LA-UR- 10-01393

Approved for public release; distribution is unlimited.

Title: Calculating System Reliability with SRFYDO

Author(s): Jerome Morzinski, Christine Anderson-Cook, Richard Klamann

Intended for: JANNAF Meeting
Colorado Springs, CO
May 3, 2010

Los Alamos NATIONAL LABORATORY
EST. 1943

CALCULATING SYSTEM RELIABILITY WITH SRFYDO

Jerome Morzinski, Christine Anderson-Cook, Richard Klamann
Statistical Sciences Group, Los Alamos National Laboratory
Los Alamos, New Mexico

ABSTRACT

SRFYDO is a process for estimating reliability of complex systems. Using information from all applicable sources, including full-system (flight) data, component test data, and expert (engineering) judgment, SRFYDO produces reliability estimates and predictions. It is appropriate for series systems with possibly several versions of the system which share some common components. It models reliability as a function of age and up to 2 other lifecycle (usage) covariates. Initial output from its Exploratory Data Analysis mode consists of plots and numerical summaries so that the user can check data entry and model assumptions, and help determine a final form for the system model. The System Reliability mode runs a complete reliability calculation using Bayesian methodology. This mode produces results that estimate reliability at the component, sub-system, and system level. The results include estimates of uncertainty, and can predict reliability at some not-too-distant time in the future. This paper presents an overview of the underlying statistical model for the analysis, discusses model assumptions, and demonstrates usage of SRFYDO.

INTRODUCTION

SRFYDO is designed to simplify and enhance the process of system reliability modeling. A classical method of reliability analysis for a missile system, for example, might assess and predict reliability based on full system (flight) data, because that is considered the "gold standard." The enhancement SRFYDO brings is the capability to incorporate other types of information into the analysis, including component test data, engineering judgment, and data from similar systems. While full system tests are the most direct measure of system reliability, they are often prohibitively expensive and may involve destructive testing which reduces the available inventory of systems. Hence using supplemental data to enhance the analysis can improve precision in a cost-effective way.

SRFYDO accommodates series systems, including multiple variants, and models reliability as a function of age and up to two other covariates, such as number of transfers, captive carry hours, etc. Users input data tables that define components of the system, associate components with test set data, specify what is known about the components for observed flight outcomes, and enter flight data, test set results, and prior information (engineering judgment).

Output from the initial exploratory data analysis consists of summaries and plots that allow the user to check for data entry errors and assess whether model assumptions are met, and provide the opportunity to modify the input tables before running the system reliability analysis. This analysis produces reliability summaries at the component and system level in both tabular and graphical form. These summaries include the median (50^{th} percentile) values as well as uncertainty estimates.

RESULTS AND DISCUSSION

OVERVIEW OF THE MODEL

SRFYDO, "System Reliability Formatter for YADAS* Data and Output," is a Python application designed to support an enhanced process for estimating reliability of complex systems. A view of the GUI interface for SRFYDO is shown in Fig. 1. A typical method for assessing reliability of a missile system might be to take all the flight data, run a probit or logistic regression model, and produce reliability estimates from that data. Those results will generally not take advantage of other information that may be available, such as data from component testing (which we will call testset data), engineering judgment,

* YADAS is a software environment for statistical analysis using Markov chain Monte Carlo.

Approved for public release; distribution is unlimited.
This work was performed under sponsorship of the DoD and DOE Joint Munitions Program at Los Alamos National Laboratory.

and results from similar systems (e.g. variants that share several components). SRFYDO incorporates all available information using peer-reviewed and published statistical methodologies. Wilson et al. (2006) and Anderson-Cook et al. (2007, 2008) give details. For details on the computing environment YADAS, see Graves (2007).

The goal of a SRFYDO analysis is to characterize the performance of a collection of missiles. Statistical methods are necessary because there is always some uncertainty in our knowledge about the state of the collection. The statistical approach used in SRFYDO is Bayesian Analysis. We start with best estimates of certain parameters, estimates that may be based on engineering judgment or experience with similar systems. Then we add the data – the results observed from flight tests and component testing, and update our prior estimates to get new values for the parameters. Starting values for the parameters are known as priors, and the updated values are the posterior (distributions). Because the forms of the parameter distributions are usually quite complex, it is necessary to use numerical methods to solve the underlying integral equations, rather than hope for closed-form solutions. The numerical method of choice is Markov chain Monte Carlo (MCMC), as implemented in the software environment YADAS.

