



## Calhoun: The NPS Institutional Archive

---

Faculty and Researcher Publications

Faculty and Researcher Publications

---

2007-03

# Team 12: Joint Dynamic Allocation of Fires and Sensors: Experimental Interface and Analysis

Ahner, D.

---



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

# Team 12: Joint Dynamic Allocation of Fires and Sensors: Experimental Interface and Analysis

## TEAM 12 MEMBERS

D. Ahner, LTC - Lead, Contact \*  
U.S. Army Training and Doctrine Command Analysis  
Center - Monterey, USA

A. Buss  
J. Freye, LT  
A Peters, CPT  
Naval Postgraduate School, USA

## INTRODUCTION

The Joint Dynamic Allocation of Fires and Sensors (JDAFS) simulation is a publicly available discrete event simulation that accounts for first order combat effects using Army approved algorithms. It couples dynamic allocation of resources such as unmanned platforms and artillery assets using optimization to a simulation to render better representations of network enabled warfare. The current configuration is not user friendly when entering the DOE factors. This is problematic for a number of reasons. First, the potential for data entry errors is significant when entering a large volume of numbers. Second, the data entry is not only time consuming but potentially expensive. Finally, an operator/analyst needs to be present as the runs are completed in order to start the next design point evolution. The potential for errors and the inherent inefficiencies warrant the development of a method to easily run a DOE if JDAFS is to be more widely used.

## JDAFS

The first team goal was to refine and test a Joint Intelligence, Surveillance, and Reconnaissance trade-off analysis scenario in JDAFS. As an initial test of functionality, the most complex ISR scenario loaded and ran. The first optimization interval took over an hour to build before the scenario could begin. The team decided to implement a much simpler scenario (240 hours) which ran to completion in a matter of minutes. A number of output and execution issues were initially noted and corrected in the latest

software build. The workshop was valuable for revealing scenario, data entry, and program anomalies for correction prior to the pending production runs.

The purpose of this research is to use the JDAFS model to simulate the effects of varying key factors associated with the operation and employment of manned and unmanned ISR platforms. An analysis of the trade-offs associated with specific platforms and basing decisions on the execution of penetrating vs. non-penetrating missions will be the outcome and main focus of this research. The analytical findings will provide a foundation for decisions regarding ISR.

## Design of Experiments

A Design-of-Experiments (DOE) will be developed based on scenarios and data provided in order to execute the Joint Dynamic Allocation of Fires and Sensors (JDAFS) simulation model. In establishing the DOE, specific factors will be identified that determine the key attributes of timely, survivable, and persistent. Once the factors have been identified, the range of levels to be taken on for the various factors will be set for examination within the DOE framework. Due to the number of factors involved, a full-factorial design, examining every possible combination of inputs would be impractical. Therefore, an efficient, robust DOE will be employed to examine the response surface. Output from the simulation runs will be analyzed to evaluate how the various input factors affect the predetermined Measures-of-Effectiveness (MOE) across the range of scenarios.

The platforms and key performance parameter used are shown in Table 1.

PLATFORM	MAX ALTITUDE (meters)	SPEED (meters/hour)	ENDURANCE (hours)	TRANSITION TIME (hours)
RQ-1	7925	148160	24	1.5
RQ-4	18288	629860	24	1.5
P-3	8839	422256	12	1.5
RC-135	10668	805629	11	1.5
U-2	21336	824140	10	1.5

Table 1 - ISR Platform Performance Parameters

\* For more information contact: [darryl.ahner@us.army.mil](mailto:darryl.ahner@us.army.mil)

Each platform had varying sensor capabilities depicted in Table 2.

PLATFORM	SENSOR	RANGE (meters)
P-3	EO/IR	75000
P-3	SAR	200000
RC-135	ELINT	300000
RQ-1	EO/IR	50000
RQ-1	SAR	200000
RQ-4	EO/IR	300000
RQ-4	SAR	75000
RQ-4	ELINT	200000
U-2	EO/IR	400000
U-2	SAR	100000
U-2	ELINT	250000

Table 2 - ISR Platform Sensor Ranges

### Scenarios

Two types of scenarios were generated for examination and comparison with JDAFS. The first scenario is a non-penetrating scenario (See Figure 1) where the ISR platforms do not penetrate the Country of Interest's national airspace.

