SANDIA REPORT
SAND2015-5453
Unlimited Release
Printed July 2015

Solving Inverse Radiation Transport Problems with Multi-Sensor Data in the Presence of Correlated Measurement and Modeling Errors

Edward V. Thomas, Christopher L. Stork, and John K. Mattingly

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	http://www.ntis.gov/search

SAND2015-5453
Unlimited Release
Printed July 2015

Solving Inverse Radiation Transport Problems with Multi-Sensor Data in the Presence of Correlated Measurement and Modeling Errors

Edward V. Thomas
Statistics and Surveillance Assessment Department

Christopher L. Stork
Computational Materials and Data Science Department

John K. Mattingly*
Nuclear Threat Science Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0829

*Currently at Department of Nuclear Engineering
North Carolina State University
Raleigh, NC 27607

Abstract

Inverse radiation transport focuses on identifying the configuration of an unknown radiation source given its observed radiation signatures. The inverse problem is traditionally solved by finding the set of transport model parameter values that minimizes a weighted sum of the squared differences by channel between the observed signature and the signature predicted by the hypothesized model parameters. The weights are inversely proportional to the sum of the variances of the measurement and model errors at a given channel. The traditional implicit (often inaccurate) assumption is that the errors (differences between the modeled and observed radiation signatures) are independent across channels. Here, an alternative method that accounts for correlated errors between channels is described and illustrated using an inverse problem based on the combination of gamma and neutron multiplicity counting measurements.

CONTENTS

FIGURES

This page is intentionally blank.

1. INTRODUCTION

Analysis of radiation measurements for nonproliferation frequently employs nonlinear regression to fit a model of the unknown source to a multivariate observed signature (e.g., gamma spectrum) [1]. Physics-based models are used to compute a radiation signature that is conditional on the values of the model parameters. Typically, the regression solver estimates the model parameters by minimizing a scalar chi-square metric that is a function of the sum of squared differences, by channel, between the observed and computed signatures. The chi-square metric weights the contribution of each channel in the radiation signature according to its error variance. However, the standard chi-square metric does not account for error covariance between the channels of the signature. The difference between the observed signature and the modeled signature is due to three sources: model specification error, measurement error, and model parameter estimation error. Treatment of model specification errors, i.e., errors arising from an incorrect choice of the model form, is beyond the scope of this paper; we will treat the latter two sources of error. The magnitudes of the remaining two error sources can be characterized by their covariances: the *measurement error covariance* and the *model parameter error covariance*. This paper focuses on incorporating estimates of the measurement error covariance and model parameter error covariance in the solution of the inverse transport problem. The methodology described herein is illustrated with an example combining gamma and neutron multiplicity counting measurements.

2. METHODOLOGY FOR SOLVING INVERSE PROBLEM

2.1 Statistical Models and Summary of Approach

Let $\{Y_i = \mu_i + \varepsilon_i,\ i = 1,2,\cdots,q\}$ and $\{\hat{Y}_i = \mu_i + \lambda_i,\ i = 1,2,\cdots,q\}$ respectively denote the observed and computed signatures. In this representation, q is the number of channels, μ_i is the true (expected) intensity in the i^{th} channel, ε_i is the measurement error of the i^{th} channel, and λ_i is the model error of the i^{th} channel. The expected value of the measurement error is zero for each channel, and the covariance of the measurement errors (across channels) is denoted by V_ε It is assumed that the model form is properly specified, so that λ_i is solely due to imprecise estimation of the model parameters. The expected value of the modeling error is zero in each channel, and the covariance of the errors in the computed signature is denoted by V_λ.

An iterative procedure is used to solve the inverse transport problem. The procedure starts with initial guesses for the physics model parameters and the two covariance matrices, V_ε and V_λ. Conditioned on the total covariance of the difference between values of $\hat{Y}$ and Y (given by $V = V_\varepsilon + V_\lambda$), generalized nonlinear least squares regression is used to re-estimate the physics model parameters by minimizing $Z = \frac{1}{q} \cdot D \cdot V^{-1} \cdot D^T$, where $D = \hat{Y} - Y$. The uncertainty of the re-estimated model parameters is propagated to the predicted signature, forming a revised estimate of V_λ (and therefore a revised estimate of V). Physics model parameters are re-estimated using the revised version of V.

The quadratic form that is minimized (Z) depends on V_ε and V_λ. While V_λ evolves as a consequence of the iterative process, V_ε must be estimated empirically. A straightforward approach for obtaining an estimate of V_ε is to utilize replicate measurements of the unknown source. Given a set of r replicate measurements, Y_r (q by r), the measurement error covariance matrix is estimated by [2]

$$\hat{V}_\varepsilon = \frac{1}{(r-1)} \cdot \sum_{k=1}^{r} (Y_k - \bar{Y}_r) \cdot (Y_k - \bar{Y}_r)^T,$$

where Y_k (q by 1) is the k^{th} replicate measured signature, and $\bar{Y}_r$ is the mean of the r replicate measured signatures.

2.2 Parameter Estimation Process

The computed signature is a function of a set of p estimated model parameters, $\hat{Y}_i = f(i; \hat{b}_1, \hat{b}_2, \cdots, \hat{b}_p)$. Assuming that the functional form of the model is correct, it is implicitly assumed that there is an unobservable set of model parameters that represent *truth*:$\{b_1, b_2, \cdots, b_p\}$. The deviations between the estimated model parameters and *true* parameter values are represented by $\{\delta_1, \delta_2, \cdots, \delta_p\}$. The associated covariance of $\{\delta_1, \delta_2, \cdots, \delta_p\}$ is denoted by V_δ. The objective is to find the set of model parameters that minimizes Z. The process begins with a set of initial estimates (guesses) for the model parameters, measurement error covariance (V_ε), and model error covariance (V_λ). Denote these estimates by $\{\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_p\}$, $\hat{V}_\varepsilon$, and $\hat{V}_\lambda$. A summary of the estimation process is as follows.