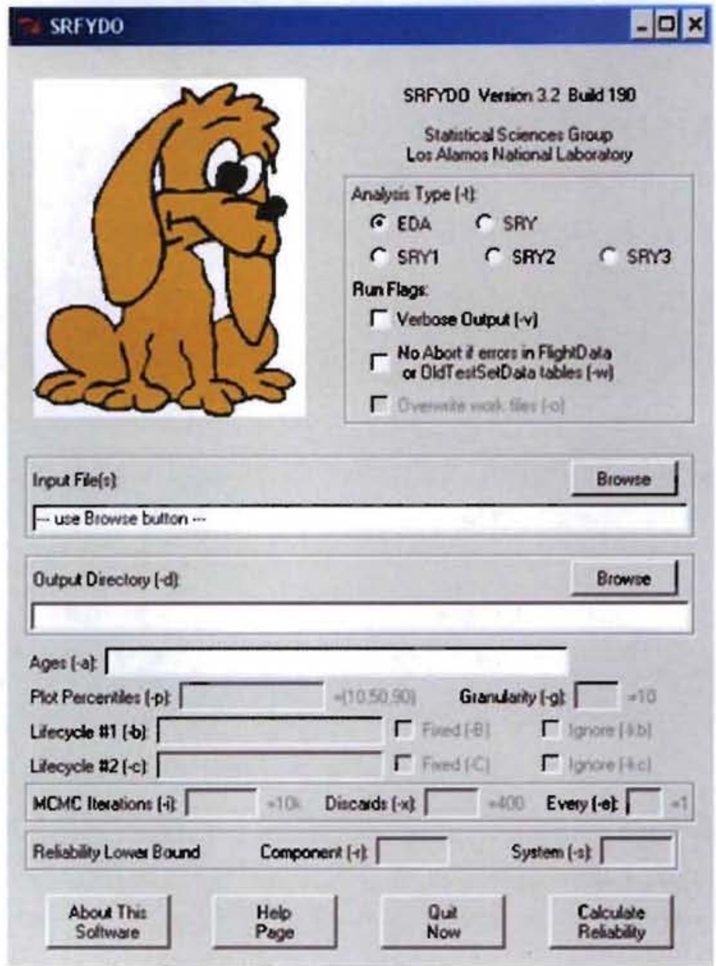

Figure 1: SRFYOD GUI

Monte Carlo methods have gained widespread acceptance in a variety of applications where it is impossible to obtain exact results deterministically. The basic steps of the Monte Carlo method are:

- Define a domain of possible inputs
- Generate inputs randomly from the domain using specified probability distributions
- Perform computations using the inputs
- Repeat many times, and aggregate results of individual computations into the final result

MCMC is based on constructing a Markov chain (a random process in which the future state of a system depends only on the present state) which guides sampling from the desired probability distributions. Details of the fundamentals of Bayesian computing to approximate the posterior distribution via Markov chain Monte Carlo can be found in Gelfand and Smith (1990), Casella and George (1992). The MCMC algorithms produce samples from the joint posterior distribution of the model parameters by sequentially updating each model parameter conditional on the current values of the other model parameters. The particular type of MCMC implemented in YADAS uses the Metropolis-Hastings algorithm [Chib and Greenberg (1995)], which generates a random walk using a proposal density and a method for rejecting proposed moves. For more details on MCMC and YADAS, see Graves (2007).

The final result of the MCMC analysis is a set of posterior distributions of model parameters that yields reliability values for the collection of missiles, including predictions out to a reasonable future time, with associated uncertainties.

SRFYDO is intended to accommodate systems with the following properties:

- Series system (only components critical to system success included)
- Multiple variants possible (several versions of system with common components)
- Can incorporate any or all of flight, continuous testset and discrete testset data
- Includes age + up to 2 other lifecycle covariates (e.g. captive carry hours, # of transfers, etc.)

Input to SRFYDO consists of tables (in MS Excel format) that do the following:

- Define the components of the system across all variants, and associate a list of components with each system variant
- Associate the type (discrete / continuous) and relationship (upper / lower bound) of test set measures (testset limit) for each component
- Define what is known about the components for each observed flight outcome (success and failure)
- Input the flight data
- Input the component test set data (i.e. quality assurance data)
- Input prior information to summarize expert knowledge about system reliability

Samples of the input tables are shown in Fig. 2. SRFYDO produces output for both a preliminary non-parametric exploratory data analysis (EDA) and for a more formal model-based parametric system reliability analysis (SRY).

