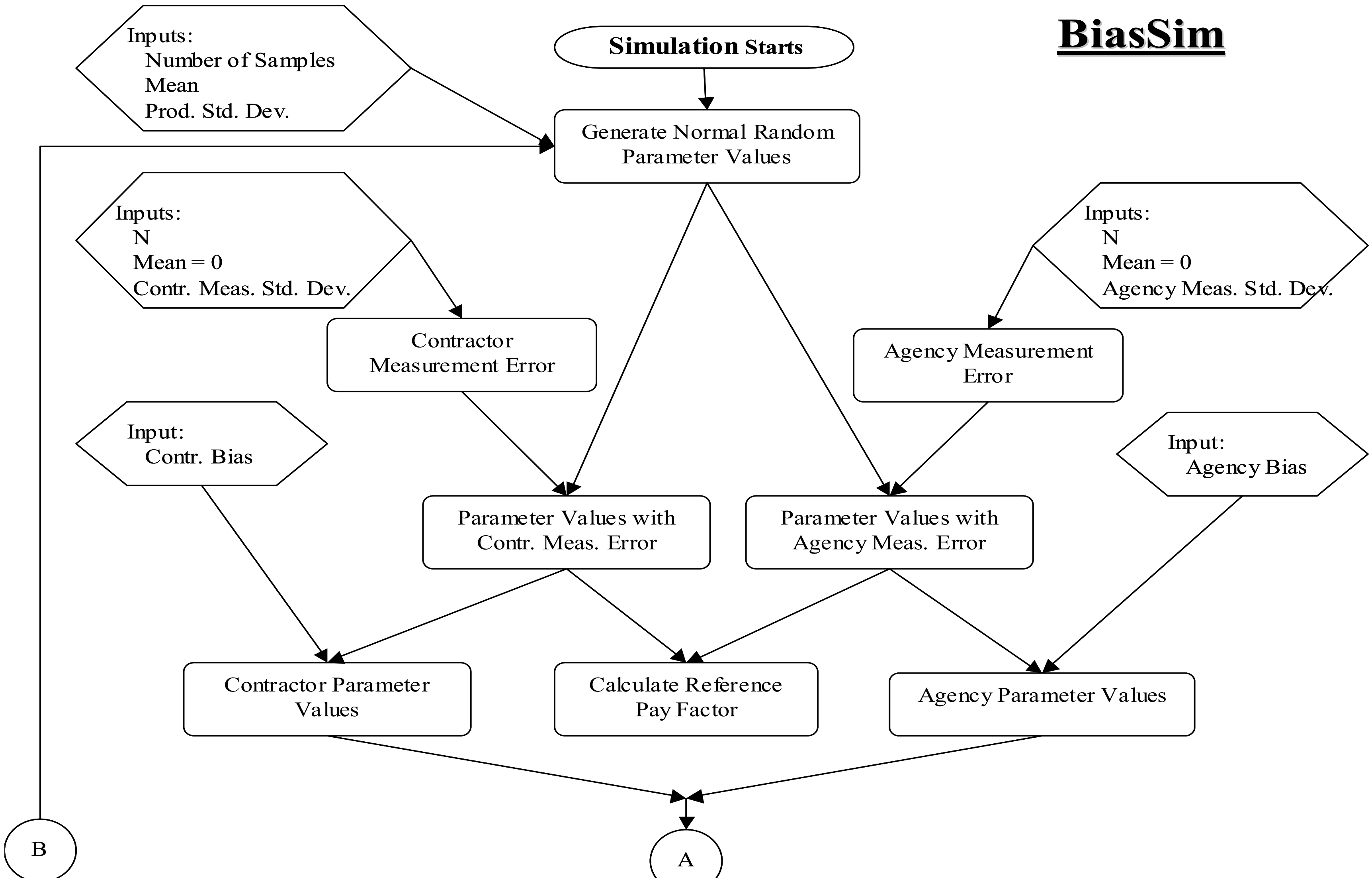
Determining Bias Magnitude

Table A3.1 shows the example of calculation of bias in a job. Suppose that 10 split samples were taken to determine the as-constructed density of a pavement in District 8. Next, the contractor and agency run each of the split samples in their own lab and the results shown in Table A3.1 are obtained. The difference between these two sets of readings is the estimate of the difference in density measurement between the contractor and the agency for the same material. Here it is assumed that density split samples are identical. This difference includes the measurement variability, which is always present. But measurement variability, being random in nature and generally normally distributedn will have a mean close to zero. But if the mean of the differences is not close to zero, an estimate of bias is obtained.

Table A3.1: Example bias calculation

Job	Contractor	Agency	Difference	Mean of Diff. (Bias)
District 8	92.6	91.8	0.8	**0.11**
	93.7	93.8	-0.2	
	93.9	93.9	0.0	
	92.8	93.4	-0.6	
	93.9	93.9	0.0	
	93.9	94.2	-0.3	
	92.8	91.8	1.0	
	95.5	95.1	0.3	
	94.2	94.5	-0.3	
	95.0	94.7	0.4	

Without additional information, it is not possible to determine how much of the bias was contributed by the contractor and how much was contributed by the agency. That notwithstanding, BiasSim can be used to study the amount of risk that bias poses to the pay factor calculation.



Key: Std. Dev.- Standard Deviation; Contr. – Contractor; Meas – Measurement; Prod. – Production; N – No. of Samples

Figure A3.1a: BiasSim Program Flowchart (1 of 2)

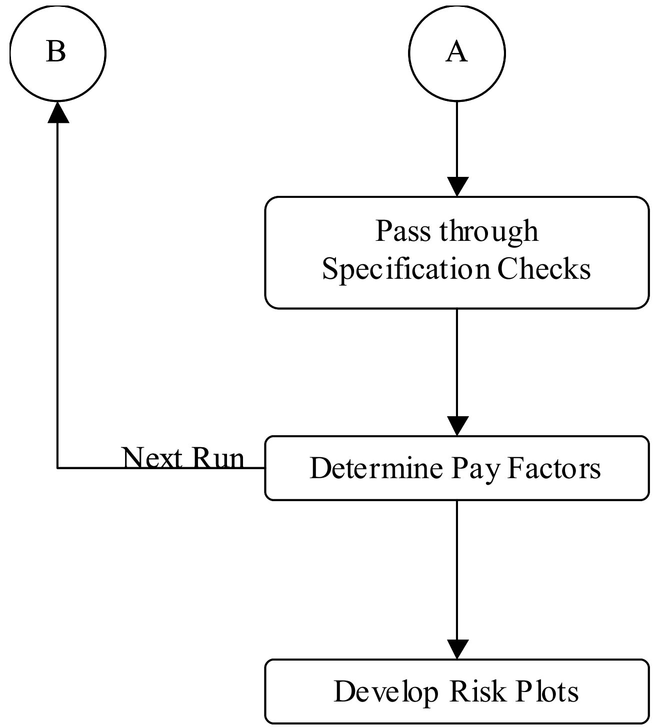
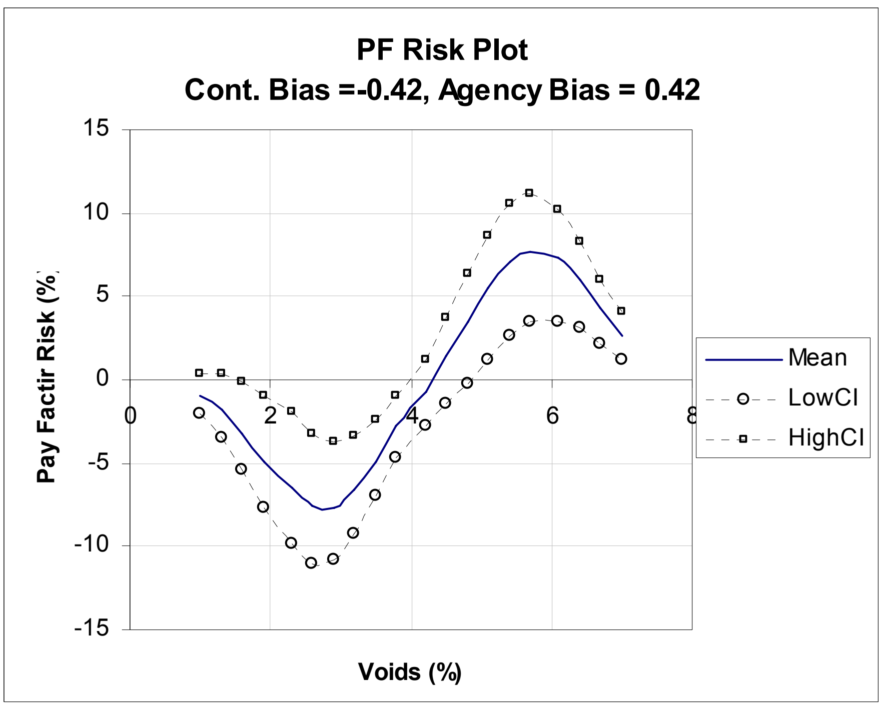


Figure A3.1b: BiasSim Program Flowchart (2 of 2)

Inputs for BiasSim

The following parameters can be studied in BiasSim:

(1) Quality Characteristic to be analyzed
(2) Production variability
(3) Device variability for contractor (multiple inputs possible)
(4) Device variability for agency (multiple inputs possible)
(5) Sample size per job
(6) Number of cases to be analyzed (for batch processing)
(7) Range of quality characteristic values for analysis
(8) Specification limits
(9) Comparison tolerances
(10) Precision desired in simulation
(11) Confidence interval



Key: Cont. – Contractor; Low CI – Lower Limit of Confidence Interval; High CI – Upper Limit of Confidence Interval

Figure A3.2: Typical risk plot obtained from BiasSim

Output from BiasSim

BiasSim was developed to perform the risk analysis for different magnitudes and signs of bias in the measurements taken by the contractor and the agency in different situations. The outputs from the simulation, therefore, are plots showing risk in pay factor (%PF) for a given set of parameters and in the range of analysis desired. In case of batch processing, the simulation runs all the cases together and then generates the plots. The simulator also has a batch processing mode wherein all the cases in the batch are executed before the generation of the various plots. The output data and the plots are stored in a separate file designated by the user. Batch processing allows for multiple sets of contractor and agency bias to be simulated. But the other parameters remain fixed for any single run. Figure A3.2 shows a typical plot generated by BiasSim. High CI and Low CI refer to the upper and lower limit of confidence interval, respectively.

Sensitivity Analysis

A senstitivity analysis using BiasSim is now presented. Risk associated in the determination of pay factor for voids in plant produced HMA is studied. Table A3.2 shows the combination of bias values that were used in this sensitivity analysis.

Table A3.2: Bias values used in the sensitivity analysis

Contractor	Agency	Contractor	Agency
0.00	0.00		
-0.42	0.00	0.00	-0.42
-0.21	0.00	0.00	-0.21
0.42	0.00	0.00	0.42
0.21	0.00	0.00	0.21
-0.21	+0.21	+0.21	-0.21
-0.42	+0.42	+0.42	-0.42

Figures A3.3-A3.6 present the results of the sensitivity analysis conducted using BiasSim. The three lines show mean risk (solid line), while the dashed lines show the limits of the 90% confidence intervals. Some important points that can be derived from these results are:

- Plot 1 (Figure A3.3) shows the risk when both contractor and agency bias are zero. This is a verification case, showing that BiasSim returns a result of zero risk when bias is input as zero.

- Plot 2 shows the risks when contractor bias is –0.42, i.e. when the contractor's void measurements are consistently 0.42% lower than the actual value, while the agency measures the correct value. Risk due to bias varies as void level increases from 1% to 7%. The maximum mean risk is 10% for both the contractor and agency depending on the void value. Since the difference between the mean agency voids and mean contractor voids is well within the prescribed comparison tolerance, their values will usually pass the comparison and therefore, the contractor's voids will be used for pay calculation.

Since the contractor is measuring less than actual value, the burden of risk falls on the contractor for lower void levels. For higher void levels, the risk shifts to the agency.

- When the contractor bias is zero but agency bias is –0.42 (reverse of the last case) (plot 3) they will still compare well most of the times because the comparison tolerance is 1%. So, contractor's values will be used most of the time, which do not have any bias, and therefore the risk is zero all through the range of voids analyzed.

- When the contractor has positive high bias (plot 5) the trend is reverse of that observed for the high negative bias.

- In the case when the contractor has high negative bias and the agency has high positive bias (plot 10) the probability of a successful QA comparison is smaller than the case shown in plots 2 and 6. As a result, the probability of agency voids being used in the pay calculation is higher, although the likelihood is still under 50%. Because in some cases the contractor voids are used (which are less than the actual value) and sometimes agency voids are used (which are higher than the actual voids), the overall mean risk is smaller. However, the 90% confidence interval is much broader than that observed in plots 2 and 6. Plot 12 shows that confidence intervals narrow when the bias magnitudes are halved, simimar to that observed in plots 4 and 8.

The source code for BiasSim, which is written in MatLab, is provided in Appendix B3.

