JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 44, No. 5, September–October 2007

Hammersley Sampling and Support-Vector-Regression-Driven Launch Vehicle Design

Mateen-ud-Din Qazi* and He Linshu†
Beijing University of Aeronautics and Astronautics, 100083 Beijing, People's Republic of China
and
Permoon Mateen‡
University of the Punjab, Lahore 54590, Pakistan

DOI: 10.2514/1.22200

Multidisciplinary design optimization of solid-fueled multistage space launch vehicles requires complex high-dimensional simulation models. To improve on the ability to efficiently explore and approximate large subspaces of these models, this research develops a new set of experimental designs for metamodeling with support vector regression. We propose to Latinize and improve the orthogonality of Hammersley sequence sampling for creating nearly orthogonal and excellent space-filling designs. Multiple measures are used to assess the quality of designed experiments. The designs are used to create metamodels of trajectory simulation of space launch vehicles using support vector regression. A hybrid genetic algorithm uses metamodels to identify minimum-launch-weight space launch vehicles. This approach resulted in an overall rapid and efficient design optimization scheme.

Nomenclature

A_e = nozzle exit area, m^2
A_t = nozzle throat area, m^2
A^* = inlet area ratio of the nozzle, $A_{\text{inlet}}/A_{\text{throat}}$
Al = percentage of aluminum powder in the grain
C = rank correlation matrix of W, penalty parameter of the error term
C_f = coefficient of thrust
C^* = characteristic velocity, m/s
D = diameter of the rocket motor, m
d = Euclidean distance
d_t = nozzle throat diameter, cm
F_i = thrust of the ith stage, N
H = Hadamard matrix
I_{sp} = specific impulse
I_{sp}^{o1} = ground specific impulse
$I_{\text{sp}}^{\text{vac}}$ = vacuum specific impulse
K = kernel function
k = number of factors or variables
k_s = propellant-grain-shape coefficient
L = stage length, m
L_{gi} = grain length, m
M = dimension of samples (number of variables)
ML_2 = modified L_2 discrepancy
m_{api} = attachment-part mass of the ith stage, kg
m_{cabi} = cable mass of the ith stage, kg
m_{cyi} = motor cylinder mass of the ith stage, kg
m_{c1}, m_{c2} = motor dome-ends mass, kg
m_{gni} = propellant mass of the ith stage, kg
$\dot{m}_{\text{gni}}$ = grain-mass-consumption rate of the ith stage, kg/s
m_{ig} = ignitor mass, kg
$m_{\text{in,cyi}}$ = cylindrical-section insulation-liner mass, kg
$m_{\text{in},c1}$, $m_{\text{in},c2}$ = fore and aft insulation-liner mass, kg
m_{j2i} = aft attachment mass of the ith stage, kg
$m_{\text{noz,ec}}$ = nozzle expansion-cone mass, kg
$m_{\text{noz,in}}$ = nozzle insulation mass, kg
$m_{\text{noz,sh}}$ = nozzle spherical-head mass, kg
m_{qi} = fore and aft skirt mass of the ith stage, kg
$m_{\text{str}i}$ = structure mass of the ith stage, kg
m_{TVC} = thrust-vector-control mass, kg
m_{01} = takeoff mass of the space launch vehicle, kg
$m_{0(i+1)}$ = takeoff mass of the upper-stage rocket, kg
N = number of samples or number of points in the design matrix, number of stages
n = number of runs, number of levels
p = integer number
p_c = chamber pressure of the rocket motor, bar
p_e = nozzle-exit-plane pressure, bar
p_h = atmospheric pressure at height h
p_0 = atmospheric pressure on the ground
Q = lower triangular matrix of size $k \times k$
R = radix notation
R_c = gas constant, $\text{J/kg} \cdot \text{mol} \cdot \text{K}$
R_{sub} = submerged ratio of the nozzle, $l_{\text{sub}}/l_{\text{nozzle}}$
R_u = curative radius of the nozzle, mm
r = kernel parameter
S_{ri} = burning surface area of the ith stage, m^2
T_c = temperature in the combustion chamber, K
t_{ki} = burn time of the ith stage, s
u_i = burning rate of the solid rocket propellant of the ith stage, m/s
W = design matrix of size $n \times k$
w = vector
X = design matrix
x_i = training vectors
$x_M(p)$ = Hammersley sequence of M-dimensional points
β = nozzle half-divergence angle, deg
γ = specific heat ratio of burned gas, kernel parameter
ε = nozzle expansion ratio
$\eta_{I\text{oa}}$ = ground impulse efficiency
$\eta_{I\text{va}}$ = vacuum impulse efficiency
η_v = volumetric loading fraction
μ_p = propellant mass fraction
ν = vector, support vector control parameter
ρ = maximum pairwise correlation
ρ_{gni} = propellant density, kg/m^3

Received 3 January 2006; accepted for publication 30 June 2006.

*Doctoral Student, School of Astronautics. Student Member AIAA.
†Professor, School of Astronautics.
‡Research Consultant, Department of Mathematics.

Γ = specific-heat-ratio function
ϕ_g = trajectory shaping factor, deg
$\phi(p)$ = inverse-radix-number function
ψ_n = smallest singular value of X^TX
ψ_1 = largest singular value of X^TX

Subscripts

c = motor chamber
e = nozzle exit
gn = propellant grain
i = stage numbers 1, 2, and 3
t = nozzle throat

Introduction

THE goal of this research is to provide new experimental designs and develop effective approximation metamodels that can enable analysts and designers to conduct more thorough investigations of complex simulation models. Conceptual designs of space launch vehicles (SLVs) cannot be conducted without the cooperation of numerous disciplines, including several analysis technologies: aerodynamic analysis, heating analysis, structural analysis, propulsion analysis, trajectory analysis, controls analysis, cost analysis, operations analysis, and so on. It is necessary to make good use of the analysis technologies in the various disciplines and to apply design optimization techniques. The application of optimization methods to multidisciplinary analysis is referred to as multidisciplinary design optimization (MDO) [1]. Many practical MDO methods have been widely applied in aerospace for overall designs of aircraft and space transportation vehicles [2]. Several analysis and design tools and integrated computer software have been proposed for space transportation systems [3,4].

Design and analysis are complementary activities. The design must support the desired analysis, and the analysis should derive as much information as possible from the allotted runs. Conducting a comprehensive experimental design on various intricate simulation models required for MDO involving high-dimensional input space is prohibitive. Despite the steady and continuing growth of computing power and speed, the computational cost of complex high-fidelity engineering analyses and simulations limit their use in important areas such as design optimization and reliability analysis. Statistical approximation techniques such as design of experiments (DOE) and response surface methodology (RSM) are the widely acknowledged solutions to minimize the computational expense of running such computer analyses and to circumvent many of these limitations.

Metamodeling is the process of building a "model of a model" to provide a fast surrogate for a computationally expensive computer code. A variety of metamodeling techniques have been developed in the past decade to reduce the computational expense of computer-based analysis and simulation codes. Reference [5] compared five experimental design types [Latin hypercube, Hammersley sampling sequence (HSS), orthogonal arrays, uniform designs, and randomly generated points] and four approximation model types (polynomial response surface, kriging, radial basis function, and multivariate adaptive response regression) in terms of their capability to generate accurate approximation for two engineering applications with typical engineering behavior and a wide range of nonlinearity.

Classical DOE approaches such as central composite design, Box–Behnken design, and full- and fractional-factorial designs generally put sample points at the extremes of the parameter space. In modern DOE, the space-filling designs such as quasi Monte Carlo (MC) sampling, orthogonal array sampling, and Latin hypercube sampling (LHS) are more commonly employed to accurately extract trend information. It is possible to exploit parallel computing, either on a multiprocessor computer or over a network, to compress the wall-clock time needed to complete the set of simulations prescribed by a DOE method.

Orthogonal arrays are an attractive class of sparse designs because they provide balanced (full factorial) designs for any projection into k factors. However, when these designs are projected into one dimension, points do overlap. Meanwhile, the classical experimental designs, in general, do not exhibit good projection capabilities, yielding several overlapping points when projected into two dimensions. However, Latin hypercube designs are better than classical designs at projecting into single dimensions.

One of the most widely used techniques for sampling is the MC sampling technique. Traditionally, MC has been applied to generate the samples of the stochastic variables. However, MC methods can result in large error bounds and variance. Of the numerous space-filling designs that exist, random Latin hypercube, random orthogonal array, maximin Latin hypercube, orthogonal-array-based Latin hypercube, uniform, orthogonal hypercube, and Hammersley sampling sequence are commonly used [6].