In EDA (Exploratory Data Analysis) mode, plots and summaries for a preliminary analysis are produced. EDA summaries are used to select a final system structure that most closely matches the assumptions of the underlying statistical model. The following summaries are created:

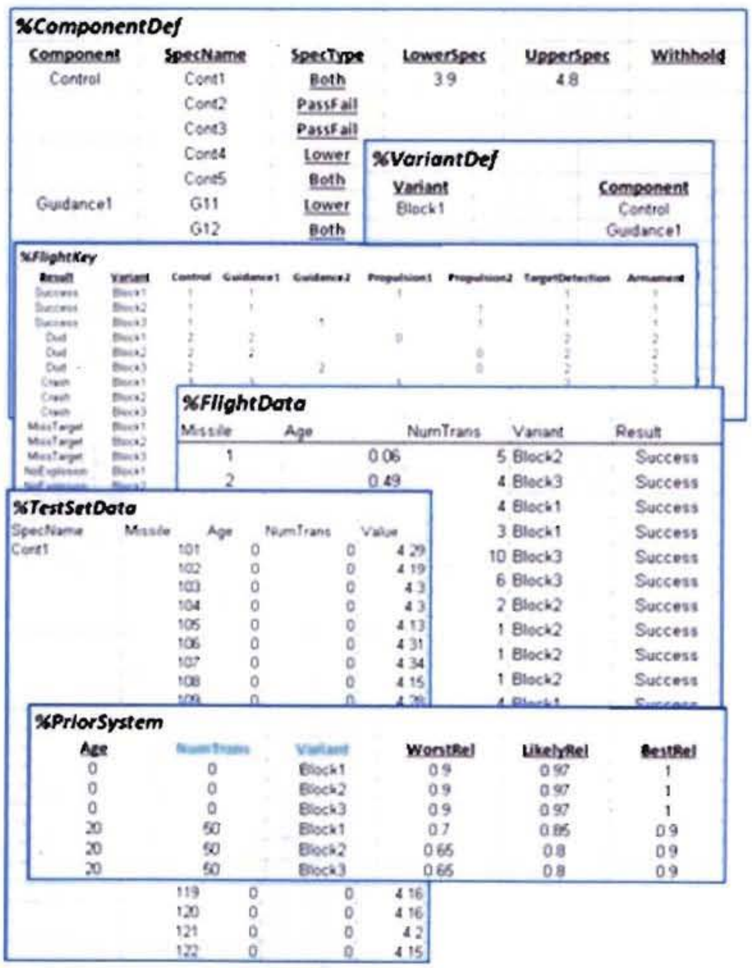

Figure 2: Samples of the Excel tables required as input for SRFYDO

- Tables of preliminary ranges for system and component reliability from data
- Tables of ranges of ages and lifecycle covariates for observed data
- Plots to examine the relationship between age and other lifecycle covariates
- Plots to approximate the usage patterns for each of the lifecycle covariates
- Plots to summarize the trends and relationship of continuous testset measures to their limits

Sample outputs from this phase are shown in Fig. 3.