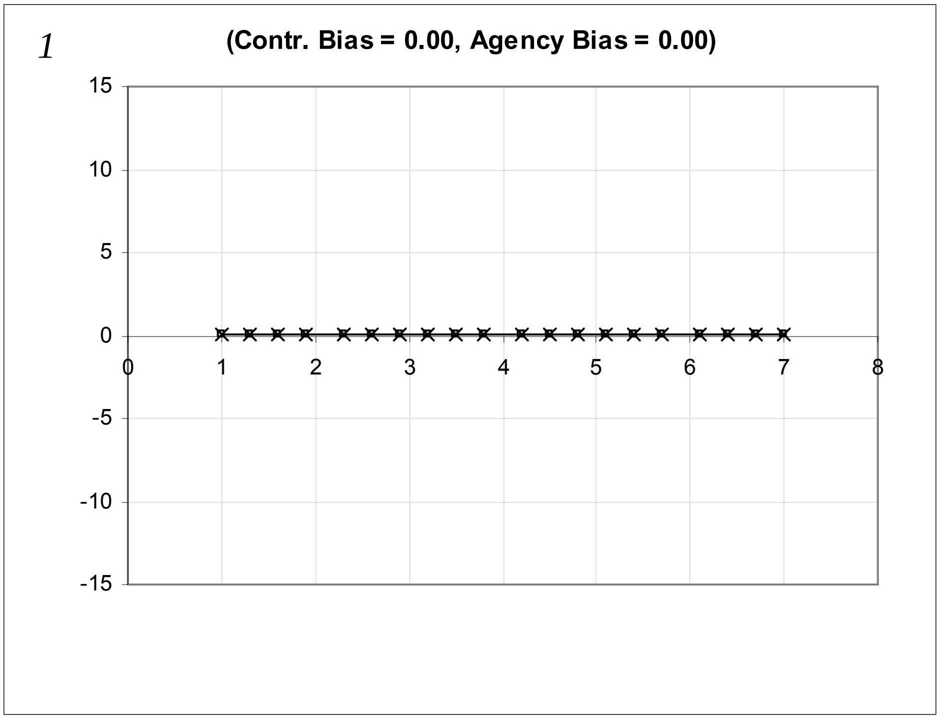
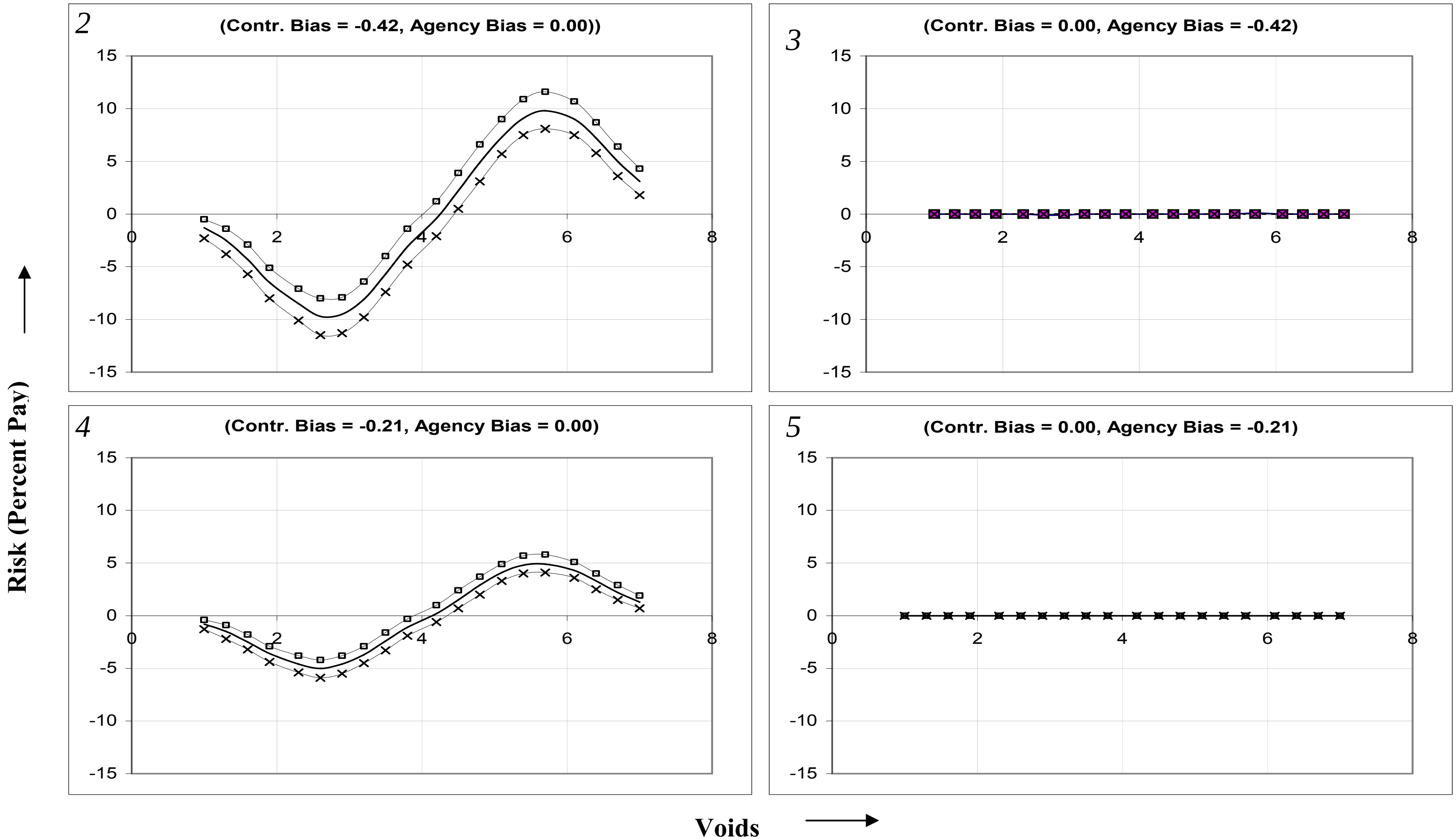


