AFRL-IF-RS-TR-2002-275
Final In-House Technical Report
October 2002

APPLYING MODEL ABSTRACTION TECHNIQUES TO THE ADVANCED LOW ALTITUDE RADAR MODEL (ALARM)

Gary Plotz and Serena Dibble

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-275 has been reviewed and is approved for publication

APPROVED:

STEVEN D. FARR
Chief, C4ISR Modeling and Simulation Branch
Information Systems Division

FOR THE DIRECTOR:

JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	OCTOBER 2002	**Final In-House, Oct 99 – Sep 02**

4. TITLE AND SUBTITLE

APPLYING MODEL ABSTRACTION TECHNIQUES TO THE ADVANCED LOW ALTITUDE RADAR MODEL (ALARM)

5. FUNDING NUMBERS

PE - 62702F
PR - 459S
TA - IH
WU - 06

6. AUTHOR(S)

Gary Plotz and Serena Dibble

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFSB
525 Brooks Road
Rome, NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-IF-RS-TR-2002-275

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFSB
525 Brooks Road
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2002-275

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Gary Plotz/IFSB/315-330-1249

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	

13. ABSTRACT ***(Maximum 200 Words)***

Modeling of real systems relies on the arduous task of describing the physical phenomena in terms of mathematical models, which often require excessive amounts of computation time when used in simulations. In the last few years there has been a growing acceptance of model abstraction whose emphasis rests on the development of more manageable models. Abstraction refers to the intelligent capture of the essence of the behavior of a model, without all the details. In the past, model abstraction techniques have been applied to complex models, such as Advanced Low Altitude Radar Model (ALARM) to simplify analysis. The scope of this effort is to apply model abstraction techniques to ALARM; a DoD prototype radar model for simulating the volume detection capability of low flying targets within a digitally simulated environment. Due to the complexity of these models, it is difficult to capture and assess the relationship between the model parameters and the performance of the simulation. Under this effort, ALARM parameters were modified and/or deleted and the impact on the simulation run time assessed. In addition, several meta-models were developed and used to assess the impact of ALARM parameters on the simulation run time. This report establishes a baseline for ALARM from which additional meta-models can be compared and analyzed.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Meta-model, model abstraction, ALARM, factorial design, regression analysis	19
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF TABLES

PAGE

1. INTRODUCTION

Modeling of real systems relies on the arduous task of describing the physical phenomena in terms of mathematical models, which often require excessive amounts of computation time when used in simulations. In the last few years there has been a growing acceptance of model abstraction whose emphasis rests on the development of more manageable models. Abstraction refers to the intelligent capture of the essence of the behavior of a model, without all the details. In the past, model abstraction techniques have been applied to complex models, such as Advanced Low Altitude Radar Model (ALARM) to simplify analysis. The scope of this effort is to apply model abstraction techniques to ALARM; a DoD prototype radar model for simulating the volume detection capability of low flying targets within a digitally simulated environment. Due to the complexity of these models it is difficult to capture and assess the relationship between the model parameters and the performance of the simulation. Under this effort ALARM parameters were modified and/or deleted and the impact on the simulation run time assessed. In addition, several meta-models were developed and used to assess the impact of ALARM parameters on the simulation run time. This paper establishes a baseline for ALARM from which additional meta-models can be compared and analyzed.

ALARM is a generic digital computer simulation designed to evaluate the capability of a ground-based radar system to detect low altitude targets[1]. The ALARM simulation incorporates four highly detailed models which the user can modify in great detail by varying multiple parameters. The four models are a radar model, atmospheric model, terrain model, and target model. ALARM simulates both Pulsed Doppler (PD) and Moving Target Indication (MTI) radar systems. A limited capability to model Continuous Wave (CW) radar is also available. ALARM can operate in either a Flight Path Analysis (FPA) mode or Detection Contour (DC) mode. In the FPA mode, ALARM requires aircraft flight data parameters to be input, and detection is determined for each defined data point along a single north to south straight-line flight path going from left to right across the radar site. The DC mode is used to more generically illustrate the radar's detection performance. ALARM has two types of DC modes. The first is the Horizontal Detection Contour (HDC) mode. In this mode, ALARM generates multiple, north to south straight-line, flight paths going from left to right across the radar site. The range and range increment along the flight path, for which the target detection is determined, is supplied by the user. The second mode is the Vertical Detection Contour (VDC) mode. In this mode the program produces Range-Height-Angle graphs representing radar detection in the vertical plane. A single aircraft speed and altitude are entered and the aircraft is assumed to fly straight and level (pitch and roll can be specified). Actual aircraft radar cross section (RCS) target data and digitized terrain data for the radar location can be loaded for either mode. The model calculates the relative angular geometry from the radar to the target and looks up the corresponding RCS of the target. In both modes a target is detected if the processed target Signal-to-Noise (S/N) ratio exceeds the detector threshold (S/N). By repeating HDC runs over a range of altitudes defined by the user, a detection volume can be generated for the given radar against the given target. For VDC mode the angle is adjusted from 0-360 degrees in increments defined by the user to generate the detection volume.