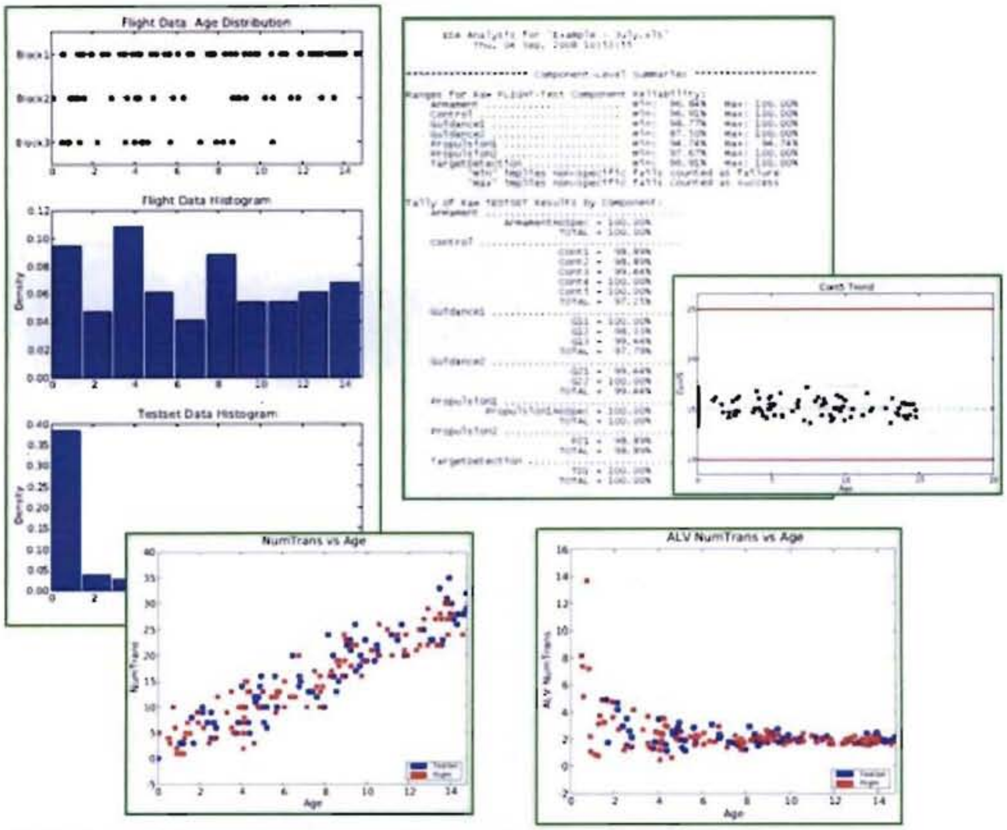

Figure 3: A sample of the numerical and graphical summaries from the EDA phase.

In SRY (System Reliability with YADAS) mode, a complete formal system reliability analysis is run, to estimate all model parameters and generate tabular and graphical summaries of the estimated system, component and testset reliabilities. The following tasks are performed:

- System structure information, flight and testset data are prepared for the YADAS analysis
- Prior information is converted into YADAS-ready form
- The analysis is carried out
- Summary results from the analysis are generated

The reliability summaries from the SRY mode of SRFYDO are in both numerical and graphical form. The values provided are the median of the posterior distribution for the reliability of interest, as well as credible intervals (see Winkler, 2003 for more details). These credible intervals capture the uncertainty of the prediction of the reliability and should be reported with the point estimate for reliability.

DESCRIPTION OF THE PROCESS

The modeling process involves integrating knowledge about the specific system with available data. The approach involves a number of assumptions about the form of the data and the structure of the system. If the assumptions are not met, it is possible to generate nonsensical results. Hence, it is important to treat the early phases of the analysis as essential stages for producing sensible results. The major stages of the analysis are shown in Fig. 4 and are listed below.

Stage 1: Understanding System and Data. In this stage, system-specific information is collected from engineering experts to describe the functionality of the system, identify key components, and how components are connected. This involves determining how available data map to components, as well as identifying and collecting relevant lifecycle measures (including age and usage measures) for all available flight and testset data. If there are several versions of the system, then we can leverage understanding about common components across these variants. For example if we have two variants: "Block 1" and