The internationally accepted buffer of 22 kilometers is respected on all flights and waypoints have been implemented to prevent ingressing and egressing aircraft from violating the COI's sovereign airspace. The second scenario (See Figure 2) assumes that conditions have changed to allow the violation of the COI's airspace. With the incursions into the COI's territory comes the

risk of engagement by air defense assets, in this case surface-to-air missiles (SAM).

For the purposes of this simulation 25 missions or targets of varying types were created and populated through out the COI. Target types consisted of command and control (C2), surface-to-air missiles (SAM), short range ballistic missiles (SRBM), medium range ballistic missiles (MRBM), long range ballistic missiles (LRBM), airfields, weapons of mass destruction facilities (WMD FAC), military facilities (MIL FAC), and ammunition storage (AMMO STOR) facilities. Each type of target can be collected against by specific types of sensors. The mission/sensor interactions must be explicitly created in the input

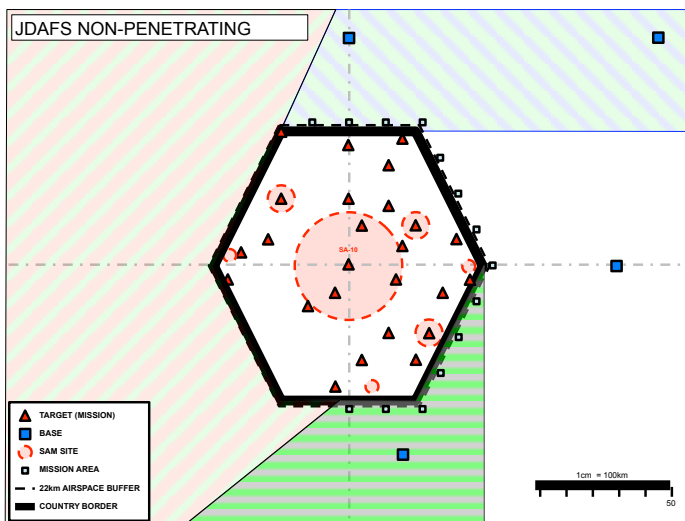
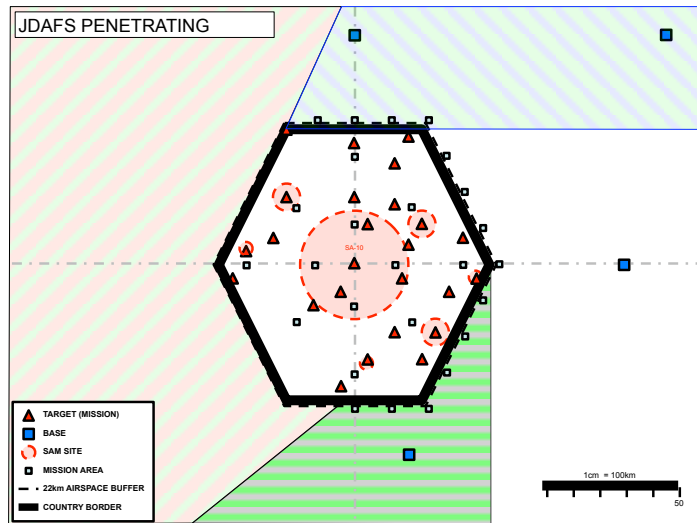


Figure 1. JDAFS Non-penetrating Scenario

tables and result in the actual creation of 76 target elements from the 25 actual missions (targets).

### Data Farming Interface

The second team goal was to develop a data farming interface (or at least requirements for one) that lends itself to analyst ease of use. A number of interesting design considerations for the design of experiment tool arose during conference discussions within the working group and with members of other teams. These discussions included whether or not a baseline scenario should be required; including a set of parameters with their types and descriptions as candidate DOE factors as a part of any DOE tool; guiding a user and allowing robust design

selection; design point generation; and experiment distribution practices to a computing cluster.