Figure A3.3: BiasSim risk plot from sensitivity analysis



Key: Contr. - Contractor

Figure A3.4: BiasSim risk plot from sensitivity analysis

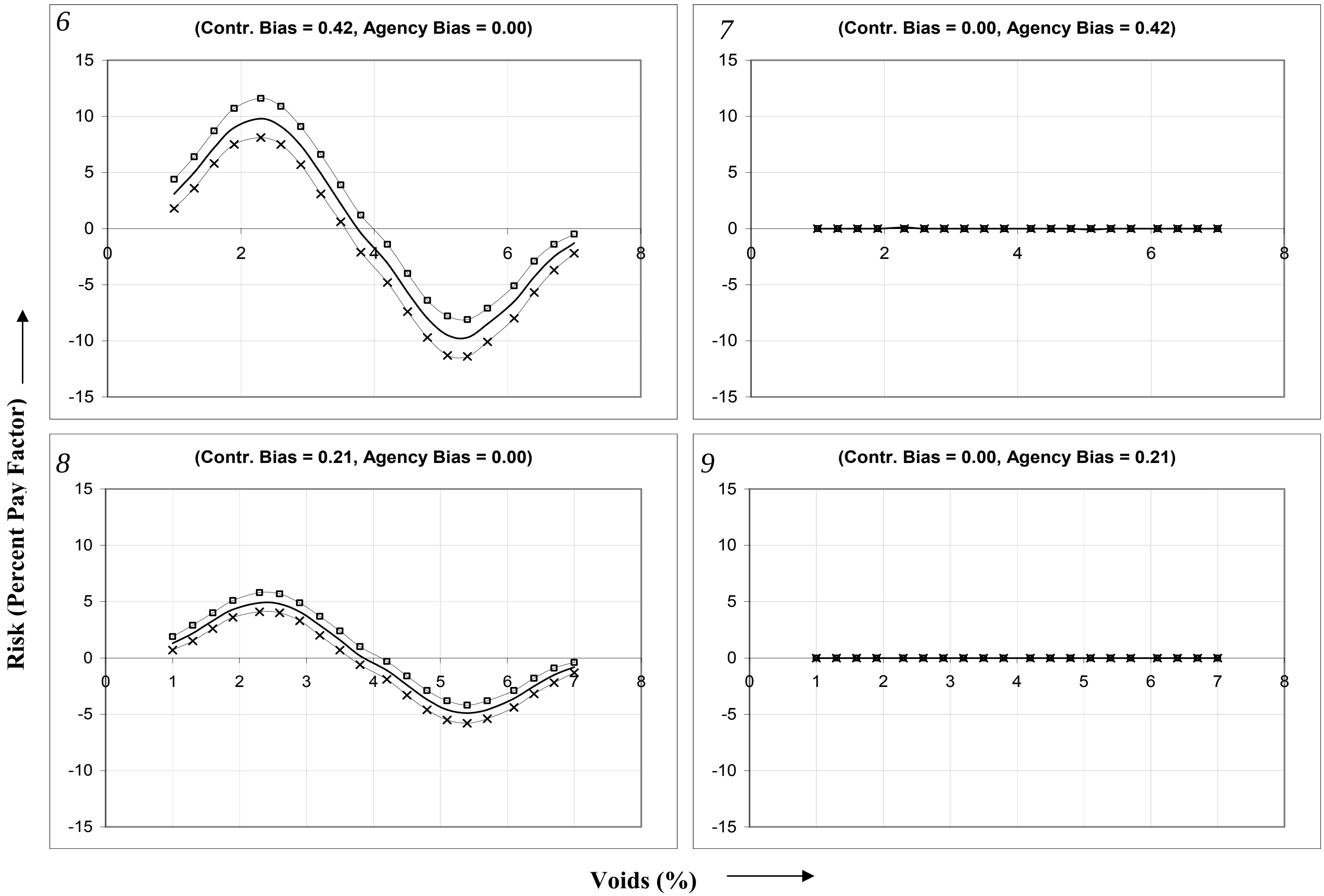


Figure A3.5: BiasSim risk plot from sensitivity analysis

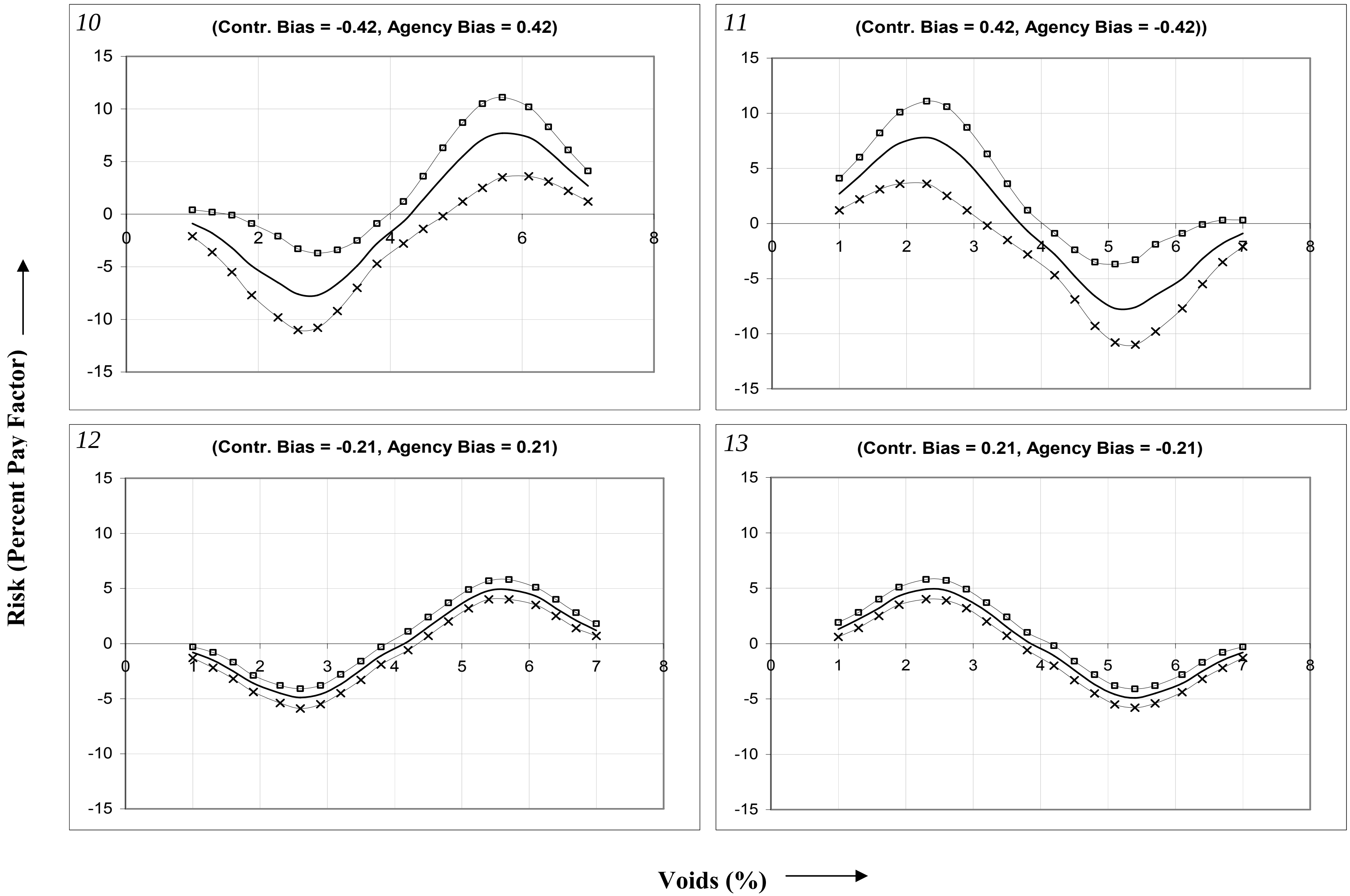


Figure A3.6: BiasSim risk plot from sensitivity analysis

APPENDIX B1

Source Code for ILLISIM

```
Dim lotavg(10) As Double
Dim lotsig(10) As Double
Dim lotpwl(10) As Double
Dim lotpay(10) As Double
Dim lotpaydiff(10) As Double
Dim lotval(15) As Double
Dim subavg(3) As Double
Dim Dev(5) As Single
Dim Devicestd(15) As Single

Function NormVal(Avg, sigma1, y)

NormVal = (1 / (sigma1 * (2 * 3.1415926) ^ 0.5)) * (2.71828 ^ -(((y - Avg) ^ 2) / (2 * sigma1 ^
2)))

End Function

Sub IntroPage()

Sheets("Intro").Activate
Range("A1").Select

End Sub

Sub InputData()

Sheets("RunData").Activate
Range("A1").Select

End Sub

Sub RunSim()

Workbooks("ILLI-SIM2.xls").Sheets("RunData").Activate
simresfilenum = Cells(17, 3)

Workbooks.Open Filename:="C:\ILLISIM\sim results template.xls"
ActiveWorkbook.SaveAs Filename:="C:\Sim Results\sim results" & simresfilenum & ".xls"

Workbooks("ILLI-SIM2.xls").Sheets("Intro").Activate

Call NewBaseline
```

```
Call Yderive

Workbooks("ILLI-SIM2.xls").Sheets("RunData").Activate
Cells(17, 3) = Cells(17, 3) + 1

Sheets("Intro").Activate

End Sub

Sub Yderive()

NumRuns = Worksheets("RunData").Range("C16")

For q = 1 To NumRuns

Sheets("ILLISIM").Activate
Range("B22:HD35").Clear

If q = 1 Then
   Sheets("RunData").Select
     Range(Cells(3, 4), Cells(14, 4)).Copy
   Sheets("ILLISIM").Activate
     Range("B1:B12").Select
   Selection.PasteSpecial Paste:=xlValues
   If Cells(3, 2) = 1 Then Cells(3, 2) = "strat"
   If Cells(3, 2) = 2 Then Cells(3, 2) = "avg"
   Cells(13, 2) = Sheets("RunData").Range("C18")

   Sheets("RunData").Activate
     Range(Cells(20, 4), Cells(23, 4)).Select
   Selection.Copy
   Sheets("ILLISIM").Activate
     Range("K9:K12").Select
   Selection.PasteSpecial Paste:=xlValues
End If
If q = 6 Then
   Sheets("RunData").Select
     Range(Cells(3, 5), Cells(14, 5)).Copy
   Sheets("ILLISIM").Activate
     Range("B1:B12").Select
   Selection.PasteSpecial Paste:=xlValues
   If Cells(3, 2) = 1 Then Cells(3, 2) = "strat"
   If Cells(3, 2) = 2 Then Cells(3, 2) = "avg"
   Cells(13, 2) = Sheets("RunData").Range("C18")
```

```
    Sheets("RunData").Activate
      Range(Cells(20, 5), Cells(23, 5)).Select
    Selection.Copy
    Sheets("ILLISIM").Activate
      Range("K9:K12").Select
    Selection.PasteSpecial Paste:=xlValues
End If
If q = 11 Then
    Sheets("RunData").Select
      Range(Cells(3, 6), Cells(14, 6)).Copy
    Sheets("ILLISIM").Activate
      Range("B1:B12").Select
    Selection.PasteSpecial Paste:=xlValues
    If Cells(3, 2) = 1 Then Cells(3, 2) = "strat"
    If Cells(3, 2) = 2 Then Cells(3, 2) = "avg"
    Cells(13, 2) = Sheets("RunData").Range("C18")

    Sheets("RunData").Activate
      Range(Cells(20, 6), Cells(23, 6)).Select
    Selection.Copy
    Sheets("ILLISIM").Activate
      Range("K9:K12").Select
    Selection.PasteSpecial Paste:=xlValues
End If
If q = 16 Then
    Sheets("RunData").Select
      Range(Cells(3, 7), Cells(14, 7)).Copy
    Sheets("ILLISIM").Activate
      Range("B1:B12").Select
    Selection.PasteSpecial Paste:=xlValues
    If Cells(3, 2) = 1 Then Cells(3, 2) = "strat"
    If Cells(3, 2) = 2 Then Cells(3, 2) = "avg"
    Cells(13, 2) = Sheets("RunData").Range("C18")

    Sheets("RunData").Activate
      Range(Cells(20, 7), Cells(23, 7)).Select
    Selection.Copy
    Sheets("ILLISIM").Activate
      Range("K9:K12").Select
    Selection.PasteSpecial Paste:=xlValues
End If

Cells(9, 8) = q

MeanValue = Worksheets("ILLISIM").Range("B1")
Processsig = Worksheets("ILLISIM").Range("B2")
```

```
Method = Worksheets("ILLISIM").Range("B3")
Devsig = Worksheets("ILLISIM").Range("B4")
NumRead = Worksheets("ILLISIM").Range("B5")
NumSubs = Worksheets("ILLISIM").Range("B6")
NumLots = Worksheets("ILLISIM").Range("B7")
NumJobs = Worksheets("ILLISIM").Range("B8")
USL = Worksheets("ILLISIM").Range("B9")
LSL = Worksheets("ILLISIM").Range("B10")
pfconst = Worksheets("ILLISIM").Range("B11")
pfslope = Worksheets("ILLISIM").Range("B12")
char = Worksheets("ILLISIM").Range("B13")

corravg = Worksheets("ILLISIM").Range("K9")
corrsig = Worksheets("ILLISIM").Range("K10")
corrpwl = Worksheets("ILLISIM").Range("K11")
corrlotpay = Worksheets("ILLISIM").Range("K12")
corrjobpay = Worksheets("ILLISIM").Range("K12")
If corrjobpay > 102 Then corrjobpay = 102

For h = 1 To NumJobs

Cells(10, 8) = h

For J = 1 To NumLots

randgen
randgen2

Sum = 0

For i = 1 To NumRead

'Calculates initial characteristic value from process deviation
   Avg = MeanValue
   sigma = Processsig
   Yfin = 100
   Area = 0
   inc = 0.05


   For Y1 = (Avg - 4 * sigma) To Yfin Step inc

      Y2 = Y1 + inc
      X1 = NormVal(Avg, sigma, Y1)
      X2 = NormVal(Avg, sigma, Y2)
      A = inc * (X1 + X2) / 2
```

```
        Area = Area + A
        If Area >= Cells(i + 3, 4) Then GoTo 10

    Next Y1

10  y = (Y1 + Y2) / 2
    lotval(i) = y

'Considering measurement device deviation
    If Devsig = "Nuke" Then
        Devicestd(i) = 0.006003 * (lotval(i)) ^ 2 - 1.116363 * (lotval(i)) + 52.616
    ElseIf IsNumeric(Devsig) Then
        Devicestd(i) = Devsig
    Else: MsgBox "Cell B3 must read 'Nuke' or '#.##'.", vbOKCancel, "Invalid Input"
    End If

'Calculates "final" density measurement
    Yfin = 100
    Area = 0
    inc = 0.05

    For Y3 = (y - 4 * Devicestd(i)) To Yfin Step inc

        Y4 = Y3 + inc
        X3 = NormVal(y, Devicestd(i), Y3)
        X4 = NormVal(y, Devicestd(i), Y4)
        A = inc * (X3 + X4) / 2
        Area = Area + A
        If Area >= Cells(i + 3, 5) Then GoTo 20

    Next Y3

20 YF = (Y3 + Y4) / 2

   lotval(i) = YF

   Sum = Sum + YF

Next i

'Caculates lot average and std based on "strat" or "avg" method
If Method = "strat" Then
    lotavg(J) = Sum / NumRead
    SumSquare = 0
    For n = 1 To NumRead
        s = (lotval(n) - lotavg(J)) ^ 2
```

```
        SumSquare = SumSquare + s
    Next n
    lotsig(J) = (SumSquare / (NumRead - 1)) ^ 0.5

ElseIf Method = "avg" Then
    SumLot = 0
    For n = 1 To NumSubs
        SumSub = 0
        For NN = 1 To CInt(NumRead / NumSubs)
            Num = NN + 5 * (n - 1)
            SumSub = SumSub + lotval(Num)
        Next NN
        subavg(n) = SumSub / CInt(NumRead / NumSubs)
        SumLot = SumLot + subavg(n)
    Next n
    lotavg(J) = SumLot / NumSubs
    SumSquare = 0
    For p = 1 To NumSubs
        s = (subavg(p) - lotavg(J)) ^ 2
        SumSquare = SumSquare + s
    Next p
    lotsig(J) = (SumSquare / (NumSubs - 1)) ^ 0.5

Else: MsgBox "Cell B3 must read 'strat' or 'avg'.", vbOKCancel, "Invalid Input"

End If

'Calculates PWL per Lot

Avg = lotavg(J)
sigma = lotsig(J)

If sigma < 0.05 Then sigma = 0.05

lotpwl(J) = PWL(Avg, sigma, NumRead, USL, LSL)

'Calculates Pay Factor and PF Difference

lotpay(J) = pfconst + pfslope * lotpwl(J)
lotpaydiff(J) = lotpay(J) - corrlotpay

Next J

lotpaysum = 0
maxoverpay = 0
maxunderpay = 0
```

```
Cells(22, 10 * (h - 1) + 2) = h

'Reports per-Lot Numbers
For K = 1 To NumLots

Cells(23, 10 * (h - 1) + K + 1) = lotavg(K)
Cells(24, 10 * (h - 1) + K + 1) = lotsig(K)
Cells(25, 10 * (h - 1) + K + 1) = lotpwl(K)
Cells(26, 10 * (h - 1) + K + 1) = lotpay(K)
Cells(27, 10 * (h - 1) + K + 1) = lotpaydiff(K)

lotpaysum = lotpaysum + lotpay(K)

If lotpaydiff(K) > maxoverpay Then maxoverpay = lotpaydiff(K)
If lotpaydiff(K) < maxunderpay Then maxunderpay = lotpaydiff(K)

Next K

'Reports per-Job Numbers
jobpay = lotpaysum / NumLots

If jobpay <= 102 Then
   jobpay = jobpay
   Else: jobpay = 102
End If

jobpaydiff = jobpay - corrjobpay

Cells(30, h + 1) = h
Cells(31, h + 1) = jobpay
Cells(32, h + 1) = corrjobpay
Cells(33, h + 1) = jobpaydiff
Cells(34, h + 1) = maxoverpay
Cells(35, h + 1) = maxunderpay

Next h

SaveResults

Next q

ReduceResults

End Sub
```

```
Sub NewBaseline()

Sheets("Baseline").Visible = True

For KK = 1 To 4

Sheets("RunData").Select
Range(Cells(3, 3 + KK), Cells(14, 3 + KK)).Copy

Sheets("ILLISIM").Activate
Range("B1:B12").Select
   Selection.PasteSpecial Paste:=xlValues
If Cells(3, 2) = 1 Then Cells(3, 2) = "strat"
If Cells(3, 2) = 2 Then Cells(3, 2) = "avg"
Cells(13, 2) = Sheets("RunData").Range("C18")

baseline

Sheets("ILLISIM").Activate
Range("K9:K12").Select
   Selection.Copy
Sheets("RunData").Activate
Cells(20, 3 + KK).Select
   Selection.PasteSpecial Paste:=xlValues
Next KK

Sheets("ILLISIM").Activate

Sheets("Baseline").Visible = False

End Sub


Dim storebook As String
Function PWL(X, y, z, USL, LSL)
   Qu = 0#
   Ql = 0#

   If z = 3 Then
   ' 3 samples

  Qu = (USL - X) / y
   Ql = (X - LSL) / y

   If Qu > -1.16 Then
```

```
      If Qu < 0 Then
         Pdu = 100 - 1 * (50 + 1.2444 * (Abs(Qu)) ^ 4 - 6.3854 * (Abs(Qu)) ^ 3 + 0.8538 *
(Abs(Qu)) ^ 2 + 38.302 * (Abs(Qu)))
         Else:
         If Qu < 1.16 Then
            Pdu = 1 * (50 + 1.2444 * (Qu) ^ 4 - 6.3854 * (Qu) ^ 3 + 0.8538 * (Qu) ^ 2 + 38.302 *
(Qu))
            Else: Pdu = 0
         End If
      End If
   Else: Pdu = 100
   End If

   If Ql > -1.16 Then
      If Ql < 0 Then
         Pdl = 100 - 1 * (50 + 1.2444 * (Abs(Ql)) ^ 4 - 6.3854 * (Abs(Ql)) ^ 3 + 0.8538 *
(Abs(Ql)) ^ 2 + 38.302 * (Abs(Ql)))
         Else:
         If Ql < 1.16 Then
            Pdl = 1 * (50 + 1.2444 * (Ql) ^ 4 - 6.3854 * (Ql) ^ 3 + 0.8538 * (Ql) ^ 2 + 38.302 *
(Ql))
            Else: Pdl = 0
         End If
      End If
   Else: Pdl = 100
   End If

   PWLtot = (100 - Pdu) + (100 - Pdl) - 100

   End If


   If z = 4 Then
    ' 4 samples


   Qu = (USL - X) / y
   Ql = (X - LSL) / y

   If Qu > -1.5 Then
      If Qu < 0 Then
         Pdu = 100 - 1 * (50 - 33.333 * (Abs(Qu)))
         Else:
         If Qu < 1.5 Then
            Pdu = 1 * (50 - 33.333 * (Qu))
            Else: Pdu = 0
```

```
        End If
     End If
  Else: Pdu = 100
  End If


  If Ql > -1.5 Then
     If Ql < 0 Then
        Pdl = 100 - 1 * (50 - 33.333 * (Abs(Ql)))
        Else:
        If Ql < 1.5 Then
           Pdl = 1 * (50 - 33.333 * (Ql))
           Else: Pdl = 0
        End If
     End If
  Else: Pdl = 100
  End If

  PWLtot = (100 - Pdu) + (100 - Pdl) - 100

  End If


  If z = 5 Then
   ' 5 samples


  Qu = (USL - X) / y
  Ql = (X - LSL) / y

  If Qu > -1.8 Then
     If Qu < 0 Then
        Pdu = 100 - 1 * (50 + 3.3742 * (Abs(Qu)) ^ 3 - 2.4068 * (Abs(Qu)) ^ 2 - 34.691 *
(Abs(Qu)))
        Else:
        If Qu < 1.8 Then
           Pdu = 1 * (50 + 3.3742 * (Qu) ^ 3 - 2.4068 * (Qu) ^ 2 - 34.691 * (Qu))
           Else: Pdu = 0
        End If
     End If
  Else: Pdu = 100
  End If

  If Ql > -1.8 Then
     If Ql < 0 Then
```

```
      Pdl = 100 - 1 * (50 + 3.3742 * (Abs(Ql)) ^ 3 - 2.4068 * (Abs(Ql)) ^ 2 - 34.691 *
(Abs(Ql)))
      Else:
      If Ql < 1.8 Then
        Pdl = 1 * (50 + 3.3742 * (Ql) ^ 3 - 2.4068 * (Ql) ^ 2 - 34.691 * (Ql))
        Else: Pdl = 0
      End If
    End If
  Else: Pdl = 100
  End If

  PWLtot = (100 - Pdu) + (100 - Pdl) - 100
  End If

  If z = 6 Then
   ' 6 samples

  Qu = (USL - X) / y
  Ql = (X - LSL) / y

  If Qu > -2.03 Then
    If Qu < 0 Then
      Pdu = 100 - 1 * (50 + 2.9406 * (Abs(Qu)) ^ 3 - 0.0022 * (Abs(Qu)) ^ 2 - 36.742 *
(Abs(Qu)))
      Else:
      If Qu < 2.03 Then
        Pdu = 1 * (50 + 2.9406 * (Qu) ^ 3 - 0.0022 * (Qu) ^ 2 - 36.742 * (Qu))
        Else: Pdu = 0
      End If
    End If
  Else: Pdu = 100
  End If

  If Ql > -2.03 Then
    If Ql < 0 Then
      Pdl = 100 - 1 * (50 + 2.9406 * (Abs(Ql)) ^ 3 - 0.0022 * (Abs(Ql)) ^ 2 - 36.742 *
(Abs(Ql)))
      Else:
      If Ql < 2.03 Then
        Pdl = 1 * (50 + 2.9406 * (Ql) ^ 3 - 0.0022 * (Ql) ^ 2 - 36.742 * (Ql))
        Else: Pdl = 0
      End If
    End If
  Else: Pdl = 100
  End If
```

```
   PWLtot = (100 - Pdu) + (100 - Pdl) - 100
   End If


   If z = 7 Then
    ' 7 samples

   Qu = (USL - X) / y
   Ql = (X - LSL) / y

   If Qu > -2.23 Then
      If Qu < 0 Then
         Pdu = 100 - 1 * (50 - 0.815 * (Abs(Qu)) ^ 4 + 5.4299 * (Abs(Qu)) ^ 3 - 1.475 *
(Abs(Qu)) ^ 2 - 37.051 * (Abs(Qu)))
         Else:
         If Qu < 2.23 Then
            Pdu = 1 * (50 - 0.815 * (Qu) ^ 4 + 5.4299 * (Qu) ^ 3 - 1.475 * (Qu) ^ 2 - 37.051 *
(Qu))
            Else: Pdu = 0
         End If
      End If
   Else: Pdu = 100
   End If

   If Ql > -2.23 Then
      If Ql < 0 Then
         Pdl = 100 - 1 * (50 - 0.815 * (Abs(Ql)) ^ 4 + 5.4299 * (Abs(Ql)) ^ 3 - 1.475 * (Abs(Ql))
^ 2 - 37.051 * (Abs(Ql)))
         Else:
         If Ql < 2.23 Then
            Pdl = 1 * (50 - 0.815 * (Ql) ^ 4 + 5.4299 * (Ql) ^ 3 - 1.475 * (Ql) ^ 2 - 37.051 * (Ql))
            Else: Pdl = 0
         End If
      End If
   Else: Pdl = 100
   End If

   PWLtot = (100 - Pdu) + (100 - Pdl) - 100
   End If


   If z = 8 Then
    ' 8 samples

   Qu = (USL - X) / y
   Ql = (X - LSL) / y
```

```
    If Qu > -2.39 Then
        If Qu < 0 Then
            Pdu = 100 - 1 * (50 - 1.155 * (Abs(Qu)) ^ 4 + 6.4174 * (Abs(Qu)) ^ 3 - 1.8227 *
(Abs(Qu)) ^ 2 - 37.415 * (Abs(Qu)))
            Else:
            If Qu < 2.39 Then
                Pdu = 1 * (50 - 1.155 * (Qu) ^ 4 + 6.4174 * (Qu) ^ 3 - 1.8227 * (Qu) ^ 2 - 37.415 *
(Qu))
                Else: Pdu = 0
            End If
        End If
    Else: Pdu = 100
    End If

    If Ql > -2.39 Then
        If Ql < 0 Then
            Pdl = 100 - 1 * (50 - 1.155 * (Abs(Ql)) ^ 4 + 6.4174 * (Abs(Ql)) ^ 3 - 1.8227 * (Abs(Ql))
^ 2 - 37.415 * (Abs(Ql)))
            Else:
            If Ql < 2.39 Then
                Pdl = 1 * (50 - 1.155 * (Ql) ^ 4 + 6.4174 * (Ql) ^ 3 - 1.8227 * (Ql) ^ 2 - 37.415 * (Ql))
                Else: Pdl = 0
            End If
        End If
    Else: Pdl = 100
    End If
    PWLtot = (100 - Pdu) + (100 - Pdl) - 100
    End If


    If z = 9 Then
     ' 9 samples

    Qu = (USL - X) / y
    Ql = (X - LSL) / y

    If Qu > -2.53 Then
        If Qu < 0 Then
            Pdu = 100 - 1 * (50 - 1.2613 * (Abs(Qu)) ^ 4 + 6.6228 * (Abs(Qu)) ^ 3 - 1.5375 *
(Abs(Qu)) ^ 2 - 37.832 * (Abs(Qu)))
            Else:
            If Qu < 2.53 Then
                Pdu = 1 * (50 - 1.2613 * (Qu) ^ 4 + 6.6228 * (Qu) ^ 3 - 1.5375 * (Qu) ^ 2 - 37.832 *
(Qu))
                Else: Pdu = 0
```

```
          End If
       End If
    Else: Pdu = 100
    End If

    If Ql > -2.53 Then
       If Ql < 0 Then
          Pdl = 100 - 1 * (50 - 1.2613 * (Abs(Ql)) ^ 4 + 6.6228 * (Abs(Ql)) ^ 3 - 1.5375 *
(Abs(Ql)) ^ 2 - 37.832 * (Abs(Ql)))
          Else:
          If Ql < 2.53 Then
             Pdl = 1 * (50 - 1.2613 * (Ql) ^ 4 + 6.6228 * (Ql) ^ 3 - 1.5375 * (Ql) ^ 2 - 37.832 *
(Ql))
             Else: Pdl = 0
          End If
       End If
    Else: Pdl = 100
    End If
    PWLtot = (100 - Pdu) + (100 - Pdl) - 100

    End If

    If z >= 10 Then
         ' 10 samples
       Qu = (USL - X) / y
       Ql = (X - LSL) / y
       If Qu > -2.65 Then
          If Qu < 0 Then
             Pdu = 100 - 1 * (50 - 1.2579 * (Abs(Qu)) ^ 4 + 6.4455 * (Abs(Qu)) ^ 3 - 0.934 *
(Abs(Qu)) ^ 2 - 38.272 * (Abs(Qu)))
             Else:
             If Qu < 2.65 Then
                Pdu = 1 * (50 - 1.2579 * (Qu) ^ 4 + 6.4455 * (Qu) ^ 3 - 0.934 * (Qu) ^ 2 - 38.272 *
(Qu))
                Else: Pdu = 0
             End If
          End If
       Else: Pdu = 100
       End If

       If Ql > -2.65 Then
          If Ql < 0 Then
             Pdl = 100 - 1 * (50 - 1.2579 * (Abs(Ql)) ^ 4 + 6.4455 * (Abs(Ql)) ^ 3 - 0.934 *
(Abs(Ql)) ^ 2 - 38.272 * (Abs(Ql)))
             Else:
             If Ql < 2.65 Then
```

```
                Pdl = 1 * (50 - 1.2579 * (Ql) ^ 4 + 6.4455 * (Ql) ^ 3 - 0.934 * (Ql) ^ 2 - 38.272 *
(Ql))
                Else: Pdl = 0
              End If
            End If
        Else: Pdl = 100
        End If

PWLtot = (100 - Pdu) + (100 - Pdl) - 100
    End If

'Take care of very slight fitting error (<0.1 error)
If PWLtot < 0 Then PWLtot = 0
If PWLtot > 100 Then PWLtot = 100

PWL = PWLtot

End Function
```