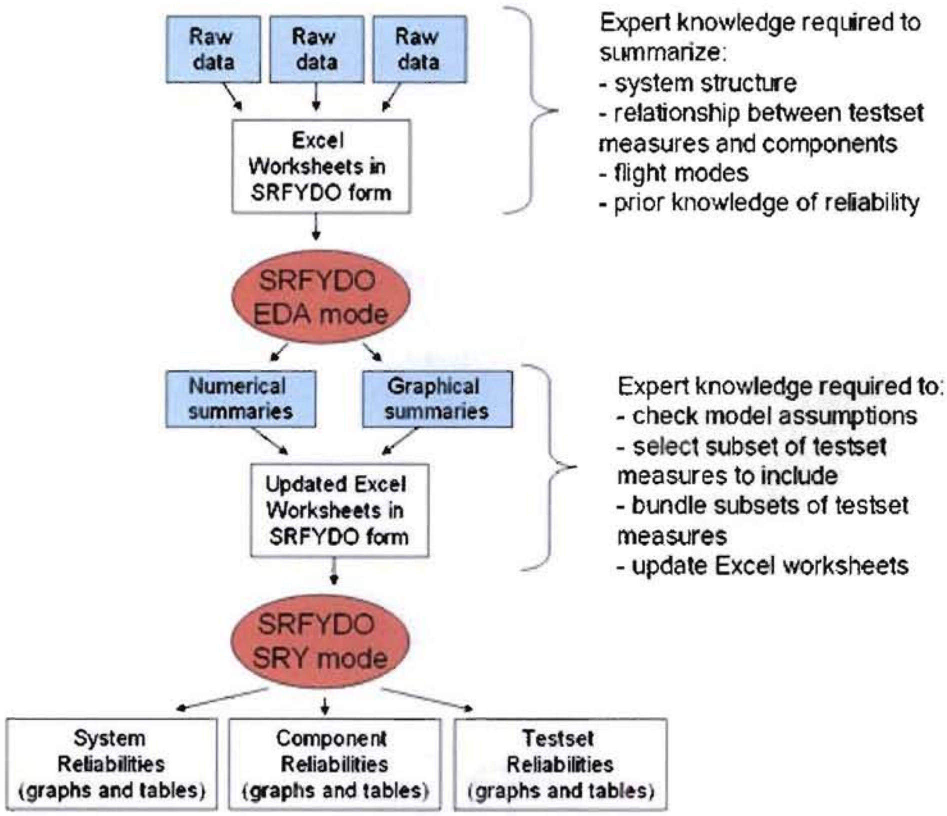

Figure 4: Process flow for using SRFYDO

"Block 2" that share several common components, then the entire set of data for both variants can be used to estimate these common components.

We also capture information about individual components from full-system tests. The assumption of series systems implies that if a successful full-system test is observed, then all of the components in the system worked. The four possible outcomes for each component are:

0 - the component failed
1 - the component worked
2 (or blank) – no information is available about this component from this test
3 - it was one of a group of components where at least one component in the group failed

Finally, the Bayesian analysis requires quantification of existing knowledge about system and/or component reliabilities. This information is based on expert judgment or engineering knowledge, and not on the data that are used in the analysis. If there is not detailed expert knowledge, then diffuse priors can be used to summarize this lack of specific information.

Stage 2: Exploratory Data Analysis. SRFYDO in EDA mode creates summaries and plots that display the raw data in order to look for unusual patterns, errors in coding, and for comparison to model assumptions. Rough numerical summaries at component and system level show reliability estimates from both flight and test set data. If the reliabilities look dramatically different, then this can indicate that data may have been coded or entered incorrectly, or that the inconsistencies between data sources should be investigated. There is also a summary showing lifecycle ranges and correlation between age and each lifecycle variable. Correlation is important to consider because if a lifecycle variable is highly correlated with age, then including that lifecycle variable will likely not help the analysis.

Several plots are also produced. The lifecycle vs. age plot (one for each lifecycle other than age) illustrates the relationship (correlation) between age and that lifecycle measure. Other plots include average lifecycle value vs. age, several kinds of distributions, and scatterplots showing trends. The goal of all these summaries and plots is to help the user check assumptions and select appropriate data for the statistical model.

Stage 3: Evaluating Model Assumptions for the System. SRFYDO is based on a statistical model which depends on certain conditions being met. If these model assumptions do not hold true, then the results (and their interpretation) may not be sensible. A brief discussion of each assumption follows.

1. Series system: if the system works, all components worked. If one component fails, the system fails.
2. Only critical testset measures are included: do not include testset measures that do not directly affect success or failure of the system.
3. Homogeneous population of systems (or we have lifecycle measures to distinguish between sub-populations): if we group sub-populations together, we must have evidence that their behavior is similar.
4. Flight tests are considered the most accurate gauge of system reliability: SRFYDO treats flight tests as the "gold standard."
5. Surrogacy: systems selected for flight and testset tests have similar lifecycle properties and can be sensibly combined into a single analysis. Getting flight test data from deployed missiles while doing component tests on missiles that have never been deployed would violate this assumption.
6. Testset limits correspond to operational limits: we compare testset limits to operational limits and not manufacturing, tolerance, or other kinds of limits.
7. Linear shift as component ages: check the scatterplots (Stage 2) to verify this assumption.
8. Approximately normal data: check the scatterplots for outliers or strange shape to the data. It may be necessary to remove outliers or rescale the data.
9. Only a single operational limit matters for failures: data outside of both upper and lower limits will cause problems for the analysis. One solution is to recast the data as discrete (pass/fail).
10. Lifecycle covariates are not highly correlated: if LC1 (a non-age lifecycle variable) is highly correlated with age, then including LC1 in the analysis will likely not help, and may lead to increased uncertainty in the analysis. Similarly, if LC1 and LC2 are highly correlated, both should not be included.
11. Failures from different testset measures are independent: if a single failure mode causes several testset measures to fail, we would be double-counting some failures.
12. System failures result from a single testset measure being out of limits: SRFYDO does not support the type of failure that could arise from several components (measured by different testset limits) showing degradation in performance.
13. Priors based on expert (engineering) knowledge: priors should be based on information separate from the data used in the current analysis. Using data from the system being evaluated to set priors would be double-counting that data.
14. Realistic priors: The ranges for the priors reflect honest expectation of system performance.