Reference [7] created an orthogonal Latin hypercube (OLH) that preserves the orthogonality among columns while attempting a good spread of points throughout the design space. The orthogonality guarantees that the quadratic and interaction effects are uncorrelated with the estimates of linear effects. It is worth noting that the orthogonality of these designs does not depend on the numerical values of the levels, hence new OLH designs can be computed by permuting the numerical values or reversing the signs of the columns. The OLH designs are constructed by purely algebraic means without the aid of computers; however, [7] does provide an algorithm to search for optimal OLH designs based on different selection criteria (minimum entropy and maximin spacing based on either Euclidean or rectangular distance).

Latin hypercube techniques are designed for uniformity along a single dimension in which subsequent columns are randomly paired for placement on a k-dimensional cube. Recently, an efficient sampling technique, the HSS, was developed in [8], which uses an optimal design scheme for placing the n points on a k-dimensional hypercube. This scheme provides a low-discrepancy experimental design and ensures that the sample set is more representative of the population, showing uniformity properties in multidimensions, unlike Monte Carlo, Latin hypercube, and its variants. It was found that the HSS technique is at least 3–100 times faster than the LHS and Monte Carlo techniques and hence is a preferred technique for uncertainty analysis, as well as for optimization under uncertainty. We modified this HSS technique and used it in our implementation of the metamodeling with support vector regression (SVR). Recently, various applications for robust designs have used HSS [9–11]. The strength of sequence methods (Halton, Hammersley, Sobol, etc.) is that they can produce fairly uniform point distributions even though samples are added one at a time to the parameter space. The one-at-a-time incremental sampling of these methods enables these methods to have better efficiency prospects than Latin hypercube sampling and centroidal Voronoi tessellation (CVT) types of methods in the area of error estimation and control [12].

We previously attempted a combination of Latin hypercube sampling and neural-network-based metamodeling for the conceptual design of the SLV [13–15]. Although neural networks are well able to approximate very complex models, they have the two disadvantages of 1) being a black-box approach and 2) having a computationally expensive training process [16].

One of the purposes of this research is to show the potential of a support vector machine (SVM) metamodeling capability for rapid design optimization of the SLV. SLVs have stringent orbital insertion conditions and flight path constraints. It was found that there is extreme complexity and strong coupling between the SLV flight trajectory and its overall system design. Simultaneous trajectory, aerodynamics, and propulsion system (including the propellant grain shape of solid-rocket-motor optimization) are some of the most important characteristics of the optimization problem in this paper.

Efficient Design of Experiments

The goal of DOE is to extract as much information as possible from a limited set of laboratory or computer experiments. To efficiently explore high-dimensional design space, the experimental designs should have the following desirable characteristics [17]:

1) approximate orthogonality of the input variables; 2) space filling, that is, the collection of experimental cases should be a representative subset of the points in the hypercube of explanatory variables; 3) ability to examine many variables efficiently; 4) flexibility in analyzing and estimating as many effects, interactions, and thresholds as possible; 5) requiring minimal a priori assumptions on the response; and 6) ease in generating the design.

Space filling or uniformity is a well-recognized property for several applications such as point placement for response surfaces (metamodeling) and sampling over the integration domain for statistical integration of function and even for irregular (non-hypercube) interpolation. Such space-filling (uniformity) measure is conceptually simple and intuitive on a qualitative level. The quantitative aspects of uniformity involve 1) the quality with which points are spaced relative to one another in the parameter space (are they all nominally the same distance from one another?) and 2) uniformity of point density over the entire domain of the parameter space (i.e., uniform coverage of the whole domain by the set of points, not just good uniformity within certain regions of the space). Achieving high sampling uniformity over the generic domain is an area of active research. Much effort has been applied to the problem of achieving uniform placement of N samples over M dimensions by modifying Monte Carlo sampling (quasi Monte Carlo), Latin hypercube sampling (distributed hypercube sampling, improved hypercube sampling, and optimal symmetric), Halton and Hammersley sampling, and CVT.

The purpose of the DOE is to gain information with the least effort, so that valid conclusions may be drawn. From a mathematical/statistical perspective, the purpose of DOE is to determine if various factors affect some response and to build models relating the response to the factors. Interaction means that the effects of one factor may often depend on the levels of another. In a well-designed (orthogonal) experiment, we can completely separate the effects of the factors. If the design is not orthogonal, the effect of the factors cannot be completely separated. Thus, orthogonal designs avoid undesirable feature interaction. Intuitively, it should be clear that one can extract the maximum amount of information regarding dependent variables from the experimental region (the region defined by the settings of the factor levels) if all factor effects are orthogonal to each other. If it is known in advance that factors of interest have negligible interaction between factors, then only space-filling (uniform) designs are desirable; otherwise (in most of the cases), orthogonal designs are desirable. A number of standard measures are used to summarize the efficiency of a design. D efficiency (D optimal) and A efficiency (A optimal) also reflect the orthogonality in the design. A-optimal and D-optimal designs are highly computing-intensive, especially for high-dimensional problems.

In short, the space filling (uniformity) and orthogonality are the two most important desirable properties in most of the cases. Two measures are used to assess the orthogonality of a design matrix. These measures are the maximum pairwise correlation and singular-valued-decomposition (SVD) condition number. The use of both measures provides a better ability to differentiate between the orthogonality of candidate designs. There are also two measures used to assess the space filling of a design matrix. These measures are the Euclidean maximum minimum distance between design points and, from uniform design theory, the modified L_2 discrepancy. The use of both measures provides a better ability to differentiate between the space-filling properties of candidate designs.

We propose a novel Latinized nearly orthogonal Hammersley sequence sampling (LNOHSS) technique. This approach provides better orthogonality and uniformity properties than other modern DOE techniques and, as a consequence, reduces the number of samples required to create effective metamodels.

Hammersley Sequence Sampling

Reference [18] gives the algorithm and pseudocode of HSS and compares the HSS and MC sampling. The algorithm that generates a set of N Hammersley points makes use of the radix-R notation of an integer. That is, a specific integer p in radix-R notation can be represented as

$$\begin{aligned} p &= p_m p_{m-1} \cdots p_2 p_1 p_0 \\ p &= p_0 + p_1 R + p_2 R^2 + \ldots + p_m R^m \end{aligned} \tag{1}$$

where $m = [\log_R p] = [(\ell_n\, p)/(\ell_n\, R)]$, and the square brackets denote the integer portion of the number inside the brackets. The inverse-radix-number function constructs a unique number on the interval [0,1] by reversing the order of the digits of p around the decimal point. The inverse-radix-number function is

$$\begin{aligned} \phi(p) &= 0 \cdot p_1 p_2 \ldots p_m \\ \phi(p) &= p_0 R^{-1} + p_1 R^{-2} + \ldots + p_m R^{-m-1} \end{aligned} \tag{2}$$

Finally, the Hammersley sequence of M-dimensional points is generated as

$$x_M(p) = \left(\frac{p}{N}, \varphi_{R_1}(p), \varphi_{R_2}(p), \ldots, \varphi_{R_{m-1}}(p) \right) \tag{3}$$

where $p = 0, 1, 2, \ldots, N-1$; and the values for $R_1, R_2, \ldots, R_{m-1}$ are the first $M-1$ prime numbers 2, 3, 5, 7, 11, 13, 17, 19, ... This approach generates a set of N points in the M-dimensional design space $[0, 1]^M$.

Measure of Orthogonality

An orthogonal design is desirable because it ensures independence among the coefficient estimates in a regression model. Orthogonality enhances our ability to analyze and estimate as many effects, interactions, and jump discontinuities as possible. Reference [7] constructed OLH to enhance the utility of Latin hypercube designs for regression analysis. In that OLH, every pair of columns has zero correlation, the elementwise square of each column has zero correlation with all other columns, and the elementwise product of every two columns has zero correlation with all other columns. These properties "ensure the independence of estimates of [the] linear effect of each variable" and the "estimates of the quadratic effects and bilinear interaction effects are uncorrelated with the estimates of the linear effect."