1. Let $\hat{V} = \hat{V}_\varepsilon + \hat{V}_\lambda$.
2. Compute the Cholesky decomposition of $\hat{V}$
    - $C = Cholesky\ Decomp\left(\hat{V}^{-1}\right)$ ➔ $C^T \cdot C = \hat{V}^{-1}$
3. Transform $Y, \hat{Y}$, and D: $Z = Y \cdot C^T$, $\hat{Z} = \hat{Y} \cdot C^T$, $D_Z = D \cdot C^T$
    - Model output is transformed: $\hat{Y} \Rightarrow \hat{Z}$
    - Measurements are transformed: $Y \Rightarrow Z$
4. Use the Levenberg-Marquardt (LM) algorithm [3] with Z, $\hat{Z}$, and the current estimate of the model parameters to update the estimated model parameters: $\{\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_p\}$
    - Use nonlinear least-squares regression of $\hat{Z}$ on Z to minimize $D_Z^T \cdot D^T$
    - The transformation (C) converts a generalized least-squares regression problem to an ordinary least squares regression problem (see below).

$$D_Z^T \cdot D_Z = \left(Y - \hat{Y}\right)^T \cdot V^{-1} \cdot \left(Y - \hat{Y}\right) = \left(Y - \hat{Y}\right)^T \cdot \left(C^T \cdot C\right) \cdot \left(Y - \hat{Y}\right) = \left(Y \cdot C^T - \hat{Y} \cdot C^T\right) \cdot \left(C \cdot Y - C \cdot \hat{Y}\right)$$

5. Obtain an estimate of V_δ (denoted by $\hat{V}_\delta$) from the LM algorithm.
6. STOP if the elements in $\{\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_p\}$ did not change significantly since the previous iteration
7. Use $\hat{V}_\delta$ and the sensitivity of $\hat{Y}$ to perturbations in the model parameters to propagate uncertainty from the physics model parameters to the computed signature in order to obtain an updated version of $\hat{V}_\lambda$. [If desired, update $\hat{V}_\varepsilon$ based on the re-computed signature.]
8. Return to 1.

2.3 Error Propagation

The following strategy can be used to provide an updated version of $\hat{V}_\lambda$.
Suppose the current values of the model parameters are $\{\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_p\}$. Select levels of perturbation for each parameter: $\{\Delta b_j\}$. First, compute the *first-order effect* of perturbing $\hat{b}_j$ on $\hat{Y}_i$ by a small amount given by Δb_j:

1. $$\Delta_{ij} = \frac{\hat{Y}_i\left(\hat{b}_1, \ldots, \hat{b}_{j-1}, \hat{b}_j + \Delta b_j, \hat{b}_{j+1}, \ldots, \hat{b}_p\right) - \hat{Y}_i\left(\hat{b}_1, \ldots, \hat{b}_{j-1}, \hat{b}_j - \Delta b_j, \hat{b}_{j+1}, \ldots, \hat{b}_p\right)}{2}$$
   for $i = 1,2,\ldots,q$ and $j = 1,2,\ldots,p$.
2. Normalize the *first-order effects*: $\alpha_{ij} = \dfrac{\Delta_{ij}}{\Delta b_j}$.
3. Assume the unknown errors in the current values of $\{\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_p\}$, given by $\delta_1, \delta_2, \ldots, \delta_p$ are distributed with $E\left(\delta_j\right) = 0, Var\left(\delta_j\right) = \sigma_j^2$, and $Cov\left(\delta_j, \delta_{j'}\right) = c_{jj'}$.

By assuming the effects of small perturbations of the model parameters ($\delta_1, \delta_2, \ldots, \delta_p$) on the computed signature are approximately linear and additive, a simple approximation of the i^{th} element of the computed signature at the perturbed conditions is given by

$\hat{Y}_i\left(\hat{b}_1 + \delta_1, \hat{b}_2 + \delta_2, \ldots, \hat{b}_p + \delta_p\right) \approx \hat{Y}_i\left(\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_p\right) + \lambda_i$, where $\lambda_i = \sum_{j=1}^{p} \delta_j \cdot \alpha_{ij}$ is the propagated error in the i^{th} element of the computed signature. The expected value of λ_i is zero. The diagonal elements of V_λ are given by

$$Var(\lambda_i) = Cov(\lambda_i, \lambda_i) = \sum_{j=1}^{p} \alpha_{ij}^2 \cdot \sigma_j^2 + 2 \cdot \sum_{\substack{j=1:p \\ j'<j}} \alpha_{ij} \cdot \alpha_{ij'} \cdot c_{jj'} \ .$$

The $(i, i')^{th}$ element of V_λ is given by

$$Cov(\lambda_i \cdot \lambda_{i'}) = \sum_{j=1}^{p} \alpha_{ij} \cdot \alpha_{i'j}\, \sigma_j^2 + \sum_{\substack{j=1:p \\ j'<j}} \left(\alpha_{ij} \cdot \alpha_{i'j'} + \alpha_{ij'} \cdot \alpha_{i'j}\right) \cdot c_{jj'} \ .$$

This simple strategy has been implemented in MATLAB (see functions *param_sens.m* and *error_prop.m* in the Appendix).

3. IMPLEMENTATION

The estimation process has been implemented in MATLAB (see Appendix).

Two external codes (*GADRAS* [4] and *nlinfit.m* [5]) were used extensively within the MATLAB implementation. GADRAS was called extensively (within MATLAB) to compute the radiation signature given values of the physics model parameters. Prior to calling GADRAS, current model parameters were written to an input file. GADRAS read the model parameters from the input file, computed the signature (based on the model parameters), and wrote the computed spectrum to an output file. When control was returned to MATLAB, the computed spectrum was retrieved (see *kernel_call_pn.m* in the Appendix).