The second part of this stage involves determining the appropriate data to include and deciding the functional form for testset measures that will be included. This may involve converting some continuous measures to pass/fail; splitting some datasets into continuous and pass/fail parts; removing some testset measures; and bundling testset measures into component-level summaries. In all cases, these modifications are made so that the data used in the analysis meets model assumptions.

Stage 4: Determine Output Choices for Final Summaries. Choices available to the user include specifying range of ages and lifecycle measures to be considered for the graphical summaries, identifying age and lifecycle measure values for final tables, specifying percentiles of the posterior distribution to be used, and choosing types of summaries to be created. When selecting maximum age for SRFYDO predictions, we recommend going no more than 50% beyond the end of the existing data. Users can choose to have SRFYDO produce additional detailed summaries and plots that display convergence properties of the MCMC analysis.

Stage 5: System Reliability with YADAS. In SRY mode, system, component and testset level reliabilities are calculated and presented. SRFYDO pre-processes the spreadsheet data into YADAS-ready form, initiates the analysis to generate model parameter estimates, and post-processes model parameter estimates to generate summaries. Users have the option of (re-)running any of the three SRY phases: SRY1, SRY2, or SRY3. Phase #1 (SRY1) consists of parsing the Excel spreadsheets and preparing input files for the YADAS-driven simulation; results from previous runs are discarded at the start of this phase. The computationally expensive Monte Carlo simulation is performed during phase #2 (SRY2); this phase might be rerun using a different number of iterations, or to recover from a machine crash or shutdown. The final computation of reliability and its visualization is performed during phase #3 (SRY3). A sample of the outputs from Stage 5 is shown in Fig. 5.