For realistic ALARM scenarios, the run time required to generate a full coverage volume can be prohibitive because of the highly detailed models. One way to reduce simulation run times is to simplify the models through the application of model abstraction techniques. In this paper three simple abstraction techniques were applied to ALARM to reduce the run time followed by an application of a more sophisticated meta-modeling technique which generates simple polynomial models which can then be used to estimate the impact of more subtle parameters on the run time.

2. MODEL ABSTRACTION TECHNIQUES

Three model abstraction techniques were chosen and applied to estimate the effects of the simulation parameters on the run time performance of ALARM: 1) Dropping Model Components, Descriptive Variables, or Interaction Rules, 2) Coarsening the Ranges and Incremental Parameter Values of Descriptive Variables, and 3) Meta-modeling.

2.1 Dropping Model Components, Descriptive Variables, and Interaction Rules

The importance of model factors can vary depending on user interest. For example, at the engineering level very finely detailed modeling may be required, while less detailed analysis is required for a Campaign Level simulation. Since all model factors are not of equal importance, a good abstraction technique is to ignore some model components, descriptive variables, or interaction rules whose impact is negligible for a given user requirement or situation. This is similar to an engineering approximation and results in a reduction in the complexity of the model by eliminating model factors which least effect the simulation response of interest.

What are the ALARM model factors that have the largest impact on run time? Based on domain expertise and experience there are two; they are the inclusion of digitized terrain and application clutter processing. Table 1 shows the relationship between the simulation run time and (a) where the terrain is stored for processing on the Hard Drive (HD) or in Random Access Memory (RAM), (b) the inclusion of clutter processing (set to on or off), and (c) the inclusion of terrain (set to on or off). The reflectivity of the terrain causes a portion of the radar transmitted power to be reflected back to the radar. This signal energy reflected from sources other than the target masks the radar's ability to see the desired target signal. Clutter processing is a technique to filter out the clutter while passing the target signal on to the detector. The filtering process used by ALARM is based on the Doppler effect. The Doppler effect is a change in transmitted frequency which is perceived to have occurred by a receiver as a result of the existence of relative radial motion between the radar's transmitted signal and the target. The magnitude of the Doppler shift is directly proportional to the rate of relative radial velocity between the radar and target. Since terrain has little or no velocity, a returning signal can be filtered out based on its small relative Doppler shift.

To fully illustrate the detection performance of ALARM it was evaluated in contour mode. Table 1 shows the impact of three ALARM components on the simulation run time: 1) the terrain switch, 2) the clutter processing switch and 3) the terrain data storage method. Two methods are available for storage and accessing the terrain data. This may be done from either a Hard Drive (HD) or from Random Access Memory (RAM). Storing the terrain data in RAM, as opposed to the Hard Drive, results in an order of magnitude decrease in run times with no loss in the detection fidelity or detection volume; as evidenced by the relative times between row 1 and 2 of table 1. A comparison of rows 2 through 4 shows that another fifty percent reduction in the required run time results when the clutter switch, terrain switch or both are turned off. Table 1 also indicates how terrain and clutter processing impact ALARM run time. First note that loading the terrain into RAM for clutter processing as opposed to processing the digitized terrain data directly from the hard drive has the largest impact in reducing the run time. The relative decrease is almost an order of magnitude. Since the terrain is the dominant factor in generating clutter, turning off either of these switches reduces the run time by about a factor of fifty percent. This technique is useful for setting an upper bound on the radar's detection capability. It is also useful for simulating the radar's detection capability against high flying targets where clutter due to terrain effects is absent.

