

Enhanced data exploration through Reduced-Order Models

Derek W. Hengeveld¹ and Adam Biskner²
LoadPath, Albuquerque, NM, 87106

A reduced-order modeling approach to predict spacecraft output responses for a set of input factors was developed. It is based on Latin Hypercube sampling and Gaussian Process regression modeling. A test case, based on a simplified Orion Crew Exploration Vehicle Thermal Desktop® model, was developed and included nine input factors and seven output responses. Output response residuals, found for predicted temperatures, hydraulic power, and pressure, had means of 1.6 K, 0.2 W, 1.6 kPa and standard deviations of 5.0 K, 1.93 W, 18.2 kPa, respectively. Additionally, a software tool was developed to more easily utilize the reduced-order model and enhance the ability to explore the data.

Nomenclature

k	=	number of input factors
\mathbf{k}_*	=	column vector of covariances
k_{**}	=	autocovariance
\mathbf{K}	=	covariance matrix
l	=	hyperparameter
n	=	number of samples
x	=	input factors
\mathbf{x}_{tr}	=	input training data matrix
y	=	output response
\mathbf{y}_{tr}	=	output training data matrix
μ	=	predicated mean value
v	=	hyperparameter
σ	=	predicted standard deviation
σ_n^2	=	noise variance
CS	=	computer simulation
GP	=	Gaussian Process
LHS	=	Latin Hypercube Sampling
ROM	=	Reduced-Order Model
TCS	=	thermal control subsystem

I. Introduction

EVALUATING thermal control subsystem (TCS) performance can be done through physical and/or computer experiments. Although physical experiments provide empirical evidence, they can be expensive. Significant costs can be incurred during fabrication (i.e. time and money) and once built, results are limited by the time it takes to complete all experiments. Additionally, physical experiments are limited by the flexibility of a test setup; consequently, parametric studies can be challenging.

Computer experiments are an attractive option to overcome the challenges of physical experiments. Constructed correctly, computer experiments can easily accommodate parametric studies and are only limited by processing power. They are especially useful during design stages, although they too have inherent costs. Development of a

¹ Senior Engineer, 2309 Renard Place SE, Suite 101, Albuquerque, NM 87106.

² Principal Engineer, 2309 Renard Place SE, Suite 101, Albuquerque, NM 87106.

nominal thermal model can take days to weeks to develop with run times on the order of hours. Comparing and evaluating multiple TCS approaches, especially important in early design stages, amplifies these timelines. Considering the myriad TCS design approaches available, computational expense can become unwieldy. For example, consider a TCS design with five design parameters of interest each evaluated at 10 levels. Evaluating each combination of parameters at all levels would require 1.0E5 simulations. At 30 minutes per simulation, this would take over 5 years of computational time. This simple example illustrates the need for alternative approaches such as reduced-order modeling that enable evaluation of large design spaces.

II. Reduced-Order Model Development

Computer simulations have been targeted as a replacement to physical experiments for many applications¹ because they are more time and cost efficient and can provide valuable insight early in the design stages. However, computer experiments are often complex and computationally expensive. When built to evaluate several variables, these costs can become unacceptable. Because of this, spacecraft thermal analysis typically bounds a problem using stacked worst cases to avoid evaluation of numerous variable combinations. The result of this approach is over design and, as a result, increased cost. When properly developed, ROMs can overcome these challenges by providing a computationally efficient surrogate that accurately captures the effects of an underlying high-fidelity model. ROMs can then provide thousands of simulation results in seconds which enables evaluation of large design spaces. Stacked worst case design evaluation can be eliminated and more optimized designs can be realized.

The following provides an overview of the development of ROMs using sampling and data fitting schemes. This approach is considerably different than nodal reduction methods in that it relies on a set of high-fidelity model runs to generate the ROM. In doing this, the proposed approach is robust and can be easily applied to other problem classes, model types, and software packages. ROMs are based on computer simulations and therefore, should be designed to capture effects of the original model with the fewest number of runs. When properly developed, these surrogate models can predict responses at untested design points very quickly; the end result is a ROM that can provide thousands of simulations in seconds. The following provides an overview of the sampling (i.e. Latin Hypercube space-filling) and data fitting methods (i.e. Gaussian Process model fitting).

A. Latin Hypercube Sampling

Although full-factorial approaches examine all combinations of variables, they do so only at extreme values (i.e. design space boundaries). Consequently, interior points are overlooked and reduced-order models can often fail far from the boundaries. Therefore, space-filling designs were utilized to efficiently identify and evaluate interior points that would provide improvements in the reduced-order model. Space-filling designs attempt to efficiently evaluate a design space for a given number of computer simulations. Design approaches include: sphere packing, Latin Hypercube, uniform design, maximum entropy, and the Gaussian-Process IMSE designs¹. However, Latin Hypercube approaches are the most commonly used for computer experiments¹ and were used as the basis for the reduced-order models presented in the current work.