MATLAB's implementation of the Levenberg-Marquardt algorithm for nonlinear regression, *nlinfit.m*, was used to estimate the model parameters given the transformed measurements (Z) and the transformed computed spectra ($\hat{Z}$).

In general, the implementation was very straightforward. However, there were some issues that required special attention. In particular, the *derivative step size* needed to assess the sensitivity of the spectrum to perturbations of model parameters (used in both *nlinfit.m* and *param_sens.m*) needed to be set at a level greater than what was originally anticipated (possibly reflective of the level of numerical precision within GADRAS). In addition, *nlinfit.m* apparently does not allow the derivative step size to vary for each model parameter.

4. ILLUSTRATIVE EXAMPLE USING A MULTI-SENSOR SIGNATURE

The proposed methodology has been used to solve inverse radiation transport problems for cases involving gamma spectra [6]. Here, we consider an example where the radiation signature consists of a combination of simulated gamma and neutron multiplicity counting measurements. Note that gamma spectroscopy and neutron multiplicity counting measurements are complementary. While gamma spectroscopy is highly specific to nuclear composition, internal features of the source cannot always be resolved due to self-shielding in special nuclear material (SNM). Neutron multiplicity counting measurements are used to estimate integral properties of special nuclear material (SNM), including: neutron source strength, neutron multiplication, and neutron generation time [7]. Hence, neutron multiplicity counting measurements can only estimate bulk system properties such as mass and multiplication. However, since neutrons transport through SNM more easily than photons, it can help resolve internal features of the source. Thus, an inverse solution based on the combined observation and analysis of a gamma spectrum and a neutron multiplicity counting measurement from the source should match features of both observables. The solution can reveal both the nuclear composition of the source as well as its bulk configuration.

For this simulated example, the source of the radiation signature is a plutonium sphere with a radius of 3.794 cm imbedded in a shell of polyethylene with thickness of 3.81 cm. The radius of the plutonium sphere and the thickness of the polyethylene reflector/moderator that is surrounding it are the parameters in the physics model to be estimated. The radiation signature consists of 270 channels: 246 gamma channels augmented with 24 Feynman-Y channels. The Feynman-Y signature, given by $F_Y(\Delta t) = \frac{\sigma^2(\Delta t)}{\mu(\Delta t)} - 1$, measures the variance observed in the neutron counting distribution that is in excess of the variance expected for a Poisson distribution [7]. It depends on the duration of the counting time (i.e., the "coincidence gate" width), Δt. Figure 5 displays the Feynman-Y signature associated with a simulated 4-second observation of the source.

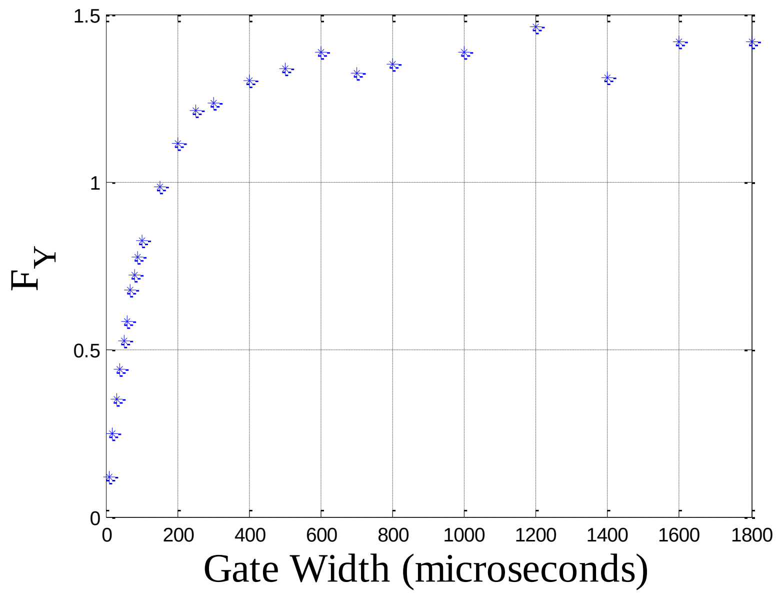

Figure 1. Feynman-Y signature of a plutonium sphere imbedded in polyethylene.

Two important features of the Feynman-Y signature are the asymptotic value achieved for long coincidence gate widths and the shape of rise from zero to the asymptotic value [7]. The Feynman-Y measurement is based on the observation of a stream of neutron count data, $\{X(1), X(2), \ldots, X(m), \ldots\}$, where $X(m)$ is the number of neutrons observed in the m^{th} time epoch. Suppose that each time epoch is of length dt. Then, $F_Y(dt) = \frac{\sigma^2(dt)}{\mu(dt)} - 1$, where $\mu(dt)$ and $\sigma^2(dt)$ are the counting statistics derived directly from $\{X(1), X(2), \ldots, X(m), \ldots\}$. Likewise, $F_Y(2 \cdot dt) = \frac{\sigma^2(2 \cdot dt)}{\mu(2 \cdot dt)} - 1$, where $\mu(2 \cdot dt)$ and $\sigma^2(2 \cdot dt)$ are the counting statistics derived from $\{X(1) + X(2), X(3) + X(4), \ldots\}$. Given this formulation, it is easy to understand how the elements within the set of Feynman-Y measurements, $\{F_Y(dt), F_Y(2 \cdot dt), F_Y(3 \cdot dt), \ldots\}$, are correlated.

As an aside, the stream of neutron count data can be approximated by a discrete-time branching process with two sources of neutrons (one source from spontaneous production, the other from chain reactions). That is, $X(m)$ is distributed as a Poisson random variable with an expected value (and variance) of $k + \lambda \cdot X(m-1)$, where k is the expected number of neutrons produced spontaneously during each epoch and $\lambda \cdot X(m-1)$ is the expected number of neutrons produced from chain reactions during the "*m - 1*" epoch. If there is no fissile material present, then $\lambda = 0$.