Sub **SaveResults**()

```
SheetNum = Worksheets("ILLISIM").Range("H9")
storesheet = "Sheet" & SheetNum
simresfilenum = Worksheets("RunData").Range("C17").Value
storebook = "sim results" & simresfilenum & ".xls"

Workbooks("ILLI-SIM2.xls").Worksheets("ILLISIM").Range("A1:B13").Copy

Workbooks(storebook).Worksheets(storesheet).Activate
Range("B3").Select
    Selection.PasteSpecial Paste:=xlValues

Workbooks("ILLI-SIM2.xls").Worksheets("ILLISIM").Range("J9:K12").Copy

Workbooks(storebook).Worksheets(storesheet).Activate
Range("B15").Select
    Selection.PasteSpecial Paste:=xlValues

Workbooks("ILLI-SIM2.xls").Worksheets("ILLISIM").Range("A29:U35").Copy

Workbooks(storebook).Sheets(storesheet).Range("B23").Activate
    Selection.PasteSpecial Paste:=xlValues

Workbooks("ILLI-SIM2.xls").Worksheets("ILLISIM").Activate
Range("A21:GS27").Select
Selection.Copy
```

```
Workbooks(storebook).Worksheets(storesheet).Activate
   Range("D32").Select
   Selection.PasteSpecial Paste:=xlValues

Workbooks("ILLI-SIM2.xls").Activate
Sheets("ILLISIM").Select
Range("A1").Select

End Sub

Sub ReduceResults()

simresfilenum = Workbooks("ILLI-SIM2.xls").Worksheets("RunData").Range("C17").Value
reducebook = "sim results" & simresfilenum & ".xls"

Workbooks(reducebook).Activate

ReduceDataPWL

End Sub

Sub ReduceDataPWL()

'************************
'BY LOT and BY JOB
'************************

For J = 1 To 4
  For K = 1 To 5

KK = (J - 1) * 5 + K
MM = (K - 1) * 400

'Assign Fixed (master) and Variable sheet names
Shnmf = "Sheet" & (J - 1) * 5 + 2
Shnmv = "Sheet" & KK

Sheets(Shnmv).Activate

Rows("52:2100").Select
Selection.ClearContents

Cells(53, 10) = "Average LOT PWL-sheet"
Cells(54, 10).Activate
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-18]C[-5]:R[-18]C[194])"
```

```
For n = 1 To 20
   Cells(40, 10 * (n - 1) + 5).Activate
   ActiveCell.FormulaR1C1 = "=AVERAGE(R[-4]C[0]:R[-4]C[9])"
Next n

Cells(53, 20) = "Average JOB PWL-sheet"
Cells(54, 20).Activate
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-14]C[-15]:R[-14]C[184])"

Next K

Sheets(Shnmf).Activate

Cells(52, 4) = "PER LOT Analysis"
Cells(53, 5) = "Over PWL"
Cells(53, 6) = "Under PWL"
Cells(54, 4) = "Max"
Cells(55, 4) = "StdDev"
Cells(53, 8) = "Overall Avg LOT PWL"
Cells(54, 8) = (Sheets("Sheet" & (J - 1) * 5 + 1).Cells(54, 10) + _
    Sheets("Sheet" & (J - 1) * 5 + 2).Cells(54, 10) + _
    Sheets("Sheet" & (J - 1) * 5 + 3).Cells(54, 10) + _
    Sheets("Sheet" & (J - 1) * 5 + 4).Cells(54, 10) + _
    Sheets("Sheet" & (J - 1) * 5 + 5).Cells(54, 10)) / 5

Cells(52, 14) = "PER JOB Analysis"
Cells(53, 15) = "Over PWL"
Cells(53, 16) = "Under PWL"
Cells(54, 14) = "Max"
Cells(55, 14) = "StdDev"
Cells(53, 18) = "Overall Avg JOB PWL"
Cells(54, 18) = (Sheets("Sheet" & (J - 1) * 5 + 1).Cells(54, 20) + _
    Sheets("Sheet" & (J - 1) * 5 + 2).Cells(54, 20) + _
    Sheets("Sheet" & (J - 1) * 5 + 3).Cells(54, 20) + _
    Sheets("Sheet" & (J - 1) * 5 + 4).Cells(54, 20) + _
    Sheets("Sheet" & (J - 1) * 5 + 5).Cells(54, 20)) / 5

Next J

For J = 1 To 4

For K = 1 To 5

KK = (J - 1) * 5 + K
MM = (K - 1) * 400
NN = (K - 1) * 40
```

```
'Assign Fixed (master) and Variable sheet names
Shnmf = "Sheet" & (J - 1) * 5 + 2
Shnmv = "Sheet" & KK

Sheets(Shnmv).Activate
ActiveWindow.LargeScroll Down:=1

For i = 1 To 200
  If Sheets(Shnmv).Cells(36, i + 4) >= Sheets(Shnmf).Cells(54, 8) _
   Then Sheets(Shnmf).Cells(MM + 55 + i, 5) = _
   Sheets(Shnmv).Cells(36, i + 4) - Sheets(Shnmf).Cells(54, 8)
  If Sheets(Shnmv).Cells(36, i + 4) >= Sheets(Shnmf).Cells(54, 8) _
   Then Sheets(Shnmf).Cells(MM + 55 + i + 200, 5) = _
   (Sheets(Shnmv).Cells(36, i + 4) - Sheets(Shnmf).Cells(54, 8)) * -1#
  If Sheets(Shnmv).Cells(36, i + 4) < Sheets(Shnmf).Cells(54, 8) _
   Then Sheets(Shnmf).Cells(MM + 55 + i, 6) = _
   Sheets(Shnmv).Cells(36, i + 4) - Sheets(Shnmf).Cells(54, 8)
  If Sheets(Shnmv).Cells(36, i + 4) < Sheets(Shnmf).Cells(54, 8) _
   Then Sheets(Shnmf).Cells(MM + 55 + i + 200, 6) = _
   (Sheets(Shnmv).Cells(36, i + 4) - Sheets(Shnmf).Cells(54, 8)) * -1#
Next i

For ii = 1 To 20
  If Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) >= Sheets(Shnmf).Cells(54, 18) _
   Then Sheets(Shnmf).Cells(NN + 55 + ii, 15) = _
   Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) - Sheets(Shnmf).Cells(54, 18)
  If Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) >= Sheets(Shnmf).Cells(54, 18) _
   Then Sheets(Shnmf).Cells(NN + 55 + ii + 20, 15) = _
   (Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) - Sheets(Shnmf).Cells(54, 18)) * -1#
  If Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) < Sheets(Shnmf).Cells(54, 18) _
   Then Sheets(Shnmf).Cells(NN + 55 + ii, 16) = _
   Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) - Sheets(Shnmf).Cells(54, 18)
  If Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) < Sheets(Shnmf).Cells(54, 18) _
   Then Sheets(Shnmf).Cells(NN + 55 + ii + 20, 16) = _
   (Sheets(Shnmv).Cells(40, 10 * (ii - 1) + 5) - Sheets(Shnmf).Cells(54, 18)) * -1#
Next ii

Next K

Sheets(Shnmf).Activate

Cells(55, 5).Activate
ActiveCell.FormulaR1C1 = "=STDEV(R[1]C:R[2000]C)"

Cells(55, 6).Activate
```

```
ActiveCell.FormulaR1C1 = "=STDEV(R[1]C:R[2000]C)*-1."

Cells(54, 5).Activate
ActiveCell.FormulaR1C1 = "=Max(R[2]C:R[2001]C)"

Cells(54, 6).Activate
ActiveCell.FormulaR1C1 = "=Min(R[2]C:R[2001]C)"

Cells(55, 15).Activate
ActiveCell.FormulaR1C1 = "=STDEV(R[1]C:R[200]C)"

Cells(55, 16).Activate
ActiveCell.FormulaR1C1 = "=STDEV(R[1]C:R[200]C)*-1."

Cells(54, 15).Activate
ActiveCell.FormulaR1C1 = "=Max(R[2]C:R[201]C)"

Cells(54, 16).Activate
ActiveCell.FormulaR1C1 = "=Min(R[2]C:R[201]C)"

Next J

End Sub
```

APPENDIX B2
Source Code for PaySim

(1) Source code for the MS Excel interface (code in Visual Basic)

```
Sub makeinp()
'
' makeinp1 Macro
' Macro recorded 8/1/2001 by Anshu Manik
'

    Sheets("Home").Select
    precision = Range("ae8").Value
    cap = Range("af8").Value
    Name = Range("F4").Value
    file = Name & "\input.csv"
    Sheets("Input").Select
    If precision = "Crude" Then prec = 0
    If precision = "Low" Then prec = 1
    If precision = "Medium" Then prec = 2
    If precision = "High" Then prec = 3
    Range("A11").Value = prec
    If cap = "Before" Then capop = 0
    If cap = "After" Then capop = 1
    Range("a12").Value = capop
    Range("A1:A13").Select
    Selection.Copy
    Sheets("Home").Select
    Range("A1").Select
    Workbooks.Add
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    Application.CutCopyMode = False
    ActiveWorkbook.SaveAs Filename:=file, _
        FileFormat:=xlCSV, CreateBackup:=False
    ActiveWorkbook.SaveAs Filename:=file, _
        FileFormat:=xlCSV, CreateBackup:=False
    ActiveWindow.Close
    Range("A1").Select
    exec = Name & "\paysim.exe"
    retval = Shell(exec, vbNormalNoFocus)
  End Sub
```

```
Sub getout()
  ci = Range("ag8").Value
  Name = Range("F4").Value
  outfile = Name & "\out.csv"
  Workbooks.Open Filename:=outfile
  Range("A1:D750").Select
  Selection.Copy
  Windows("PaySim.xls").Activate
  Sheets("Out.CSV").Select
  Range("A2").Select
  ActiveSheet.Paste
  ActiveWindow.WindowState = xlMinimized
  ActiveWindow.Close
  ActiveWindow.WindowState = xlMaximized
  Windows("PaySim.xls").Activate
  Sheets("Home").Select
  Range("A1").Select

  ActiveSheet.Unprotect
  Para = Range("f5").Value
    Start = 2
  endplot = Start + Range("f14").Value
  ActiveSheet.ChartObjects("Chart 2").Activate
  ActiveChart.ChartArea.Select
  ActiveChart.Axes(xlCategory).AxisTitle.Select
  Selection.Characters.Text = Para
  Selection.AutoScaleFont = False
  With Selection.Characters(Start:=1, Length:=5).Font
      .Name = "Arial"
      .FontStyle = "Bold"
      .Size = 9.25
      .Strikethrough = False
      .Superscript = False
      .Subscript = False
      .OutlineFont = False
      .Shadow = False
      .Underline = xlUnderlineStyleNone
      .ColorIndex = xlAutomatic
  End With

ActiveChart.ChartArea.Select
  ActiveChart.ChartTitle.Select

  ci = ci * 100
```

```
    Selection.Characters.Text = "Risk Analysis" & Chr(10) & "(Average Risk with " & ci & "%
CI)"
    Selection.AutoScaleFont = False
    With Selection.Characters(Start:=1, Length:=40).Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 11.25
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With

        ActiveSheet.ChartObjects("Chart 2").Activate
    ActiveChart.ChartArea.Select
    ActiveChart.Axes(xlCategory).Select
    With ActiveChart.Axes(xlCategory)
        .MinimumScale = Range("f15").Value
        .MaximumScale = Range("f16").Value
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlAutomatic
        .ReversePlotOrder = False
        .ScaleType = xlLinear
        .DisplayUnit = xlNone
    End With
    ActiveChart.Axes(xlValue).Select
    With ActiveChart.Axes(xlValue)
        .MinimumScale = -20
        .MaximumScale = 20
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlAutomatic
        .ReversePlotOrder = False
        .ScaleType = xlLinear
        .DisplayUnit = xlNone
    End With
    ActiveChart.ChartArea.Select

    ActiveChart.SeriesCollection(1).XValues = "=Out.CSV!R2C1:R" & endplot & "C1"
    ActiveChart.SeriesCollection(1).Values = "=Out.CSV!R2C2:R" & endplot & "C2"
    ActiveChart.SeriesCollection(2).XValues = "=Out.CSV!R2C1:R" & endplot & "C1"
    ActiveChart.SeriesCollection(2).Values = "=Out.CSV!R2C3:R" & endplot & "C3"
```

```
    ActiveChart.SeriesCollection(3).XValues = "=Out.CSV!R2C1:R" & endplot & "C1"
    ActiveChart.SeriesCollection(3).Values = "=Out.CSV!R2C4:R" & endplot & "C4"
    ActiveWindow.Visible = False
    Windows("PaySim.xls").Activate
    ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
    ActiveWindow.Visible = False
    Windows("PaySim.xls").Activate

    Sheets("Home").Select



End Sub
```