Two measures are used to assess the degree of orthogonality. One measure is the maximum pairwise correlation ρ of the column of a design matrix. The maximum pairwise correlation between two vectors, $\boldsymbol{v} = [v_1, v_2, \ldots, v_M]^T$ and $\boldsymbol{w} = [w_1, w_2, \ldots, w_M]^T$, is found by calculating the absolute value of Eq. (4) for all pairs of column vectors in the design matrix and then selecting the maximum of these values. A value of 0 is best (signaling orthogonality), and a value of 1 is worst (indicating that at least one column is a linear combination of the remaining columns).

$$\frac{\sum_{i=1}^{M} [(v_i - \bar{v})(w_i - \bar{w})]}{\sqrt{\sum_{i=1}^{M} (v_i - \bar{v})^2 \sum_{i=1}^{M} (w_i - \bar{w})^2}} \tag{4}$$

The second measure of orthogonality is a condition number of $X^T X$. The condition number is commonly used in linear algebra applications to examine the sensitivities of a linear system. Additionally, it can reveal the orthogonality of the proposed design matrix. A companion condition number is used in this research work, as shown in Eq. (5), generated from the SVD, defined using the 2-norm of the design matrix [17]. An orthogonal design matrix has a condition number of 1. A nonorthogonal design matrix has a condition number greater than 1.

$$\text{cond}_2(X^T X) = \frac{\psi_1}{\psi_n} \tag{5}$$

Measure of Space Filling

Reference [19] defines a *uniform design* as a design "that allocates experimental points [that are] uniformly scattered on the domain" and classifies uniform designs as space-filling designs. Such designs

are not concentrated in clusters or solely at corner points of the regions and thus provide coverage of the entire experimental region, and this facilitates broad exploration of the model. The modified L_2 discrepancy (ML_2, shown next) is used to measure space filling [19]. Given two designs, the design with a smaller ML_2 discrepancy has better space-filling properties.

$$ML_2 = \left(\frac{4}{3}\right)^k - \frac{2^{1-k}}{n}\sum_{d=1}^{n}\prod_{i=1}^{k}\left(3 - x_{di}^2\right) + \frac{1}{n^2}\sum_{d=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{k}[2 - \max(x_{di}, x_{ji})] \quad (6)$$

The second measure used in assessing space-filling properties of a design is the Euclidean maximin distance [17]. For a given design; define a distance list $\boldsymbol{d} = (d_1, d_2, \ldots, d_{[n(n-1)]/2})$, where the elements of $\boldsymbol{d}$ are the Euclidean intersite distance of the n design points, ordered from smallest to largest. The Euclidean maximin distance is defined as d_1, for which a larger value is better. A larger value of d_1 means that no two points are close to (within d_1 of) each other.

Development of New Experimental Designs

The HSS is more volumetrically uniform than the LHS set; the LHS points project more uniformly onto the coordinate axes. The projection of standard HSS will be improved by adopting the uniform projection onto the coordinate axes, just as with LHS. Thus, the new experimental design is termed as *Latinized* because it borrows this feature from LHS. Therefore, Latinized means hybridization of HSS and LHS, to have both lower discrepancy and higher volumetric uniformity (space filling).

A sequence of steps is devised that lead to an effective tradeoff between the concepts of near-orthogonality and space filling. This activity is not computer-intensive, and designs are created effectively. The general plan is to follow the main steps given in [7,17] with some modifications (applied to Hammersley sequence sampling instead of the Latin hypercube). The designs are constructed by adopting a few steps from current algorithms [7,17] (followed in our steps 4–10). We expanded on the number of variables that these designs can have through newly proposed steps 1–3. By doing this, one is able to significantly increase the number of variables that can be examined within a fixed number of runs (see Table 1). The number of runs required by [7,17] increases nonlinearly as the number of factors increases; however, in our designs, the number of runs only increases linearly. Thus, our proposed experimental designs are extremely useful for high-dimensional problems. The proposed algorithm generalizes for any arbitrary number of variables.

The difference in the number of variables examined in the LNOHSS and other similar designs grows dramatically as the number of variables to be explored increases. For example, the approach in [7] requires 4907 runs and [17] requires 129 runs to build an OLHC for 22 variables. The new LNOHSS designs require only 49 runs to examine 22 variables and are thus capable of examining many more variables than can other designs, while maintaining orthogonality.

The algorithm for creating a LNOHSS is summarized as follows:

Step 1) Determine the number of factors k, where k is an integer greater than two and k exists only if $\text{rem}(k, 4) = 0$ and only for the cases in which k, $k/12$, or $k/20$ is a power of 2. If the number of variables is other than 4, 8, 12, 16, 20, 24, 32, 40, 48, 64, ..., then round the required number of variables up to the nearest one of these numbers. Also determine the total number of levels n for each variable as $n = 2k + 1$.

Step 2) Generate a HSS (design matrix) by first using $k - 1$ prime numbers in Eq. (3).

Step 3) Construct matrix $\boldsymbol{M}$ by Latinizing the HSS; that is, for each coordinate direction k, simply sort the points by that coordinate, and then overwrite the original values by the values $1/(2 \times N)$, $3/(2 \times N), \ldots, (2 \times N - 1)/(2 \times N)$. Our hope is that the process of adjusting the point coordinates does not too severely damage the nice dispersion properties inherent in the HSS point placement. A Latinized HSS attempts to achieve good dispersion in two opposing senses: first in the Latin hypercube sense (which considers each dimension separately) and second in the HSS sense, which considers dispersion in the original high-dimensional space.

Step 4) Construct matrix $\boldsymbol{S}$ from the Hadamard matrix of size k. Hadamard matrices are matrices of 1 and -1 for which the columns are orthogonal ($H' \times H = n \times I$). A k-by-k Hadamard matrix with $k > 2$ exists only if $\text{rem}(k, 4) = 0$. The Hadamard function in MATLAB handles only the cases in which k, $k/12$, or $k/20$ is a power of 2.

Step 5) Construct matrix $\boldsymbol{T}$ as an element-by-element product of the matrices $\boldsymbol{M}$ and $\boldsymbol{S}$. The orthogonal design matrix is completed by augmenting $\boldsymbol{T}$ with its mirror image and center point row of zeros (appending below the original $\boldsymbol{T}$).

Step 6) Calculate the pairwise correlations and condition number.

Step 7) Improve the orthogonality of the design matrix by the following procedure: Each column element of the design is replaced with the element's rank $(1, 2, \ldots, n)$ within one column. Consider only those realizations of $\boldsymbol{W}$ for which $\boldsymbol{C}$ is positive definite. The basic idea is to transform $\boldsymbol{W}$ into a set of uncorrelated variates. A Cholesky factorization scheme is used to determine $\boldsymbol{Q}$. Then let $\boldsymbol{D} = \boldsymbol{Q}^{-1}$ and $\boldsymbol{C} = \boldsymbol{Q}\boldsymbol{Q}^T$ such that $\boldsymbol{D}$ has the property $\boldsymbol{D}\boldsymbol{C}\boldsymbol{D}^T = \boldsymbol{I}$. The original $\boldsymbol{W}$ is then transformed into a new matrix, $\boldsymbol{W_B}$ ($n \times k$ matrix), using $\boldsymbol{W_B} = \boldsymbol{W}\boldsymbol{D}^T$. The elements in each column of matrix $\boldsymbol{W_B}$ are replaced by their rank order $(1, 2, \ldots, n)$. Repeat this process (step 7 only) until there is no further decrease in the maximum pairwise correlation. Finally, to reconstruct the design matrix, the ordered ranks in the final $\boldsymbol{W}$ are then mapped back into the original input-variable values.

Step 8) Calculate the maximin distance and ML_2 discrepancy for the design matrix.

Step 9) If a number of variables other than 4, 8, 12, 16, 20, 24, 32, 40, 48, 64, ... is required, construct each of the possible combination of columns (having the appropriate number of desired variables) and calculate the maximin distance and ML_2 discrepancy. Choose the design with the minimal rank sum over the two measures. The purpose is to remove the worst columns.

Step 10) Additional design points can also be generated from the original design matrix such that near-orthogonality is maintained and space filling is improved. The original design matrix has its column permuted. This permuted design matrix is then appended vertically to the original design matrix containing n runs. The center point run is redundant and not repeated. Thus, the number of runs is increased to $2n - 1$. The encouraging result is the likely reduction in the maximum pairwise correlation. Thus, when selecting columns to permute, it seems wise to emphasize space filling. To find the best permutation of columns to be appended, calculate the maximin distance and ML_2 discrepancy for each permutation. Rank the designs according to these measures. Choose the design with the minimum rank sum over the two measures.

Because the authors are unaware of any published literature on uniform designs with this number of variables and levels, no comparison can be made, but the proposed designs does exhibit excellent orthogonality and space-filling properties. As a means of comparison, 1000 Latin hypercube with 20 variables, each with 41 levels, are generated and their average orthogonality and space-filling properties are compared with LNOHSS. The LNOHSS design is considerably better in all measures than an average Latin hypercube, as shown in Table 2.