However, turning off the terrain and hence eliminating clutter is unrealistic in the real world, except perhaps for a few special cases; such as a very calm sea state or very flat featureless terrain. Therefore, terrain and clutter processing will be maintained and other abstraction techniques will be investigated to further reduce the run time.

TABLE 1

Changing the resolution by Coarsening the Ranges and Incremental Values of Descriptive Variables

Terrain Loaded	CLUTTER SW	Terrain SW	Simulation Run Time (min)	Relative Time (Sec)
HD	on	on	186 min	1
RAM	on	on	26 min 33 sec	0.14
RAM	on	off	13 min 23 sec	.072
RAM	off	on	13 min 10 sec	.071
RAM	off	off	9 min 3 sec	.048

Reducing the maximum value for ALARM factors and decreasing the simulation resolution by increasing the Incremental Values of ALARM factors can be as simple as a straight reduction in variable range or considering a reduced set of allowable values for the variables. Which ALARM factors impact the run time the most? From the results of table 1 it will be those factors that affect the clutter processing. These include both the terrain and clutter processing components and the capability of the processing hardware such as the processor speed, number of processors, and how the terrain data is stored on the HD or in RAM. ALARM performs a target detection calculation, which includes clutter processing, for each target position. How many target positions are there? It depends on the maximum range (RMAX) and range increment (RINC). For the example presented in table 2 row 2 below the maximum radar range RMAX is set to 80 Kilometers and the range increment RINC to 500 meters. This requires 1 detection calculation every 500 meters from 0-80 Kilometers or 160 discrete target detection calculations for each flight path. To cover the total area each horizontal flight path is repeated at 500 meter increments from out to the maximum range again. This results in a total of 160 times 160 or 25,600 calculations for each altitude. To fill in the volume coverage, we repeat these calculations over an altitude range of (0-3200) feet at a resolution of 100 feet. Thus to determine the volume coverage for 80 square kilometers at 500 meter resolution in azimuth and 100 feet in elevation requires about 800,000 detection calculations. Clutter processing can be adjusted independently of the radar maximum range by setting the clutter maximum range to a smaller value or clutter resolution increment to a larger value. This is logical since as the radar signal propagates outward in a straight line there will come a point at which the signal no longer intersects the earth (which curves downward). Thus beyond this range there will be no clutter signal caused by terrain.

The effect of these factors on the simulation run time is presented in table 2. First, storing the terrain in RAM reduces the simulation execution time by almost a factor of ten. Rows 2-6 show how cutting the maximum radar range or clutter processing range in half or doubling the sampling increment results in cutting the simulation run time in half. Furthermore rows 7-10 show how adjusting two or more factors simultaneously results in a reduction in the simulation run time by factor of four or more. In addition, the run time can be reduced by about 60 percent each time the processor speed is doubled (see rows 11-12). Finally, by increasing the processor speed and distributing the simulation across multiple processors simultaneously the simulation time can be reduced from tens of minutes to less the ten minutes. Thus by increasing the processor speed, distributing the processing across multiple processors and intelligently adjusting parameters which impact detection volume and the density of sample points, a detection volume can be generated in several minutes instead of hours.

TABLE 2

RUN TIME VS ALARM VARIABLES

Time Min/Sec	Terrain Storage	RMAX KM	Clutter MAX KM	Range Inc. Meters	Azimuth Width Degrees	CPU Speed MHz	No. Proc	No. Alt Cuts
186’	HD	80	50	500	360	350	1	1
26’ 33”	RAM	80	50	500	360	350	1	1
13’10”	RAM	40	50	500	360	350	1	1
14’ 9”	RAM	80	25	500	360	350	1	1
13’ 49”	RAM	80	50	1000	360	350	1	1
13’ 16”	RAM	80	50	500	180	350	1	1
4’ 48”	RAM	80	25	1000	360	350	1	1
1’29”	RAM	80	50	500	20	350	1	1
18”	RAM	80	25	1000	20	350	1	1
10”	RAM	40	25	1000	20	350	1	1
11’	RAM	80	50	500	360	700	1	1
6’	RAM	80	50	500	360	1600	1	1
11”	RAM	80	50	500	360	1600	32	1
5.6”	RAM	80	50	500	360	1600	64	1
3’	RAM	80	50	500	360	1600	64	32
6’	RAM	80	50	500	360	1600	64	64
1’ 7”	RAM	80	25	1000	360	1600	64	64
14”	RAM	80	25	500	20	1600	64	64