MISSION	TYPE	EO	SAR	ELINT
1	C2			X
2	SA-2	X	X	X
3	WMD FAC	X	X	
4	IRBM	X	X	X
5	AIRFIELD	X	X	X
6	SRBM	X	X	X
7	C2			X
8	C2	X	X	X
9	SA-2	X	X	X
10	SA-2	X	X	X
11	C2			X
12	AMMO STOR	X	X	
13	AMMO STOR	X	X	
14	IRBM	X	X	X
15	LRBM	X	X	X
16	SRBM	X	X	X
17	MIL FAC	X	X	
18	WMD FAC	X	X	
19	MIL FAC	X	X	
20	MIL FAC	X	X	
21	IRBM	X	X	X
22	MIL FAC	X	X	
23	C2			X
24	AIRFIELD	X	X	X
25	SA-10	X	X	X

Table 3 - Targets and Required Sensors for Collection

For JDAFS, a method was developed so that the analyst can specify which input factors are to be varied, match those factors with a design, and generate the corresponding input database. For each design point, the DesignGenerator program uses SQL statements to create a corresponding input file. JDAFS writes its output reports to the same database that was used for input, so there is always a direct association between input and output values.

Execution of JDAFS is oriented towards a single run with a single input file; there is no built-in capability in JDAFS itself to run experimental designs. Yet executing JDAFS with various input factors set

based on efficient designs is essential to its effective use. It is also critical to being able to use JDAFS for data farming. The normal input to a JDAFS scenario is currently via an Access database; in the near-term, support for other databases such as MySQL, Oracle, and Derby will be added.

A method was developed so that the analyst can specify which input factors are to be varied, match those factors with a design, and generate the corresponding input database. This was done so that there was no impact whatsoever on the basic input database, and so that the specific experimental design used was completely swappable. That is, the analyst could keep the same basic input scenario and same set of factors, but swap in any experimental design scheme that was desired with only a minimal amount of changes.

Given a database input format and instance (scenario), any possible data element in the database is a potential design factor. Practically speaking, however, for a model such as JDAFS, large groups of data elements together can constitute a single factor. For example, the maximum range for a particular type of sensor is one possible factor. Setting that factor requires more than changing a single value; all “max\_range” columns corresponding to that sensor must be changed to correctly set the factor to its correct level.

To have a concrete use-case, we chose the nearly-orthogonal Latin hypercube design implemented by the spreadsheet by Susan Sanchez (Sanchez, 2007, [http://harvest.nps.edu/LinkedFiles/NOLHdesigns\\_v4.xls](http://harvest.nps.edu/LinkedFiles/NOLHdesigns_v4.xls)). For simplicity, the only types of factors initially considered were continuous and integer. The values from the spreadsheet for each set of factors were normalized to a range of [0, 1] and saved in plain-text comma-separated (csv) files. Conceptually, these files constitute tables in a virtual database (and future versions will be implemented as such); therefore, we represent these tables of values as a database called “Designs.” Different schemes can be used to generate these tables of values without modifying any other part of the input or the code.

The Factors database consists of two tables: one to identify factors in the template database by table and column, together with a user-specified minimum and maximum value, and a second table that maps the number of factors (determined by the number of rows in the first table) to one of the design point files. That table is then read, and the corresponding values

applied to the minimum and maximum values to produce an input value. The "DesignGenerator" program that does this is written in Java and uses Java Database Connectivity (JDBC) to access the databases. Hence, using different databases will only require that a different JDBC driver be installed.

For each design point, the DesignGenerator program uses SQL statements to create a corresponding input file, which is numbered <name>xxxx.mdb, where "xxxx" is an index corresponding to the particular design point index. All the input from the template database is copied, and those entries that are factors are modified using the SQL UPDATE query to be set to the design point values. An additional table is written that identifies the particular names and values of the input factors. This information will be used after JDAFS is executed to extract the independent and dependent variables for statistical analysis.

