STAT COE-Report-7-2018

Computer Experiments: Space Filling Design and Gaussian Process Modeling

Best Practice

Authored by: Cory Natoli

Sarah Burke, Ph.D.

30 March 2018

The goal of the STAT COE is to assist in developing rigorous, defensible test strategies to more effectively quantify and characterize system performance and provide information that reduces risk. This and other COE products are available at www.afit.edu/STAT.

STAT Center of Excellence
2950 Hobson Way – Wright-Patterson AFB, OH 45433

Table of Contents

Executive Summary

Computer simulations provide an interesting change to classic design choices because the responses are typically deterministic or have very little noise. More importantly, test designs should exhibit this characteristic, as the test design always reflects the type of model used to fit the response. In computer experiments, higher order terms are frequently necessary to adequately characterize a response. Therefore, space filling designs are often excellent design choices because the design points are spread out evenly throughout the design region. These designs provide flexibility to form appropriate models to characterize the system.

Keywords: Space filling designs, computer simulations, experiments, deterministic response

Introduction

As computer processing power rapidly increases, computer simulations are being utilized in a number of scenarios, especially when physical testing is too challenging. For example, computer experiments may be preferred when testing the physical system is expensive, the physical system is difficult to operate, there are safety concerns, or the computer model allows for faster exploration of alternatives/development of prototypes (Montgomery, 2017). In general, simulations are capable of running a larger sample size than we could on the physical system; however, in some computer experiments each run may still take a long time to run due to the complex nature of the simulation. A well-designed, efficient test is still, therefore, required for these types of computer experiments.

The response of a computer experiment is often deterministic, meaning the computer experiment will always produce the same output when the initial input conditions remain constant. The model we fit on the data from a computer experiment, therefore, must reflect that there is little to no noise in the response. This model also informs the type of design used for the test itself. In classical designs, the three foundations of design of experiments (DOE) (randomization, replication, and blocking) are emphasized when selecting a design for a given test. However, these principles are irrelevant for deterministic models, because the response will always be the same for the same factor combination regardless of when tested. In other words, replicates waste resources and the effect of potential noise factors (e.g., day) are no longer a concern. Another characteristic of computer experiments with a deterministic output is that these models often require higher order terms, particularly with the interaction terms between factors. Higher order terms are included because the computer simulations are typically an attempt to replicate a complex process. The goal of these tests is generally characterization over comparison in order to understand the properties of the true physical system. Therefore, choosing only corner points (like in a factorial design) will not be sufficient to characterize the response due to its more complex nature. Rather than just using corner points, the designs for computer experiments should fill the design space. This allows us to fit many types of models that can explain the complexity of these systems (Montgomery, 2017).

Space Filling Designs

Space filling designs are recommended for tests with deterministic models because the design points are spread out evenly throughout the design region. In order to use these design choices, an important assumption is necessary: the computer simulation must reflect the true physical system. This paper does not explore the ability for space filling designs to help validate a simulation with physical test data, but does explain using these designs on a validated simulation to characterize a physical system.

Due to the spread out points, space filling designs are able to capture the different behaviors of responses in different areas of the design region. Because of the complex nature of these simulations, the behavior of the response can change dramatically across the design space. Space filling designs typically seek to not have replicates in k dimensions (where k is the number of factors). If the dimension of the design space is reduced upon finding factors with no statistical effect on the response, an ideal space filling design will not have replicate points in the lower dimension design space. For instance, if a factor is determined to be not significant and the design is projected onto the lower dimension of remaining factors, replicates will not be present. This allows us to gain more information on the system since replicated test points do not provide additional information and the response is deterministic.

The design diagnostics for space filling designs are not as extensive (or meaningful) as compared to classical designs. Metrics such as power, prediction variance, and the alias properties of the design are no longer meaningful for models with a deterministic response. Space filling design metrics include the minimum distance between points and discrepancy. These diagnostics are meaningful when comparing multiple designs. The larger the minimum distance between points, the better, for a fixed sample size. Discrepancy is a metric for how evenly spaced the design points are throughout the design region. The smaller the discrepancy, the better, for a fixed sample size, as this indicates a more uniformly spaced design. Four different space filling designs are discussed in the next sections.

Sphere Packing Design

Sphere packing designs choose design points such that the minimum distance between pairs of design points is maximized. All factors must be continuous to create a sphere packing design. The algorithm for creating a sphere packing design uses min-max optimization. This design type does not consider discrepancy, so sphere packing designs typically have large values for the minimum distance between points and discrepancy. Figure 1 shows an example of a sphere packing design for three factors.

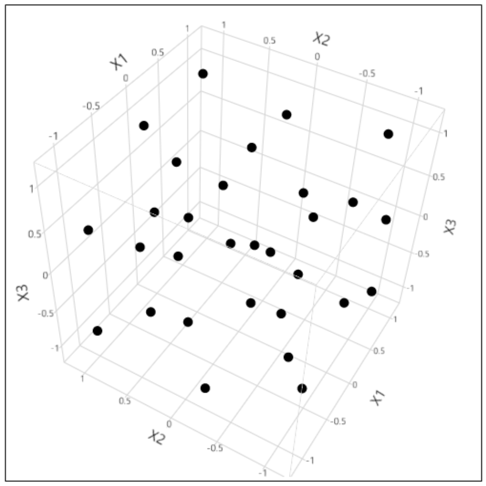


Figure 1: Sphere packing design

Simply observing the design in all dimensions may be difficult to interpret. To help clarify, often two variables will be plotted against each other, or a scatterplot matrix will be created. Both can be seen in Figure 2.