Table 1 Comparison of the number of variables examined by different methods

Number of experiments	LNOHSS	OLHC ([17])	OLHC ([7])	Percentage increase
17	**8**	7	6	14%
33	**16**	11	8	45%
65	**32**	16	10	100%
129	**64**	22	12	191%

Table 2 Comparison of LNOHSS with Latin hypercube; 20 variables, 41 levels, $(N_O^{LHSS})_{20}^{41}$

Property	LNOHSS	Latin hypercube (average of 1000)	Percent better
Maximum pairwise correlation	0.0267	0.45	94%
Average condition number	1.089	4.59	76%
Maximin distance	2.326	1.15	102%
ML_2 discrepancy	47.6	54.9	13.3%

Table 3 Nearly orthogonal designs for fewer than 20 variables derived from $(N_O^{LHSS})_{20}^{41}$

Desired variables	Deleted columns	Maximum pairwise correlation	Condition number	ML_2	Maximin distance
19	10	0.0267	1-0884	30.1364	2.2478
18	4,19	0.0251	1.0830	19.0732	2.0646
17	1, 13, 14	0.0267	1.0817	12.1043	2.0056

Our proposed LNOHSS is deterministically set and it can be created offline and tabulated. The LNOHSS experimental design is mainly noniterative; only step 7 is repeated for improvement of orthogonality, which quickly converges in a few iterations. Thus, our designs are easier to generate than the designs of [7,17], because their designs are nondeterministic and use iterative approaches. They select the final design after comparing and evaluating an extremely large possible set of candidate designs. Our resulting designs do have near-orthogonality and excellent space-filling properties. Finally, before the experiment, there are no assumptions made concerning the variables.

Step 9 (e.g., the cases in which fewer than 20, but more than 16, variables are required) is considered next. Each of the possible combinations of columns from design matrix is examined by calculating the minimax distance and ML_2 discrepancy as the columns are deleted. Table 3 summarizes the results for the 41-run case when the 17 to 19 variables are desired. Table 3 provides designs that have good orthogonality and space-filling properties. These experimental designs are now used to generate effective metamodels for design optimization of the SLV.

Multidisciplinary Analysis and Design of the SLV

The SLV considered for this study is made up of three sequentially staged solid-fueled rocket motors. The mission is to insert a payload of 500 kg to a circular low Earth orbit (LEO) of 500 km. Table 4 shows the 19 design variables that we have to optimize: geometric parameters; propulsion parameters, including the grain-shape parameter; and flight-trajectory parameters. One of the features of this study is that we also optimize ascent trajectory at the conceptual design phase.

Figure 1 shows the overall analysis process. If the weight and volume of payload to orbit and a set of values of design variables are known, each analysis is conducted sequentially, as exhibited in Fig. 1. Analysis in this paper refers not only to the calculation of output values for the following analysis based on the values of design variables and the input values obtained from the preceding analysis, but it also refers to the computation of the function values of constraint conditions, which the variables must satisfy. The objective function of the optimization is to minimize the launch weight. Thus, we give a performance index under some constraints and define an optimization problem. The following paragraphs summarize the analysis methods and constraint conditions in the optimization problem.

Table 4 Design variables of vehicle configuration, propulsion and flight trajectory

Variable	Description
μ_{pi}	Propellant mass fraction
D_i	Diameter of the ith stage motor, m
ϕ_g	Trajectory shaping factor, deg
p_{ci}	Chamber pressure of the rocket motor, bar
p_{ei}	Nozzle-exit-plane pressure, bar
u_i	Burning rate of the solid propellant, mm/s
k_{si}	Propellant-grain-shape coefficient

Vehicle Definition

Solid propellant grain is considered for all three sequentially stacked stages of the SLV. The shape of payload fairing is already assumed and not optimized. Rocket motors have ellipsoidal dome ends. Though the number of stages should also be one of the design variables, in this paper, it is fixed as three. We constrain the flight performance parameters of the maximum value of axial overload and maximum angle of attack to be below 10 g and 5 deg, respectively.

Propulsion Analysis

The burning surface area of the propellant grain mainly dictates the performance of the propulsion system in a solid-fuel rocket motor. We were not restricted to a particular shape of grain at the conceptual design level; rather, a variable k_{si} is used to represent the S_{ri} of grain as a function of L_i and D_i:

$$k_{si} = \frac{S_{ri}}{L_i D_i} \tag{7}$$

The amount of propellant is calculated from the design variables and propulsion analysis. The burn time and grain-mass-consumption rate are also calculated:

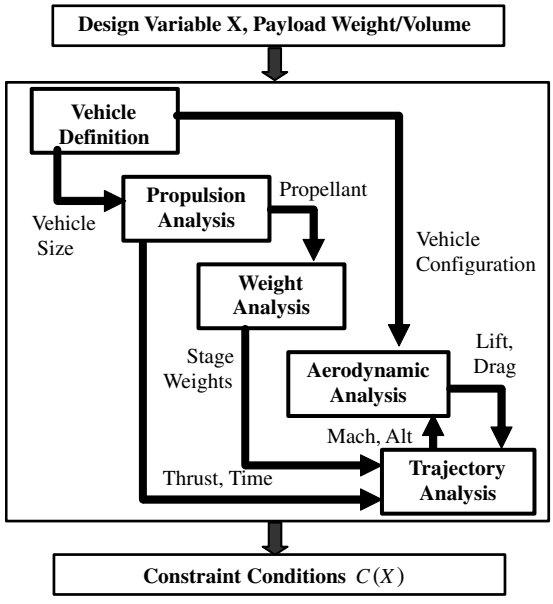

Fig. 1 Analysis process.

$$m_{\mathrm{gn}i} = \frac{\pi}{4}\rho_{\mathrm{gn}i}\eta_i L_i D_i^2 \tag{8}$$

$$t_{ki} = \frac{\pi\eta_i D_i}{4u_i k_{si}} \tag{9}$$

$$\dot{m}_{\mathrm{gn}i} = \rho_{\mathrm{gn}i} u_i k_{si} L_i D_i \tag{10}$$

Theozzle throat area, expansion ratio, and exit area are calculated as

$$A_t = \frac{\rho_{\mathrm{gn}i} u_i A_{b,\max}}{\Gamma_o p_{c,\max}}\sqrt{R_c T_c} \tag{11}$$

$$\varepsilon = \frac{\Gamma_o}{(p_e/p_c)^{\frac{1}{\gamma}}\sqrt{2\gamma/\gamma - 1[1-(p_e/p_c)^{\frac{\gamma-1}{\gamma}}]}} \tag{12}$$

$$A_e = A_t\varepsilon \tag{13}$$

$$\Gamma_o = \sqrt{\gamma}(2/\gamma+1)^{\frac{\gamma+1}{2(\gamma-1)}} \tag{14}$$

It is recommended to use mass flow rate and specific impulse for the calculation of thrust:

$$F_1 = I_{\mathrm{sp}gi}^{o1}\dot{m}_{gi} + (p_o - p_h)A_{e1} \tag{15}$$

$$F_{2\cdots N} = I_{\mathrm{sp}}^{\mathrm{vac}}\dot{m}_{gi} - p_h A_{ei} \tag{16}$$

$$I_{\mathrm{sp}}^{01} = \eta_{I\mathrm{oa}} C^* C_f \quad \text{and} \quad I_{\mathrm{sp}}^{\mathrm{vac}} = \eta_{I\mathrm{va}} C^* C_f \tag{17}$$

$$C^* = \frac{\sqrt{R_c T_c}}{\Gamma} \tag{18}$$

$$C_f = \Gamma\sqrt{\frac{2\gamma}{\gamma-1}\left[1-\left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right]} + \frac{A_e}{A_t}\left(\frac{p_e - p_c}{p_c}\right) \tag{19}$$

The ground impulse efficiency and vacuum impulse efficiency are taken as

$$\eta_{I\mathrm{oa}} = 1.0502 + 0.0111\,\ell n\, d_t - 0.0328\,\ell n\,\beta - 0.254Al - 0.00617\,\ell n\,\varepsilon \tag{20}$$

$$\eta_{I\mathrm{va}} = 1.949 + 0.0122\,\ell n\, d_t - 0.0465\,\ell n\,\beta - 0.230Al - 0.00448\,\ell n\,\varepsilon + 0.0153\,\ell n(1-R_{\mathrm{sub}}) - 0.00842\,\ell n\,A^* - 0.005\,\ell n\,P_c + 0.00333\,\ell n\,R_u \tag{21}$$

Weight Analysis

The weight analysis computes the component weights of the rocket motor and their combined gross weight using the geometric and performance parameters in the design variables; the amount of propellant is given by the propulsion analysis. The takeoff mass of the multistage solid launch vehicle is written as the sum of the weights of grain and structure mass [20].

$$m_{0i} = m_{0(i+1)} + m_{\mathrm{gn}i} + m_{\mathrm{str}i} \tag{22}$$

Introducing propellant mass fraction $\mu_{pi} = m_{\mathrm{gn}i}/m_{0i}$ as the ratio of the grain mass of the stage to the takeoff mass of the rocket and putting it in preceding equation, we get

$$m_{\mathrm{gn}i}\left(\frac{1}{\mu_p} - 1\right) - m_{0(i+1)} - m_{\mathrm{str}i} = 0 \tag{23}$$