The long-term process is stationary (sub-critical) only if $\lambda < 1$. See Pazsit and Pal [8] for detailed information on the statistics of neutron fluctuations.

4.1 Estimation of measurement error covariance

Solution of the inverse problem requires initial estimates for the measurement error covariance, V_ε. Here, due to the independence of the gamma and neutron multiplicity measurements, $V_\varepsilon = \begin{bmatrix} V_\varepsilon(gamma) & 0 \\ 0 & V_\varepsilon(neutron) \end{bmatrix}$. As described in [6], the errors in the gamma measurements are approximately Poisson-distributed and independent. Thus, $V_\varepsilon(gamma)$ can be estimated via replicate measurements of the source. Here, initial estimates for the measurement error covariance specific to the net gamma spectrum, $V_\varepsilon(gamma)$, are based on the initial estimates for the model parameters and a reference background spectrum (B). The observed gamma spectrum (O) consists of the combination of background and source signals. The net gamma spectrum (Y_{gamma}) consists of a background-corrected spectrum: $Y_{gamma} = O - B$, where B is the reference background spectrum. From the initial estimates of the model parameters, we obtain an initial estimate of the net spectrum, $\hat{Y}_{gamma}(initial)$. The combination of B and $\hat{Y}_{gamma}(initial)$ was taken to be the expected intensity for O. Then, assuming independent Poisson counting statistics, $\hat{V}_\varepsilon(gamma) = diag\left(B + \left(B + \hat{Y}_{gamma}(initial)\right)\right)$. Throughout the iterative process, $\hat{V}_\varepsilon(gamma)$ is modified to incorporate updated versions of $\hat{Y}_{gamma}$.

A simple way to estimate the measurement error covariance for the Feynman-Y signature is via replicate measurements. Another way to estimate the measurement error covariance is to use the bootstrap re-sampling procedure [9] to create an ensemble of pseudo-replicates from a single observation. Figure 2 displays the estimated measurement error covariance obtained by the bootstrap method for the Feynman-Y signature displayed in Figure 1. In this case 200 pseudo-replicates of the Feynman-Y signature were created. Each pseudo-replicate was constructed by concatenating forty randomly selected 100 milli-second blocks (from the original 4-second simulated observation) into a 4-second observation. The starting position for each block was randomly selected from a uniform distribution across the original 4-second observation. The block size is believed to be sufficiently large given that the steady-state variance (Feynman-Y asymptote) is reached within several milliseconds. The measurement error covariance was estimated directly from these 200 pseudo-replicates.

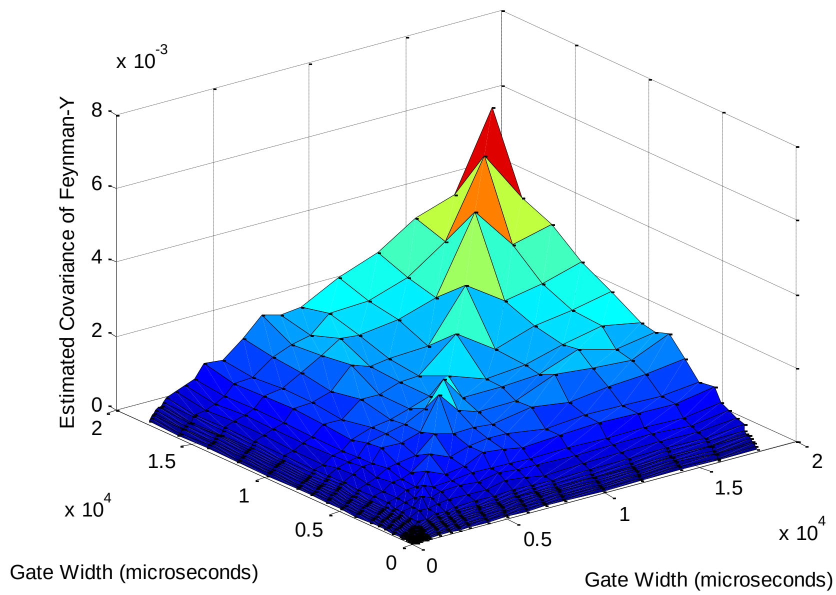

Figure 2. Estimated covariance of Feynman-Y measurement error based on bootstrap.

4.2 Results

The iterative estimation process was initiated several times, each with a different set of initial guesses for the physics model parameters. In each case, the initial estimate of $V_\varepsilon(gamma)$ was based on the initial values of the model parameters. In contrast, $\hat{V}_\varepsilon(neutron)$ was based on a simulated stream of neutron count data. Throughout the iterative process, $\hat{V}_\varepsilon(gamma)$ was modified to incorporate updated versions of $\hat{Y}$. The estimated model error covariance ($\hat{V}_\lambda$) was initially set to be the zero matrix. In each case, the estimation process converged rapidly, within 3 iterations.

Figures 3 displays the evolved estimates of the diagonal elements of $V_\varepsilon(gamma)$ and $V_\lambda(gamma)$. Figure 4 displays the diagonal elements of $\hat{V}_\varepsilon(neutron)$ and the evolved estimates of the diagonal elements of $V_\lambda(neutron)$. Except for a few gamma channels, the model error variances are inconsequential when compared to measurement error variances. So, in this particular case, consideration of the model errors does not further constrain the solution. This can be seen by comparing Figures 5 and 6. These figures show the quadratic form, *Z*, as a function of the model parameters. The two surfaces (relating to the initial and final [third iteration] estimate of *V*) are substantially the same.