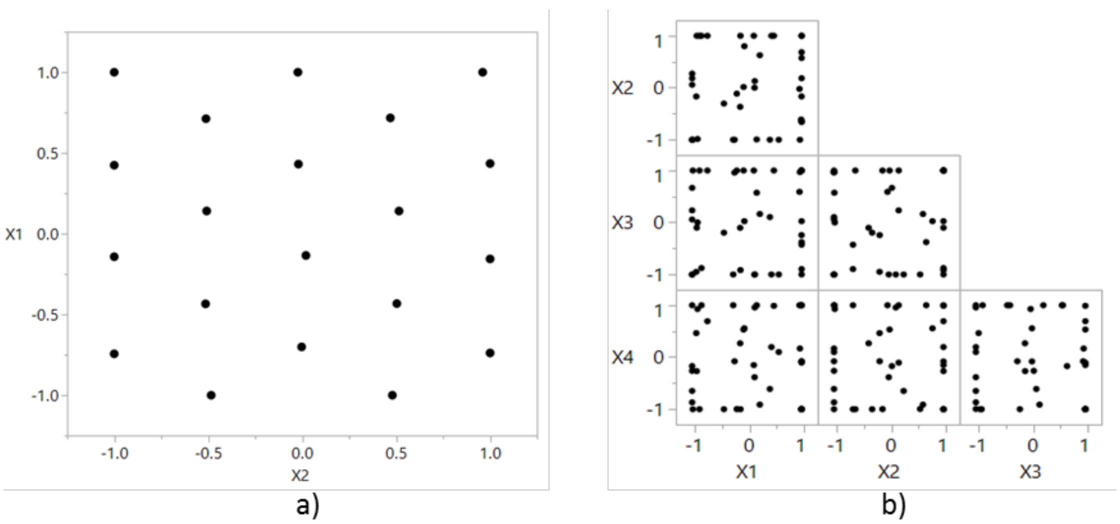


Figure 2: Sphere packing design for (a) two factors and (b) three factors

These graphs help determine what areas of the design space have been covered. In observing the scatterplot matrix, it is clear to see that sphere packing designs favor the edges of the design space before putting points into the middle of the design space. This approach helps to maximize the minimum distance between points. Sphere packing designs clearly favor areas of the design space and are not uniformly spread throughout which is why there are large discrepancy values.

Uniform Design

Uniform designs choose points that are uniformly scattered throughout the design region. Once again, all factors must be continuous to create this type of design. Uniform designs seek to minimize the discrepancy, but do not consider the minimum distance between points metric. Therefore, uniform designs typically have small values for the minimum distance between points and small values for discrepancy. Figure 3 shows an example of a uniform design for two factors, three factors, and the scatterplot matrix for the three factor design.

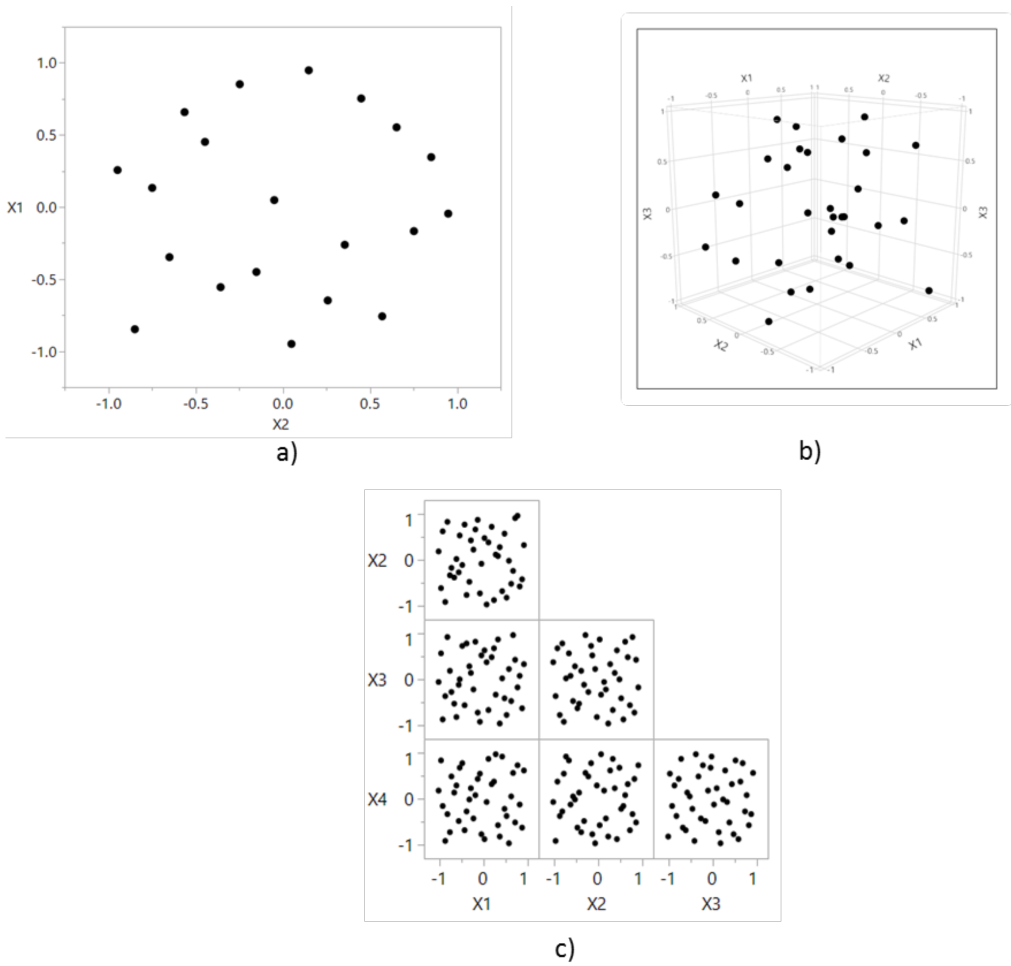


Figure 3: Uniform space filling design for (a) two factors (b) three factors (c) three factors illustrated using a scatterplot matrix

The scatterplot matrix of the uniform design in Figure 3c shows a much more even spread when compared to the sphere packing design. However, the points can be very close together which may indicate a lack of information in certain areas in the design region. For example, in Figure 3a and 3b, the corner points are not captured in the uniform design, which may be important areas of interest. Both the sphere packing design and uniform design are very effective at optimizing one of the two main metrics for space filling designs. Another type of space filling design attempts to balance the two metrics.