Source code for paysim.exe (compiled in C)

```
#include <math.h>
#include "rngs.h"
#include "rvgs.h"
#include <io.h>
#include <string.h>
#include <FCNTL.H>
#include <stdio.h>
#include <stdlib.h>



  double Uniform(double a, double b)
// Returns a uniformly distributed real number between a and b.
// NOTE: use a < b

{
 return (a + (b - a) * Random());
}

  double Exponential(double m)
// Returns an exponentially distributed positive real number.
// NOTE: use m > 0.0

{
 return (-m * log(1.0 - Random()));
}



  double Normal(double m, double s)
// Returns a normal (Gaussian) distributed real number.
// NOTE: use s > 0.0
//
// Uses a very accurate approximation of the normal idf due to Odeh & Evans,
// J. Applied Statistics, 1974, vol 23, pp 96-97.

{
 const double p0 = 0.322232431088;     const double q0 = 0.099348462606;
 const double p1 = 1.0;                const double q1 = 0.588581570495;
 const double p2 = 0.342242088547;     const double q2 = 0.531103462366;
 const double p3 = 0.204231210245e-1;  const double q3 = 0.103537752850;
```

```
  const double p4 = 0.453642210148e-4;  const double q4 = 0.385607006340e-2;
  double u, t, p, q, z;

  u   = Random();
  if (u < 0.5)
    t = sqrt(-2.0 * log(u));
  else
    t = sqrt(-2.0 * log(1.0 - u));
  p   = p0 + t * (p1 + t * (p2 + t * (p3 + t * p4)));
  q   = q0 + t * (q1 + t * (q2 + t * (q3 + t * q4)));
  if (u < 0.5)
    z = (p / q) - t;
  else
    z = t - (p / q);
  return (m + s * z);
}



   double Chisquare(long n)
// Returns a chi-square distributed positive real number.
// NOTE: use n > 0

{
  long   i;
  double z, x = 0.0;

  for (i = 0; i < n; i++) {
    z  = Normal(0.0, 1.0);
    x += z * z;
  }
  return (x);
}




void simulate()
{
    float dstd=0;                                           //read parameter values from file
from input.csv file
            float pstd=0;
            float std;
            int   nsamp=0,njob=0,npoint=0;
            float lplot=0,uplot=0;
```

```
	float ulimit=0,llimit=0,pcap=0;
	int capopn=0;
	int   i,j,k;
	int precision = 0;                                              //for determinig the precision
of calculation of area under normal curve
	float p;
	float cifact, cilow, ciup;
	double areanorm(double x, int precsn);
	void sort (double risk[10000], int numelt);
	//read input file created by paysim.xls
	FILE *infile, *outfile;

	infile=fopen("input.csv","r");
	if (infile==NULL) printf("did not read");
		//else printf("read");
	i = fscanf(infile,"%f\n %f\n %i\n %i\n %i\n %f\n %f\n %f\n %f\n %f\n %i\n %i\n
%f\n",&dstd,&pstd,&nsamp,&njob,&npoint,&lplot,&uplot,&llimit,&ulimit,&pcap,&precision,&
capopn, &cifact);

   fclose(infile);
	//input file closed
	std=pow(pow(dstd,2)+pow(pstd,2),0.5);

	printf("Device Std Dev           = %f\nProcess Std Dev          = %f\n",dstd,pstd);
	printf("Number of Samples        = %i\nNumber of Sublots        = %i\n",nsamp,njob);
	printf("Points to be Plotted     = %i\nLower Limit of Plot      = %f\nUpper Limit of Plot
= %f\n",npoint,lplot,uplot);
	printf("Lower Specification Limit = %f\nUpper Specification Limit = %f\nPay Cap
= %f\n",llimit,ulimit,pcap);
	printf("Precision                = %i  (0=Crude, 1=low, 2=Med, 3=High)\n",precision);
	printf("Pay Cap Option           = %i  (0=Before averaging over sublot, 1=After averaging
over sublot\n",capopn);
	printf("Confidence Level         = %f\n",cifact);




	long x;                                                         //actual simulation starts here
	const int num=6000;
	double rn=0;
	double chi[10000];
	double norm[10000];
	double avg;
	double pwl0=0;
	double target=0;
```

```
        double Chisquare(long n);
        void PutSeed(long x);
        void SelectStream(int index);
        double Normal(double m, double s);

        for (j=1;j<=num+1;j++)                    //initialise the arrays
                {
                        chi[j]=0;
                        norm[j]=0;
                }

        SelectStream(0);                    /* select the default stream */
   PutSeed(-1);                       /* and set the state to 1    */

        outfile=fopen("out.csv","w");
        for (i=1;i<=num;i++)
                {
                        chi[i]=Chisquare(nsamp-1);  //generating chi sq and normal rnd nos
                        norm[i]=Normal(0,1);
                }

        cilow=0.5*(1-cifact);               //for getting lower limit of confidence interval
        ciup =cifact+0.5*(1-cifact);  //for getting upper limit of confidence interval

        float rangex =uplot-lplot;
        float increment=rangex/npoint;
        double left, right;
        double meanai, halfwidth,lowci=0, highci=0;
        double pay[10000]=0;
   if (capopn == 0)                               //pay cap put before averaging over sublot
        {
        for (i=0;i<=npoint;i++)
                {
                        p=lplot + increment * i;
                        pwl0=areanorm((ulimit-p)/pstd,precision)-areanorm((llimit-
p)/pstd,precision);
                        target=55+50*pwl0;                          //ideal percent within limits pay
factor
                        if (target > pcap) target=pcap;           //account for pay cap
        //              printf("i=%f target= %f\n",p,target);
                        double meanrisk=0;
                        double risk[10000]=0;
                        double cumrisk=0;
                        double var=0;
                        for (j=1;j<=num;j++)
```

```
					{
						left =((ulimit-p)*pow(nsamp,0.5)-
norm[j]*std)/(std*pow(chi[j],0.5));
						right=((llimit-p)*pow(nsamp,0.5)-
norm[j]*std)/(std*pow(chi[j],0.5));
						pay[j]=55 + 50 * (areanorm(left,precision)-
areanorm(right,precision));
						if (pay[j] > pcap) pay[j]=pcap;			//pay cap to be
applied at for each step here
						risk[j] = pay[j] ;
						cumrisk = cumrisk + risk[j];
					}
		sort(risk, num);
				int q1=(num*cilow);
				int q3=num*ciup;
				meanrisk = cumrisk/num;

				for (j=1;j<=num;j++)			//find out 90% confidence intervals
					{
						var = var + pow((risk[j]-meanrisk),2);
					}

				halfwidth= cifact * pow((var / (num-1)),0.5);
				lowci  = meanrisk - halfwidth - target;
				highci = meanrisk + halfwidth - target;
		printf("%i%%  complete\n",(100*i/npoint));

				fprintf(outfile,"%f ,  %f , %f , %f\n", p, meanrisk - target, risk[q1]-target,
risk[q3]-target);
			}
		  }   //capopn = 0 ends here


		 if (capopn == 1)				//pay cap put after averaging over sublot
		{
		for (i=0;i<=npoint;i++)
			{
				p=lplot + increment * i;
				pwl0=areanorm((ulimit-p)/pstd,precision)-areanorm((llimit-
p)/pstd,precision);
				target=55+50*pwl0;			//ideal percent within limits pay
factor
				if (target > pcap) target=pcap;		//account for pay cap
		//		printf("i=%f target= %f\n",p,target);
				double meanrisk=0;
				double risk[10000]=0;
```

```
			double cumrisk=0;
			double var=0;
			int pos=1;
			int q1,q3;
			int reps=num/nsamp;


		for (j=1;j<=num;j++)
				{
					left =((ulimit-p)*pow(nsamp,0.5)-
norm[j]*std)/(std*pow(chi[j],0.5));
					right=((llimit-p)*pow(nsamp,0.5)-
norm[j]*std)/(std*pow(chi[j],0.5));
					pay[j]=55 + 50 * (areanorm(left,precision)-
areanorm(right,precision));
					risk[j] = pay[j] ;

				}

		int s=1;
		float temp=0.0;
		double avgpay[10000]=0;
		double subpay=0;
		for (j=1;j<=num;j++)
			{
				temp=0.0;
				for (k=1;k<=nsamp;k++)
					{
						temp=temp+risk[j];
						if (k < nsamp) j=j+1;
					}
				subpay=temp/nsamp;
				if (subpay > pcap) subpay=pcap;
				cumrisk = cumrisk + subpay;
				avgpay[s]=subpay;
				s=s+1;
			}


			meanrisk = cumrisk/reps;



			sort(avgpay, reps);				//sorting to get upper 95th percentile
and lower 5th percentile
			q1=(reps*cilow);
```

```
                    q3=reps*ciup;
                    int mid=reps*0.5;
        printf("%i%%  complete\n",(100*i/npoint));

                    fprintf(outfile,"%f ,  %f , %f , %f\n", p, meanrisk - target , avgpay[q1]-
target, avgpay[q3]-target);
                }
        }   //capopn = 1 ends here

                fclose(outfile);
}


double areanorm(double x, int precsn)
{
        int i, n;
        if (x < -10) x=-10;
        double incre, start;

        double area=0.0, area1=0.0, area2=0.0;
        float min=-10.0;
        incre=1.0;
        start=min+incre/2;                              //dx=incre for integration of area
        if (x > -3)
                {

                    start=min+incre/2;
                    for (i=0;i <=6;i++)
                        {
                            area1 = area1 + exp(-pow((start+i*incre),2)/2);
                        }
                }

   area1 = (incre*area1);

        start=-3;

   if (precsn==0) incre=0.05;
        if (precsn==1) incre=0.04;                                      //dx=incre for
integration of area
        if (precsn==2) incre=0.03;
        if (precsn==3) incre=0.004;
        n=(x-(start))/(2*incre);
        n=n*2;
        if (x < -3) n=0;
        area=area+exp(-pow(start,2)/2) + 4*exp(-pow((start + incre),2)/2);
```

```
	for (i=2;i<=n-1;i=i+2)
		{
			area=area+2*exp(-pow((start+i*incre),2)/2) + 4*exp(-
pow((start+(i+1)*incre),2)/2);

		}
		//printf("x=%f  #   ",start+n*incre);
		area=(area1+(incre/3)*(area + exp(-
pow((start+n*incre),2)/2)))/(pow((2*3.1415926),0.5));
		return area;
}

double areanorm(double x, int precsn)				//calculates area under
normal curve
										//from -
infinity to x
{
	//area is integrated over two ranges
	//first from z=-10 to -3 and then from z=-3 to x
	//printf("*** %i ***",precsn);
	int i;
	double area=0.0, area1=0.0, area2=0.0;
	float min=-10.0;
	float incre;
	incre=1.0;
	float start=min+incre/2;				//dx=incre for integration of
area
	if (x < -10) x=-10;
	if (x > -3)
		{

			start=min+incre/2;
			for (i=0;i <=6;i++)
				{
					area1 = area1 + exp(-pow((start+i*incre),2)/2);
				}
		}

   area1 = (incre*area1);

	min=-3.0;
	float width=(min-x);		//consider -10 =-infinity
	if (x < -3) width=0;
   if (precsn==0) incre=0.05;
	if (precsn==1) incre=0.02;				//dx=incre for
integration of area
```

```
        if (precsn==2) incre=0.01;
        if (precsn==3) incre=0.005;
  // printf("incre = %i",precsn);
   start=min+incre/2;
        int nsteps=abs(width*(1/incre))-1;
        for (i=0;i<=nsteps;i++)
                {
                        area2 = area2 + exp(-pow((start+i*incre),2)/2);
                }

        area2=(incre*area2);
        area = (area1 + area2)/(pow((2*3.1415926),0.5));
        //if (x <=-7)  area= area;
        //else area=0.0;
        //printf("area1 = %f\narea2 = %f\n",area1,area2);
        //area=1;
        return area;
}


void sort (double unsorted[10000], int numelt)
{
        int i,j;
        double temp;
        for (i=1; i<=numelt; i++)
        {
                for (j=1; j<=(numelt-i); j++)
                {
                        if (unsorted[j] > unsorted[j+1])
                                {
                                        temp=unsorted[j+1];
                                        unsorted[j+1]=unsorted[j];
                                        unsorted[j]=temp;
                                }
                }

        }


}
```

Code for random number generation (used in the simulation)

```
#include <stdio.h>
```

```
#include <time.h>
#include "rngs.h"

#define MODULUS    2147483647
#define MULTIPLIER 48271
#define CHECK      399268537
#define STREAMS    256
#define A256       22925
#define DEFAULT    123456789

static long seed[STREAMS] = {DEFAULT};
static int  stream        = 0;
static int  initialized   = 0;


   double Random(void)
// Random returns a pseudo-random real number uniformly distributed
// between 0.0 and 1.0.

{
  const long Q = MODULUS / MULTIPLIER;
  const long R = MODULUS % MULTIPLIER;
        long t;

  t = MULTIPLIER * (seed[stream] % Q) - R * (seed[stream] / Q);
  if (t > 0)
    seed[stream] = t;
  else
    seed[stream] = t + MODULUS;
  return ((double) seed[stream] / MODULUS);
}


   void PlantSeeds(long x)
// Use this function to set the state of all the random number generator
// streams by "planting" a sequence of states (seeds), one per stream,
// with all states dictated by the state of the default stream.
// The sequence of planted states is separated one from the next by
// 8,367,782 calls to Random().

{
  const long Q = MODULUS / A256;
  const long R = MODULUS % A256;
        int  j;
        int  s;
```

```
  initialized = 1;
  s = stream;                        // remember the current stream
  SelectStream(0);                   // change to stream 0
  PutSeed(x);                        // set seed[0]
  stream = s;                        // reset the current stream
  for (j = 1; j < STREAMS; j++) {
    x = A256 * (seed[j - 1] % Q) - R * (seed[j - 1] / Q);
    if (x > 0)
      seed[j] = x;
    else
      seed[j] = x + MODULUS;
   }
}


   void PutSeed(long x)
// Use this function to set the state of the current random number
// generator stream according to the following conventions:
//    if x > 0 then x is the state (unless too large)
//    if x < 0 then the state is obtained from the system clock
//    if x = 0 then the state is to be supplied interactively

{
  char ok = 0;

  if (x > 0)
    x = x % MODULUS;                       /* correct if x is too large  */
  if (x < 0)
    x = ((unsigned long) time((time_t *) NULL)) % MODULUS;
  if (x == 0)
    while (!ok) {
      printf("\nEnter a positive integer seed (9 digits or less) >> ");
      scanf("%ld", &x);
      ok = (0 < x) && (x < MODULUS);
      if (!ok)
        printf("\nInput out of range ... try again\n");
    }
  seed[stream] = x;
}


   void GetSeed(long *x)
// Use this function to get the state of the current random number
// generator stream.

{
```

```
  *x = seed[stream];
}


   void SelectStream(int index)
// Use this function to set the current random number generator
// stream -- that stream from which the next random number will come.

{
  stream = ((unsigned int) index) % STREAMS;
  if ((initialized == 0) && (stream != 0))   /* protect against        */
    PlantSeeds(DEFAULT);                     /* un-initialized streams */
}
```