1. Overview

Latin hypercube sampling selects n different values from each of k variables X_1, \dots, X_k . The range of each variable is divided into n non-overlapping intervals on the basis of equal probability. One value from each interval is selected at random with respect to the probability density in the interval. The n values thus obtained for X_1 are paired in a random manner (equally likely combinations) with the n values of X_2 . These n pairs are combined in a random manner with the n values of X_3 to form n triplets, and so on, until n k -tuplets are formed.² This is the Latin hypercube sample. It is convenient to think of this sample (or any random sample of size n) as forming a $n \times k$ matrix of input where the i^{th} row contains specific values of each of the k input

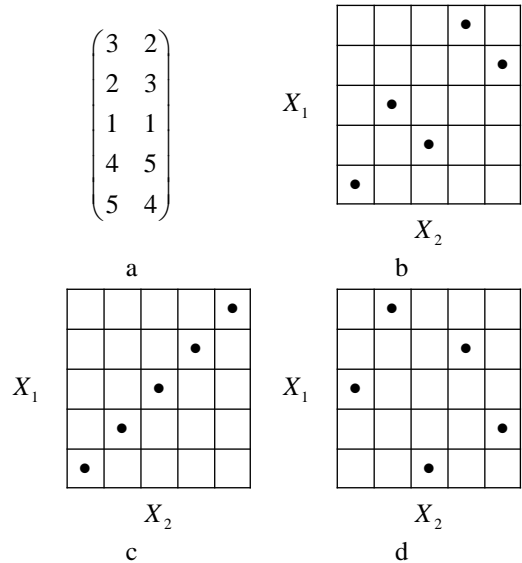


Figure 1. Sample Latin Hypercube. Many Latin Hypercube designs can be developed for a specific application, but some designs (d) work better than others (c) at filling a design space.

variables to be used on the i^{th} run of the computer model. Given $n=5$ and $k=2$, a sample (Figure 1a) is illustrated in Figure 1b. However, there are samples that fill the space better (Figure 1c) than others (Figure 1d).

2. Algorithm

A Latin Hypercube Sampling (LHS) algorithm was developed based on concepts of the Maximin Method and tested using Matlab. Through research and analysis, the Maximin Method has proven to be the best and most efficient method,³ as it is a simple and effective design to implement and the linearity of the method results in short run times. Sampling testing was completed as shown in Fig. 2. Each example includes two input factors and both 8 and 100 sampling points.

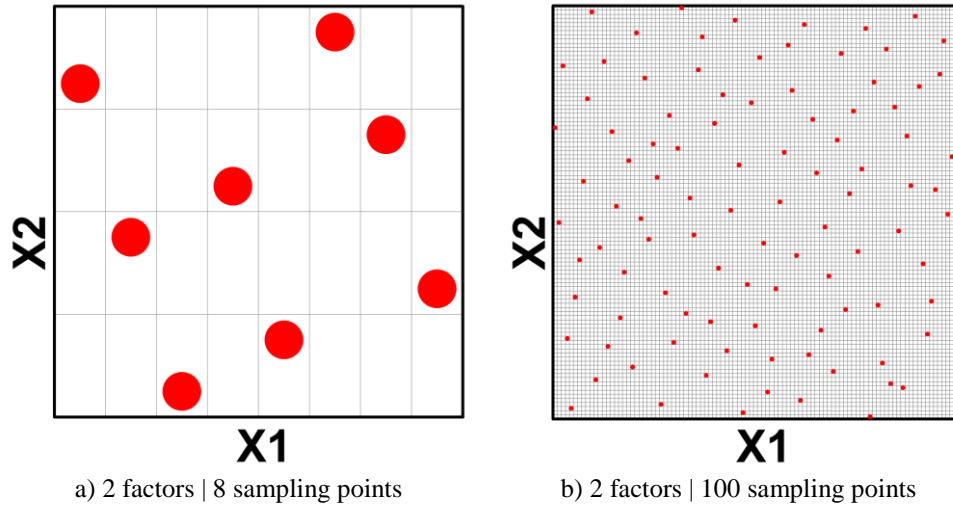


Figure 2. Results of Latin Hypercube sampling algorithm. A robust sampling algorithm was developed for multiple input factors and a broad range of sampling points.

B. Gaussian-Process Data Fitting

The use of Gaussian processes (GPs) for regression is a relatively new concept. In 1996, Williams and Rasmussen⁴ introduced the use of GPs to high dimensional problems that have been traditionally tackled using other modeling techniques such as neural networks and decision trees. GP modeling does not impose a specific model structure on the underlying function, $f(x)$, being modeled⁵. Instead, a Gaussian prior is placed on the range of possible functions that could represent the mapping of input factors x to output responses y . The Gaussian prior incorporates knowledge about the underlying function in the data where available, and is specified using the GP covariance function. As such, GP modeling is considered to be a non-parametric modeling technique, where the training data are used to discover the model properties in a supervised manner. However, some basic assumptions must be made about $f(x)$ and these are specified in the GP covariance function.

1. Overview

Consider an experiment with training data evaluated at n locations each defined by a k -dimensional vector (i.e. k input factors). For training data at the i -th location, $\mathbf{x}_{\text{tr}_i} = (x_{\text{tr}_i1}, \dots, x_{\text{tr}_ik})$, a given response is denoted by $y_{\text{tr}_i} = y(\mathbf{x}_{\text{tr}_i})$. Consequently, there is a $n \times k$ training data matrix, \mathbf{x}_{tr} . Outputs of these trials, $\mathbf{y}_{\text{tr}} = y(\mathbf{x}_{\text{tr}}) = (y_{\text{tr}_1}, \dots, y_{\text{tr}_n})'$, is an n -dimensional vector. For any location, the output is modeled as shown in Equation (1).

$$y(x_i) = \mu + z(x_i) \quad i = 1, \dots, n \quad (1)$$

The value μ is the overall mean and $z(x_i)$ is a Gaussian Process with $E(z(x_i)) = 0$, $\text{Var}(z(x_i)) = \sigma_z^2$, and $\text{Cov}(z(x_i), z(x_j)) = \sigma_z^2 R_{ij}(\mathbf{X}, \theta)$ ⁶. The resulting prediction equation contains one model term for each design point in the original experiment (i.e. training data). Introduced for computer experiments by Sacks, Welch, Mitchell, and Wynn⁷, this approach is desirable in computer experiments since they provide an exact fit to the training data and require only $k+1$ parameters.