The idea is to calculate $m_{\mathrm{gn}i}$ and $m_{\mathrm{str}i}$ as functions of grain length. This single nonlinear equation with only one unknown variable is then solved, thus determining the required L_{gi}. The structure mass shown in Eq. (24) is composed of component masses. The expressions for these individual components are given in [15].

$$m_{\mathrm{str}i} = m_{\mathrm{cy}i} + m_{c1i} + m_{c2i} + m_{qi} + m_{j2i} + m_{\mathrm{in},c1i} + m_{\mathrm{in},c2i} + m_{\mathrm{in,cy}i} + m_{\mathrm{noz,ec}i} + m_{\mathrm{noz,sh}i} + m_{\mathrm{noz,in}i} + m_{igi} + m_{\mathrm{TVC}i} + m_{\mathrm{cab}i} + m_{\mathrm{ap}i} \tag{24}$$

Aerodynamic Analysis

The study involves estimating the vehicles' aerodynamic properties in the different flowfield regions that the SLV encounters during atmospheric flight that ranges from subsonic to hypersonic speeds. The aerodynamic analysis incorporates the U.S. Air Force Missile DATCOM 1997 (digital). It is capable of quickly and economically estimating the aerodynamics of a wide variety of configuration designs, and it has the predictive accuracy suitable for conceptual and preliminary designs. The trajectory analysis depends on the aerodynamic analysis; that is, the aerodynamic analysis sends the trajectory analysis to the results obtained from interpolation of the values in the tables, and these results are used to compute the flight trajectories. The approach used in this study is simpler and of lower fidelity than the latest sophisticated computation methods, but they are useful in that the aerodynamic analysis, by which the optimization process computes the aerodynamic characteristics repeatedly, needs only a short computation time [21].

Trajectory Analysis

This study implements a three-degree-of-freedom trajectory analysis [22]. State variables are altitude, velocity, flight path angle ϑ, and mass. The trajectory analysis computes state variables by integrating equations of motion and examining constraint conditions during the flight. One of the characteristics in this study is to optimize the flight trajectory with the other design variables, as already described. The trajectory analysis, however, must deal with dynamic variables that are dependent on time, whereas the other design variables are static and independent of time [23].

The flight trajectory of the SLV is composed of several phases, as described next. Each stage shuts down, one after another, and separates to shed inert mass. Because the equations of motion change discontinuously at these shutdown points, the trajectory is divided into intervals, the number of which corresponds to the number of stages. Initial and terminal times of the burn intervals are given by the propulsion analysis.

After vertical launch, the SLV accelerates by the power of the first-stage rocket motor and flies vertically for 5 s before the next phase of pitchover. The pitchover phase lasts up to the start of the transonic phase. The angle of attack is constrained to approach zero before the start of the transonic phase (Mach = 0.8) and remain zero until the end of the transonic phase (Mach = 1.3). The launch-maneuver control angle ϕ_g is one of the design variables, and it represents the pitch angle at 40 s. A fourth-order curve is fitted from 90 deg at 5 s to the launch-maneuver control angle at 40 s. This curve represents the programmed pitch angle for the launch-maneuver phase.

The powered-ascent phase starts from the end of the transonic phase up to the shutdown of the first- and second-stage rocket motors. We constrain the flight path angle of the SLV to decrease monotonously. At the end of the second stage, the final stage and payload separates and zooms up, with no thrust in an elliptical orbit, for which the apogee is 500 km. Finally, an apogee kick puts the payload into a 500-km circular LEO. The constraint conditions in these phases are the angle of attack, dynamic pressure, normal overload, and body axial overload.

Support-Vector-Regression-Based Metamodeling

The simulation-based design approach may rely heavily on complex computer analysis and simulation codes such as finite element analysis and computational fluid dynamic analyses to improve the product design. These time-consuming and expensive analyses are repeatedly invoked during the optimization process, often driving the design exploration and multidisciplinary optimization execution times to prohibitively long durations. Thus, metamodels, or approximation of the most time-consuming analysis, with lower computational requirements are an attractive option.

Any function that represents the trends of a response over the range of the design variable is called a response surface. A response surface is the true unknown response trend, and response surface approximation denotes a user-defined function that models the response trend. Synonyms for response surface approximation include model, metamodel, surrogate model, and approximation model.

The RSM offers a systematic flexible approach for studying the parameter space efficiently by fitting a polynomial to system output. For complex systems with a large number of design variables, the RSM has limited applications. Artificial neural networks (ANNs) are considered as universal approximators, but they have a computationally expensive training process and they may be trapped in local optima, resulting in poorly trained networks. A kriging model consists of two parts: The first part, global behavior, is typically a linear regression model. The second part, the local contribution, is a statistical process that takes into account the covariance among sampled points within the immediate neighborhood. Because the global term is a linear regression approach, it requires a model to be preassumed, which is an inherent potential drawback. Selection of the size of neighborhood for the second part is also an issue. RSM, neural networks, and kriging have a curse of dimensionality.

This research investigates the potential of SVR for complex and high-dimensional metamodels such as most time-consuming trajectory analysis modules in SLV design. In the last section, we compared the performance of the SVM and ANNs. Reference [24] reviewed computational theory behind the SVM and its approximation is compared against the four metamodeling techniques (response surface methodology, kriging, radial basis functions, and multivariate adaptive regression splines) using a testbed of 26 engineering analysis functions.

Several representative low-fidelity analysis mare integrated in our approach. The final results are, of course, only as good as the supporting analysis code. These analysis modules are primarily nonlinear in nature and can be replaced by high-fidelity codes. Our proposed methodology can also handle these high-fidelity codes in the same manner. Our proposed method is capable of capturing nonlinearity in high-dimensional problems. In the next section, a trajectory metamodel using the support vector machine is developed to replace the most time-consuming trajectory analysis module. The trajectory analysis module uses a set of differential equations that are solved for whole flight duration for each candidate design. Trajectory optimization at the system design level is one of the significant features of this work.

Support Vector Machine

The SVM is a principled and very powerful method that, in the few years since its introduction, has already outperformed most other systems in a wide variety of applications.

The SVM features machine learning that stems from the statistical learning theory (SLT) framework and has proven successful in many real world problems. The performance of model is proportional to the performance on the actual experimental data, but weighted by a complexity term (e.g., the number of parameters of the model). The SLT provides a quantitative value for this complexity term and therefore allows one to effectively balance the complexity of the model with its ability to correctly fit the data. The main features of the method are sketched in the following, and the underlying theory and details can be found in [25].

Given a training set of instance-label pairs (x_i, y_i), $i = 1 \cdots l$, where $x_i \in R^n$ and $y\{1, -1\}^l$, the SVM requires the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \begin{cases} y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \tag{25}$$

where training vectors x_i are mapped into a higher (may be infinite) dimensional space by the function ϕ. Then the SVM finds linear separating hyperplanes with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is called the kernel function. The four basic kernels are linear $K(x_i, x_j) = x_i^T x_j$; polynomial $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$; radial basis function (RBF) $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$; and sigmoid kernel $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.

The problem is cast into a constrained quadratic programming problem that has one minimum that can be found in polynomial time if a positive definite kernel is used. This is a craved peculiarity that favors the method with respect to the classical ANNs. In fact, the ANN is plagued by local minima. One of the main features of the method is that the solution of the optimization problem leads to a sparse representation of the solution with respect to the kernel basis. Only some of the weights realizing the minimum are nonzero, and the related data points are called the *support vectors*.

The architecture of the system does not have to be determined before training. Input data of any arbitrary dimensionality can be treated with only linear cost in the number of input dimensions. A unique solution is found after training as a solution of the quadratic programming problem. Capacity can be controlled effectively through the regularization functional.