APPENDIX B3

Source code for BiasSim (Matlab code)

```
%This is the main simulation engine for BiasSim software.
%It is used to simulate the risk due to bias in the
%measurements of a any quality characteristic in highway construction.
%It uses input file "input.txt" generated by AMSimBias.xls.
%All ouputs are put in the PF.xls file generated by this software.

fid = fopen('input.txt', 'rt');   %Open Input file made by Excel (AMSimBias.xls)

[A count] = fscanf(fid, '%g %g',[2,inf]);       %Get all input data into A
A = A';
fclose(fid);

%open ouput file and put in the parameter values used
fid = fopen('PF.csv', 'w');            %This is to reset the file i.e. delete all previous entries

%assign all input values to appropriate variables
N = A(4,1);                            %Number of samples in each job
Col = 1;
NRuns = A(13,1);                           %Number of jobs with similar statistics
NPoints = A(6,1);

ProdSigma = A(3,1);                    %Production variability
MeasureSigmaCont = A(1,1);                 %Measurement variability for contractor
MeasureSigmaAgency = A(2,1) ;              %Measurement variability for agency
%MeasureSigmaThparty = 0.5;                %Measurement variability for third party

SpecLimit1   = A(11,1) ;                        %spec limit for qc/qa comparision for PF (N=1
comparision)
SpecLimit3   = A(12,1) ;                        %spec limit for qc/qa comparision for PF (N=3
comparision)
%BiasThparty = 0.1*ones(N, Col);           %Bias in third party's density data from actual density
UpperSpec = A(10,1);
LowerSpec = A(9,1);

NBias = A(15,1);
bias = A(16:35,1:2);

CI = A(14,1);
fclose(fid);                           %close PF.csv; parameter values written so far

fid =fopen('PF.csv', 'a');
```

```
fprintf(fid, 'Input Parameter Values used in the simulation:\n');
fprintf(fid, 'Production Variability = %8.4f\nContractor Measurement Variability = %8.4f\n',
ProdSigma, MeasureSigmaCont);
fprintf(fid, 'Agency Measurement Variability = %8.4f\nN = 1 Spec Limit = %8.2f\nN = 3 Spec
Limit = %8.2f\n', MeasureSigmaAgency, SpecLimit1, SpecLimit3');
fprintf(fid, 'Upper Spec = %8.2f\nLower Spec = %8.2f\n', UpperSpec, LowerSpec);

for z = 1:NBias                     %loop for batch processing

Mu = A(7,1);    %to 100                 %mean density, later to be put in For loop
Width = A(8,1);                        %range on the x-axis
Width = Width - Mu;

Bias1 = bias(z,1);                  %choose bias from the batch
Bias2 = bias(z,2);

BiasCont   = Bias1*ones(N, Col);           %Bias in contractor's density data from actual density
BiasAgency = Bias2*ones(N, Col);              %Bias in Agency's density data from actual density

fprintf(fid, 'Contractor Bias= %8.2f,  ,  ,Agency Bias= %8.2f\n', Bias1, Bias2);
fprintf(fid, ' X-Value,  Mean, LowCI, HighCI\n');
Mu = Mu - (Width/(NPoints-1));

%initializing the variables
MeanRisk = zeros(NPoints,1);
LowCI    = zeros(NPoints,1);
HighCI   = zeros(NPoints,1);

for p = 1:NPoints                   %# points for sweep across the range of analysis
   Mu = Mu + (Width/(NPoints-1));

   PFUb = zeros(NRuns,1);                    %for storing pay factors during each run
   PFB  = zeros(NRuns,1);

   %initializing the variables
   NormalRandom = zeros(N, Col);              %simulating density with prod variability
   MeasureRandomCont     = zeros(N, Col);
   MeasureRandomAgency   = zeros(N, Col);
   DensityUnbiasedCont   = zeros(N, Col);
   DensityUnbiasedAgency = zeros(N, Col);
   DensityCont           = zeros(N, Col);
   DensityAgency         = zeros(N, Col);
   DensityUnbiasedPF     = zeros(N, Col);
   DensityPF             = zeros(N, Col);
```

```
   for j = 1:NRuns

      %Generating simulated density data
      %this is without data and without measurement variability
      NormalRandom = normrnd(Mu, ProdSigma, N, Col);                  %two sets for introducing
two bias values

      %introducing measurement variability
      MeasureRandomCont     = normrnd(0, MeasureSigmaCont, N, Col);
      MeasureRandomAgency   = normrnd(0, MeasureSigmaAgency, N, Col);

      %Unbiased emasurements
      DensityUnbiasedCont    = NormalRandom + MeasureRandomCont;
      DensityUnbiasedAgency  = NormalRandom + MeasureRandomAgency;

      %Biased measurements
      DensityCont    =  DensityUnbiasedCont + BiasCont;
      DensityAgency  =  DensityUnbiasedAgency + BiasAgency;

      %Apply specs to the biased density data
      %Applying to first biased data
      OneComp1   = 0;             %to count N=1 pass comparisions
      ThreeComp1 = 0;             %to count N=3 pass comparisions
      ThreeFailComp1 = 0;         %to count N=3 fail comparisions


      for i = 1:5:N-4
         %specs comparision for the first biased data
         if abs(DensityCont(i) - DensityAgency(i)) <= SpecLimit1
            DensityPF(i:i+4) = DensityCont(i:i+4);
            OneComp1 = OneComp1 + 1;
         elseif abs(mean(DensityCont(i+1:i+3)) - mean(DensityAgency(i+1:i+3))) <= SpecLimit3
            DensityPF(i:i+4) = DensityCont(i:i+4);
            ThreeComp1 = ThreeComp1 + 1;
         else
            DensityPF(i:i+4) = DensityAgency(i:i+4);
            ThreeFailComp1 = ThreeFailComp1 + 1;
         end


      end    %end of loop for spec comaprisions

      for i = 1:5:N-4
         %specs comparision for the first unbiased data
         if abs(DensityUnbiasedCont(i) - DensityUnbiasedAgency(i)) <= SpecLimit1
            DensityUnbiasedPF(i:i+4) = DensityUnbiasedCont(i:i+4);
```

```
        elseif                      abs(mean(DensityUnbiasedCont(i+1:i+3))                    -
mean(DensityUnbiasedAgency(i+1:i+3))) <= SpecLimit3
            DensityUnbiasedPF(i:i+4) = DensityUnbiasedCont(i:i+4);

        else
            DensityUnbiasedPF(i:i+4) = DensityUnbiasedAgency(i:i+4);

        end


    end    %end of loop for spec comaprisions

    %determine pay factor

    %determine percent within limits for unbiased data
    AvgUb = mean(DensityUnbiasedPF);
    StdUb = std(DensityUnbiasedPF);
    PWLUb = 100*(normcdf(UpperSpec, AvgUb, StdUb) - normcdf(LowerSpec, AvgUb,
StdUb));

    %determine PF
    PFUb(j) = 55 + 0.5*PWLUb;


    %determine percent within limits for biased data
    AvgB = mean(DensityPF);
    StdB = std(DensityPF);
    PWLB = 100*(normcdf(UpperSpec, AvgB, StdB) - normcdf(LowerSpec, AvgB, StdB));

    %determine PF
    PFB(j) = 55 + 0.5*PWLB;

    %fprintf(fid, '%8.1f %8.1f %8.1f\n', Mu, PF1, PF2);


  end             %end of 1 to NRuns loop; for one mean point in the sweep

%Calculate mean PF and CI for PF at mean value
Risk = PFB - PFUb;          %positive risk mean contractor got more pay than actual
MeanRisk(p) = mean(Risk);

%determine confidence interval (5th percentile and 95th percentile)
SRisk = sort(Risk);
LowCI(p) = SRisk(round(0.5*(1-CI)*NRuns));
HighCI(p) = SRisk(round((CI*NRuns));        %90% CI being formed
```

```
Run = z
Progress = 100*p/NPoints

%print the results into a file

fprintf(fid, '%8.1f, %8.1f, %8.1f, %8.1f\n', Mu, MeanRisk(p), LowCI(p), HighCI(p));
end            %end of loop for the full sweep across range

fprintf(fid, '\n');

end            %end of loop for batch processing

fclose(fid);
```

APPENDIX B4

Source code for SRA (Matlab code)