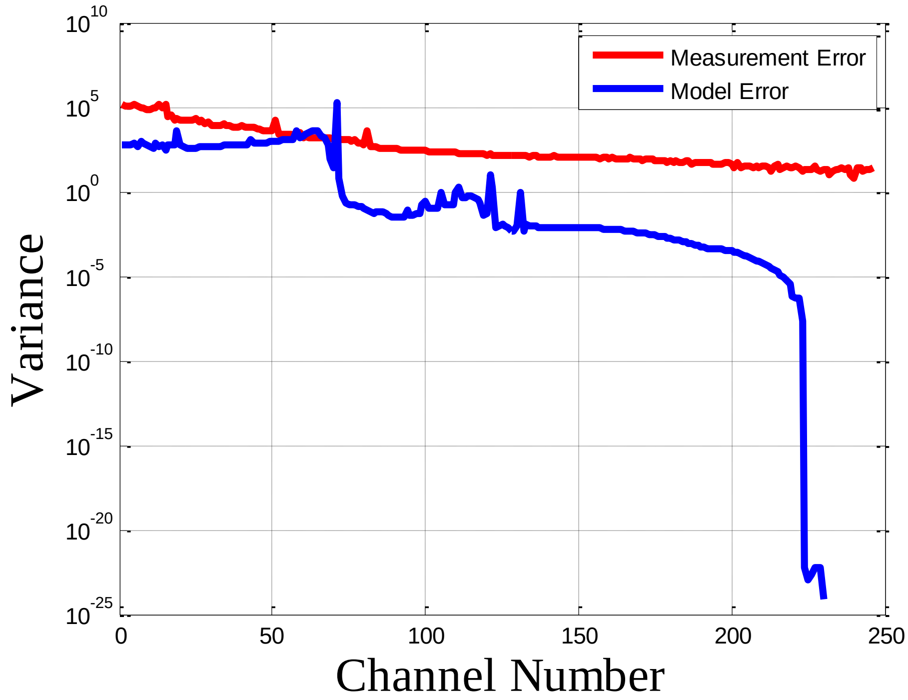

Figure 3. Estimates of Measurement Error and Model Error Variances (gamma channels).

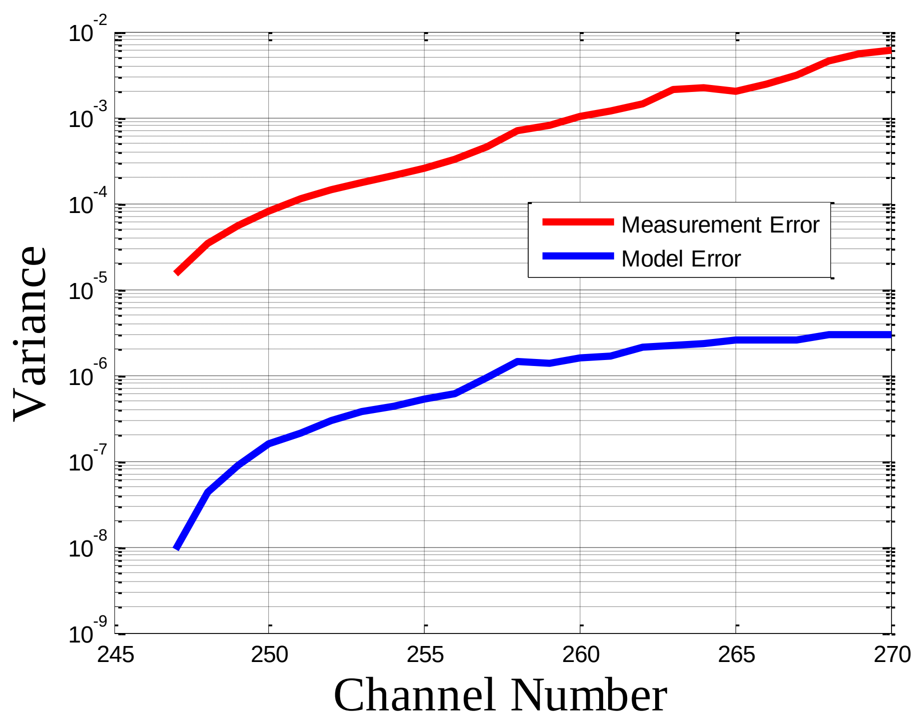

Figure 4. Estimates of Measurement Error and Model Error Variances (Feynman-Y channels).

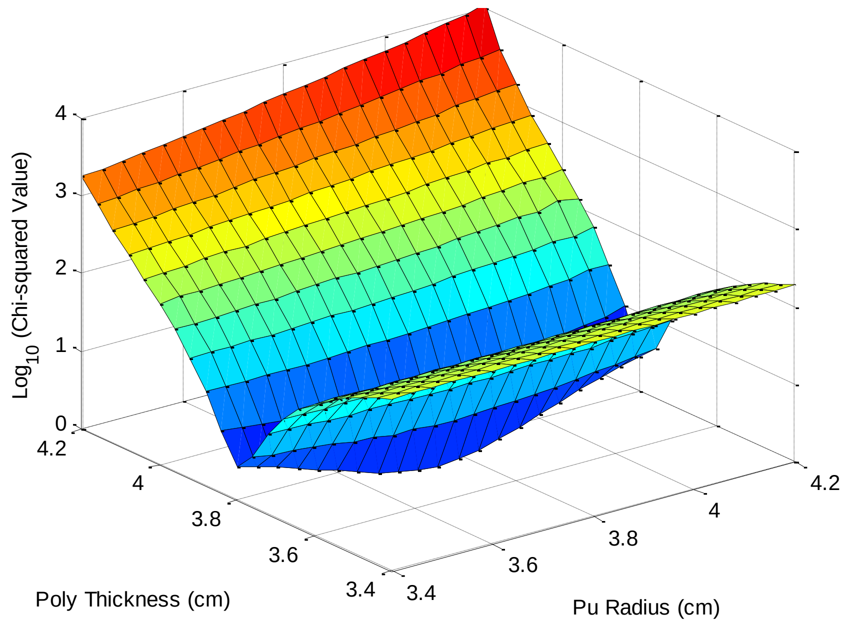

Figure 5. Z versus physics model parameters for initial iteration.