Support Vector Regression (ν – SVR)

SVR is a particular implementation of SVM. Given a set of data points $\{(x_1, z_1), \ldots, (x_l, z_l)\}$ such that $\boldsymbol{x}_i \in R^n$ is an input and $\boldsymbol{z}_i \in R^1$ is a target output, use ν to control the number of support vectors. The primal form of support vector regression is

$$\min_{w,b,\xi,\xi^*,\varepsilon} \frac{1}{2} w^T w + C\left(\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right) \times \begin{cases} (w^T \phi(x_i) + b) - z_i \leq \varepsilon + \xi_i \\ z_i - (w^T \phi(x_i) + b) \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \qquad i = 1, \ldots, l, \varepsilon \geq 0 \end{cases} \tag{26}$$

The dual problem solved is shown in Eq. (27), and the approximate function is given in Eq. (28):

Table 5 MAE and MSE of metamodels trained with 656 samples of LNOHSS

Metamodels	Training samples	Tuning parameters			Performance	
		C	γ_{RBF}	ν	MAE	MSE
Final velocity	656	36	.5	0.58	0.0462	0.0133
Final height		500	5	0.8	0.007	0.0005
Axial overload (maximum)		130	5	0.8	0.0187	0.0088

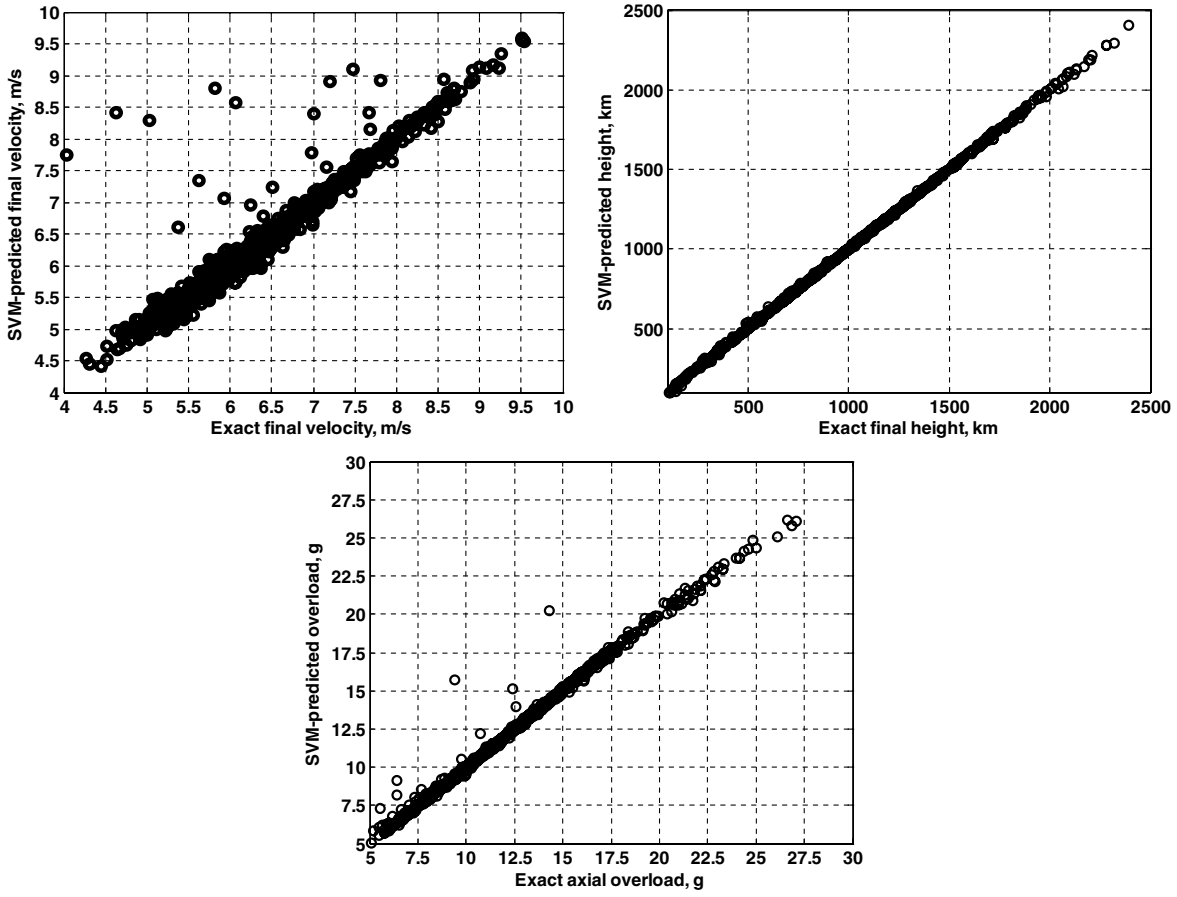

Fig. 2 Trajectory analysis metamodel testing (1000 random validation samples).

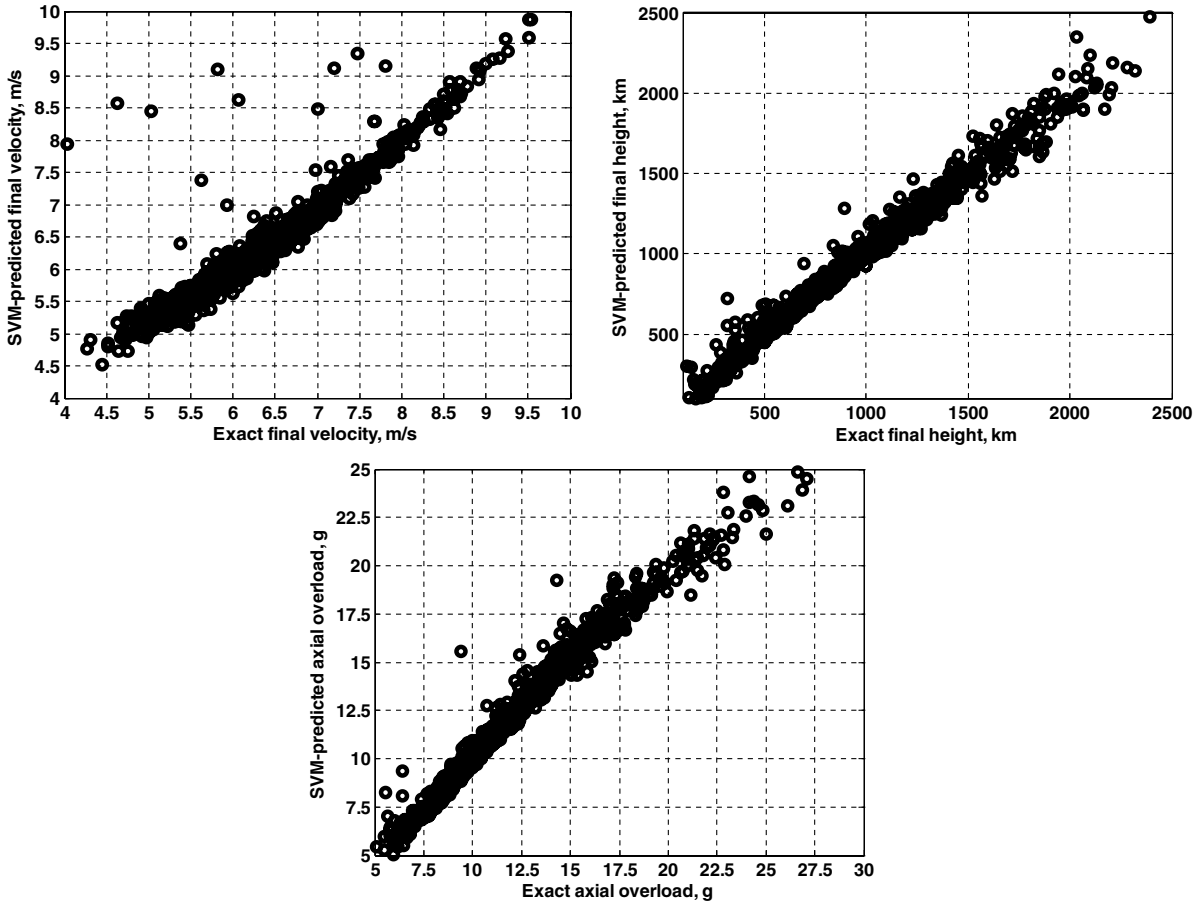

Fig. 3 Trajectory metamodel using LNOHSS sampling and SVM regression.