3. ALARM META-MODEL DESIGN

A meta-model, as defined by Caughlin [2], is a mathematical approximation of the system relationships, defined by another, more detailed model. The meta-model approximates the causal time dependent behavior of a complex simulation model, and allows assessment of individual factors on the performance of the simulation. A good meta-model should satisfy one or more of the following four characteristics. First and most important, the meta-model should mirror the output of the original model to within some desired accuracy. Second, it should maintain that accuracy over the range of interest. Third, the implementation should be simpler than the original model. Fourth, the computational time required to generate an output with the meta-model should be reduced.

3.1 Meta-model techniques

The ALARM meta-models presented here are based on factorial designs [3]. Factorial designs are based on polynomial expressions whose terms involve the set of input factors or variables that are carefully chosen by a model domain expert, or as determined by a sensitivity analysis. For each factor chosen a range of levels (or values) is chosen. The values for each factor are applied to the simulation and the output is tabulated. The simplest factorial designs use only two levels for each of ‘n’ input factors resulting in 2^n unique combinations of input vectors. Thus, for a design using four factors and two levels, $4^2 = 16$ outputs are generated. The output chosen to demonstrate the factorial meta-model design for ALARM is the simulation run time. The four ALARM variables chosen to represent the meta-model are: the target altitude (x_1), the number of Pulse Doppler (PD) filters (x_2), the PD filter band width (x_3), and the target velocity (x_4). Several meta-models were generated using a set of test runs of the ALARM simulation. Each meta-model was derived from the data generated by the ALARM simulation, and was used to capture the run time relationship imposed by those factors. Table 3 lists the values and ranges of each of the four factors used in the generation of the meta-models. The comparison of the meta-models was based on two statistics measuring the goodness of fit. For each meta-model the output is directly compared with the actual ALARM model output. A general overview of the ALARM meta-model designs is presented next, followed by a description of the test cases used in the evaluation of the meta-models.

A full factorial model was initially chosen based on the requirement that each of the four factors has two input levels. In order to determine which interactions among the four factors were statistically the most significant, a normal probability plot of the factor effects was applied and an Analysis of Variance (ANOVA) table was used to confirm the results [4]. The factors and interactions that were found significant are included in the general form of the meta-model below:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{23}x_2x_3 + \varepsilon$$

TABLE 3
ALARM META-MODEL DATA

Tgt Alt	# PDF	Filt BW	Tgt Speed	Time (sec)
100	8	78.125	300	82.69
100	8	78.125	150	85.71
100	8	156.25	300	85.73
100	8	156.25	150	91.6
100	64	78.125	300	129.77
100	64	78.125	150	129.53
100	64	156.25	300	193.14
100	64	156.25	150	193.07
500	8	78.125	300	89.19
500	8	78.125	150	92.96
500	8	156.25	300	92.84
500	8	156.25	150	99.99
500	64	78.125	300	146.24
500	64	78.125	150	146.55
500	64	156.25	300	223.08
500	64	156.25	150	223.12

The coefficients were determined using a least squares estimation. The method of least squares requires that a polynomial be fitted to the set of response data points such that the sum of the squares of the distance of the points to the fitted line is minimized. The general form of the meta-model using this solution yields:

$$\acute{y} = 81.084 + 0.012546\ x_1 - 0.47921\ x_2 - 0.056430\ x_3 + 0.00071685\ x_1x_2 + 0.014898\ x_2x_3$$

(Meta-model 1)

Least Squares Estimation is based on the assumption that the data is from a normal distribution. However, plotting the residuals on a normal probability plot revealed that the data is not normal. Therefore, to normalize the data a Box-Cox Transformation was applied to the data. The form of the transformed meta-model is

$$\acute{y} = \mathrm{sqrt}(2*(4155.0 + 0.36180\, x_1 - 156.38\, x_2 - 15.778\, x_3 + 0.16140\, x_1x_2 + 2.6877\, x_2x_3) + 1).$$