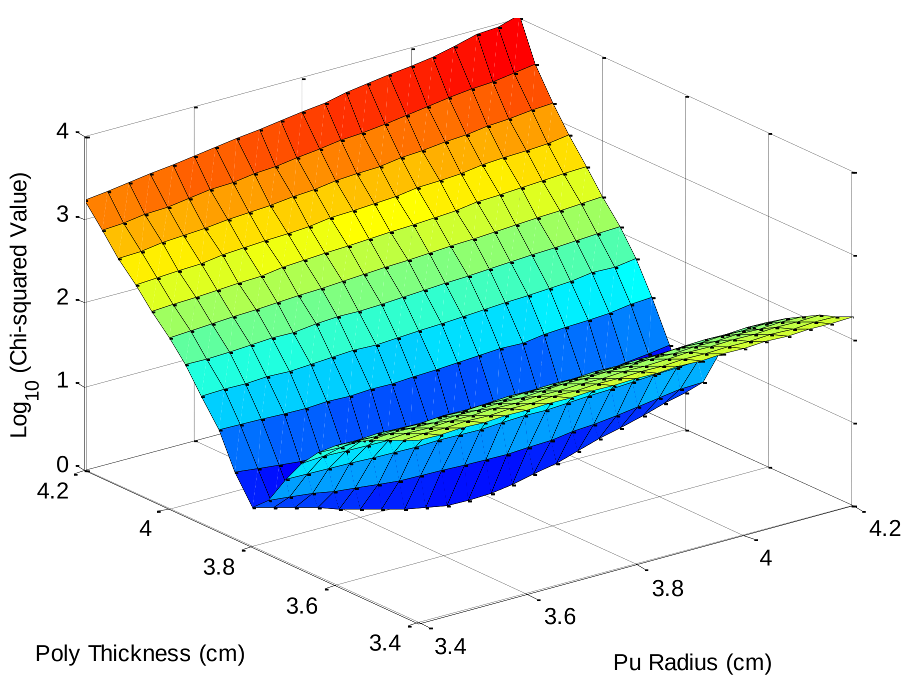

Figure 6. Z versus physics model parameters for final (third) iteration.

Figure 7 displays the surface of the quadratic form, *Z*, for the final iteration of *V* when only gamma channels are used to solve this inverse problem. Again, the surfaces displayed in Figures 6 and 7 are substantially similar. Thus, there is apparently no added benefit of including neutron measurements to solve this particular inverse problem. However, it is likely that other problems might benefit from this approach.

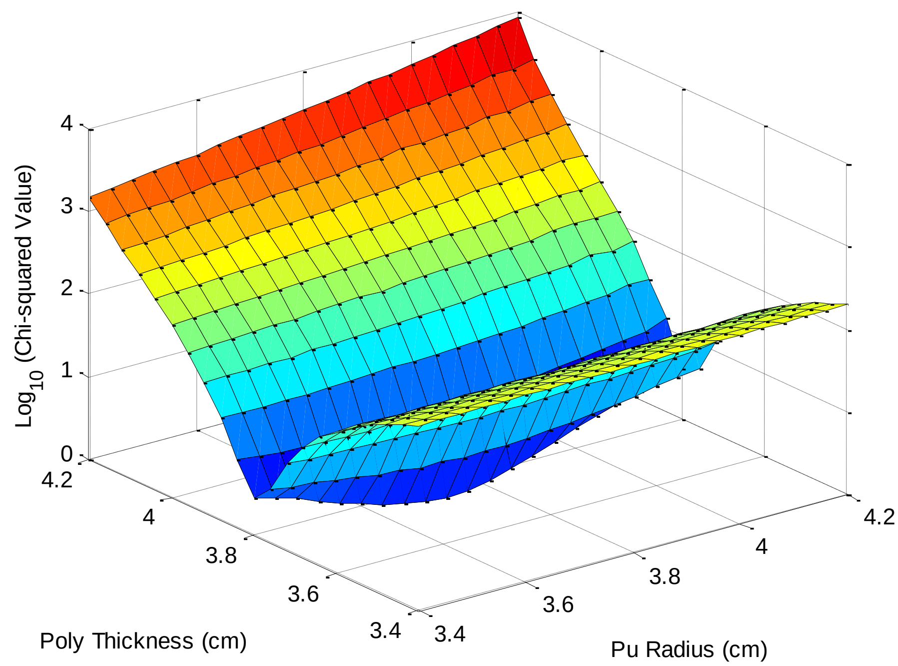

Figure 7. Z versus physics model parameters for third iteration (gamma channels only).

REFERENCES

[1] J. Mattingly and D. J. Mitchell, “A framework for the solution of inverse radiation transport problems,” IEEE Nuclear Science Symposium, Dresden, Germany, October 2008.
[2] M.N. Leger, L. Vega-Montoto and P.D. Wentzell, “Methods for systematic investigation of measurement error covariance matrices,” Chemometrics and Intelligent Laboratory Systems, 2005. 77(1-2): 181-205.
[3] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," SIAM Journal on Applied Mathematics, 1963, 11: 431–441.
[4] D. J. Mitchell, *GADRAS-95 User's Manual,* Sandia National Laboratories Systems Research Center 5900 Report (1995).
[5] MATLAB, http://www.mathworks.com/
[6] E. V. Thomas, C. L. Stork, and J. K. Mattingly, “Accounting for correlated errors in inverse radiation transport problems,” IEEE Nuclear Science Symposium, Knoxville, Tennessee, November 2010.
[7] J. Mattingly, “Computation of neutron multiplicity statistics using deterministic transport,” IEEE Nuclear Science Symposium, Orlando, FL, October 2009.
[8] I. Pazsit and L. Pal, “Neutron Fluctuations,” Amsterdam: Elsevier, 2008.
[9] B. Efron and R. J. Tibshirani, “An Introduction to the Bootstrap,” London, U.K.: Chapman and Hall, 1993.