The covariance function provides a relationship between training data. Although several approaches can be utilized for this correlation structure, the approach used was the squared exponential (SE) covariance function one of the most commonly used covariance functions¹ shown in Equation (2).

$$k(v, l, x_{tr})_{i,j} = v^2 \cdot \exp\left(\frac{-(x_{tr_i} - x_{tr_j})^2}{2 \cdot l^2}\right) \quad i = 1 \dots n \quad j = 1 \dots n \quad (2)$$

Here, v and l are hyperparameters that define the properties of the covariance function. The SE covariance function assumes that input points that are close together in the input space correspond to outputs that are more correlated than outputs corresponding to input points which are further apart. For example, if $x_i \cong x_j$, $k(x_i, x_j)$ tends towards its maximum, v^2 ; conversely if $x_i \neq x_j$, $k(x_i, x_j)$ tends towards its minimum, 0. The covariance matrix, \mathbf{K} , includes all training data as shown in Equation (3).

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & \dots & k_{1,j} \\ \vdots & \ddots & \vdots \\ k_{i,1} & \dots & k_{i,j} \end{bmatrix} \quad (3)$$

This equation becomes,

$$\mathbf{K} = \begin{bmatrix} v^2 \cdot \exp\left(\frac{-(x_{tr_1} - x_{tr_1})^2}{2 \cdot l^2}\right) & \dots & v^2 \cdot \exp\left(\frac{-(x_{tr_1} - x_{tr_n})^2}{2 \cdot l^2}\right) \\ \vdots & \ddots & \vdots \\ v^2 \cdot \exp\left(\frac{-(x_{tr_i} - x_{tr_1})^2}{2 \cdot l^2}\right) & \dots & v^2 \cdot \exp\left(\frac{-(x_{tr_i} - x_{tr_n})^2}{2 \cdot l^2}\right) \end{bmatrix}. \quad (4)$$

Consider training data consisting of four (i.e. $n = 4$) combinations of inputs, \mathbf{x}_{tr} , and outputs, \mathbf{y}_{tr} as shown in Equation (5).

$$\mathbf{x}_{tr} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad \mathbf{y}_{tr} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} \quad (5)$$

Assuming hyperparameters of $v = 1$ and $l = 1$, the covariance matrix, \mathbf{K} , becomes,

$$\mathbf{K} = \begin{bmatrix} 1 & 0.607 & 0.135 & 0.011 \\ 0.607 & 1 & 0.607 & 0.135 \\ 0.135 & 0.607 & 1 & 0.607 \\ 0.011 & 0.135 & 0.607 & 1 \end{bmatrix}. \quad (6)$$

This demonstrates the strong correlation between like values (i.e. training data near one another). Covariances between test and training data points is then defined as shown in Equation (7).

$$k_*(v, l, x_{tr})_i = v^2 \cdot \exp\left(\frac{-(x - x_{tr_j})^2}{2 \cdot l^2}\right) \quad i = 1 \dots n \quad (7)$$

A column vector of covariances between test and training data points is defined as shown in Equation (8).

$$\mathbf{k}_* = \begin{bmatrix} k_*(v, l, x_{tr})_1 \\ \vdots \\ \vdots \\ k_*(v, l, x_{tr})_n \end{bmatrix} \quad (8)$$

Finally, the autocovariance of the test input is defined as shown in Equation (9).

$$k_{**}(v, l, x) = v^2 \cdot \exp\left(\frac{-(x-x)^2}{2 \cdot l^2}\right) \quad (9)$$

Using the previously defined covariance values, the predicted mean and standard deviation for a given input value are found using Equations (10) and (11).

$$\mu(x) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}_{tr} \quad (10)$$

$$\sigma^2(x) = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_* + \sigma_n^2 \quad (11)$$

2. Hyperparameter Optimization

Before the output response y for a given input is found, the unknown hyperparameters of the covariance function (i.e. l , v , and a noise variance σ_n^2 assumed to be zero) must be optimized to suit the training data. This is performed via minimization of the log marginal likelihood⁸ given by

$$\log(p(\mathbf{y}_{tr} | \mathbf{x}_{tr})) = \frac{1}{2} \mathbf{y}_{tr}^T \mathbf{K}^{-1} \mathbf{y}_{tr} + \frac{1}{2} \log(|\mathbf{K}|) + \frac{n}{2} \log(2\pi) \quad (12)$$

The log marginal likelihood can be used to choose between different models. This equation is made up of a combination of $\frac{1}{2} \mathbf{y}_{tr}^T \mathbf{K}^{-1} \mathbf{y}_{tr}$ that determines the success of the model at fitting the output data, a model complexity penalty $\frac{1}{2} \log(|\mathbf{K}|)$, and a constant term that depends on the training data set size $\frac{n}{2} \log(2\pi)$.