Latin Hypercube Design

Latin hypercube designs provide a balance between the objectives of sphere packing and uniform designs. The algorithm to select design points is constrained optimization: the objective is to maximize the minimum distance between design points (like the sphere packing design), but requires points to be evenly spaced (like the uniform design). This design type also requires that all factors be continuous. Figure 4 shows a Latin hypercube design for three factors.

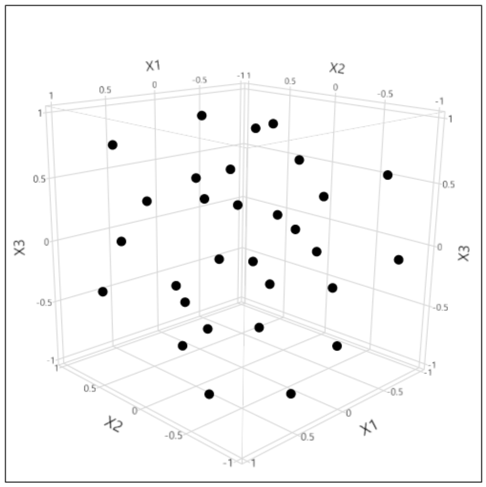


Figure 4: Latin hypercube design

Latin hypercube designs are most commonly used in practice due to its balanced nature. The Latin hypercube design is the most effective design to capture the different behaviors of responses in different regions of the design region. Figure 5 shows a Latin hypercube design for two and three factors, respectively. Note that this method captures the boundaries and interior of the space.

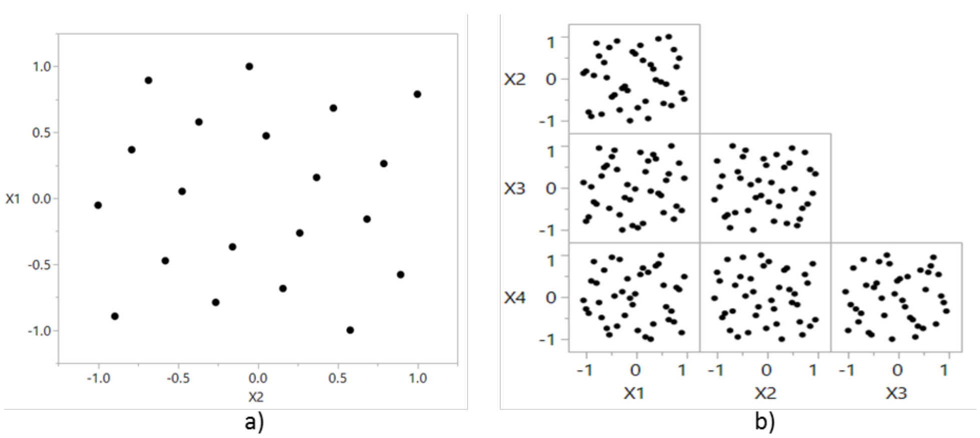


Figure 5: Latin hypercube design for (a) two factors and (b) three factors

The Latin hypercube design best captures the design space by selecting both points on the outsides of the design region and in the center. The Latin hypercube is the most frequently used space filling design when all factors are continuous. However, when there are categorical factors, a different type of design is required.

Fast Flexible Filling

The three methods so far are applicable only when all factors are continuous. In contrast, the fast flexible filling design is utilized when there is one or more categorical factor. In fact, this is the only space filling design option currently available in JMP (the statistical software that is used in this paper) which allows categorical factors. This design uses a clustering approach to select design points. This method also best demonstrates why a space filling design seeks to not have replicates in k dimensions (where k is the number of factors). Among the different levels of a categorical variable, a fast flexible filling design will not place the continuous factors at the same level. The reasoning behind this choice is that if the categorical factor is not significant, the design will project to an unreplicated design in the lower dimension design space. Figure 6 demonstrates this concept where X1 is a continuous factor and X2 is a two-level categorical factor. When the design is projected down from the categorical factor, it results in no replicated points of X1, seen on the right side of Figure 6. The distance between each consecutive design point is also the same in the lower dimensional space providing the most insight in the fewest number of points.

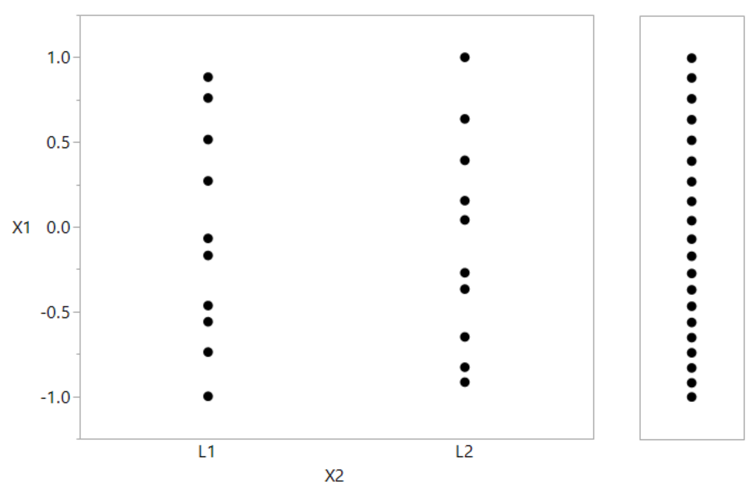


Figure 6: A fast flexible design with a 2-level categorical factor (X2) vs. a continuous factor (X1)

This method also extends to categorical factors with more than two levels. Figure 7 shows a fast flexible filling design for X1, a continuous factor, and X2, a 4-level categorical factor. Projecting this design to the X1 space, there are no replicated observations.