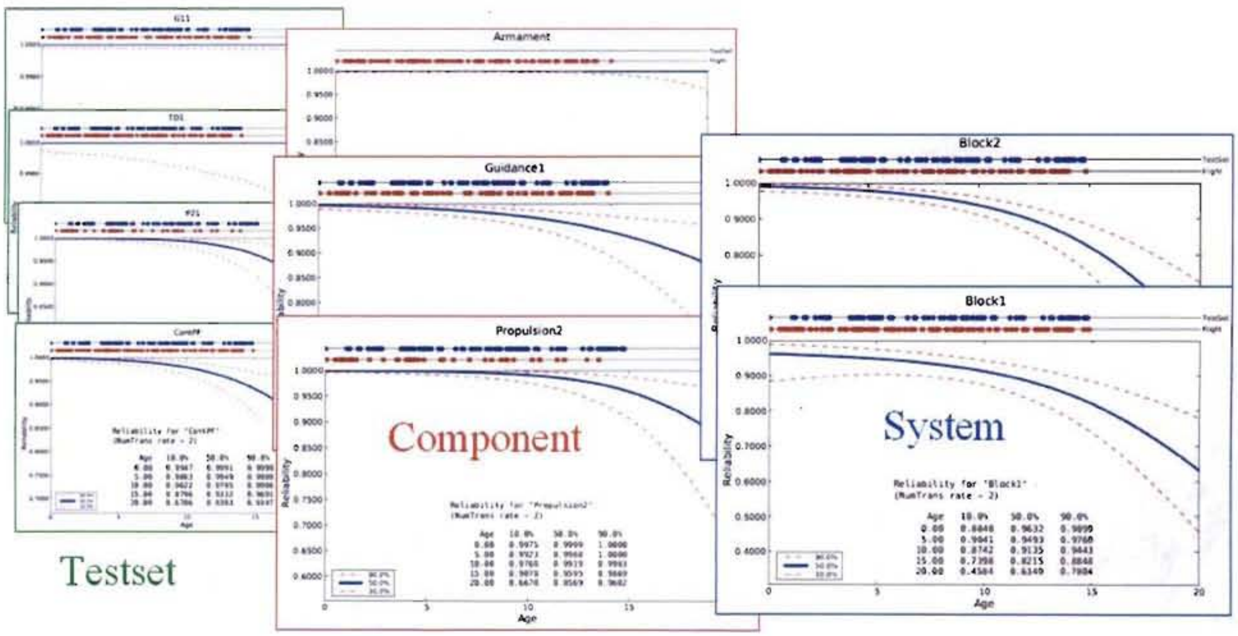

Figure 5: A sample of outputs from SRY phase of SRFYDO

INTERPRETING SRFYDO OUTPUT

SRFYDO produces a log file that records and summarizes all the input. In EDA mode, these summaries are used to check assumptions and make any necessary changes to the input files. Other EDA output consists of plots similar to Fig. 6.

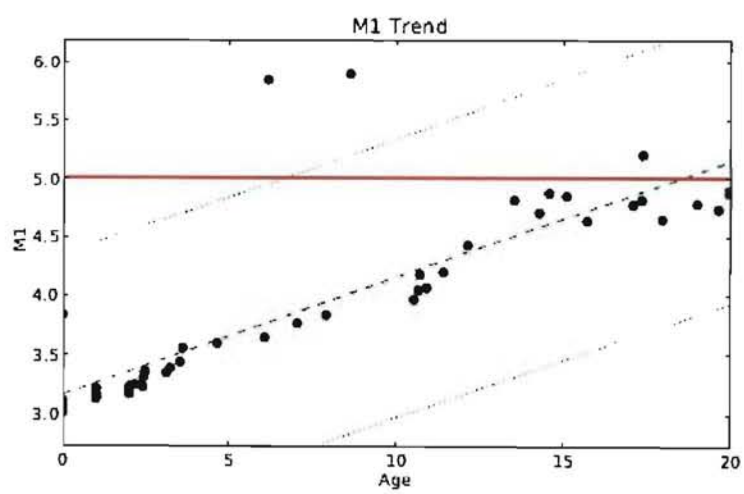

Figure 6: M1 trend.

Figure 6 shows measurements for component M1 from ages 0 – 20. The horizontal line at 5.0 indicates the upper limit for this component. Three measurements exceed the upper limit. Two of those would be considered outliers, because they do not match the pattern of the rest of the data. One measurement at age 17.5 also exceeds the upper limit, but it is consistent with the pattern of the rest of the data. Before running SRY mode, the user would have to modify the input by removing the outliers and converting them to a new discrete (pass/fail) measure. The rest of the data is appropriately modeled as continuous with an upper limit of 5.

Standard output from SRY mode consists of plots such as the system reliability plot shown in Fig. 7 below. The two lines above the graph show the ages at which we had testset and flight data. The solid line between the dashed lines gives the median value for our reliability estimate. The dashed lines are the 10th and 90th percentiles of our estimate for reliability of this system. Note that the spread of the dashed lines increases substantially after age 20. That reflects the fact that as we go beyond the age where we have data, our uncertainty in the prediction gets much larger.