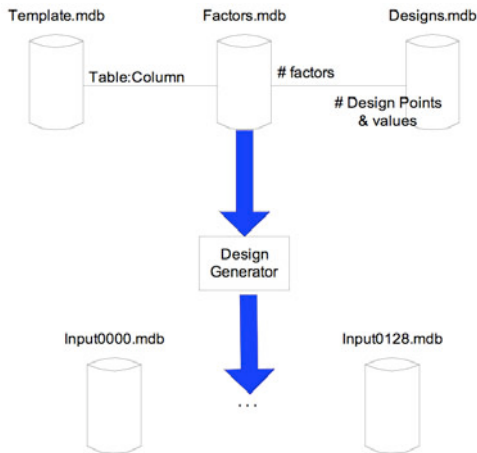


Figure 3. DesignGenerator Producing Input Files

JDAFS can write its output reports to the same database that was used for input, so there is always a direct association between input and output values.

The next step is to take the set of databases containing input and output and process them to produce a file or database suitable for statistical analysis; that is, in a suitable input format for a statistical package such as JMP. This will essentially be the reverse of the flow in Figure 3.

## RESULTS

The JDAFS simulation provides for the generation of a number of Measures of Effectiveness including, Coverage, Coverage by Type, and Attrition. Sensor coverage is the focus for this analysis. An initial plot of

the distribution and summary statistics for 274 design points in the non-penetrating scenario is shown in Figure 4. The coverage results appear to be roughly normally distributed with a mean of 0.48. The five outlying data points at the bottom of the outlier box plot warrant further examination.

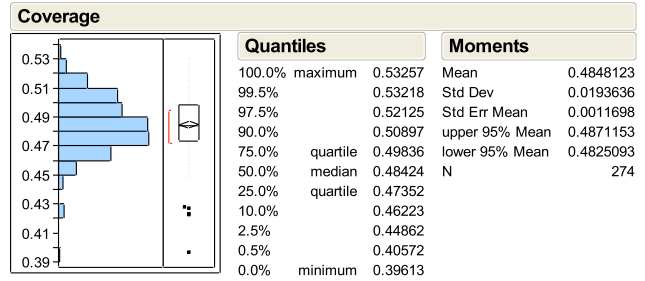


Figure 4. Distribution and Summary Statistics of Mission Area Coverage

An examination of the data reveals that four of the outlying data points 1, 2, 4, and 265 are the result of design points having only a single base available for operations. Even though each operating location had a full compliment of aircraft (21 total airframes), the mean coverage ranked at the bottom of all the results. When ranked from greatest to least coverage, D, C, A, then B, this ordering is not surprising when compared to the mean distance of the bases to their mission areas in the pre-simulation analysis. Having a single operating base available to ISR platforms in a region of the size encompassed in this study is unrealistic. Therefore, the design points that include only one operating location will not be considered for the remainder of this analysis.

For comparison with future models, a full quadratic model with the outlying data points was constructed. Figure 5 shows this model. Note that the R-squared value for this model is 0.73.

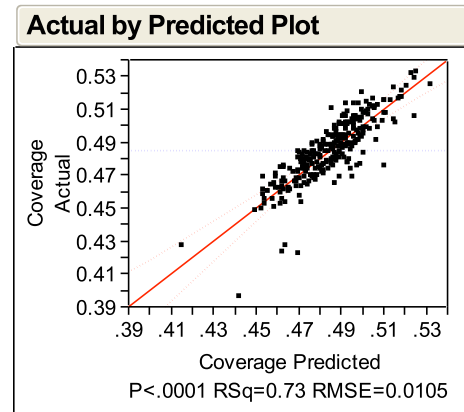


Figure 5. Actual by Predicted Coverage Plot for Full Quadratic Model with Single base Outliers included.



After removing the single base outliers, the distribution and summary statistics for the non-penetrating scenario data were recalculated. The increase in the mean coverage is negligible but the remaining data more closely approximates a normal distribution.