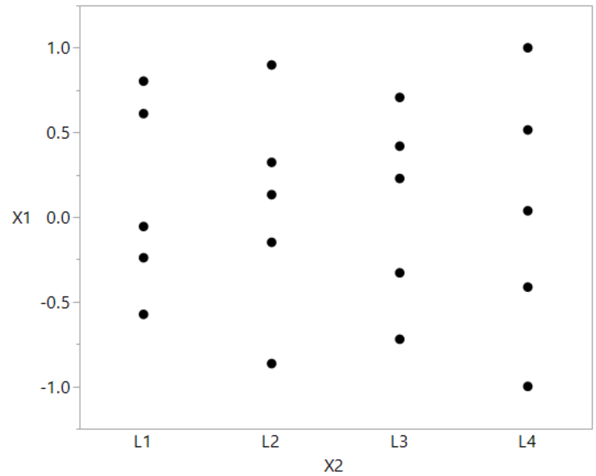


Figure 7: A fast flexible design with a 4-level categorical factor (X2) vs. a continuous factor (X1)

Sample Size

Creating a space filling design must balance the cost (time, money, resources, etc.) of executing the computer experiment and the number of runs necessary to estimate the model of interest. In general, the recommended run size is approximately 10 runs per number of factors (Montgomery, 2017). However, the more complex the model, the more runs that are necessary to suitably estimate it. Comparing several designs for different run sizes and using scatterplot matrices provides additional information for design choices. The designs can also be compared using the minimum distance between points and discrepancy.

Gaussian Process Modeling

Gaussian process models are one of the most frequently used methods to model the response of a deterministic computer experiment. These models seek to find an exact fit to the observations since there is no (or little) noise. In linear regression, there is a parameter for every model term. Because deterministic models tend to be complex, in that they require high order terms such as quadratic, cubic, or high order interactions terms, the number of parameters to estimate can be large. In Gaussian process models, the number of parameters is equal to the number of factors, so there are fewer parameters to estimate.

The true relationship between the response and the factors is complex. Therefore, there is no assurance that a Gaussian process model will interpolate well in a design region where no data was collected (Montgomery, 2017). For example, in Figure 8, if the space filling design did not contain points in the corner of the design region, the model would not capture the change in shape in this region.

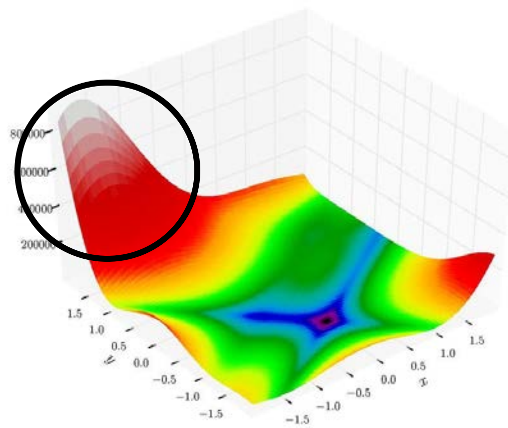


Figure 8: The Goldstein Price function[1]

The Gaussian process model is complex and requires a computer to fit the model. The form of the model is:

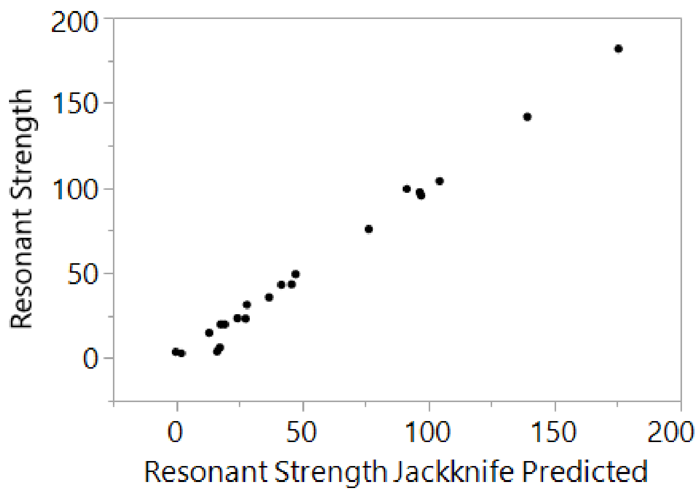
$$y = \mu + z(\boldsymbol{x})$$

where $z(\boldsymbol{x})$ is a Gaussian process with covariance matrix $\sigma^2\mathbf{R}(\theta)$.

More information on the mathematical details of the Gaussian process model can be found in Montgomery (2017).

Interpreting Gaussian process models requires observing a number of effects. Software will provide information on theta, total sensitivity, main effect, and interaction effects. Theta represents the model parameter estimates. Factors that have small theta values have little (or no) impact on prediction. The main effect is the total variation due to the factor alone. The interaction effects show the variation due to the interaction of one factor with other factors. The total sensitivity is the sum of the main effect and all interaction terms for each factor. This is a measure of the influence a factor and all its interactions have on the response. A common visual representation of the model is plotting the actual Y on the y-axis and the Y jackknife predicted on the x-axis. The Y jackknife predicted is the predicted value where the row is excluded from the prediction model for each associated response. Ideally, this will be a straight line. Figure 9 shows an example of this graph.

1: https://en.wikipedia.org/wiki/File:Goldstein_Price_function.pdf



a)

Model Report

Column	Theta	Total Sensitivity	Main Effect	Diameter Interaction	Height Interaction
Diameter	0.0312514	0.7025057	0.1256223	.	0.5768834
Height	0.0020268	0.8743777	0.2974943	0.5768834	.

μ	σ^2
196.48607	22479.793

-2*LogLikelihood
189.77644

Fit using the Gaussian correlation function.

b)

Figure 9: (a) Actual vs. Predicted graph for a Gaussian process model and (b) Model Report for a Gaussian process model

Gaussian process modeling is incredibly powerful for modeling complex systems, but sometimes a computer experiment can be modeled using stepwise regression. Stepwise regression can fit models that include interactions, quadratic terms, and higher order terms. These models can be easier to interpret than a Gaussian process model, but may not adequately characterize the system across the entire design space. If the stepwise regression technique does not yield a good fitting model, Gaussian process modeling allows us to model the increased complexity in the system.

Conclusion

Space filling designs are powerful designs for computer systems with deterministic responses. These designs allow for a spread of points encompassing the entire design space. A variety of different space filling designs exist depending on the needs of the experiment. In order to model complex deterministic systems, Gaussian process models seek to find an exact observation to these systems. These methods allow for the use of rigorous statistical procedures to model computer simulations in a number of scenarios when physical testing is challenging.