APPENDIX – MATLAB CODES

Driver script

```
clear all;
% get net gamma spectrum (truth)
load gspectrum_truth.txt;
y_gam=gspectrum_truth(12:257);

% Get background gamma spectrum
load background.gammaspectrum.txt;

% background count time = 3X signal count time
y_gamma_bkg=background_gammaspectrum(12:257);

% form observed foreground and background gamma measurements
obs_foreground = poissrnd (y_gam + y_gamma_bkg / 3.3);
obs_bkg = poissrnd (y_gamma_bkg);

% form observed net gamma measurement
y_gamma = obs_foreground -obs_bkg/3.3;

% get Feynman-Y measurement (truth)
y_neu=neutron_interpolate('fspectrum_truth.txt');

% get covariance matrix of Feynman-Y signature
load cvar;
c=chol(cvar);

% form observed Feynman-Y signature
randn('seed',843809);
err_neu=randn(1,24)*c;
y_neutron=y_neu+err_neu';

% display net gamma measurement and observed Feynmann-Y signature
figure; semilogy((1:246),y_gamma,'-*'); grid;
xlabel('Channel Number','FontSize',20,'FontName','Times New Roman');
ylabel('Gamma Intensity','FontSize',20,'FontName','Times New Roman');
gate=[10 20 30 40 50 60 70 80 90 100 150 200 250 300 400 500 600 700 800 1000 1200 1400 1600 1800]';
figure; plot(gate,y_neutron,'*'); grid;
xlabel('Gate Width (microseconds)','FontSize',20,'FontName','Times New Roman');
ylabel('F_Y','FontSize',20,'FontName','Times New Roman');

% concatenate gamma and Feynman-Y signatures
s=[y_gamma; y_neutron];
% Specify initial guess for parameters with truth: b0=[3.794; 3.810]
b0=[4; 4];

% Get estimate of spectrum based on initial guess of model parameters
z_raw=kernal_call_raw(b0);

% Set initial model error covariance
v_lambda=zeros(270);

% covar of feynman_Y
load cvar;
v_eps=zeros(270);
bb(:,1)=b0;

% Set options for nlinfit
options=statset('Display','iter','DerivStep',.0003);

for i=2:5;

    % form measurement error covaraiance
    v_eps(1:246,1:246)=diag(z_raw(1:246) + 2*y_gamma_bkg);
    v_eps(247:270,247:270)=cvar;

    if i > 2;
        % Evaluate model output sensitivity to small perturbations in parameters
        d_beta=[.0003; .0003];
        alpha=param_sens(b0,d_beta);

        % Propagate model parameter error covariance into model spectrum error covariance
        v_lambda=error_prop(alpha,covb);
    end;

    % Total covariance is sum of measurement/model error covariances
    v_tot=v_eps+v_lambda;

    % Transform net spectrum
    c=chol(v_tot);
    z_trans=c*s;

    % Call nlinfit with
    [beta,dum1,dum2,covb]=nlinfit(c,z_trans,@kernal_call_pn,b0,options);

    % Retain current parameter estimates
    bb(:,i)=beta;
    b0=beta;

    % Get new estimate of spectrum based on current estimates of model parameters
    z_raw=kernal_call_raw(b0);
end;
```

kernel_call_pn

```
function z=kernal_call_pn(beta,c)
%
%
%   PURPOSE:
%   Provides transformed model spectrum.
%
%   SYNTAX:
%   z=kernal_call_pn(beta,c);
%
%   OUTPUT VARIABLES:
%   -(1) z-->q-vector of transformed model spectrum
%
%   INPUT VARIABLES:
%   -(1) beta-->p-vector of model parameters
%   -(2) c-->qxq matrix defining Cholesky transformation matrix
%
%   SEE ALSO:
%
%   FULL DESCRIPTION:
%   1. Writes model parameter values into "source.params" file
%   2. Calls Gadras kernal (using "source.params" as input)
%   3. Retrieves gamma spectrum derived from Gadras from "gspectrum.txt"
%   4. Retrieves feynman_y neutron spectrum derived from Gadras from "fspectrum.txt"
%   5. Interpolates neutron spectrum into specific gates
%   6. The retrieved spectra (gamma augmented with neutron) is transformed by c
%   7. Transformed augmented spectrum is output

fid=fopen('c:\gadras\detector\kernel\source.params','w');
fprintf(fid,'pu_thickness=%16.10E\r\n',beta(1));
fprintf(fid,'poly_thickness=%16.10E\r\n',beta(2));
fclose(fid);

! cd c:\gadras\detector\kernel
! C:\gadras\Kernel\Sandia.GadrasKernel.Application.exe source.params berp1p5.template gspectrum.txt fspectrum.txt
pause(.01);

load gspectrum.txt
y_gamma=gspectrum(12:257);
y_neutron=neutron_interpolate('fspectrum.txt');
y=[y_gamma; y_neutron];

z=c*y;
```