sra.m

```
function retval = sra()
clear;

fid2 = fopen('input.txt', 'rt');              %Open Input file made by Excel
(SRA.xls)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

%will need to be changed for AC and Density analysis
% for density pl. enter qcc = 1
%     voids   pl. enter qcc = 2
%     AC      pl. enter qcc = 3

qcc = 2;

switch qcc
case 1
    qc = 'Density (% Gmm)';
    AllowMSigma = 0.50;
case 2
    qc = 'Air Voids (%)';
    AllowMSigma = 0.23;
case 3
    qc = 'AC (%)';
    AllowMSigma = 0.05;
end


%open output file
fid = fopen('ExpDes2.csv', 'w');                  %This is to reset the file
i.e. delete all previous entries
fclose(fid);                                 %close the file and delete the
parameter values written so far

fid =fopen('ExpDes2.csv', 'a');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
tic
[A Count] = fscanf(fid2, '%g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g
%g %g %g %g %g %g %g %g %g %g %g %g',[30,inf]); %Get all input data into A
A = A';
fclose(fid2);

%chekc the number of cases to be run from the input file
Count = 0;
for i=1:30
```

```
    if A(1,i)==1
        Count = Count + 1;
        Cases(Count) = i;
    end
end

% setting up structure of real time plotting window
% no. of cases = Count

mnpf = 4;                % no. of plots per figure
% number of figure windows
nf = floor(Count/mnpf) + 1;
if Count <= 3
    ncf = 1;
    nrf = Count;
else
    ncf = 2;
    nrf = 2;
end

%Start batch processing i.e. run each case

%initialise summary result variables
RiskFactorBNeg = zeros(30,1);
RiskFactorBPos = zeros(30,1);
RiskFactorB    = zeros(30,1);
SweetB         = zeros(30,1);
PeakBiased     = zeros(30,1);
TroughBiased   = zeros(30,1);

%unbiased portion removed
RiskFactorUbNeg = zeros(30,1);
RiskFactorUbPos = zeros(30,1);
RiskFactorUb    = zeros(30,1);
SweetUb         = zeros(30,1);
PeakUnBiased     = zeros(30,1);
TroughUnBiased   = zeros(30,1);

%############################################################################
#########################
% loop for each case run starts
for w = 1:Count

%+++++++++++++++++++++++++++++ Change Comment for output file
++++++++++++++++++++++++++++++++++++++++=
fprintf(fid, 'Allowable measurement variability = 0.23. Experimental Design
Runs\n');
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++=

%assign all input values to appropriate variables
N = A(8,Cases(w));                                  %Number of samples in
each job
Col = 1;
```

```
NRuns = A(2,Cases(w));                              %Number of jobs with
similar statistics
NPoints = A(16,Cases(w));

ProdSigma           = A(7,Cases(w));                 %Production variability
MeasureSigmaCont    = A(4,Cases(w));                 %Measurement variability
for contractor
MeasureSigmaAgency  = A(5,Cases(w)) ;                %Measurement variability
for agency
MeasureSigmaThparty = A(6,Cases(w)) ;                %Measurement variability
for third party

SpecLimit1 = A(14,Cases(w)) ;                        %spec limit for qc/qa
comparision for PF (N=1 comparision)
SpecLimit3 = A(15,Cases(w)) ;                        %spec limit for qc/qa
comparision for PF (N=3 comparision)

            %Bias in third party's density data from actual density
UpperSpec  = A(13,Cases(w));
LowerSpec  = A(12,Cases(w));

if qcc == 3
    RejQU = UpperSpec + 0.17;
    RejQL = LowerSpec - 0.17;
elseif qcc == 1
    RejQU = 98.5;
    RejQL = 87;
elseif qcc == 2
    RejQU = 1.5;
    RejQL = 6.5;
else
    msgbox('Please Enter Valid Quality Characteristic Code!');
    break
end

MidSpec    = (UpperSpec + LowerSpec)/2;
NBias      = Count;

AcceptableRisk = 5;

%Desired Confidence Interval
CI         = A(3,Cases(w));


fprintf(fid, 'Input Parameter Values used in the simulation:\n');
fprintf(fid, 'Production Variability = %8.4f\nContractor Measurement
Variability = %8.4f\n', ProdSigma, MeasureSigmaCont);
fprintf(fid, 'Agency Measurement Variability = %8.4f\nThird Party Measurement
Variability = %8.4f\nN = 1 Spec Limit = %8.2f\nN = 3 Spec Limit = %8.2f\n',
MeasureSigmaAgency, MeasureSigmaThparty, SpecLimit1, SpecLimit3');
fprintf(fid, 'Upper Spec = %8.2f\nLower Spec = %8.2f\n', UpperSpec,
LowerSpec);
fprintf(fid, 'Sample Size = %8.0f\n', N);
```

```
LRange = A(18,Cases(w));                            %mean density, later to be
put in For loop
URange = A(19,Cases(w));                                %range on the x-axis
Width  = URange - LRange;
Mu     = LRange;

Bias1  = A(9,  Cases(w));                            %choose bias from the
batch
Bias2  = A(10, Cases(w));
Bias3  = A(11, Cases(w));

Nmore = 3*N;
BiasCont    = Bias1;   %*ones(Nmore, Col);            %Bias in contractor's
density data from actual density
BiasAgency  = Bias2;   %*ones(Nmore, Col);            %Bias in Agency's
density data from actual density
BiasThparty = Bias3;   %*ones(Nmore, Col);            %Bias in Third Party's
density data from actual density

fprintf(fid, 'Contractor Bias= %8.2f\nAgency Bias= %8.2f\nThird Party Bias=
%8.2f\n', Bias1, Bias2, Bias3);
fprintf(fid, 'Risk with Original Data, , , , , , , , , , , , , , , , , , Risk
with Bias Removed\n');

fprintf(fid, 'Mu, MeanRisk, LowCI(p), HighCI(p), PFBm, LowCIPF, HighCIPF,
PFBLm,');
fprintf(fid, 'PercentOneComp, PercentThreeComp, PercentThreeFailComp,');
fprintf(fid, 'MeanRiskOld(p), LowCIOld(p), HighCIOld(p),PFBOldm, LowCIPFOld,
HighCIPFOld, PFBLOldm,');
fprintf(fid, 'MeanRiskUb(p), LowCIUb(p), HighCIUb(p), PFUbm, LowCIPFUb,
HighCIPFUb,');
fprintf(fid, 'PercentOneCompUb, PercentThreeCompUb,
PercentThreeFailCompUb,');
fprintf(fid, 'MeanRiskUbOld, LowCIUbOld, HighCIUbOld,PFUbOldm, LowCIPFUbOld,
HighCIPFUbOld\n');


Incre = Width/(NPoints-1);


PointDensity = A(17, Cases(w));
Mu = Mu - Incre;


%SweetSpot  = zeros(NPoints2,2);
SweetStart1= 0;                             %starting and ending points of
sweet spot inside the spec limits
SweetEnd1  = 0;
SweetStart2= 0;
SweetEnd2 = 0;

p          = 0;

%initializing values for real-time plotting
```

```
    x(1)  = 0;
    y1(1) = 0;
    y2(1) = 0;
    y3(1) = 0;

    x1(1)  = 0;
    y11(1) = 0;
    y12(1) = 0;
    y13(1) = 0;
    y14(1) = 0;

    x11(1) = 0;
    y21(1) = 0;
    y22(1) = 0;
    y23(1) = 0;
 %clear rp;
%for p = 1:NPoints                            % points for sweep across the
range of analysis
while Mu < URange
    p = p + 1;

    %determining increment for mean quality charcteristic for plotting
    if (p > 3)
        SlopeHighCI = abs((HighCI(p-1)-HighCI(p-2))/(SweetSpot(p-1,1)-
SweetSpot(p-2,1)));
        SlopeLowCI  = abs((LowCI(p-1)-LowCI(p-2))/(SweetSpot(p-1,1)-
SweetSpot(p-2,1)));

        %average the last two slopes to get slope
        %this is done to dampen the effect of oscilations (in case of noisy
output)
        if (p > 4)
            SlopeHighCI2= abs((HighCI(p-2)-HighCI(p-3))/(SweetSpot(p-2,1)-
SweetSpot(p-3,1)));
            SlopeLowCI2 = abs((LowCI(p-2)-LowCI(p-3))/(SweetSpot(p-2,1)-
SweetSpot(p-3,1)));
            SlopeHighCI = (SlopeHighCI + SlopeHighCI2)/2;
            SlopeLowCI  = (SlopeLowCI  + SlopeLowCI2 )/2;
        end

        %the slope is futher dampened because the scale on y axis is not same
as that one the x-axis
        Slope       = ((URange - LRange)/40)*(max(SlopeHighCI,
SlopeLowCI))^2/80;

        Increment   = Incre/(PointDensity*max(Slope,0.001));
        Increment   = max(Incre/PointDensity, Increment);
        Increment   = min(Incre, Increment);

    else
        Increment = Incre;
    end
```

```
    Mu = Mu + Increment;
    if (Mu  > URange-0.055)
        Mu = URange;
    end

    Mu
%  %unbiased portion removed
    PFUb = zeros(NRuns,1);                     %for storing pay factors during
each run
    PFB  = zeros(NRuns,1);

    %initializing the variables
    TotNum = Nmore*NRuns;

   % Density                = zeros(N, Col)
    DensityCont            = zeros(N, Col);
    DensityAgency           = zeros(N, Col);
    DensityThparty          = zeros(N, Col);

    %unbiased portion removed
    DensityUnbiasedPF       = zeros(N, Col);
    DensityPF               = zeros(N, Col);

        OneComp1   = 0;                     %to count N=1 pass comparisions
        ContAcceptsDept = 0;                %to count N = 1 fail and
contractor accepting dept results
        ThreeComp1 = 0;                     %to count N=3 pass comparisions
        ThreeFailComp1 = 0;                 %to count N=3 fail comparisions

  %unbiased portion removed
        OneCompUb = 0;                         %to count N=1 pass comparisions
for unbiased data
        CAcceptsDUb = 0;                     %to count N = 1 fail and
contractor accepting dept results for unbiased data
        ThreeCompUb = 0;                         %to count N=3 pass
comparisions for unbiased data
        ThreeFailUb = 0;                         %to count N=3 fail
comparisions for unbiased data
    TotNum          = Nmore*NRuns;
    ReqNum          = N    *NRuns;
    BufNum          = TotNum - ReqNum;

    DensityC        = normrnd(Mu, ProdSigma, TotNum, Col);
    DensityContC    = DensityC + normrnd(0, MeasureSigmaCont, TotNum, Col) +
BiasCont;
    DensityAgencyC  = DensityC + normrnd(0, MeasureSigmaAgency, TotNum, Col)
+ BiasAgency;
    DensityThpartyC  = DensityC + normrnd(0, MeasureSigmaThparty, TotNum,
Col) + BiasThparty;

    cntC = 0;        %counts no. of rejectable sublots for contractor data
    cntA = 0;        %counts no. of rejectable sublots for Agency data
    cntT = 0;        %counts no. of rejectable sublots for Third Party data
%determine how many sublots will need to be rejected
        for rq = 1:ReqNum
```

```
                %replace contractor data with acceptable data
                while DensityContC(rq) < RejQL
                    cntC = cntC+1;

                    %warn if too many rejcted sublots are coming up
                    if cntC > BufNum          %0.5*N / 5
                        string1 = sprintf('Too many rejected sublots!!!');
                        string2 = sprintf('One possiblity is that analysis is
being done outside realisitc range of parameter values!');
                        %disp(string1);
                        %disp(string2);
                    break;
                    end

                    DensityContC(rq) = DensityContC(ReqNum+cntC);
%replaces the rejectable value
                end


                while DensityContC(rq) > RejQU
                    cntC = cntC+1;
                    %warn if too many rejcted sublots are coming up
                    if cntC > BufNum          %0.5*N / 5
                        string1 = sprintf('Too many rejected sublots!!!');
                        string2 = sprintf('One possiblity is that analysis is
being done outside realisitc range of parameter values!');
                        %disp(string1);
                        %disp(string2);
                    break;
                    end
                    DensityContC(rq) = DensityContC(ReqNum+cntC);
%replaces the rejectable value
                end

                %replace Agency data with acceptable data
                while DensityAgencyC(rq) < RejQL
                    cntA = cntA+1;
                    %warn if too many rejcted sublots are coming up
                    if cntA > BufNum          %0.5*N / 5
                        string1 = sprintf('Too many rejected sublots!!!');
                        string2 = sprintf('One possiblity is that analysis is
being done outside realisitc range of parameter values!');
                        %disp(string1);
                        %disp(string2);
                    break
                    end
                    DensityAgencyC(rq) = DensityAgencyC(ReqNum+cntA);
%replaces the rejectable value

                end

                while DensityAgencyC(rq) > RejQU
                    cntA = cntA+1;
                    %warn if too many rejcted sublots are coming up
                    if cntA > BufNum          %0.5*N / 5
```

```
                        string1 = sprintf('Too many rejected sublots!!!');
                        string2 = sprintf('One possiblity is that analysis is
being done outside realisitc range of parameter values!');
                        %disp(string1);
                        %disp(string2);
                    break;
                    end
                    DensityAgencyC(rq) = DensityAgencyC(ReqNum+cntA);
%replaces the rejectable value
                end

                %replace Third Party data with acceptable data
                while DensityThpartyC(rq) < RejQL
                    cntT = cntT+1;
                    %warn if too many rejcted sublots are coming up
                    if cntT > BufNum          %0.5*N / 5
                        string1 = sprintf('Too many rejected sublots!!!');
                        string2 = sprintf('One possiblity is that analysis is
being done outside realisitc range of parameter values!');
                        %disp(string1);
                        %disp(string2);
                    break;
                    end
                    DensityThpartyC(rq) = DensityThpartyC(ReqNum+cntT);
%replaces the rejectable value

                end

                while DensityThpartyC(rq) > RejQU
                    cntT = cntT+1;
                    %warn if too many rejcted sublots are coming up
                    if cntT > BufNum          %0.5*N / 5
                        string1 = sprintf('Too many rejected sublots!!!');
                        string2 = sprintf('One possiblity is that analysis is
being done outside realisitc range of parameter values!');
                        %disp(string1);
                        %disp(string2);
                    break;
                    end
                    DensityThpartyC(rq) = DensityThpartyC(ReqNum+cntT);
%replaces the rejectable value
                end
            end         %end of loop for replacing rejectable quality data
with acceptable quality values


    for j = 1:NRuns




        %Biased measurements
        iStart = (j-1)*N +1;
        iEnd   = j*N;
        DensityCont    =  DensityContC(iStart:iEnd);
```

```
        DensityAgency  =  DensityAgencyC(iStart:iEnd);
        DensityThparty =  DensityThpartyC(iStart:iEnd);

        flag = 0;



        %Apply specs to the biased density data
        for i = 1:5:N

            rem = N-i;
            if rem <=4
                adv = rem;
            else
                adv = 4;
            end

            %specs comparision for the biased data
            if abs(DensityCont(i) - DensityAgency(i)) <= SpecLimit1
                DensityPF(i:i+adv) = DensityCont(i:i+adv);
                OneComp1 = OneComp1 + 1;
            elseif abs(DensityAgency(i) - MidSpec) < abs(DensityCont(i) -
MidSpec)
                DensityPF(i:i+adv) = DensityAgency(i:i+adv);
                ContAcceptsDept = ContAcceptsDept + 1;
            elseif adv >= 2 & abs(mean(DensityCont(i:i+2)) -
mean(DensityThparty(i:i+2))) <= SpecLimit3
                DensityPF(i:i+adv) = DensityCont(i:i+adv);
                ThreeComp1 = ThreeComp1 + 1;
            else
                DensityPF(i:i+adv) = DensityThparty(i:i+adv);
                ThreeFailComp1 = ThreeFailComp1 + 1;
            end



        end    %end of loop for spec comaprisions

        %simulating the situation when an attempt is first made to remove the
bias and then apply
        %comparision limits on them to calculate pay factors


        Diff = (DensityCont -DensityAgency);

     %unbiased portion removed
         Bias = mean(Diff);
         UnbiasedDiff = Diff - Bias;

        ThDiff = (DensityCont -DensityThparty);
        ThBias = mean(ThDiff);
        UnbiasedThDiff = ThDiff - ThBias;
        for i = 1:5:N
            %specs comparision for the data from which bias has been removed
```

```
            if abs(UnbiasedDiff(i)) <= SpecLimit1
                DensityUnbiasedPF(i:i+adv) = DensityCont(i:i+adv);
                OneCompUb = OneCompUb + 1;
            elseif abs(DensityAgency(i) - MidSpec) < abs(DensityCont(i) -
MidSpec)
                DensityUnbiasedPF(i:i+adv) = DensityAgency(i:i+adv);
                CAcceptsDUb = CAcceptsDUb + 1;
            elseif adv >= 2 & abs(mean(UnbiasedThDiff(i:i+2))) <= SpecLimit3
                DensityUnbiasedPF(i:i+adv) = DensityCont(i:i+adv);
                ThreeCompUb = ThreeCompUb + 1;
            else
                DensityUnbiasedPF(i:i+adv) = DensityThparty(i:i+adv);
                ThreeFailUb = ThreeFailUb + 1;
            end


        end    %end of loop for spec comaprisions




        %determine pay factor

        %unbiased portion removed
        %determine percent within limits for unbiased data
        AvgUb = mean(DensityUnbiasedPF);
        StdUb = std(DensityUnbiasedPF);
        PWLUb = 100*(normcdfam(UpperSpec, AvgUb, StdUb) -
normcdfam(LowerSpec, AvgUb, StdUb));

        %determine PF
        PFUb(j) = 53 + 0.5*PWLUb;
        PFUbOld(j) = 55 + 0.5*PWLUb;
        if PFUbOld(j) > 103
            PFUbOld(j) = 103;
        end

        %determine percent within limits for biased data
        AvgB = mean(DensityPF);
        StdB = std(DensityPF);




        PWLB(j) = 100*(normcdfam(UpperSpec, AvgB, StdB) -
normcdfam(LowerSpec, AvgB, StdB));

        %determine PF
        PFB(j) = 53 + 0.5*PWLB(j);
        PFBOld(j) = 55 + 0.5*PWLB(j);
        if PFBOld(j) > 103
            PFBOld(j) = 103;
        end
        %fprintf(fid, '%8.1f %8.1f %8.1f\n', Mu, PF1, PF2);
```

```
    end                                      %end of 1 to NRuns loop; for
one mean point in the sweep

 %determine how many times N= 1 passed and how many times failed
 TotComp = OneComp1 + ContAcceptsDept + ThreeComp1 + ThreeFailComp1;
 PercentOneComp = 100*OneComp1/TotComp;
 PercentContAcceptsDept = 100*ContAcceptsDept/TotComp;
 PercentThreeComp = 100*ThreeComp1/TotComp;
 PercentThreeFailComp = 100*ThreeFailComp1/TotComp;

 %determine how many times N= 1 passed and how many times failed when rel.
bias removed
 PercentOneCompUb = 100*OneCompUb/TotComp;
 PercentCAcceptsDUb = 100*CAcceptsDUb/TotComp;
 PercentThreeCompUb = 100*ThreeCompUb/TotComp;
 PercentThreeFailCompUb = 100*ThreeFailUb/TotComp;

 %determine means and CI of PF
        PFBm         = mean(PFB);
        SPFB         = sort(PFB);
        LowCIPF      = SPFB(round(((1-CI)/2)*NRuns));
        HighCIPF     = SPFB(round(((1+CI)/2)*NRuns));     % CI being formed

%Old PF case
%determine means and CI of PF
        PFBOldm         = mean(PFBOld);
        SPFBOld         = sort(PFBOld);
        LowCIPFOld      = SPFBOld(round(((1-CI)/2)*NRuns));
        HighCIPFOld     = SPFBOld(round(((1+CI)/2)*NRuns));     % CI being
formed


 %Ideal pay factor (Base line)
 %Base line is calculated for data mean + production variability
 NBL = 40000;
NBLmore = 3*NBL;
DensityBL = zeros(NBLmore,1);
DensityBL = normrnd(Mu, ProdSigma, NBLmore, 1);
% generate random numbers corresponding to allowable measurement variability
allowmvar = normrnd(0, AllowMSigma, NBLmore, 1);
DensityBL = DensityBL + allowmvar;
%replace rejectable quality data with acceptable data
cnt = 0;
for rq = 1:NBL
    while DensityBL(rq) < RejQL | DensityBL(rq) > RejQU
        cnt = cnt + 1;
        if cnt > (NBLmore-NBL)
            break;
        else
            DensityBL(rq) = DensityBL(NBL + cnt);
        end
    end
end
```

```
%determine mean and std of baseline PF
DensityBLm = mean(DensityBL(1:NBL));
StdDensityBL = std(DensityBL(1:NBL));
PWLBL =  100*(normcdfam(UpperSpec, DensityBLm, StdDensityBL) -
normcdfam(LowerSpec, DensityBLm, StdDensityBL));
PFBLm = 53 + 0.5*PWLBL;
PFBLOldm = 55 + 0.5*PWLBL;

%apply cap (here pay for each parameter is capped. This is different from
earlier specs.)
 if PFBLOldm > 103
      PFBLOldm = 103;
  end

%Calculate mean PF and CI for PF at mean value
RiskBiased  = PFB - PFBLm;                             %positive risk mean
contractor got more pay than actual
MeanRisk(p) = mean(RiskBiased);

%determine confidence interval ((100-alpha)th percentile and alph th
percentile)
SRisk     = sort(RiskBiased);
LowCI(p)  = SRisk(round(((1-CI)/2)*NRuns));
HighCI(p) = SRisk(round(((1+CI)/2)*NRuns));     % CI being formed

    %Old PF case
    %Calculate mean PF and CI for PF at mean value
    RiskBiasedOld  = PFBOld - PFBLOldm;                        %positive risk
mean contractor got more pay than actual
    MeanRiskOld(p) = mean(RiskBiasedOld);

    %determine confidence interval ((100-alpha)th percentile and alph th
percentile)
    SRiskOld     = sort(RiskBiasedOld);
    LowCIOld(p)  = SRiskOld(round(((1-CI)/2)*NRuns));
    HighCIOld(p) = SRiskOld(round(((1+CI)/2)*NRuns));     % CI being formed

SweetSpot(p,1) = Mu;

% Evlpts2(w,p,1)   = Mu;
% Evlpts2(w,p, 2:NRuns+1) = SRisk;

%unbiased portion removed
%Calculate mean PF and CI for PF at mean value (rel. bias removed)
RiskUnbiased  = PFUb - PFBLm;                          %positive risk means
contractor got more pay than actual
MeanRiskUb(p) = mean(RiskUnbiased);


%determine confidence interval ((100-alpha)th percentile and alph th
percentile) (rel. bias removed)
SRiskUb     = sort(RiskUnbiased);
LowCIUb(p)  = SRiskUb(round(((1-CI)/2)*NRuns));
HighCIUb(p) = SRiskUb(round(((1+CI)/2)*NRuns)); % CI being formed
```

```
% mean PF and LCL and UCL when rel bias removed
PFUbm = mean(PFUb);
SPFUb = sort(PFUb);
LowCIPFUb = SPFUb(round(((1-CI)/2)*NRuns));
HighCIPFUb = SPFUb(round(((1+CI)/2)*NRuns)); % CI being formed on PF when
rel. bias removed

RiskUnbiasedOld  = PFUbOld - PFBLOldm;                    %positive risk
means contractor got more pay than actual
MeanRiskUbOld = mean(RiskUnbiasedOld);

%determine confidence interval ((100-alpha)th percentile and alph th
percentile) (rel. bias removed)
SRiskUbOld     = sort(RiskUnbiasedOld);
LowCIUbOld  = SRiskUbOld(round(((1-CI)/2)*NRuns));
HighCIUbOld = SRiskUbOld(round(((1+CI)/2)*NRuns)); % CI being formed

% mean PF and LCL and UCL with old eq. when rel bias removed
PFUbOldm = mean(PFUbOld);
SPFUbOld = sort(PFUbOld);
LowCIPFUbOld = SPFUbOld(round(((1-CI)/2)*NRuns));
HighCIPFUbOld = SPFUbOld(round(((1+CI)/2)*NRuns)); % CI being formed

%Progress indicator calculation
Run         = w
Progress    = round(100*(Mu-LRange)/(URange-LRange));
Prog = sprintf('%8.0f %% Completed', Progress);
disp(Prog);

%print the results into a file

fprintf(fid, '%8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f,  ', Mu,
MeanRisk(p), LowCI(p), HighCI(p), PFBm, LowCIPF, HighCIPF, PFBLm);
fprintf(fid, '%8.2f, %8.2f, %8.2f,  ', PercentOneComp, PercentThreeComp,
PercentThreeFailComp);
fprintf(fid, '%8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f,  ',
MeanRiskOld(p), LowCIOld(p), HighCIOld(p),PFBOldm, LowCIPFOld, HighCIPFOld,
PFBLOldm);
fprintf(fid, '%8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f, ', MeanRiskUb(p),
LowCIUb(p), HighCIUb(p), PFUbm, LowCIPFUb, HighCIPFUb);
fprintf(fid, '%8.2f, %8.2f, %8.2f,  ', PercentOneCompUb, PercentThreeCompUb,
PercentThreeFailCompUb);
fprintf(fid, '%8.2f, %8.2f, %8.2f, %8.2f, %8.2f, %8.2f\n ', MeanRiskUbOld,
LowCIUbOld, HighCIUbOld,PFUbOldm, LowCIPFUbOld, HighCIPFUbOld);

%store all the results printed in the file in a matrix as well

%results corresponding to New PF Eq.
row(1, 1:8)   = [Mu, MeanRisk(p), LowCI(p), HighCI(p), PFBm, LowCIPF,
HighCIPF, PFBLm];
row(1, 9:11)  = [PercentOneComp, PercentThreeComp, PercentThreeFailComp];

%results corresponding to Old PF Eq.
```

```
row(1, 12:18) = [MeanRiskOld(p), LowCIOld(p), HighCIOld(p),PFBOldm,
LowCIPFOld, HighCIPFOld, PFBLOldm];

%results corresponding to New PF Eq. with rel. bias removed
row(1, 19:24) = [MeanRiskUb(p), LowCIUb(p), HighCIUb(p), PFUbm, LowCIPFUb,
HighCIPFUb];
row(1, 25:27) = [PercentOneCompUb, PercentThreeCompUb,
PercentThreeFailCompUb];

%results corresponding to Old PF Eq. when rel. bias removed
row(1, 28:33) = [MeanRiskUbOld, LowCIUbOld, HighCIUbOld,PFUbOldm,
LowCIPFUbOld, HighCIPFUbOld];

rp(w, p, 1:33) = row;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% uncomment this section for activating real time plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fg = floor((w-1)/mnpf) + 1;
% figure(fg);
% subplotnum = w-(fg-1)*mnpf;
% if subplotnum == 1
%     clf;
% end
%
% subplot(nrf, ncf, subplotnum);
% if p == 1
%     cla;
% end
%
% hold on; grid on;
% xlabel(qc);
% ylabel('Risk (% Bid Amount)');
%
% as = num2str(AllowMSigma, '%6.2f');
% sp = num2str(ProdSigma, '%6.2f');
% sm = num2str(MeasureSigmaCont, '%6.2f');
% ns = num2str(N, '%5.0f');
% sl1= num2str(SpecLimit1, '%6.2f');
% sl3= num2str(SpecLimit3, '%6.2f');
%
% if BiasCont == 0
%     bs = 'Low ';
% else
%     bs = 'High';
% end
%
% ttl = ['Sig-P=', sp, '   Sig-M=', sm, '   N=', ns, '  Bias=', bs];
% title(ttl);
% axis([2, 6, -30, 30]);
%
% h = plot(x, y1, 'k');
% h2 = plot(x, y2, 'b-x');
% h3 = plot(x, y3, 'b-x');
%
```

```
% set(h, 'erasemode', 'none');
% set(h2, 'erasemode', 'none');
% set(h3, 'erasemode', 'none');
%
%
%     drawnow
%     x  = rp(1:p, 1);
%     y1 = rp(1:p, 2);
%     y2 = rp(1:p, 3);
%     y3 = rp(1:p, 4);
%     set(h, 'xdata', x, 'ydata', y1);
%     set(h2, 'xdata', x, 'ydata', y2);
%     set(h3, 'xdata', x, 'ydata', y3);
%
%
% %Draw mean PF, confidence limits and base line pay
% fg1 = fg + 100;
% figure(fg1);
% subplotnum = w-(fg1-101)*mnpf;
% if subplotnum == 1
%     clf;
% end
%
% subplot(nrf, ncf, subplotnum);
% if p ==1
%     cla;
% end
% hold on; grid on;
% xlabel(qc);
% ylabel('Pay Factor (%)');
%
% as = num2str(AllowMSigma, '%6.2f');
% sp = num2str(ProdSigma, '%6.2f');
% sm = num2str(MeasureSigmaCont, '%6.2f');
% ns = num2str(N, '%5.0f');
%
% ttl = ['Sig-P=', sp, '   Sig-M=', sm, '   N=', ns, '  Bias=', bs];
% title(ttl);
% axis([2, 6, 60, 105]);
%
% h11 = plot(x1, y11, 'k', 'linewidth', 2);
% h12 = plot(x1, y12, 'r--');
% h13 = plot(x1, y13, 'r--');
% h14 = plot(x1, y14, 'k-*');
%
% set(h11, 'erasemode', 'none');
% set(h12, 'erasemode', 'none');
% set(h13, 'erasemode', 'none');
% set(h14, 'erasemode', 'none');
%
%
%     drawnow
%     x1  = rp(1:p, 1);
%     y11 = rp(1:p, 5);
%     y12 = rp(1:p, 6);
%     y13 = rp(1:p, 7);
%     y14 = rp(1:p, 8);
```

```
%     set(h11, 'xdata', x1, 'ydata', y11);
%     set(h12, 'xdata', x1, 'ydata', y12);
%     set(h13, 'xdata', x1, 'ydata', y13);
%     set(h14, 'xdata', x1, 'ydata', y14);
%
%     legend('Mean PF', 'Low CL PF', 'Up CL PF', 'Base Line', 0);
%
%   %Draw mean risk, LCL, and UCL when rel Bias is removed
% fg2 = fg + 200;
% figure(fg2);
% subplotnum = w-(fg2-201)*mnpf;
% if subplotnum == 1
%     clf;
% end
%
% subplot(nrf, ncf, subplotnum);
% if p ==1
%     cla;
% end
%
% hold on; grid on;
% xlabel(qc);
% ylabel('Risk (% Bid Amount)');
%
% as = num2str(AllowMSigma, '%6.2f');
% sp = num2str(ProdSigma, '%6.2f');
% sm = num2str(MeasureSigmaCont, '%6.2f');
% ns = num2str(N, '%5.0f');
% sl1= num2str(SpecLimit1, '%6.2f');
% sl3= num2str(SpecLimit3, '%6.2f');
%
% if BiasCont == 0
%     bs = 'Low ';
% else
%     bs = 'High';
% end
%
% ttl = ['Sig-P=', sp, '   Sig-M=', sm, '   N=', ns, '  Bias=', bs];
% title(ttl);
% axis([2, 6, -30, 30]);
%
% h21 = plot(x11, y21, 'k');
% h22 = plot(x11, y22, 'b-x');
% h23 = plot(x11, y23, 'b-x');
%
% set(h21, 'erasemode', 'none');
% set(h22, 'erasemode', 'none');
% set(h23, 'erasemode', 'none');
%
%
%     drawnow
%     x11 = rp(1:p, 1);
%     y21 = rp(1:p, 19);
%     y22 = rp(1:p, 20);
%     y23 = rp(1:p, 21);
%     set(h21, 'xdata', x11, 'ydata', y21);
%     set(h22, 'xdata', x11, 'ydata', y22);
```

```
%     set(h23, 'xdata', x11, 'ydata', y23);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end                         %end of loop for the full sweep across range
clear risk;
risk(:, :) = rp(w,1:p,1:4);
%risk = risk'
nrband(w) = nrb(risk, [LowerSpec UpperSpec]);
rindices(1:5, 1:3, w) = msrisk(risk, [LowerSpec UpperSpec]);
plotnum = num2str(w, '%2f');
%matname = ['ExpDes', w, '.mat'];
%save matname rp;

LenB = p;   %length(HighCI);
LenU = p;   %length(HighCIUb);
%SweetSpot = zeros(max(LenB, LenU),3);




fprintf(fid, '\n');

fprintf(fid, 'Narrow Risk Band Width =, %5.2f\n', nrband(w));
fprintf(fid, ', Q Char , Low CI, High CI\n');
fprintf(fid, 'Mid Point, %8.2f, %8.2f, %8.2f\n', rindices(1,1,w),
rindices(1,2,w), rindices(1,3,w));
fprintf(fid, 'Lower Spec LImit, %8.2f, %8.2f, %8.2f\n', rindices(2,1,w),
rindices(2,2,w), rindices(2,3,w));
fprintf(fid, 'Upper Spec Limit, %8.2f, %8.2f, %8.2f\n\n', rindices(3,1,w),
rindices(3,2,w), rindices(3,3,w));
fprintf(fid, 'Maximum Negative Risk =, %5.2f, for the quality characteristic
value =, %5.2f\n', rindices(4, 2, w), rindices(4, 1, w));
fprintf(fid, 'Maximum Positive Risk =, %5.2f, for the quality characteristic
value =, %5.2f\n\n\n', rindices(5, 2, w), rindices(5, 1, w));

PTime = toc;
fprintf(fid, 'Processing Time = %5.2f\n\n\n', PTime);
end                                             %end of loop for batch
processing

fprintf(fid, '\n\n\n');


fclose(fid);
nrband
rindices
%save 'ExpDes2.mat' rp;
%save 'EvalnPtsExpDes2.mat' Evlpts2;
totaltime= toc

retval = 0
```