(Meta-model 2)

If a model is to account for an n^{th}-order effect in a factor, then the experimental design must have n + 1 different levels in that factor. Since a 2^4 factorial design was used, the data only had two levels (-1 and 1, or low and high) and could only account for first-order effects. In order to determine whether the model had some quadrature that was not being accounted for, the residuals were plotted against the predicted values. The points on this plot suggested a violation of linearity (fitting data that are nonlinearly related with a linear model).

The next step was to make a model that contained squared terms in order to account for the quadrature. A Central Composite Design (CCD) was chosen. A CCD uses the data from the factorial model, a center run, and axial points (all of the factors are held at the center while one is set at a positive and negative distance from the center). With the central composite design, there are three levels (-1, 0, and 1, or low, center, and high) so the model can estimate second-order single-factor effects (curvature in each factor). In this case a meta-model with the axial points scaled at +1 and -1 was developed. The coefficients in the CCD models were solved for in the same manner, using least squares estimation:

$$\acute{y} = 221.04 - 0.095502\, x_1 - 0.65088\, x_2 - 0.37889\, x_3 - 1.0365\, x_4 - 0.000060268\, x_1x_2 + 0.0000035200\, x_1x_3 - 0.0000018333\, x_1x_4 + 0.011941\, x_2x_3 + 0.000091964\, x_2x_4 - 0.00026161\, x_3x_4 + 0.0000062400\, x_1x_2x_3 - 0.00000022917\, x_1x_2x_4 - 0.00000014263\, x_1x_3x_4 + 0.0000037143\, x_2x_3x_4 + 0.0000000036952\, x_1x_2x_3x_4 + 0.00018459\, x_1^2 + 0.0045198\, x_2^2 + 0.0015457\, x_3^2 + 0.0022895\, x_4^2$$

(Meta-model 3)

A normal probability plot of the residuals from meta-model 3 revealed that a transformation on the data was needed. The form of the transformed meta-model is:

$$\acute{y} = (-2*(\, 0.49996 - 1.4918\text{e-}008\, x_1 + 1.0413\text{e-}006\, x_2 + 3.9367\text{e-}007\, x_3 - 5.4993\text{e-}007\, x_4 - 4.6875\text{e-}011\, x_1x_2 - 2.0937\text{e-}011\, x_1x_3 - 4.7619\text{e-}012\, x_1x_4 + 1.1949\text{e-}009\, x_2x_3 + 2.2798\text{e-}010\, x_2x_4 - 3.3685\text{e-}010\, x_3x_4 - 1.3429\text{e-}012\, x_1x_2x_3 - 2.9762\text{e-}014\, x_1x_2x_4 + 8.4419\text{e-}014\, x_1x_3x_4 + 5.0667\text{e-}012\, x_2x_3x_4 - 6.8571\text{e-}016\, x_1x_2x_3x_4 + 7.2195\text{e-}011\, x_1^2 - 7.4390\text{e-}009\, x_2^2 - 1.0336\text{e-}009\, x_3^2 + 1.1907\text{e-}009\, x_4^2\,) + 1)^{-1/2}$$

(Meta-model 4)

The outliers in the residual plots of both meta-models 3 and 4 corresponded to factor x_1 and factor x_4 being held constant and x_2 and x_3 changing.

TABLE 4
META-MODELS LEVELS OF EFFECTIVENESS

	MAE	R^2
Meta-model 1	2.89	99.63%
Meta-model 2	4.22	99.09%
Meta-model 3	3.88	98.04%
Meta-model 4	1.05	99.88%

4. RESULTS AND CONCLUSIONS

The test statistics R^2 and MAE for each test case, shown in Table 4, provide insight into the behavior of the meta-models over varied input data selections. The R^2 statistic describes how much of the variation is actually attributed to the model and the MAE statistic describes, on average, how far off the estimates are from the model. The fidelity of the meta-models was determined by comparing their run times against ALARM run times. Based on these two statistics, the best meta-model to estimate the simulation run time would be meta-model 4, which is a transformed Central Composite Design (CCD) model. This CCD model takes into account second-order single-factor effects (curvature in factors) and the transformation adjusts the data points so the distribution is Gaussian**.**