Example

Design

Suppose we have a computer simulation for two separate sensor models we wish to compare: Model 1 vs Model 2. The goal of these models is to detect enemy targets. The test objective for this phase of testing is to "characterize and show differences by using a higher fidelity model." In other words, the goal is to characterize the performance of Model 1 and how it compares to Model 2 under a variety of test conditions. The first step is to identify the possible factors. For simplicity, numerical factors will be considered continuous and can take on any numerical value within a given range. This test used:

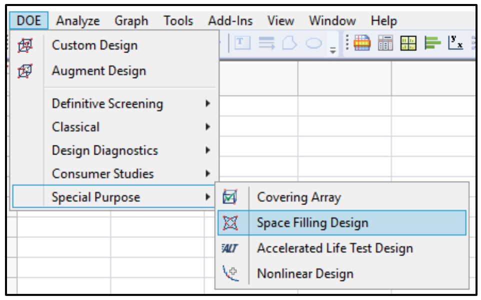
- Number of decoys – targets that are intentionally designed by Red forces to deceive Blue forces into diverting blue resources from the designed mission (Continuous)
- Number of confusers – decoy targets in the scenario that emit a signal similar to that of the target(s) of interest (Continuous)
- Number of sythentic aperture radars (SAR) – the number of autonomous vehicles equipped with synthetic aperture radars (SAR) (Continuous)
- Number of electronic support measures (ESM) – the number of autonomous vehicles equipped with ESM(Continuous)
- Split - Whether the initial formation of the autonomous vehicles is in a split or co-located format (Categorical)
- Decoy Distance – max distance between decoys and target(s) of interest (Continuous)
- Laydown – different patterns of red forces placements (Categorical)
- Model – the sensor model being utilized (Categorical)

Table 1: Summary of Factor Levels

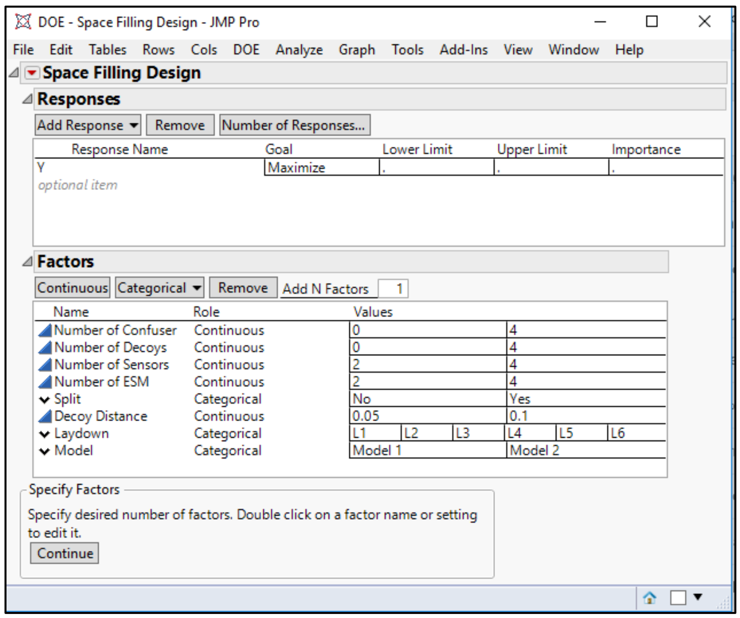
Factor	Number of Confusers	Number of Decoys	Number of Sensors	Number of ESM	Split	Decoy Distance	Laydown	Model
Levels	0	0	2	2	No	.05	L1	Model 1
	4	4	4	4	Yes	.1	L2	Model 2
							L3	
							L4	
							L5	
							L6	

We know this system is a deterministic model, so we elect to use a space filling design. Since we have categorical variables, we use a fast flexible design. The following steps show how to create this design in JMP V.13.

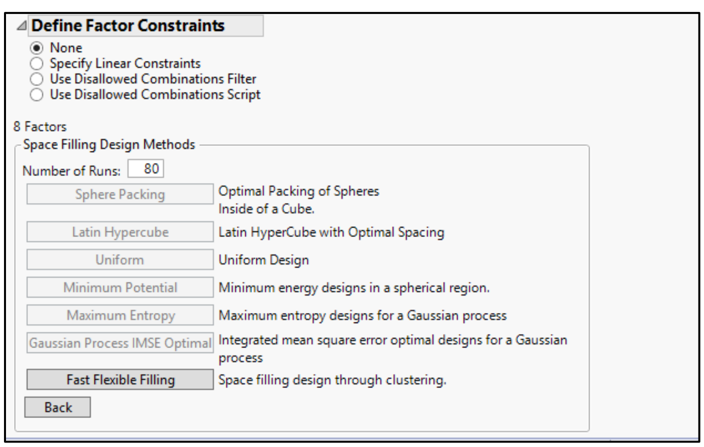
1. Open a New Data table
2. Select "DOE -> Special Purpose -> Space Filling Design"



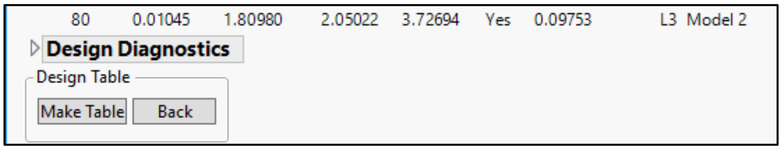
3. Enter the factors into the factor window. Select the role of the variable and the number of levels and enter in the appropriate values.