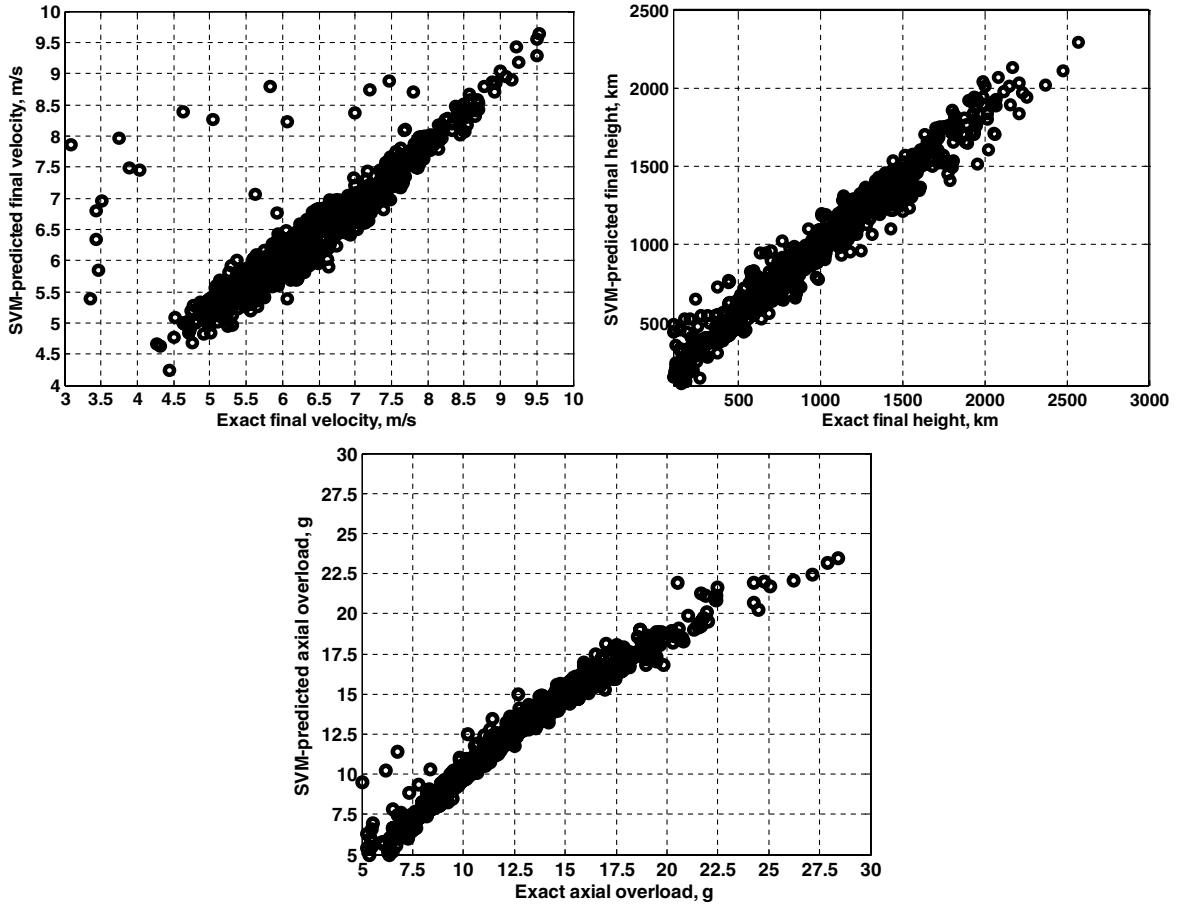

Fig. 4 Trajectory metamodel using Latin hypercube sampling and SVM regression.

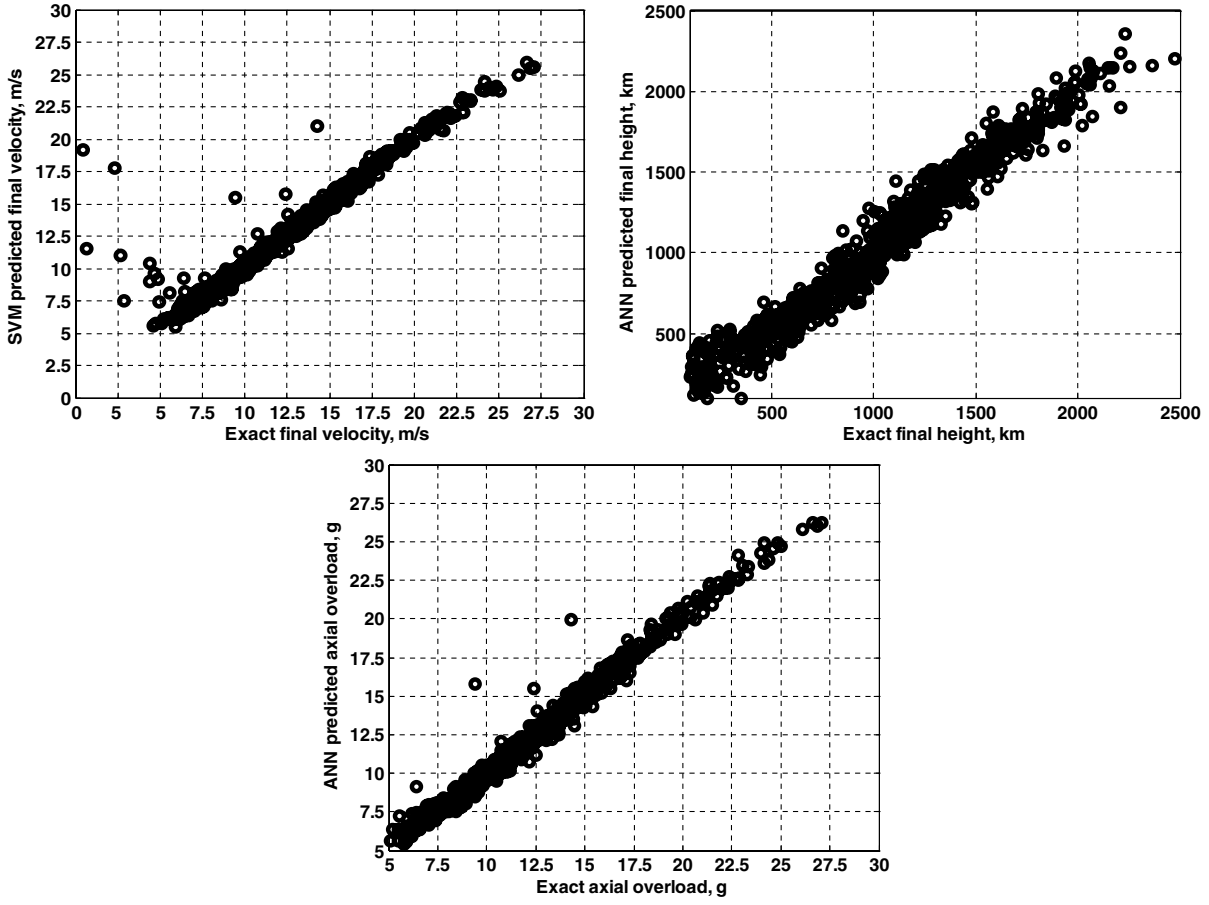

Fig. 5 Trajectory metamodel using LNOHSS sampling and ANN regression.

Table 6 Comparison of sampling and metamodeling strategy

Metamodels	Training samples	Case 1 LNOHSS sampling SVM metamodels MAE	Case 1 LNOHSS sampling SVM metamodels MSE	Case 2 LHS sampling SVM metamodels MAE	Case 2 LHS sampling SVM metamodels MSE	Case 3 LNOHSS sampling ANN metamodels MAE	Case 3 LNOHSS sampling ANN metamodels MSE
Final velocity	41	0.08166	0.02984	0.1188	0.0567	0.0466	0.0174
Final height		0.0619	0.0112	0.08802	0.01478	0.0787	0.0103
Axial overload (maximum)		0.07202	0.03273	0.0863	0.0436	0.0610	0.0287

$$\min_{\alpha,\alpha^*} \frac{1}{2}(\alpha-\alpha^*)^T Q(\alpha-\alpha^*) + z^T(\alpha-\alpha^*) \begin{cases} e^T(\alpha-\alpha^*)=0, & e^T(\alpha-\alpha^*)=Cl\nu \\ 0 \le \alpha_i, \alpha_i^* \le C, & i=1,\ldots l \end{cases} \tag{27}$$

$$\sum_{i=1}^{l}(-\alpha_i+\alpha_i^*)K(\mathrm{x}_i,\mathrm{x})+b \tag{28}$$

Analysis Metamodels and Their Validity

The LIBSVM library for the support vector machine is used in this research work.[§] The procedure adopted is as follows:

1) Scale training, validation, and testing data to the range $[-1, 1]$.
2) Consider the RBF kernel $K(x, y) = e^{-\gamma\|x-y\|^2}$.
3) Form a grid from a range of C, γ, and ν.
4) Train SVM using values of C, γ, and ν from the grid.
5) Test SVM using testing data and measure the mean square error (MSE) and maximum absolute error (MAE).

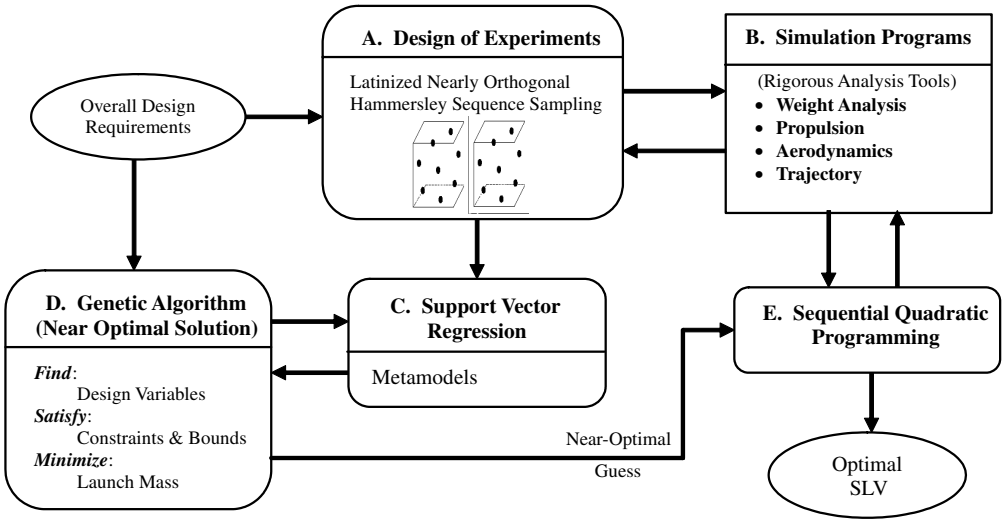

Fig. 6 Flowchart of the data flow and optimization approach.