In order to compare the meta-models with test points, random values were used for the inputs in the ALARM system. The ALARM responses were compared to the meta-model estimations for the same data. The statistics used to analyze these test points were the MAE, which was previously described, and a percent accuracy. “Average Percent Accuracy” is the average of the percent of the simulation run time within which the meta-model correctly estimated within. “Percent Accuracy Within 10%” is the percentage of the test points that were estimated within 10% of the actual simulation run time. “Percent Accuracy Within 5%”

is the percentage of the test points that were estimated within 5%. Below is a chart of the results:

TABLE 5

MEASURES OF META-MODEL ACCURACY

	MAE	Average Percent Accuracy	Percent Accuracy Within 10%	Percent Accuracy Within 5%
Meta-model 1	16.9	14.6%	50%	20%
Meta-model 2	24.1	16.0%	30%	20%
Meta-model 3	10.6	8.2%	90%	40%
Meta-model 4	9.82	7.7%	80%	50%

Points corresponding to factor x_2 and factor x_3 stood out in most of the residual plots, and a meta-model that contained only these two factors estimated the simulation run time fairly well. Therefore, it is safe to assume that factor x_2 and factor x_3 (the number of pulse Doppler filters and the filter bandwidth) have the most effect, of the chosen factors, on the simulation run time.

One measure in determining the meta-models' ability to pattern the behavior of the ALARM benchmark model involves comparing test case 2 (transformed or normalized) with test case 1. For each meta-model the comparison of the AARE (and R^2) statistics compares favorably and the differences that resulted were not significant. These comparisons support the inference that the meta-models pattern the behavior of the benchmark model for input values that are within the chosen input data ranges that were used in designing the meta-model itself.

5. FUTURE META-MODEL DEVELOPMENT AND ANALYSES

The meta-model analyses could be expanded in the following manner:

1. Perform additional simulations, similar to test cases 3 and 4, to determine a more accurate estimate of the test statistics on the meta-models. The intent would be to determine

statistics that were not highly dependent on the choice of the subset of input data values that were used to generate the meta-model.

2. Perform sensitivity analyses to determine the level of influence that each input factor has on the behavior of the model. This would provide more insight regarding the meta-model itself.
3. Incorporate different types of meta-models, such as those based on adaptive neural networks or higher level CCD designs, into the baseline that was established within this paper. Through in-depth analyses more robust meta-models could be identified, along with the general rules of thumb to exploit the strengths of each model.

6. SUMMARY

This paper focused on applying model abstraction techniques to ALARM; a DoD prototype radar model for simulating the volume detection capability against low flying targets within a digitally simulated environment. Due to the complexity of these models it is difficult to capture and assess the relationship between the parameters and the performance of the simulation. Under this effort ALARM parameters were modified and/or deleted and the impact on the simulation run time assessed. In addition, several meta-models were developed and used to assess the impact of ALARM parameters on the simulation run time. The impact of the terrain switch, the clutter processing switch, and the terrain data storage method, on the simulation run time were evaluated. Several ALARM meta-models, based on factorial designs, were developed and applied. The meta-model test cases show that the run time execution can be estimated using the meta-models for a given target altitude, velocity, and set of Doppler Filters and bandwidth. The test case results indicated a strong correlation between the simulation run times predicted by the meta-models and run time for the actual ALARM simulations. This paper establishes a baseline for ALARM from which additional meta-models can be compared and analyzed to determine the effect of ALARM factors on the simulation run time.

REFERENCES

1. "Software User's Manual for the Advanced Low Altitude Radar Model (ALARM) V 4.3," Air Force Research Laboratory

2. D. Caughlin, "Verification, Validation, and Accreditation (VV&A) of Models and Simulations Through Reduced Order Meta-modeling". Proceedings of the 1995 Winter Simulation Conference, pp. 1405-1411, 1995.

3. R. Myers, *Response Surface Methodology*, Edwards Brothers, Ann Arbor Michigan, 1976.

4. R. Myers, *Classical and Modern Regression with Applications*, PWSC-Kent, Boston, 1990.