kernel_call_raw

```
function y=kernal_call_raw(beta)
%
%
%   PURPOSE:
%   Provides model spectrum.
%
%
%   SYNTAX:
%   z=kernal_call_raw(beta);
%
%   OUTPUT VARIABLES:
%   -(1) y-->q-vector of model spectrum
%
%   INPUT VARIABLES:
%   -(1) beta-->p-vector of model parameters
%
%   SEE ALSO:
%
%   FULL DESCRIPTION:
%   1. Writes model parameter values into "source.params" file
%   2. Calls Gadras kernal (using "source.params" as input)
%   3. Retrieves gamma spectrum derived from Gadras from "gspectrum.txt"
%   4. Retrieves neutron spectrum derived from Gadras from "fspectrum.txt"
%   5. Interpolates neutron spectrum into specific gates
%   6. Concatenated signature is output
%
%
%   REFERENCES:
%
%
%
fid=fopen('c:\gadras\detector\kernel\source.params','w');
fprintf(fid,'pu_thickness=%16.10E\r\n',beta(1));
fprintf(fid,'poly_thickness=%16.10E\r\n',beta(2));
fclose(fid);

! cd c:\gadras\detector\kernel
! C:\gadras\Kernel\Sandia.GadrasKernel.Application.exe source.params berp1p5.template gspectrum.txt fspectrum.txt
pause(.03);

load gspectrum.txt
y_gamma=gspectrum(12:257);
y_neutron=neutron_interpolate('fspectrum.txt');
y=[y_gamma; y_neutron];
```

param_sens

```
function alpha=param_sens(beta,d_beta)
%
%   PURPOSE:
%   Computes partial derivatives of spectrum with respect to model parameters
%
%
%   SYNTAX:
%   alpha=param_sens(beta,d_beta);
%
%   OUTPUT VARIABLES:
%   -(1) alpha--> qxp matrix of partial derivatives
%
%   INPUT VARIABLES:
%   -(1) beta-->p-vector of model parameters
%   -(2) d_beta-->p-vector of derivative step sizes for model parameters
%
%   SEE ALSO:
%
%
%
%   FULL DESCRIPTION:
%   Computes partial derivatives of spectrum with respect to model parameters
%   Uses call to GADRAS kernal that outputs untransformed spectra
%
%   REFERENCES:
%
[n,param_dim]=size(beta);

b_upp=beta+d_beta;
b_low=beta-d_beta;

for j=1:n
    b=beta;
    b(j)=b_upp(j);
    y_upp=kernal_call_raw(b);
    b(j)=b_low(j);
    y_low=kernal_call_raw(b);
    alpha(:,j)=(y_upp-y_low)./(2*d_beta(j));
end;
```

error_prop

```
function v=error_prop(alpha,covb)
%
%   PURPOSE:
%   Propagates uncertainty from model parameters to computed spectrum
%
%
%   SYNTAX:
%   v=err_prop(alpha,covb)
%
%   OUTPUT VARIABLES:
%   -(1) v-->qxq-matrix representing covariance of computed spectrum
%
%   INPUT VARIABLES:
%   -(1) alpha-->qxp-vector of sensitivity of computed spectrum to
%        perturbations of model parameters (obtained from param_sens.m)
%   -(2) covb-->pxp covariance matrix associated with model parameter
%        estimates (obtained from nlinfit.m)
%
%
%   SEE ALSO:
%   para_sens.m
%   nlinfit.m
%
%   FULL DESCRIPTION:
%   Uses first-order sensitivity of computed spectrum to perturbations
%   of model parameters with uncertainty in model parameters to propagate
%   uncertainty to computed spectrum
%
%   REFERENCES:
%
[q,p]=size(alpha);
v=zeros(q,q);
for i=1:q
    for j=1:q
        for k=1:p
            v(i,j)=v(i,j)+alpha(i,k)*alpha(j,k)*covb(k,k);
            if k>1
                for kp=1:k-1
                    v(i,j)=v(i,j)+(alpha(i,k)*alpha(j,kp)+alpha(i,kp)*alpha(j,k))*covb(k,kp);
                end;
            end;
        end
    end;
end;
```

neutron_interpolate

```
function y_neutron=neutron_interpolate(fname)
%
%
%   PURPOSE:
%   Takes GADRAS neutron output (with variable gates - in fname) and reformats to the
%   indicated gate structure indicated in x by linear interpolation
%
%   SYNTAX:
%   y_neutron=neutron_interpolate(fname);
%
%   OUTPUT VARIABLES:
%   -(1) y_neutron-->24-vector of reformatted Feynman-Y signature
%
%   INPUT VARIABLES:
%   -(1) fname-->filename containing output Feynman-Y signatue from GADRAS
%
%   SEE ALSO:
%
%   FULL DESCRIPTION:
%   1. Inputs Feynman-Y spectrum output by GADRAS
%   2. Interpolates Feynman-Y spectrum output by GADRAS to fixed gates
%   3. Outputs reformatted Feynman-Y signature
%
%
%   REFERENCES:
%


x=[10 20 30 40 50 60 70 80 90 100 150 200 250 300 400 500 600 700 800 1000 1200 1400 1600 1800]';

[nx,dum]=size(x);
y_neutron=zeros(nx,1);

fid = fopen(fname);
a=fscanf(fid, '%g');
n=a(1);
nn=2*n;
aa=reshape(a(2:nn+1),2,n)';
fclose(fid);

gate=aa(:,1);
fy=aa(:,2);

low_ind=1;
for i=1:nx
    while x(i) > gate(low_ind)
        low_ind=low_ind+1;
    end;
    y_neutron(i)=fy(low_ind-1) + ( (x(i)-gate(low_ind-1)) / …
        (gate(low_ind) - gate(low_ind-1)) )*(fy(low_ind)-fy(low_ind-1));
end;
```

Distribution

1 John K. Mattingly (electronic copy)
Department of Nuclear Engineering
North Carolina State University
Raleigh, NC 27607

1	MS0428	JF Nagel	00400 (electronic copy)
1	MS0434	Michelle Stevens	00410 (electronic copy)
1	MS0829	Jennifer Franklin	00415 (electronic copy)
1	MS0829	Edward Thomas	00415 (electronic copy)
1	MS0886	Chris Stork	01814 (electronic copy)
1	MS0899	Technical Library	09536 (electronic copy)

Sandia National Laboratories