3. Example case

A Gaussian Process Regression algorithm was developed in Fortran. This algorithm was tested using nominal training data consisting of eleven (i.e. $n = 11$) combinations of inputs, \mathbf{x}_{tr} , and outputs, \mathbf{y}_{tr} as shown in Equation (13).

$$\begin{aligned} \mathbf{x}_{tr} &= [0 \ 10 \ 20 \ 30 \ 40 \ 50 \ 60 \ 70 \ 80 \ 90 \ 100]^T \\ \mathbf{y}_{tr} &= [80 \ -20 \ 5 \ 30 \ 5 \ 5 \ 30 \ 40 \ 50 \ 60 \ 90]^T \end{aligned} \quad (13)$$

Using a hyperparameter combination $v = 500 \mid l = 500$, the Fortran algorithm was run from $x = 0$ to 100 . The results, showing training data and Gaussian Process results, are shown in Fig. 3a.

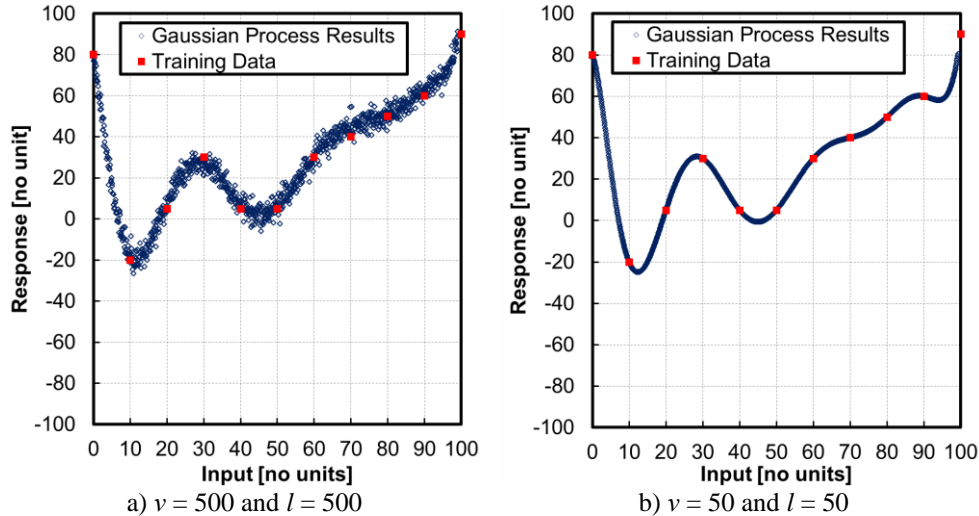


Figure 3. Sample Gaussian Process algorithm results. A robust data fitting algorithm was developed based on Gaussian Process methods and is sensitive to the selection of hyperparameters.

Based on minimization of Equation (12), the hyperparameters were found to be approximately $l = 50$ and $\nu = 50$. Using these values, and using estimates of standard deviation, the prediction plot is shown in Fig. 3b. These figures illustrate good agreement between training data and the Fortran algorithm. However, it's clear that changing hyperparameter values impacts this relationship; the results of Fig. 3b are clearly better than those of Fig. 3a and shows the importance of hyperparameter optimization.

III. Results and Discussion

As a test case, a simplified Orion Crew Exploration Vehicle (CEV) thermal model, developed in Thermal Desktop®, was utilized. This model was selected as it includes complexities that would provide a good test of the ROM approach. Although not presented here, the ROM that was developed could be used to evaluate optimal working fluids, impact of space temperatures, regenerator designs, radiator architectures, and the impact of applied heat loads.

The CEV thermal model consists of an external fluid loop and detailed heat rejection system (i.e. radiators) (Figure 4a). Simulating internal heat development of the crew module is done through a single heat source (i.e. symbol QLOAD). The fluid loop setpoint (i.e. temperature of FLOW.487) is controlled via a PID to control the flow (via a bypass loop) through a regenerative heat exchanger. Heat dissipation is rejected to a constant temperature environment.

The Orion CEV thermal model consists of several thermal submodels (e.g. radiator submodel) and one fluid submodel (i.e. FLOW). Figure 4b illustrates the thermal model and specifies key FLOW submodel components (e.g. lumps and paths). Details of components within the thermal model are illustrated in Figure 4b. Included are lump and path numbers at key locations within the model.

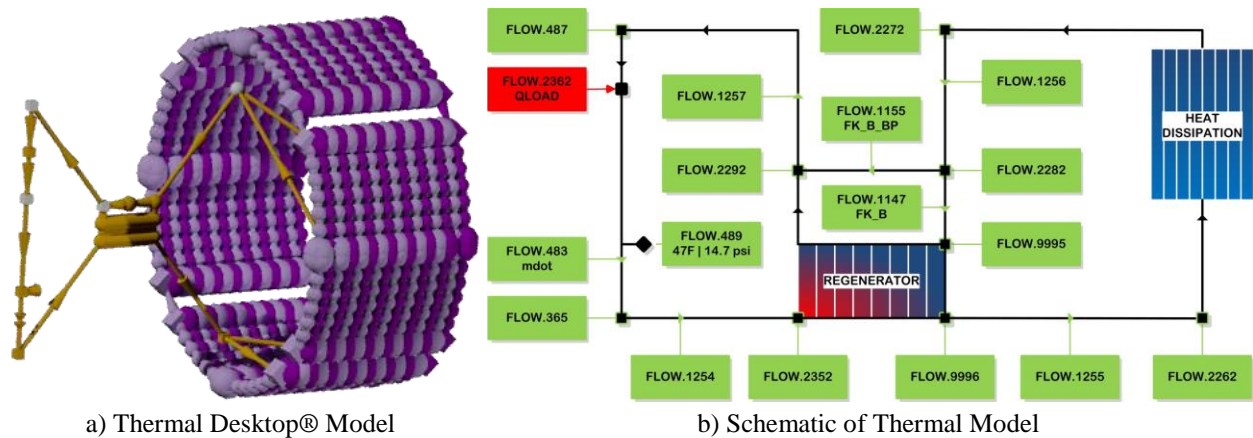


Figure 4. Illustration of Simplified Orion CEV Thermal Desktop® Model. A CEV thermal model, rendering shown in a), was used to demonstrate the utility of sampling and data fitting methods for developing ROMs. An illustration of critical nodes, showing locations of primary output responses, is shown in b).

A thorough analysis of potential system fluids was completed. This was done to help build new fluid property files for use during simulations. In order to provide accurate data when modeling each fluid, a mass flow rate (Mdot) was calculated. By using known values of mass flow rates from the 50/50 PGW, Multitherm -58, Galden, and HFE 7000, a common flow capacitance (Mdot*Cp) was calculated and compared. This average value (from Galden, 50/50, and Multitherm) turned out to be approximately 380 Btu/hr-R at ~20 °C. The flow capacitance from HFE 7000 was calculated. A list of mass flow rates and flow capacitances are shown in Table 1.

Table 1. Summary of Inputs to Thermal Desktop

Symbol	50/50 PGW	MultiTherm - 58	Dynalene HC-50	Anhydrous Ammonia	HFE 7000**	Galden HT170
Cp @ 20 °C (J/kg-K)	3395.90	2590.0	2701.0	4745.0	1288.2	957.0
Mdot* (lbm/hr)	468.0	610.8	588.4	334.9	1235.0	1665.0
Flow Capacitance (Mdot*Cp)	379.6	377.8	379.6	379.6	380.0	380.6

* Mass flow rates are changed in order to maintain the same flow capacitance.

** Temperature at 21.1 °C for HFE 7000.

Based on discussion with NASA personnel, evaluation of the thermal model, and results of a factor screening effort, the following input factors and corresponding ranges were selected for use in subsequent ROM efforts (Table 2). Also included are nominal values (i.e. values that were utilized in the supplied thermal model) and justification for selection.

Table 2. Summary of Input Factors

No.	Input Factor	Symbol Name	Range (Nominal Value)	Justification
1	Working Fluid	Not Applicable	Dynalene HC 50, Galden HT 170, HFE 7000	Discussion with NASA Technical Contacts
2	Regenerator Area per Node	Aheat_HFC	0.5 to 2.0 m ²	Bounds the range of values used within the thermal model
3	Space Temperature	TEMP_SPACE	0 K to 300 K	Discussion with NASA Technical Contacts (low temperature value)
4	Radiator Emissivity	Opt_Epsilon	0.7 to 1.0	---
5	Radiator Fin Efficiency	rad_fin_eff	0.7 to 1.0	---
6	Tube Inside Diameter	RadTubD	0.003175 m (0.125") to 0.005080 m (0.200")	Bounds the range of values used within the thermal model.
7	Fin-to-Tube Conductance	TContact	50 to 1000 (285)	Provides a large range around the nominal value. Range is much larger than that used for sensitivity study.
8	Regenerator Thermal Mass per Node	HX_THERMAL_MASS	500 to 4,000 J/K (1,450 J/K)	Provides a large range around the nominal value. Range is much larger than that used for sensitivity study.
9	Heatload	QLOAD	0 to 4,000 W	Provides a large range to help identify minimum/maximum heatloads that a system can accommodate.

Based on discussion with NASA personnel and evaluation of the thermal model, the following primary output responses were selected for use in subsequent reduced-order modeling efforts (Table 3).

Table 3. Summary of Primary Output Responses

No.	Output Response	Symbol Name	Description
1	Set-point Temperature	FLOW.487	Temperature of FLOW.487 at end of simulation. This will be either 1) steady-state temperature or 2) temperature at end of maximum simulation time.
2	Fluid Hydraulic Power	Varies	Calculated fluid hydraulic power based on 1) lump pressure differential, 2) densities, and 3) flow rate.
3	Pressure	FLOW.365	Pressure at FLOW.365
4	Pressure	FLOW.2262	Pressure at FLOW.2262
5	Pressure	FLOW.2272	Pressure at FLOW.2272
6	Flow Rate	---	System flow rate
7	Average Radiator ΔT	Varies	Average ΔT as a result of TContact of 7 radiators

In addition, the impact of QLOAD on a particular system configuration must be understood. This includes: A) the minimum heat load (i.e. QLOAD) to maintain set point (i.e. FLOW.487), B) the maximum heat load to maintain set point, and C) the ratio of these two values. Consequently, it was important to include a broad enough QLOAD range to capture the limiting conditions (i.e. A and B from above). A QLOAD range of 0 to 4,000 W was selected based on results of the sensitivity analysis.

Based on prior experience, a good approximation of number of sampling points is 2^k , where k is the number of input factors. Further, the regenerator area per node input factor (Aheat_HFC) was a FLOW (i.e. fluid submodel)

based variable. Because of this, updates to this variable in FloCAD could not be done under Thermal Desktop's dynamic mode.⁹ As a result, separate runs had to be initiated by a user to ensure that changes to Aheat_HFC were adequately captured. Consequently, this variable was modeled at three distinct levels. At each combination of Aheat_HFC level and fluid type, 2⁸ (i.e. 256) samples were simulated for a total of 768 samples per fluid. LH sampling points were found utilizing JMP® 11 by SAS Institute, Inc. Based on LH sampling and the developed high-resolution thermal model, training data was obtained. This data provided the foundation upon which the ROM was developed.

A. HFE 7000 Model Test Results

The HFE 7000 ROM predicted temperatures (i.e. set-point and average radiator ΔT) with a maximum residual mean of 0.6 K and standard deviation of 3.7 K. The model predicted fluid hydraulic power with a maximum residual mean of 0.02 W and standard deviation of 0.09 W. Finally it predicted pressures with a maximum residual mean of 0.08 kPa and standard deviation of 0.6 kPa with a maximum percent difference standard deviation of 0.6%. The ROM did not perform well in capturing time to steady-state and percent bypass as indicated by high residuals and % difference values. In fact, maximum and minimum percent bypass values were unrealistic (i.e. greater than 100% and less than 0%, respectively). However, these values were not as critical as others such as fluid hydraulic power and pressure values. Further results can be found in Table 4 and Figure 5 and Figure 6.

Table 4. HFE 7000 CS versus ROM Results for Six Output Responses (768 LH Sample Points)

	Set-point Temperature [K]	Fluid Hydraulic Power [W]	Pressure (FLOW.365) [Pa]	Pressure (FLOW.2262) [Pa]	Pressure (FLOW.2272) [Pa]	Average Radiator ΔT [K]	Time to Steady State [s]	Percent Bypass [%]
Max ROM	303.0	2.00	114,578	113,828	106,513	4.8	4,280	122.8
Min ROM	220.8	0.08	111,066	110,238	102,812	0.3	686	-23.0
Max CS	306.1	2.09	114,441	113,659	106,325	4.8	3,600	99.9
Min CS	224.6	0.14	111,414	110,635	103,337	0.1	760	0.3
Residuals (ROM-CS)								
Max	9.5	0.28	1,700	1,609	1,466	0.5	1,743	92.9
Min	-10.0	-0.36	-2,825	-2,833	-2,961	-0.3	-2,206	-59.1
Mean	0.6	0.02	81	80	76	0.0	-26	1.8
Std Dev	3.7	0.09	597	586	578	0.1	645	13.5
% Difference								
Max	3.6	30.2	1.5	1.5	1.4	425.5	200.1	35,527.2
Min	-3.6	-41.8	-2.5	-2.5	-2.8	-17.5	-61.5	-8,545.0
Mean	0.2	1.7	0.1	0.1	0.1	9.6	9.4	470.0
Std Dev	1.4	10.4	0.5	0.5	0.6	49.6	44.1	3,536.0

B. Factor Sweeps

To further assist in evaluating the accuracy of the models, factor sweeps were carried out to examine ROM performance over a range of input factor values. Included here are a sample of factor sweeps for Galden HT 170; others were carried out but not included here for brevity. For each factor sweep all factors were set at nominal values (Table 5) while one factor was varied over its entire range. This was done for QLOAD.

Table 5. Summary of Nominal Parameter Values for Factor Sweeps

Aheat_HFC [m ²]	TEMP_SPACE [K]	Opt_Epsilon [---]	rad_fin_eff [---]	RadTubD [m]	TContact [---]	HX_THERMAL_MASS [J/K]	QLOAD [W]
1.25	150	0.85	0.85	0.004128	525	2250	2000

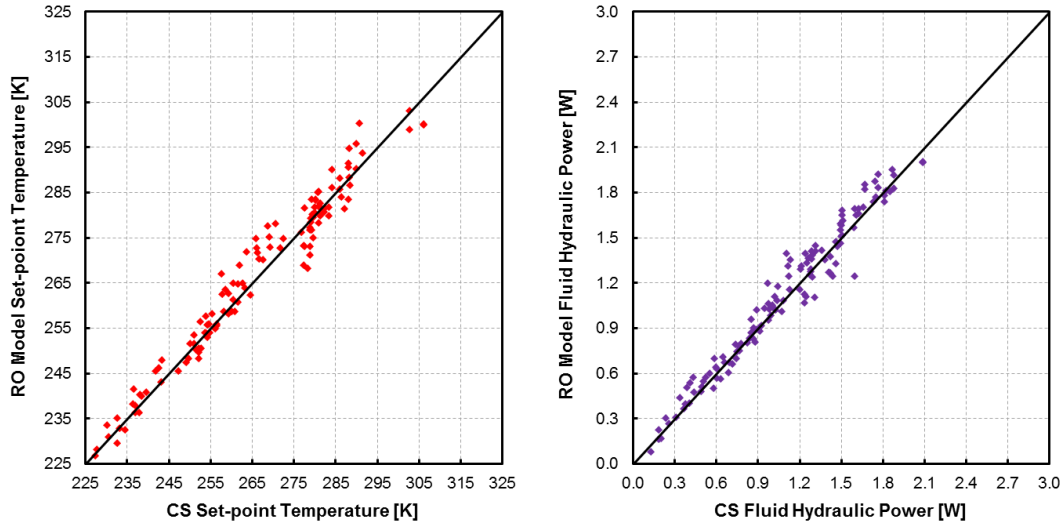


Figure 5. HFE 7000 ROM versus CS Plots for Two Output Responses: Set-point Temperature and Fluid Hydraulic Power (768 LH Sample Points). *The HFE 7000 ROM predicted temperatures and fluid hydraulic power with maximum residual means and standard deviations of 0.6 K and 0.02 W and 3.7 K and 0.09 W, respectively.*

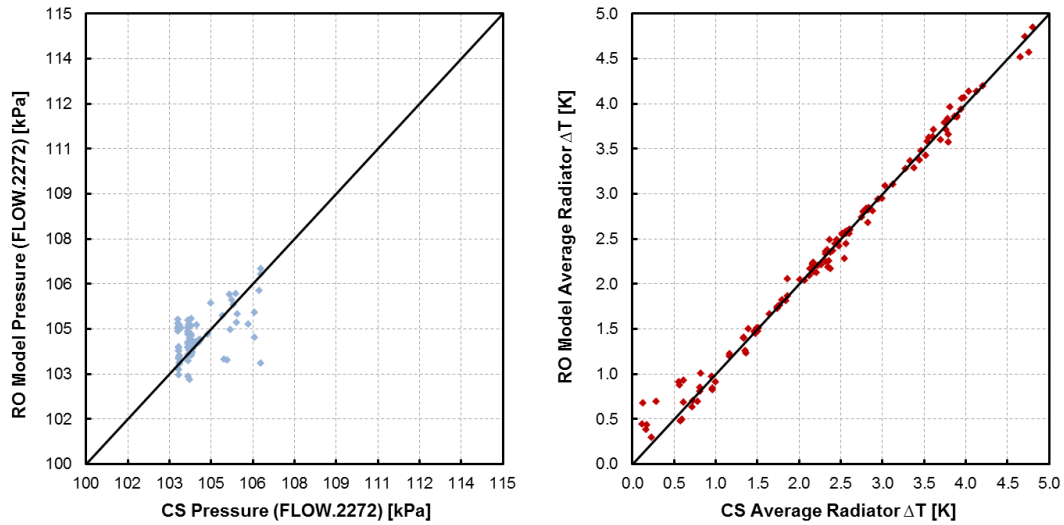


Figure 6. HFE 7000 ROM versus CS Plots for Two Output Responses: Pressure (FLOW.2272) and Average Radiator ΔT (768 LH Sample Points). *The HFE 7000 ROM predicted pressures with a maximum residual mean of 0.08 kPa and standard deviation of 0.6 kPa.*

For each factor sweep, an output response was calculated using the ROM and CS results. The comparisons are shown in Figure 7. These figures illustrate that the ROM provides a useful surrogate for smooth functions. However, it was found that discontinuities (e.g. Time to Steady State output response) challenges the ROM predictions. The same discontinuities were apparent with other working fluids (i.e. Dynalene HC 50 and HFE 7000).

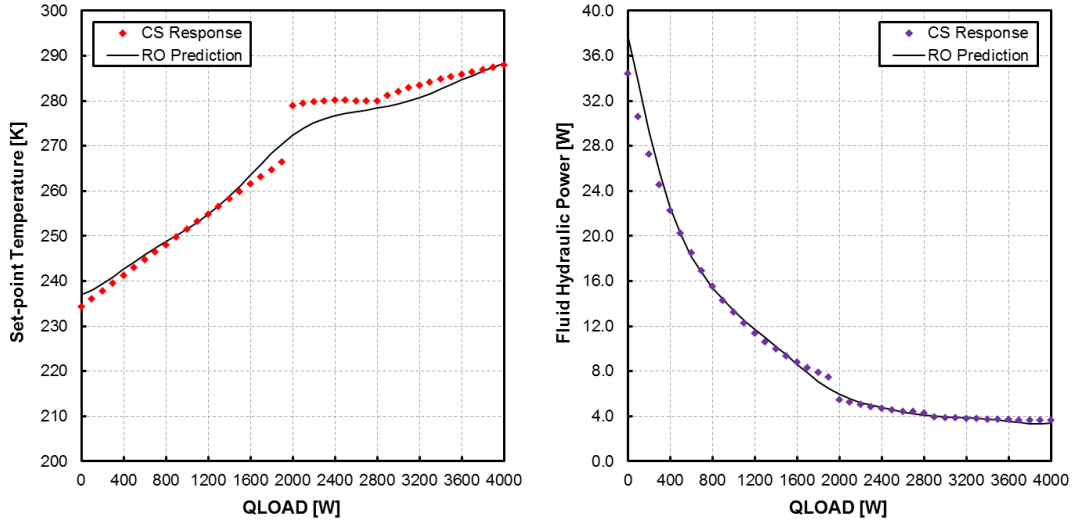


Figure 7. Galden HT 170 Set-point Temperature and Fluid Hydraulic Power CS Response and ROM Prediction versus QLOAD Input Factor. *Discontinuities challenged the prediction performance of the ROM.*

C. Sampling Size

An investigation into sampling size was started by adjusting the number of samples for the Galden HT 170 ROM. Samples were doubled from 768 to 1536 to determine the impact on ROM performance. As shown in Figure 8, increased sample numbers improved set-point temperature prediction. In fact the maximum residual means improved from 0.3 K to 0.2 K while standard deviation improved from 5.0 K to 3.8 K. However, achieving this improvement required twice as many runs and the associated computational expense. In the future, additional investigation into sample sizes and associated penalties will be carried out.

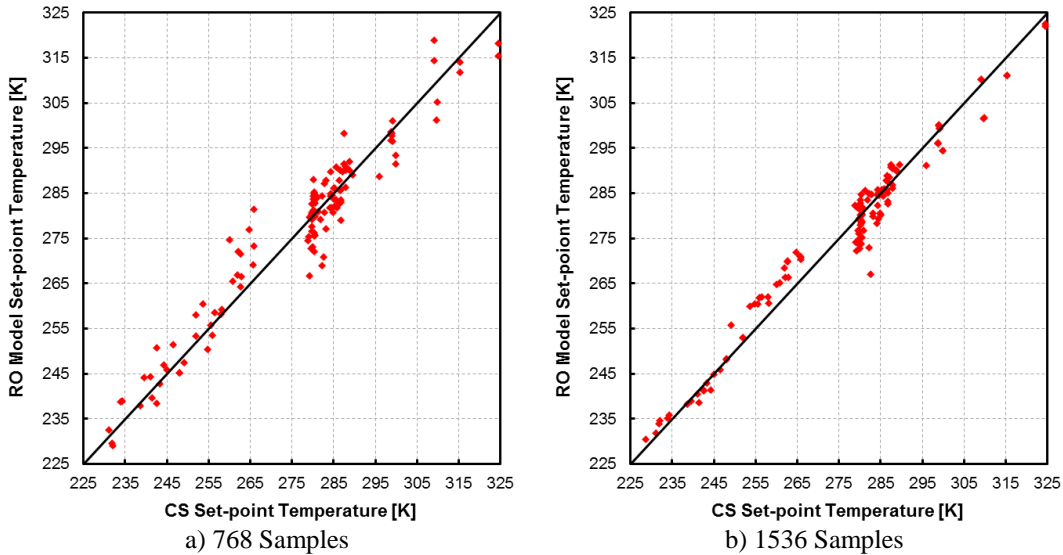


Figure 8. Galden HT 170 ROM versus CS Plots for Set-point Temperature Output Response (768 and 1536 LH Sample Points).

IV. Conclusions

The NASA CEV ROM for Dynalene HC 50, HFE RT 170, and HFE 7000 provide a useful surrogate to more computationally expensive computer simulations. Several observations were made:

- The ROM did a good job replicating temperature output responses. Across the three fluids, the ROM predicted temperatures (i.e. set-point and average radiator ΔT) with a maximum residual mean of 1.6 K and standard deviation of 5.0 K.
- The ROM did a good job replicating the hydraulic power output response. Across the three fluids, the ROM provided predictions with a maximum residual mean of 0.2 W and standard deviation of 1.93 W.
- The ROM did a good job replicating pressure output responses. Across the three fluids, the ROM predicted pressure (i.e. FLOW.365, FLOW.2262, and FLOW.2272) with a maximum residual mean of 1.6 kPa and standard deviation of 18.2 kPa with a maximum percent difference standard deviation of 7.3%.
- The ROM did a poor job of replicating output responses with discontinuities. This includes time to steady-state and percent bypass.

Although increasing sample size did improve performance of the ROM (i.e. set-point temperature), the improvement was small. ROM set-point temperature predictions improved from a maximum residual mean of 0.3 K and standard deviation of 5.0 K to 0.2 K and 3.8 K, respectively. This indicated that the 2^k sample size was an appropriate balance of precision/accuracy and computational expense.

The approach presented here provides a robust method for generating ROMs for a wide-range of models. These ROMs can then be used to more effectively explore a design space by leveraging their computational efficiency. Although not presented here, the NASA CEV ROM that was developed could be used to evaluate optimal working fluids, impact of space temperatures, regenerator designs, radiator architectures, and the impact of applied heat loads.

Acknowledgments

This material is based upon work supported by Small Business Innovative Research projects with NASA.

References

- ¹Jones, B., and Johnson, R. T. "Design and analysis for the Gaussian process model," *Quality and Reliability Engineering International* Vol. 25, No. 5, 2009, pp. 515-524.
- ²Swiler, L. P., and Wyss, G. D. "A User's Guide to Sandia's Latin Hypercube Sampling Software: LHS UNIX Library Standalone Version." Sandia Technical Report SAND2004-2439, 2004.
- ³Deutsch, J. L., and Deutsch, C. V. "Latin hypercube sampling with multidimensional uniformity," *Journal of Statistical Planning and Inference* Vol. 142, No. 3, 2012, pp. 763-772.
- ⁴Williams, C. K., and Rasmussen, C. E. "Gaussian processes for regression," 1996.
- ⁵Ebden, M. "Gaussian processes for regression: A quick introduction," *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 2008.
- ⁶Ranjan, P., Haynes, R., and Karsten, R. "A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data," *Technometrics* Vol. 53, No. 4, 2011, pp. 366-378.
- ⁷Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. "Design and analysis of computer experiments," *Statistical science*, 1989, pp. 409-423.
- ⁸Lynn, S. "Virtual metrology for plasma etch processes." NATIONAL UNIVERSITY OF IRELAND, MAYNOOTH, 2011.
- ⁹Panczak, T. D., Ring, S. G., Welch, M. J., Johnson, D., Cullimore, B. A., and Bell, D. P. "Thermal Desktop User's Manual, Version 5.8." C&R Technologies, 2015.