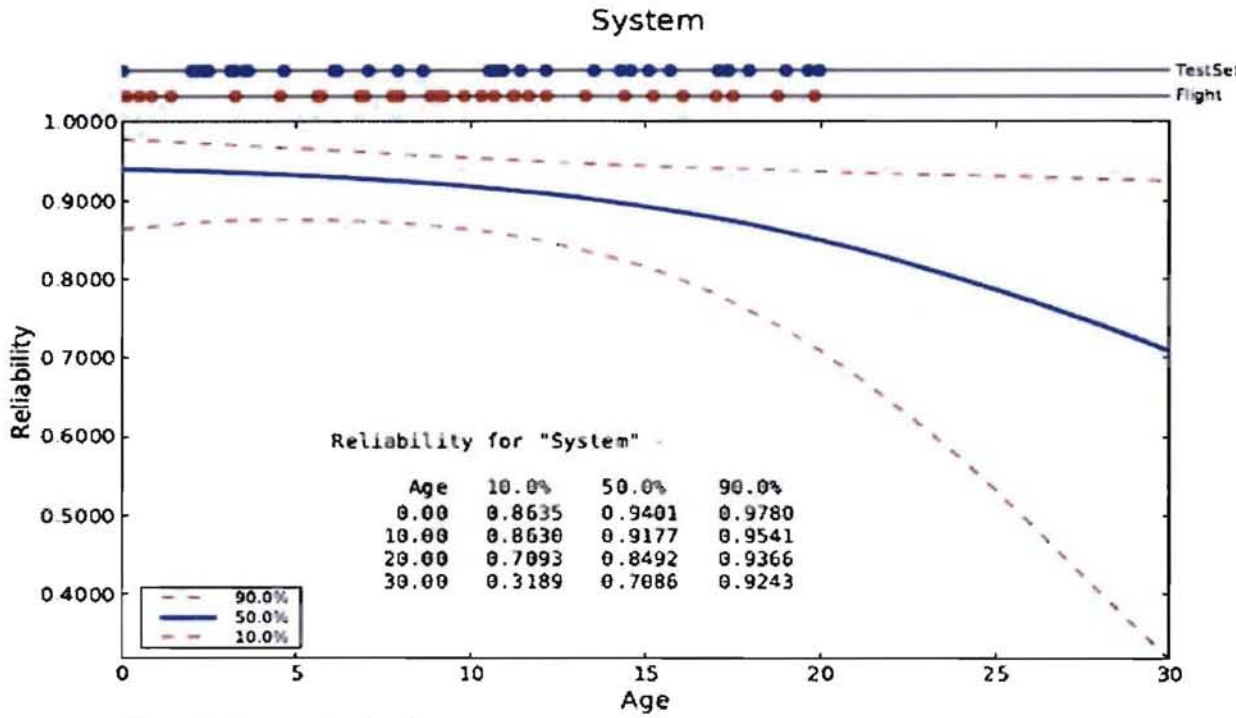

Figure 7: System Reliability

SUMMARY AND CONCLUSIONS

We have described the process of using SRFYDO to estimate the reliability of complex systems. The advantage of using SRFYDO comes from the capability to use information from sources other than flight tests, such as testset data, engineering judgment, and data from similar systems. Hence, this approach of data combination has following advantages:

- It uses relevant data already available to predict reliability
- It provides a check on consistency of information from different sources
- There is flexibility to incorporate partial information into model, if there is some ambiguity in diagnosing failures to a particular component
- By following the trends in continuous measures of condition, we gain the ability to predict failure before being observed in full-system test
- By modeling at the component level for reliabilities, we are able to identify key drivers of reliability, compare replacement parts to their predecessors, and leverage data from related version of systems which share common components.

We have validated SRFYDO output through multiple (hundreds) of simulations with known underlying reliability values based on a variety of:

- System architectures: # of components, specs, variants
- Data structures: mix and number of testset vs. flight data points; uncertain knowledge of single vs. multiple failures
- Other input factors: priors, # of iterations, distribution parameters, etc.

Additionally, we have compared SRFYDO output against results from other approaches using only a portion of the data. In all cases we find excellent agreement between SRFYDO results and results using standard methods.

SRFYDO is currently being used for reliability estimation of munitions systems by several branches of the U.S. military. For details or more information, including the possibility of introductory training classes, write to SRFYDO@lanl.gov.

REFERENCES

1. Anderson-Cook, C.M., Graves, T., Hamada, M., Hengartner, N., Johnson, V., Reese, C.S., Wilson, A.G. (2007) "Bayesian Stockpile Reliability Methodology for Complex Systems". *Journal of the Military Operations Research Society* 12: 25-37.

2. Anderson-Cook, C.M., Graves, T., Hengartner, N., Klamann, R., Wiedlea, A.K., Wilson, A.G., Anderson, G., Lopez, G. (2008) "Reliability Modeling using Both System Test and Quality Assurance Data". *Journal of the Military Operations Research Society* 13 5-18.

3. Casella, G., and George, E. (1992), "Explaining the Gibbs Sampler," *The American Statistician*, 46, 167-174.

4. Chib, S., and Greenberg, E. (1995), "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, 49, 327-335.

5. Gelfand, A.E., and Smith, A.F.M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398-409.

6. Graves, T.L. (2007), Design Ideas for Markov Chain Monte Carlo Software. Journal of Computational and Graphical Statistics 16:24-43.

7. Wilson, A.G., Graves, T.L., Hamada, M.S., Reese, C.S. (2006) "Advances in Data Combination, Analysis and Collection for System Reliability Assessment". *Statistical Science* 21: 514-531.