riskPmeas.m

```
% this code is used to read a csv output file generated by sra.m for
% experimental design runs. the gaps between successive runs had to be
% removed for easier reading by matlab.
% This code calculates various measures of goodness of the risk plots for
% determining the effect of various parameters in risk.
clear;
resolution = 100;
speclimits = [2.65, 5.35];
load ED6.csv;
out = ED6;

%identify the sets of data for each case run
n = length(out);
count = 0;      %no. of cases in the output
for i = 1:n
    if out(i, 1) == 2
        count = count + 1;
        %starting point of each case
        caseind(count,1) = i;

        %ending point of each run
        if count > 1
            caseind(count-1,2) = i-1;
        end
    end

end

%ending point of entire data set
caseind(count,2) = n;

%setting resolution to be an even number
if (rem(resolution,2) == 1)
    resolution = resolution - 1;
end

for i = 1:count
    %select data for each case
    risk = out(caseind(i, 1):caseind(i, 2), 1:4);
    npoints = caseind(i, 2) - caseind(i, 1) + 1;

    % interpolate to greater resolution
    xx = linspace(risk(1,1), risk(npoints, 1), resolution+1);
    irisk(1:resolution+1, 1) = xx';
    irisk(1:resolution+1, 3) = spline(risk(:, 1), risk(:, 3), xx)';
    irisk(1:resolution+1, 4) = spline(risk(:, 1), risk(:, 4), xx)';

    % narrow risk band
    nrband(i) = nrb(irisk, speclimits);
    % risk at mid point and spec limits and maximum and minmum risk
    mrisk(1:5, 1:3, i) = msrisk(irisk, speclimits);

%     figure(i)
%     clf;
%     hold on;
```

```
%     axis([2 6 -30 30]);
%     grid on;
%     plot(irisk(:, 1), irisk(:,3), irisk(:,1), irisk(:,4));
%     plot(risk(:,1), risk(:,2), 'r');

end


%put the results in another output file in a summary form
fid = fopen('ExpDesIndices6.txt', 'w');
fprintf(fid, 'NRB   MidRisk-LCL MidRisk-UCL LSLRisk-LCL LSLRisk-UCL LSLRisk-
LCL LSLRisk-UCL  LocMN  MaxNegRisk    LocMP  MaxPosRisk\n');

for i = 1:count
    fprintf(fid, '%8.2f    %8.2f      %8.2f    ', nrband(i), mrisk(1, 2, i),
mrisk(1, 3, i));
    fprintf(fid, '%8.2f    %8.2f      %8.2f     %8.2f    ', mrisk(2, 2, i),
mrisk(2, 3, i), mrisk(3, 2, i), mrisk(3, 3, i));
    fprintf(fid, '%8.2f    %8.2f      %8.2f     %8.2f    \n', mrisk(4, 1, i),
mrisk(4, 2, i), mrisk(5, 1, i), mrisk(5, 2, i));

end

fclose(fid);

nrband
mrisk
```

ptrindex4.m

```
% calculates riskindex at a single point of evaluation
% it takes a representative set of risk values and takes their moment
% about the mean and returns that value.
%This file is same as ptrindex.m except that the exponent is
% proportional to the magnitude of the risk value rather than
% relative to its position within the band.

function retval = ptrindex3(risk)

nmmt = 100;
%sort risk values
risk = abs(risk);
risk = sort(risk);

%select nmmt representative values from the sorted list for taking moment
n = length(risk);       %number of evaluations at a point
incre = n/nmmt;

for i = 1:nmmt
    trim(nmmt-i+1) = risk(n-floor((i-1)*incre));
end

range = 50;
```

```
rind = zeros(1,2);
for i = 1:nmmt

    %first pt risk index. weight changes exponentially
    if range == 0
        expo = 1;
    else
        expo(1) = 0.5 + trim(i)/range;
    end
    rind(1) = rind(1) + trim(i)^expo;

    %second pt risk index. weight changes linearly
    factor = 1   + trim(i)/range;
    rind(2) = rind(2) + trim(i)*factor;



end

retval = (1/nmmt)*rind;
```

---

nrb.m

```
%this code calculates the narrow risk band in a risk plot

function bandwidth = nrb(irisk, specwidth)
resolution = 100;
clear band;
band(:, 2) = irisk(:, 4) - irisk(:, 3);
band(:, 1) = irisk(:, 1);

% find out no. of evaluated points
[m, n] = max(irisk(:,1))

mid = floor(n/2)+1;

%initialize the variables
startnrb = band(mid, 1);
endnrb = band(mid, 1);
split = mid;
for i = mid:-1:1
    if band(i, 2) <= 5
        split = i;
        %startnrb = band(i, 1);
    else
        break;
    end
end
split
if split > 2
    x = band(split-2:split+2, 1);
    y = band(split-2:split+2, 2);
else
    switch split
```

```
        case 1
            x = band(split:split+4, 1);
            y = band(split:split+4, 2);
        case 2
            x = band(split-1:split+3, 1);
            y = band(split-1:split+3, 2);
    end
end


yy(:,1) = linspace(x(1), x(5), resolution)';
yy(:, 2) = spline(x, y, yy(:,1));

for i = resolution:-1:1
    if yy(i, 2) <= 5
        startnrb = yy(i, 1);
    else
        break;
    end
end

split = mid;
for i = mid:n
    if band(i, 2) <= 5
        split = i;
        endnrb = band(i, 1);
    else
        break;
    end
end

if split < n-2
    x = band(split-2:split+2, 1);
    y = band(split-2:split+2, 2);
else
    switch split
        case n-1
            x = band(split-3:split+1, 1);
            y = band(split-3:split+1, 2);
        case n
            x = band(split-4:split, 1);
            y = band(split-4:split, 2);
    end
end

yy(:,1) = linspace(x(1), x(5), resolution)';
yy(:, 2) = spline(x, y, yy(:,1));

for i = 1:resolution
    if yy(i, 2) <= 5
        endnrb = yy(i, 1);
    else
        break;
    end
end
```

```
bandwidth = endnrb - startnrb;
```

msrisk.m

```
% this code takes a risk plot and spec limits and returns risk at the mid
% spec and at spec limits.
% structure of returning matrix
% retval =
%   [mean       lowrisk     highrisk
%    lspec      lowrisk     highrisk
%    uspec      lowrisk     highrisk
%    minpos     minrisk         0
%    maxpos     maxrisk         0    ]


function retval = msrisk(irisk, speclimits)

xrisk = [mean(speclimits), speclimits ];

% quality characteristic
retval(1:3, 1) = xrisk';
% low at mid spec and spec limits
splow = spline(irisk(:,1), irisk(:,3));
retval(1:3, 2)= ppval(splow, xrisk)';
% high at mid spec and spec limits
sphigh = spline(irisk(:,1), irisk(:,4));
retval(1:3, 3)= ppval(sphigh, xrisk)';
% maximum positive risk
xx = linspace(speclimits(1), speclimits(2), 200);
finerisk(1,:) = ppval(splow, xx);
finerisk(2,:) = ppval(sphigh, xx);
[ymin, imin] = min(finerisk(1,:));
[ymax, imax] = max(finerisk(2,:));
retval(4,1:2) = [xx(imin) ymin];
retval(5,1:2) = [xx(imax) ymax];
```

Illinois Department of Transportation

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN