|  |
| --- |
| Sustainment in Deployment Optimization Tool (SiDOT) |
| EMIS 7305 Final Report |
|  |
| **Evan McCaleb, Jesse Mooney, Daniel Stovall** |
| **December 4, 2015** |

Table of Contents

[1 Executive Summary 3](#_Toc436753816)

[2 Introduction 4](#_Toc436753817)

[2.1 Need 4](#_Toc436753818)

[2.2 Problem 4](#_Toc436753819)

[2.3 Objective 4](#_Toc436753820)

[2.4 Approach 5](#_Toc436753821)

[3 Ground Rules & Assumptions 6](#_Toc436753822)

[4 Analysis/Study/Evaluation 7](#_Toc436753823)

[5 Results 8](#_Toc436753824)

[6 Summary & Conclusions 9](#_Toc436753825)

[7 References 10](#_Toc436753826)

[8 Appendix A – Project Plan 11](#_Toc436753827)

[9 Appendix B – Project Presentation 11](#_Toc436753828)

**(INSERT TABLE TO FIGURES)**

# Executive Summary

# Introduction

Almost every company utilizes some sort of database or program that manages inventory, maintenance, documentation, or for archiving historical records. Oftentimes, companies are so dependent on these complex databases that they come to a work stoppage whenever there is an issue with the database. There are always alternative options set in place such as the use of paper documentation; however, sometimes it is better to wait on the database to be repaired due to the extra time required to manually input the data back into the database once it becomes available. Most of the more pronounced companies alleviate this issue by having multiple databases where the company isn’t at a 100% work stoppage. The problem with this risk mitigating step is that there is not a good alternative for a backup supply inventory system. Once the supply database goes down there is no effective way in determining accurate parts quantity and inventory, and this becomes even more of a concern when the database covers multiple locations or sites. For the same reason there is also a concern when a properly working database cannot be relied on due to input error, poor tracking or accountability.

In this project we took this idea and applied it to a sustaining engineering scenario. It extremely difficult for engineers to build repairs for customers without knowing what materials they have available, or materials that they can obtained quickly, in a quick manner in order to keep availability high and repair time low with databases that are unreliable. We also took upon the initiative to improve the entire situation by not only providing a more efficient parts inventory system but also taking the engineers out of the equation completely. This improvement will drastically improve availability, reliability, and decrease maintainability time.

## Need

Develop a tool/database that will aid in the determination of the optimum solution of a simple sustainment repair problem in the program deployment phase. The initial tool is tailored to the Lockheed Martin F-35 program, but it may be easily customized to other programs. The tool compares multiple sustainment solutions for a given problem/discrepancy and provides a direct comparison of the different solutions in terms of cost, schedule, and performance impact of each solution. The comparison will then allow the customer the option of choosing their preferred solution based on their specific reliability and available needs.

## Problem

F-35 repair and modifications are often needed to effectively maintain deployable field assets in order to meet program reliability and availability requirements. However, multiple solutions can often be implemented to solve a given problem. Due to the unknowns of local part availability, support equipment, maintainer experience for types of repairs, and accurate urgency requirements it is often difficult for sustainment engineers to provide quick repair solutions. Therefore, a need exists to systematically analyze the optimal solution depending on material availability and operational need.

## Objective

Deliver a user-friendly tool that takes a wide variety of F-35 minor product anomalies and outputs an approved solution that fits the optimal solution for cost, and availability. Provide a tool program that is easy to learn, maintainable, and usable by many customers at multiple locations. Supply the best logistical, most engineered solution in an expedited time frame in order to increase product availability. The tool will essentially decrease product downtime due to the normal period of time it takes to obtain an approved repair disposition, since the logistical constraints such as materials, new components, consumables, required hazmat, and maintainer experience levels will all be known.

## Approach

A software tool capable of simulating fleet availability with multiple

# Ground Rules & Assumptions

The following is a list of ground rules and assumptions for the analysis:

* Problems are all known
* Tool considers problems that are repairable
* All consumables are readily available
* All required support equipment is available and reliable
* All repaired elements have the same reliability
* All aircraft are at the same location
* Aircraft have single discrepancies with single solutions at one time
* Each aircraft has an identical exponential reliability curve
* The list of materials in the inventory are not replaced over the time period of program analysis
* Aircraft has background reliability curve in series with known problem reliability curve
* All aircraft are currently treated as if they fly a fixed average number of hours per day
* All aircraft are currently treated as if starting with same number of hours on the airframe
* Aircraft currently resume exact same curve after repair from baseline aircraft reliability failure
* Each aircraft is examined individually by the code so most of these assumptions could be easily loaded from a database with ~500 lines additional code per item stated above

# Analysis/Study/Evaluation

# Results

While the case study below provides a simple demonstration of the analysis required for an F-35 squadron, the primary product of this project is the tool itself. Section X describes the tool from the perspective of capabilities and user interfaces, and Section X.2 provides an example wherein the tool was used to analyze different repair options for a damaged panel on an F-35.

# Final Tool

SiDOT was intended to be operated by the user through a graphical user interface (GUI), which allows the user input a problem into the tool efficiently and effectively. Per the project plan, the delivered version of SiDOT is a demonstration version. However, the vast majority of the final capability is included in the demonstration version.

The SiDOT GUI was developed using the MATLAB GUI Editor. The GUI has multiple windows, which are prompted by user selections. The primary problem inputs are located in the main GUI window, which is presented in Figure 1. This main GUI window contains inputs for the discrepancy description, reliability model inputs, and output options.

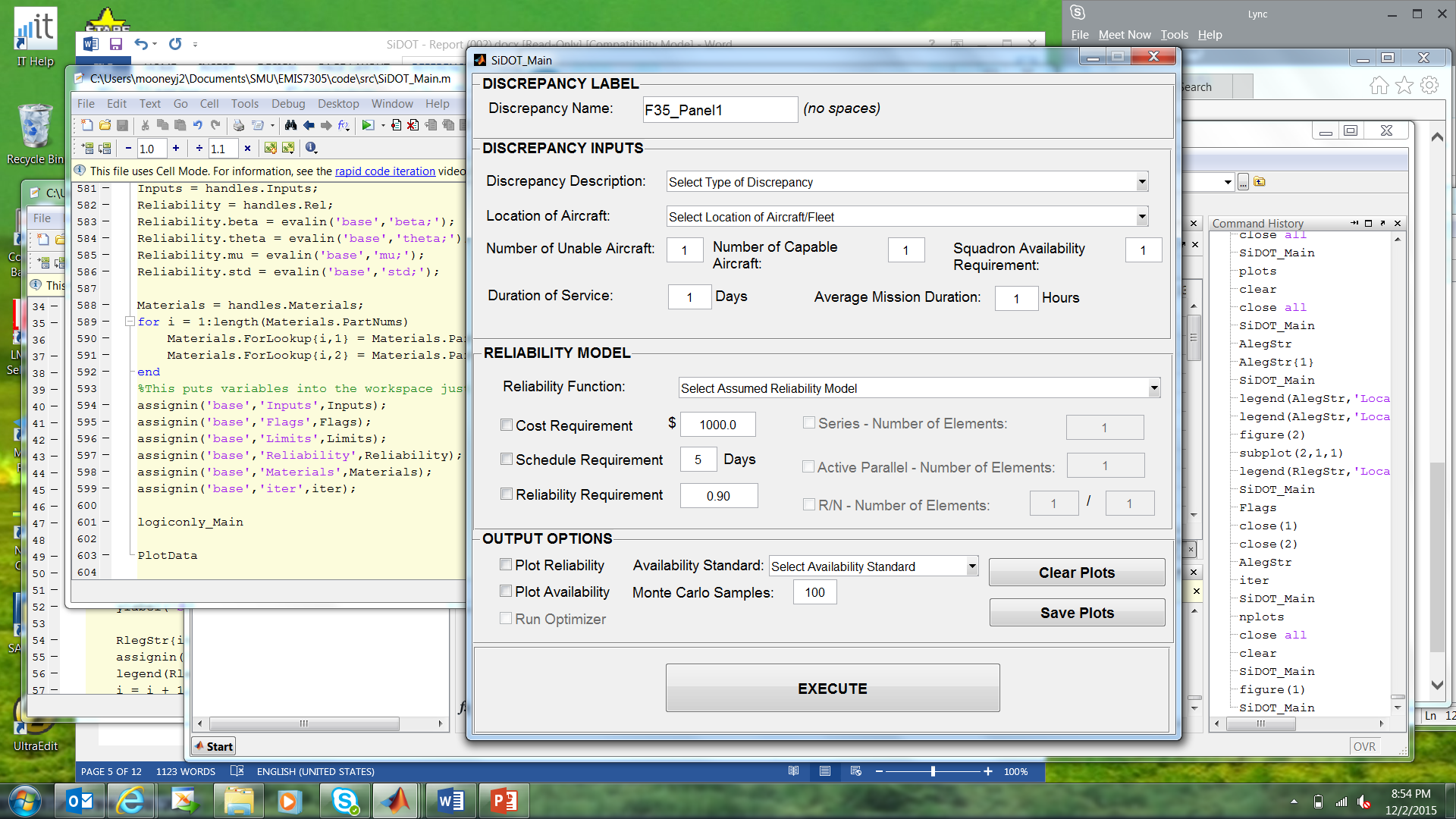


Figure : SiDOT Main GUI Interface

In the DISCREPANCY INPUTS section, the user must select from a list of common discrepancies or define a custom discrepancy. Each discrepancy is linked to a list of materials required for that type of repair and an expected time for the repair to complete. For a custom repair, the user must manually input these variables manually using the Custom Discrepancy GUI (Figure 2), which is prompted by the selection of custom discrepancies.

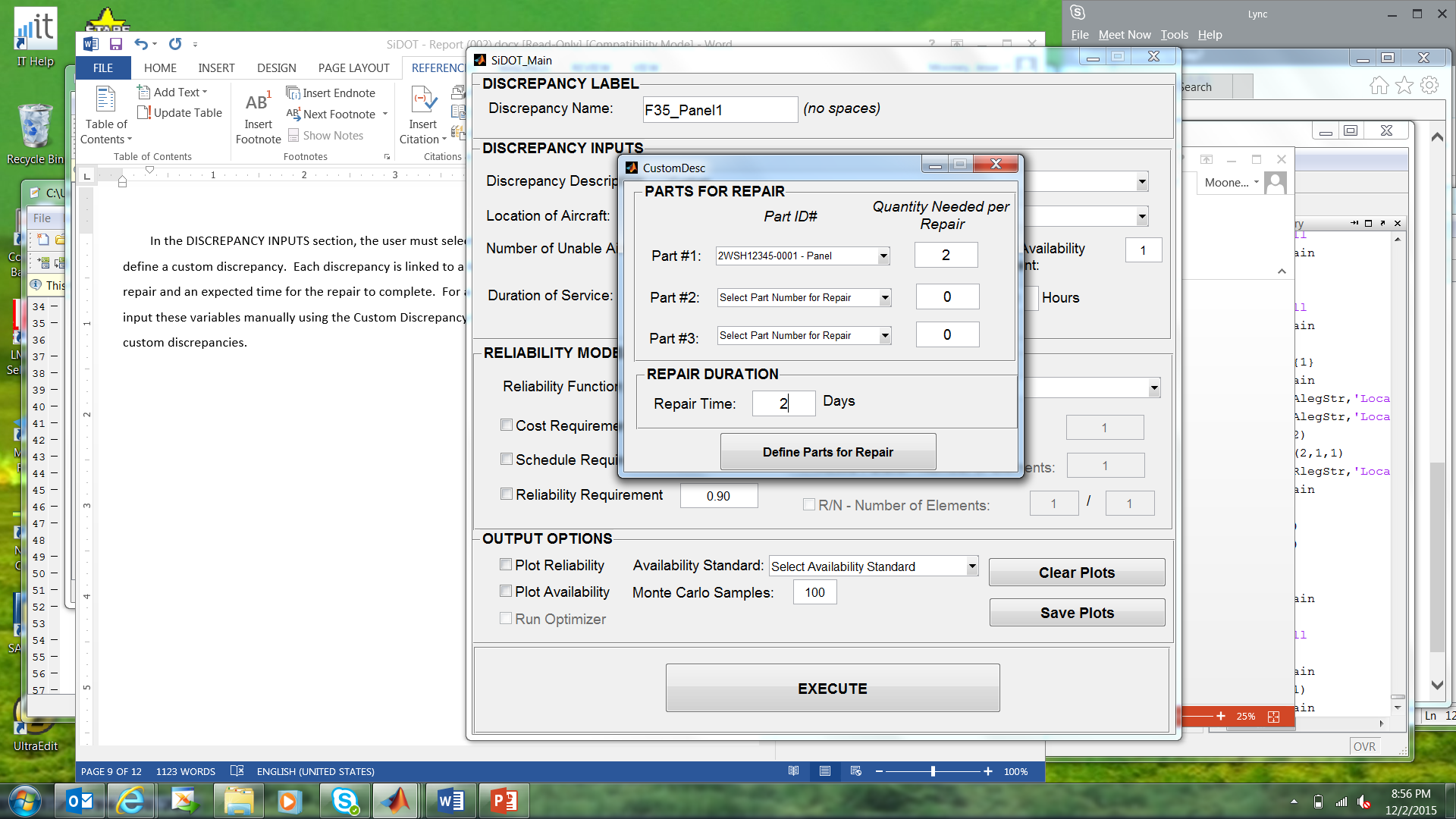


Figure : Custom Discrepancy Inputs

Within The discrepancy input section, the user must also input the number of aircraft to be repaired, number of working aircraft in the squadron, minimum availability requirement for the squadron to carry out missions, duration of concern, and average mission duration.

The user must then define reliability inputs for the repair. Reliability models can be defined using exponential, normal, lognormal, and Weibull functions. Once a function is selected, the user is prompted to input the corresponding parameters for the selected function in a separate GUI window. Figure 3 demonstrates the user inputs for a normal distribution, where the other functions are disabled. Also included in the RELIABILITY MODEL section are user-defined constraints, which will be used in the output plots. Note that the serial, active parallel, and R-out-of-N inputs are greyed out due to being omitted for the demonstration version of SiDOT.

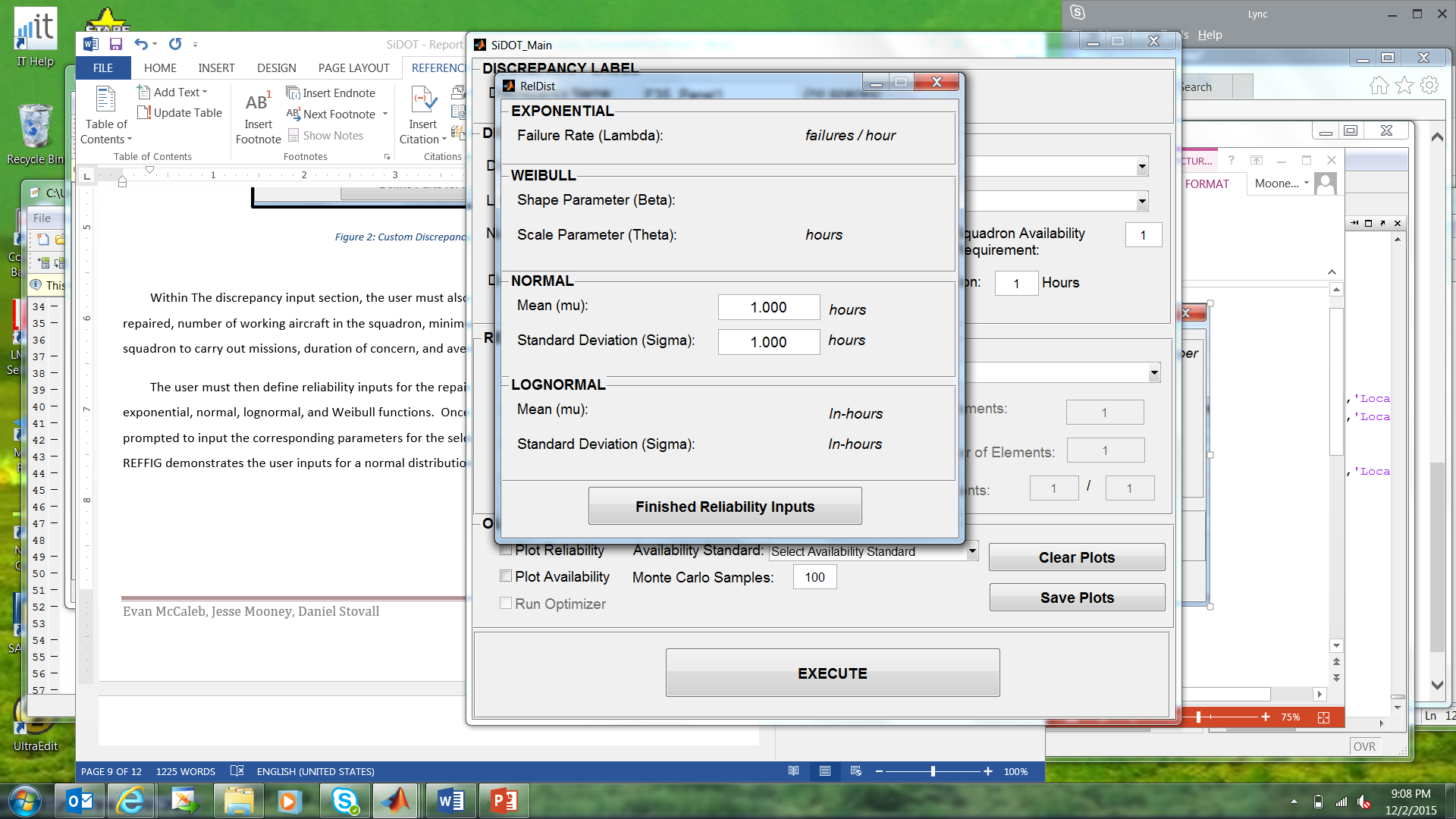


Figure : Reliability Function Input Window

Finally, the user must define the desired outputs in the OUTPUT OPTIONS section. Here the output plots are determined, with reliability and availability plots being optional. The user must also define the Availability Standard by which the output should be judged, where options include the mean or various probabilities. The user must also define the number of Monte Carlo simulation runs, although the default 100 is recommended to provide a large enough sample size without overrunning the memory. At any time the user may clear the plots and start over using the “Clear Plots” button.

Once all inputs have been completed, the user may begin the analysis by clicking the “EXECUTE” button. This button begins the simulation and plots the results based on the stated user desires. The simulation output plot will always appear, which presents the results of each Monte Carlo simulation run in terms of aircraft availability per day along with the minimum required capable aircraft. Depending on user selections, a second plot may be generated showing the reliability curves of the baseline aircraft and the repaired part, as well as the availability as a function of the user-defined standard or percentile. These plots are presented for the case study below.

The user may change any of the available inputs and replot using the main GUI window. The resulting plot will add the newest case to the previous plots for comparison, although the user may clear the plots and start over using the “Clear Plots” button.

# Damaged Panel Case Study

A common reported discrepancy on the F-35 has been damaged panels, often due to mis-drilled rivet or fastener holes. The repair is common and an approved method is generally accepted, which made this case particularly fitting for analysis by the SiDOT tool. The damaged aircraft and the local parts warehouses were both assumed to be collocated in the United States, which negates the need for cost and schedule penalties for overseas shipping. It was also assumed that 4 aircraft out of a squadron of 30 aircraft had reported this damage, while 28 aircraft were required to keep the squadron at operational strength.

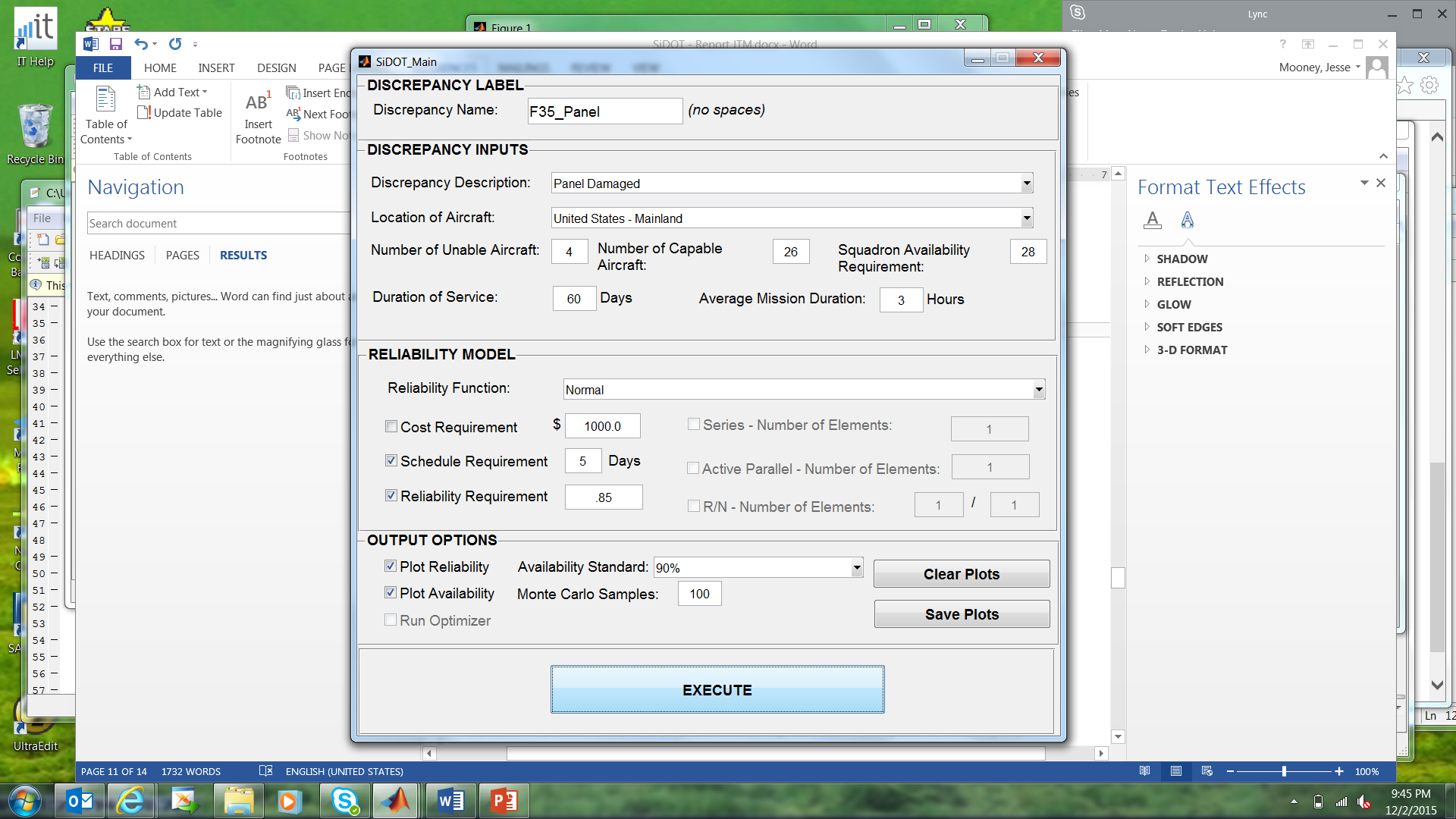


Figure : F-35 Panel Case Study Inputs

From previous data (created in place of actual F-35 data), it was assumed that the repaired panel had a normal distribution, with a mean of 20 flight hours between failures, and a standard distribution of 6 flight hours. SiDOT assumes this reliability function is applied in series with the overall aircraft reliability. The remaining inputs are presented from the SiDOT main GUI in REFFIG. It was assumed that the squadron has a schedule requirement of 5 days until the squadron must meet the availability requirement. The first attempt at the analysis desired a 90% availability standard. The results of the Monte Carlo simulation is presented in Figure 5, and the reliability and availability plots are presented in Figure 6.



Figure : F-35 Case Study Monte Carlo Results - Normal Distribution



Figure : F-35 Case Study Reliability and Availability Outputs

It can be seen both in the Monte Carlo results and the 90th-Percentile analysis that the squadron does not meet the availability requirement, regardless of schedule. A second look at the mean availability in Figure 7 shows that the availability requirement is only briefly met.

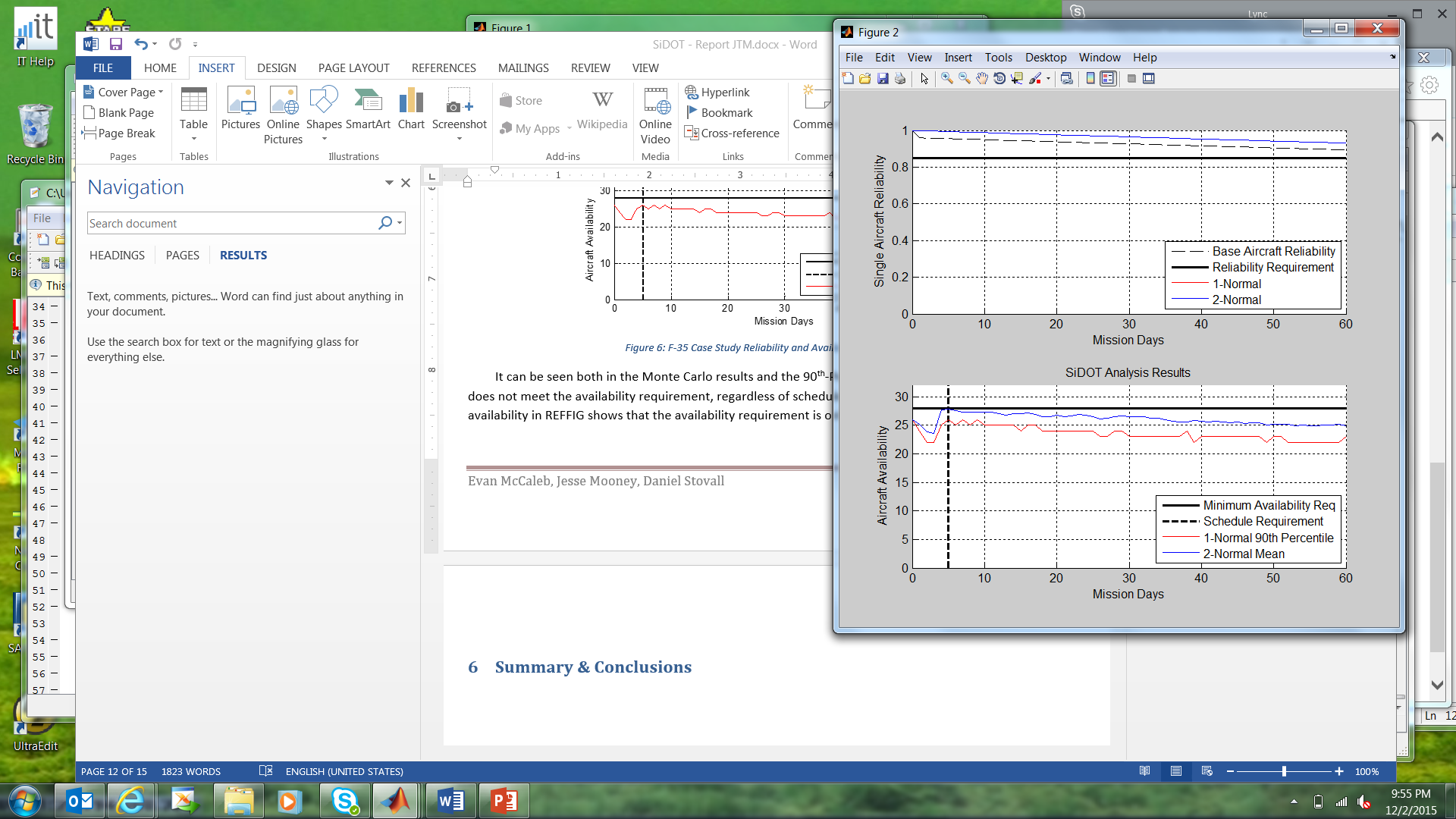


Figure : F-35 Case Study Mean Availability

At this point, the program would have multiple options including changing the repair type, lowering the availability limit, or supplying additional aircraft. In this case, it was determined that the 90th Percentile was critical. At this point, it was determined…

# Summary & Conclusions

# References

# Appendix A – Project Plan



# Appendix B – Project Presentation