From the final 270 design point dataset a full quadratic model with main effect, interaction, and polynomial terms was created. The Stepwise Regression Control settings within JMP were as follows:

- Probability to enter: 0.01
- Probability to leave: 0.01
- Direction: Mixed

The Construct Model Effects macros for Factorial to Degree and Polynomial to Degree were set at 2 to add two-way interaction and quadratic terms.

The resultant model achieves an R-squared of 0.78 and contains 12 main effect terms, 4 interaction terms, and 3 second order terms. See Figures 6 and 7 for the regression plot and a list of the regression model terms.

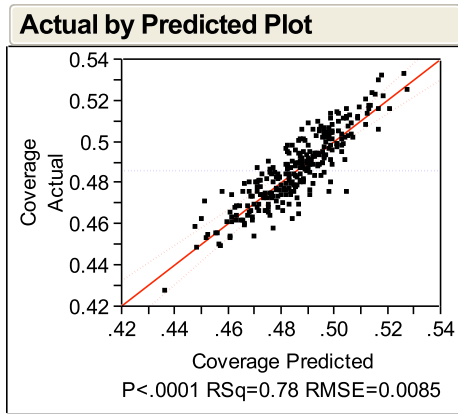


Figure 6. Actual by Predicted Coverage Plot for Full Quadratic Model with Single Base Outliers Removed

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	0.4762323	0.002843	167.51	<.0001*
A RQ-4	-0.002767	0.000534	-5.18	<.0001*
A RQ-1	-0.000885	0.000285	-3.10	0.0021*
A RC-135	0.0039884	0.000535	7.45	<.0001*
A U-2	0.0056309	0.000536	10.50	<.0001*
B RQ-4	-0.002767	0.000541	-5.12	<.0001*
B RC-135	-0.000835	0.000538	-1.55	0.1219
B U-2	0.0035851	0.000534	6.71	<.0001*
C RQ-4	-0.002511	0.000531	-4.73	<.0001*
C U-2	0.0058404	0.000531	10.99	<.0001*
D RQ-4	-0.001957	0.000529	-3.70	0.0003*
D U-2	0.0074843	0.000532	14.07	<.0001*
OPT INT	-0.004388	0.000361	-12.15	<.0001*
(A RQ-4-1.52963)*(B RC-135-1.52963)	-0.002434	0.000522	-4.66	<.0001*
(A RC-135-1.52963)*(D RQ-4-1.52963)	-0.001769	0.000518	-3.41	0.0007*
(B RC-135-1.52963)*(D RQ-4-1.52963)	-0.001793	0.000535	-3.35	0.0009*
(C RQ-4-1.52963)*(D U-2-1.52963)	-0.002549	0.000504	-5.05	<.0001*
(B RQ-4-1.52963)*(B RQ-4-1.52963)	0.001986	0.000574	3.46	0.0006*
(C U-2-1.52963)*(C U-2-1.52963)	-0.001624	0.00057	-2.85	0.0047*
(D RQ-4-1.52963)*(D RQ-4-1.52963)	0.0030399	0.000559	5.44	<.0001*

Figure 7. Full Quadratic Coverage Model Terms

As expected, the most capable platforms, the U-2 and RQ-4, show up as terms from each base. The RC-135 is reflected twice from Bases A and B. The RQ-1 from Base A is the only entry for that platform. Interestingly, the P-3 is not represented in the model at all. In addition to the aircraft factors, the optimization term is also included in the final model.

To test the validity of the Coverage Full Quadratic Regression model the regression assumptions were verified.

## CONCLUSIONS

A scenario was developed, largely within JDAFS, that credibly represented the effects of joint sensors and was able to measure the trade-offs of varying platforms using JDAFS network-enabled assignment optimization.

A data farming interface was developed that began the process of enabling the power of JDAFS. In the coming months, this interface will be refined for JDAFS but, more importantly, generalized for a wide variety of applications to provide analysts who use our simulation the power to conduct exploratory studies and develop credible response surfaces.