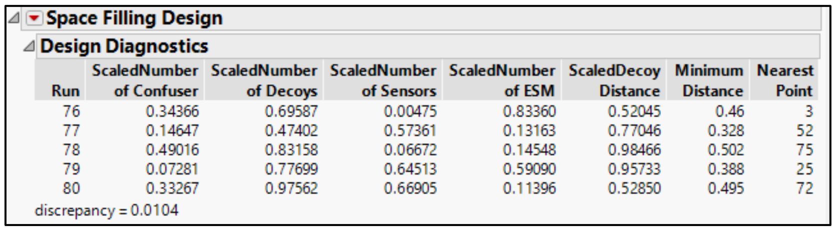
4. Select "Continue"
5. Specify the number of runs (80 since we have 8 factors) and Fast Flexible Filling design. There is an option for factor constraints, of which we have none, in this window.



6. The following window will show the design, but first to view the diagnostics, scroll to the bottom and expand the design diagnostics tab.



7. Observe the discrepancy (located near the bottom of the window) and the minimum distances to compare this design with others.



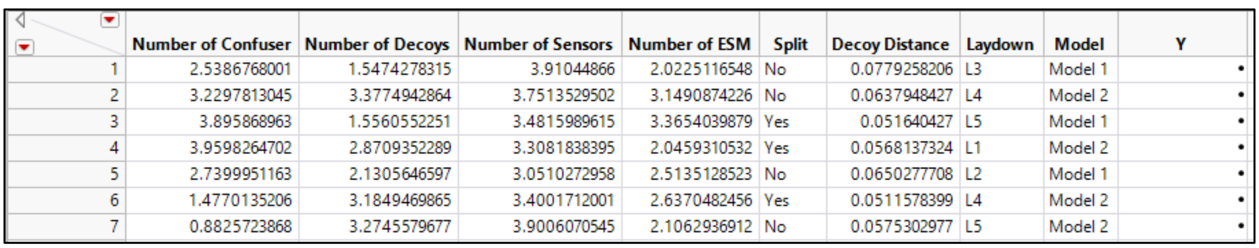
Space Filling Design

Design Diagnostics

Run	ScaledNumber of Confuser	ScaledNumber of Decoys	ScaledNumber of Sensors	ScaledNumber of ESM	ScaledDecoy Distance	Minimum Distance	Nearest Point
76	0.34366	0.69587	0.00475	0.83360	0.52045	0.46	3
77	0.14647	0.47402	0.57361	0.13163	0.77046	0.328	52
78	0.49016	0.83158	0.06672	0.14548	0.98466	0.502	75
79	0.07281	0.77699	0.64513	0.59090	0.95733	0.388	25
80	0.33267	0.97562	0.66905	0.11396	0.52850	0.495	72

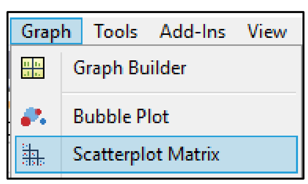
discrepancy = 0.0104

8. Select "Make Table" at the bottom of the window to show the design in a data table now. An example of the first 7 observations is shown below.

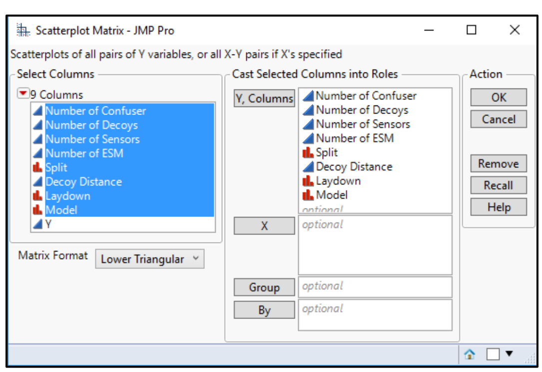


	Number of Confuser	Number of Decoys	Number of Sensors	Number of ESM	Split	Decoy Distance	Laydown	Model	Y
1	2.5386768001	1.5474278315	3.91044866	2.0225116548	No	0.0779258206	L3	Model 1	•
2	3.2297813045	3.3774942864	3.7513529502	3.1490874226	No	0.0637948427	L4	Model 2	•
3	3.895868963	1.5560552251	3.4815989615	3.3654039879	Yes	0.051640427	L5	Model 1	•
4	3.9598264702	2.8709352289	3.3081838395	2.0459310532	Yes	0.0568137324	L1	Model 2	•
5	2.7399951163	2.1305646597	3.0510272958	2.5135128523	No	0.0650277708	L2	Model 1	•
6	1.4770135206	3.1849469865	3.4001712001	2.6370482456	Yes	0.0511578399	L4	Model 2	•
7	0.8825723868	3.2745579677	3.9006070545	2.1062936912	No	0.0575302977	L5	Model 2	•

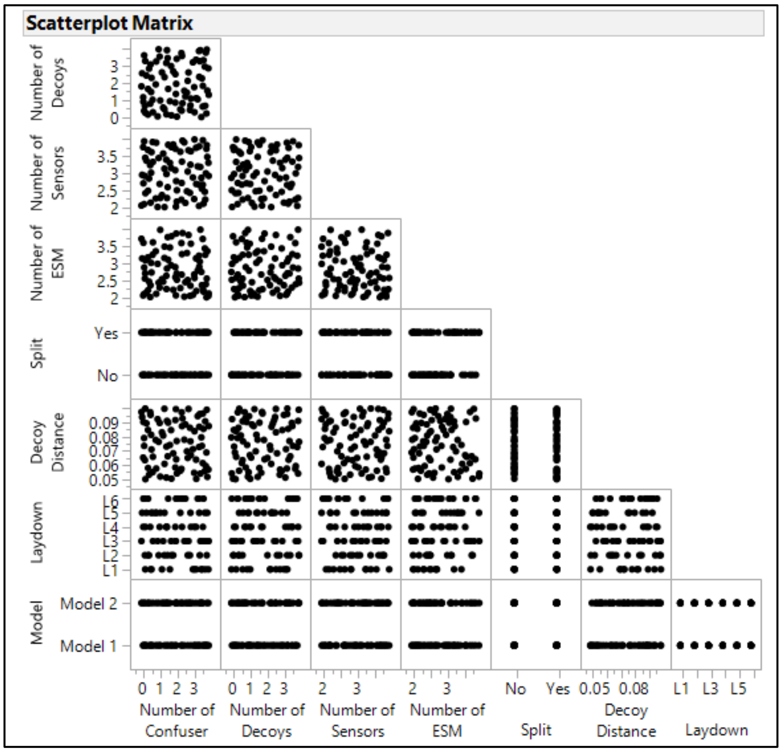
9. In order to observe the design space captured by this design, we can create a scatterplot matrix. Select "Graph -> Scatterplot Matrix."



10. Load the 8 factors into the "Y, Columns" window and select "OK."



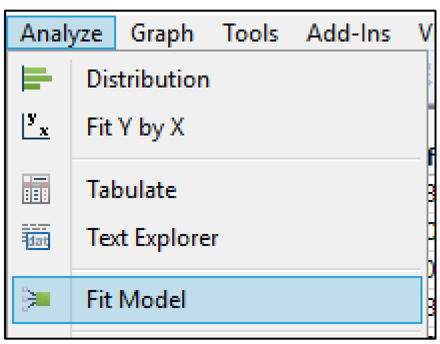
11. Observe the scatterplot matrix and make sure the design appropriately covers the design space. Jitter may be selected and can be removed by selecting the red drop down and unchecking "points jittered."



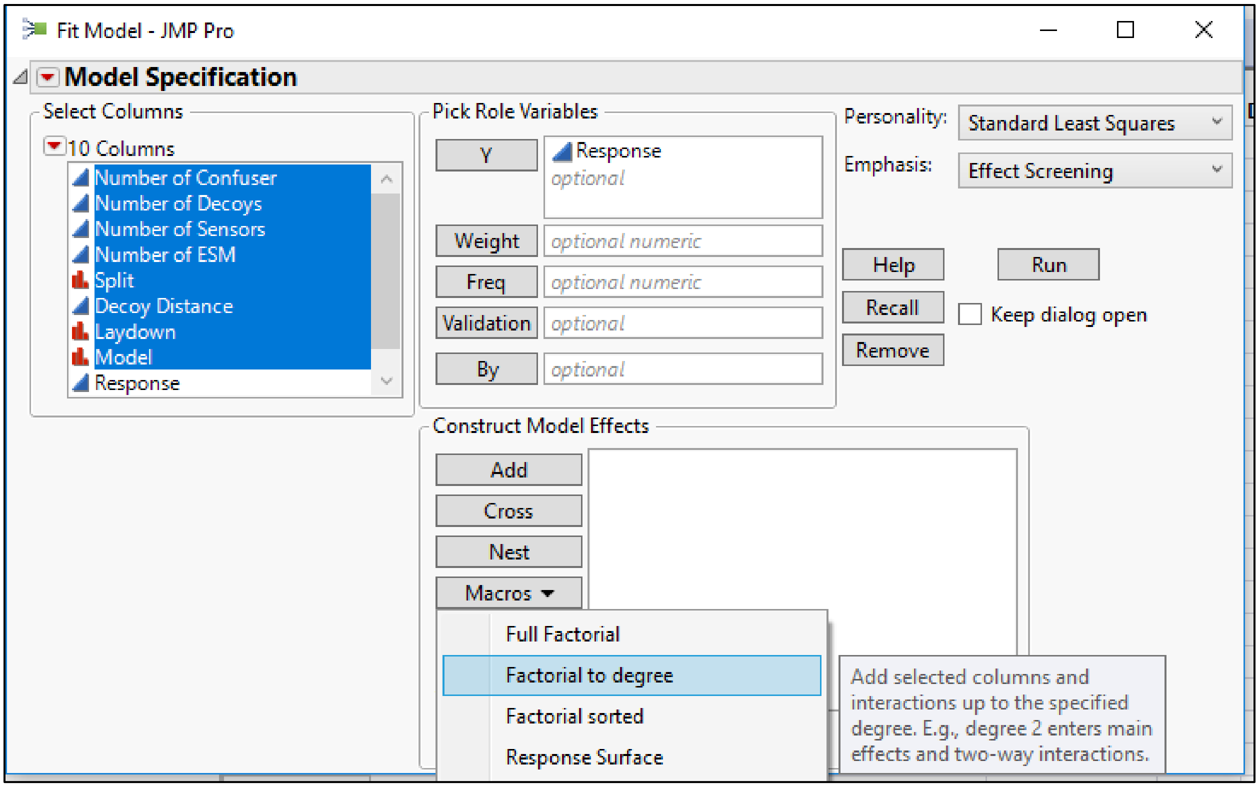
Analysis

Now, let's assume we collected data on this experiment and wish to fit a model. The traditional method would be to use the "Fit Model" command in JMP.

1. Select "Analyze -> Fit Model"



2. Select the response and load it into the Y. Note: Data has been simulated for the response column for this example. Select all of the factors and select the dropdown "Macros -> Factorial to degree." Degree will be set at 2 by default which will add in all of the main effects and two-factor interactions.



3. Select "Run" to build the model.
4. We can look at the Effect Summary of the model and remove insignificant terms at a specified significance level, however this model has two terms that cannot be estimated since laydown has many terms associated with it for the main effect and interactions due to the 6 different levels of laydown. This requires a large number of runs to be able to estimate all of these terms in the model. This means that there are more terms to fit than are estimable by our design. We might instead try to fit a Gaussian process model.

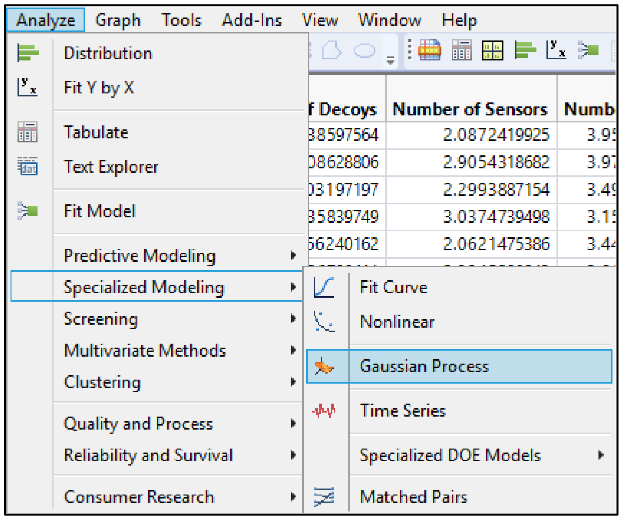
Effect Summary

Source	LogWorth	PValue	
Number of Confuser(0,4)	1.211	0.06152	
Number of Confuser*Number of Decoys	0.954	0.11128	
Laydown	0.888	0.12942	
Number of ESM(2,4)	0.766	0.17157	
Number of Decoys*Split	0.633	0.23277	
Number of Confuser*Number of ESM	0.532	0.29344	
Decoy Distance(0.05,0.1)	0.423	0.37758	
Number of Confuser*Split	0.408	0.39040	
Number of Sensors*Number of ESM	0.384	0.41260	
Number of ESM*Model	0.342	0.45481	
Split*Model	0.257	0.55301	
Number of Sensors*Split	0.237	0.57983	
Number of ESM*Decoy Distance	0.213	0.61172	
Number of Decoys(0,4)	0.189	0.64786	^
Split*Laydown	0.181	0.65867	
Number of Decoys*Number of Sensors	0.168	0.67928	
Decoy Distance*Model	0.167	0.68023	
Number of Confuser*Model	0.165	0.68426	
Number of ESM*Split	0.142	0.72073	
Number of Decoys*Decoy Distance	0.131	0.73991	
Number of Confuser*Decoy Distance	0.126	0.74837	
Number of Sensors*Laydown	0.122	0.75437	
Number of Sensors*Model	0.117	0.76467	
Number of Sensors*Decoy Distance	0.107	0.78172	
Split*Decoy Distance	0.093	0.80641	
Number of Confuser*Laydown	0.085	0.82138	
Number of ESM*Laydown	0.062	0.86731	
Decoy Distance*Laydown	0.060	0.87106	
Number of Decoys*Model	0.041	0.91070	
Number of Decoys*Number of ESM	0.020	0.95495	
Number of Confuser*Number of Sensors	0.009	0.97977	
Number of Sensors(2,4)	0.005	0.98784	^
Number of Decoys*Laydown	0.005	0.98884	
Laydown*Model	0.003	0.99407	
Model	.	.	
Split	.	.	

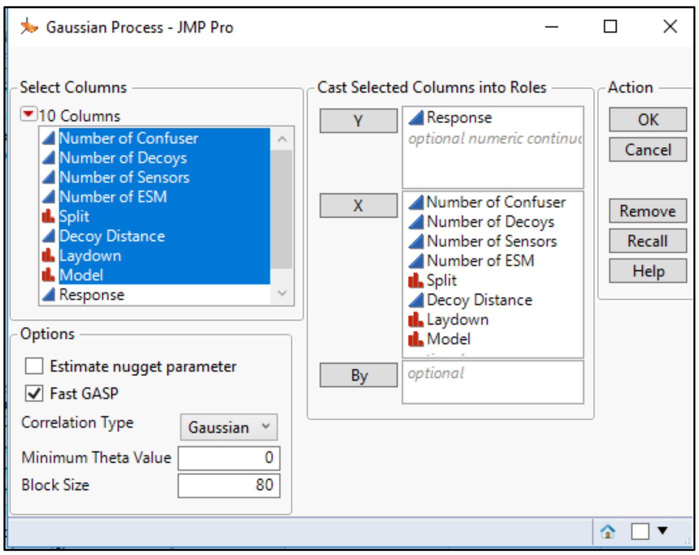
Remove Add Edit ☐ FDR ('^' denotes effects with containing effects above them)

The two terms with "." are inestimable.

5. To fit the Gaussian process model, select "Analyze -> Specialized Modeling -> Gaussian Process"

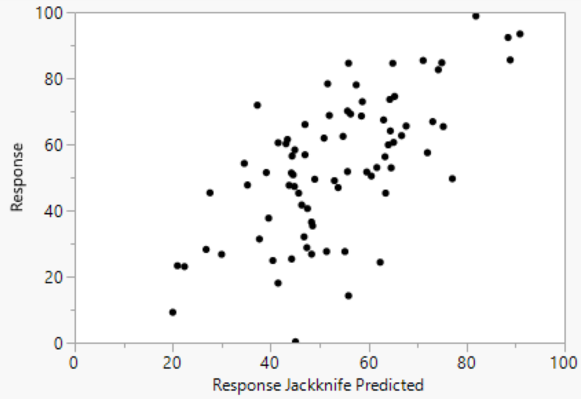


6. Load the Response into "Y" and the factors into "X". Select "OK."



Gaussian Process Model of Response

Actual by Predicted Plot



Model Report

Column	Theta	Total Sensitivity	Main Effect	Number of Confuser Interaction	Number of Decoys Interaction	Number of Sensors Interaction	Number of ESM Interaction	Decoy Distance Interaction
Number of Confuser	3.938579	0.790036	0.584207	.	1.222e-12	0.144308	0.018639	0.042882
Number of Decoys	3.3894e-7	3.664e-12	1.66e-12	1.222e-12	.	5.414e-13	8.612e-15	2.321e-13
Number of Sensors	2.5400625	0.2412184	0.0898922	0.144308	5.414e-13	.	0.0030171	0.0040011
Number of ESM	0.1709209	0.0316853	0.009179	0.018639	8.612e-15	0.0030171	.	0.0008502
Decoy Distance	2035.0444	0.0503701	0.0026369	0.042882	2.321e-13	0.0040011	0.0008502	.

When analyzing the Gaussian process model, the theta is the first parameter of interest. Recall, terms with small thetas have little impact on the prediction formula. Number of Decoys and Number of ESM have little effect on the response in this model. Next, main effect shows the total variation due to the factor alone and Number of Confuser adds the most variation. We can also look at the graph provided, which would be a straight line if the model fit perfectly. While the line is not perfect, it is clear to see a positive trend and a relatively strong correlation which indicates this model is a good fit.

References

Montgomery, D. C. (2017). *Design and analysis of experiments*. John Wiley & Sons.