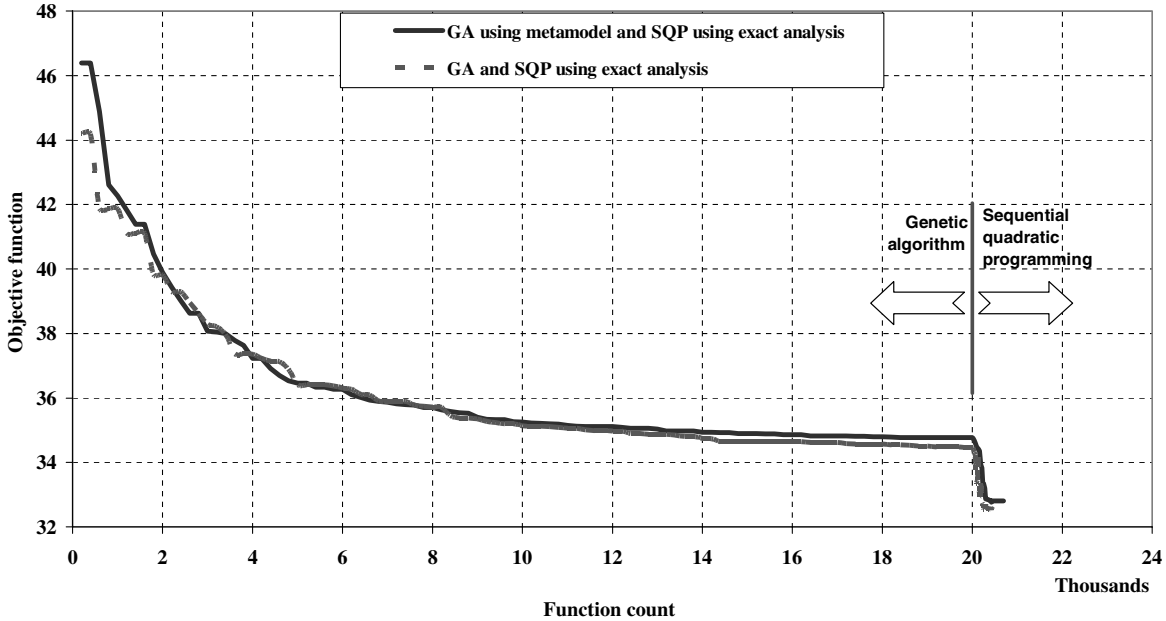

Fig. 7 History of GA convergence with and without the SVR metamodel.

[§]Data available online at http://www.csie.ntu.edu.tw/~cjlin/libsvm [retrieved 27 June 2006]

Table 7 Comparison of function counts with the SVR metamodel

GA using SVR metamodel and SQP using exact analysis				
Run	Function count in SVM training	GA function count	SQP function count	Optimal solution, Mg
1	656	nil	226	**33.2**
			410	**32.5**
			227	**33.3**
			240	**33.6**
			246	**33.5**

6) Retrain SVM with all combination of parameters (i.e., grid search).

7) Use the best parameters C, γ, and ν (corresponding to the minimum MSE and MAE).

The range of tuning parameters used are $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$, $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$, and $\nu = 0, \ldots, 1$. A two-level grid search, from coarse to fine, is adopted in this study. Three separate metamodels of final velocity, final height, and axial overload are generated using support vector regression. The SVMs are trained with experimental designs based on the LNOHSS generated earlier. One thousand randomly generated validation data points are collected to assess the accuracy of each metamodel over the region of interest. For each set of validation points, the MAE and MSE are computed. The MSE provides good estimates of the *global* error over the regions of interest. The MAE gives a good estimate of the *local* error by measuring the worst error within the region of interest, in which a good approximation will have low MSE and low MAE values.

The performance of these metamodels is validated by 1000 randomly generated samples, and the corresponding MAE and MSE are calculated. The trajectory metamodel used in optimization is trained with 656 samples of the LNOHSS. The resulting MAE and MSE of this metamodel are shown in Table 5 and the performance is plotted in Fig. 2.

A comparison is made between the LNOHSS and LHS for sampling effectiveness, and the SVR and ANN are compared for metamodel effectiveness. Three cases are developed: 1) LNOHSS sampling with the SVR metamodel, 2) LHS sampling with the SVR metamodel, and 3) LNOHSS sampling with the ANN metamodel. These cases are shown in Figs. 3–5 and are compared in Table 6. Sample size for training in all cases is fixed as 41 to judge the difference at a lower number of runs.

Cases 1 and 2 can be compared for sampling efficiency; they show that the proposed LNOHSS performs better than the LHS. Cases 1 and 3 can be compared for metamodeling effectiveness and they show that the performance of the SVM is consistent. As anticipated, earlier ANN metamodels may be trapped in the local optimum, just as for the final height metamodel of case 3.

Optimization and Results

Optimization Using Hybrid Genetic Algorithm

The overall data flow and optimization approach is shown in Fig. 6. The user defines the overall requirement and identifies the design parameters. Experiments (or simulations) are designed in stage A (design of experiment) using the proposed LNOHSS to get the most accurate metamodel with the minimum sampling. Based on the experimental specifications, designers perform experiments or simulations and get the result data in stage B (simulation programs). In stage C (support vector regression), designers build an approximation metamodel based on the obtained training data. Stage D [genetic algorithm (GA)] performs global optimization and identifies a near-optimal guess. The GA calls SVR metamodels of trajectory for fitness evaluation instead of actual trajectory analysis. Stage E [sequential quadratic programming (SQP)] performs local optimization, starting with the provided near-optimal guess; it calls actual analyses and identifies the optimal SLV.

The total gross weight of the SLV with a payload is defined as the objective function to be minimized. Training data for SVR are generated by 656 samples of the LNOHSS. One thousand randomly generated validation samples are used to evaluate the performance of SVR metamodels. The GA uses a population size of 200 and the maximum number of generations to be evolved is 100.

The convergence history of optimization with and without using metamodels is shown in Fig. 7 as two curves representing the average of five runs of each strategy. Each optimization run of hybrid GA is seeded with a different nine-digit random number. The elite solution at the 20,000 function count (200 population size times 100 generations) is passed on to SQP as the near-optimal guess. The SQP then performs local convergence to identify the minimum mass SLV. The number of exact function evaluations is compared in Tables 7 and 8.

Table 9 Optimal design variables

Design variables		Stage 1	Stage 2	Stage 3
Propellant mass fraction	μ_{pi}	0.68	0.68	0.68
Stage diameter, m	D_i	1.3	0.6	0.7
Trajectory control angle, deg	ϕ_g	63.7	-	-
Chamber pressure, bar	p_{ci}	55	41.6	34.3
Nozzle-exit-plane pressure, bar	p_{ei}	0.53	0.25	0.19
Burning rate, mm/s	u_i	6.2	7	6.5
Grain-shape parameter	k_{si}	1.67	1.63	1.53

Table 10 Specifications of the optimal SLV

Characteristic	Stage 1	Stage 2	Stage 3
Stage diameter, m	1.3	0.9	0.7
Grain length, m	10.3	5.5	2.4
Gross weight, Mg	**32.5**	8.5	2.1
Propellant weight, Mg	22.1	5.7	1.4
Propellant rate, kg/s	249	103	29
Thrust, kN	585	285	79
Nozzle exit diameter, m	1.1	1.1	0.6
Throat diameter, m	0.30	0.22	0.13
Velocity, km/s	2.3	5.2	7.61
Height, km	43	127	500
Flight path angle, deg	30	23	
Angle of attack (max), deg	5.0	0	0
Axial overload, g	6.7	10.7	12
Burn time, s	89	56	49
Specific impulse, s	239	282^{vac}	274^{vac}
Weight-to-thrust ratio, ν_o	0.54	0.29	0.26

Table 8 Comparison of function counts without the SVR metamodel

Solutions from GA and SQP using exact analysis				
Run	GA function count	SQP function count	Total function count	Optimal solution, Mg
1	20,000	1307	21,307	**34.5**
2	20,000	691	20,691	**32.7**
3	20,000	918	20,918	**32.5**
4	20,000	1033	21,033	**32.6**
5	20,000	419	20,419	**32.7**

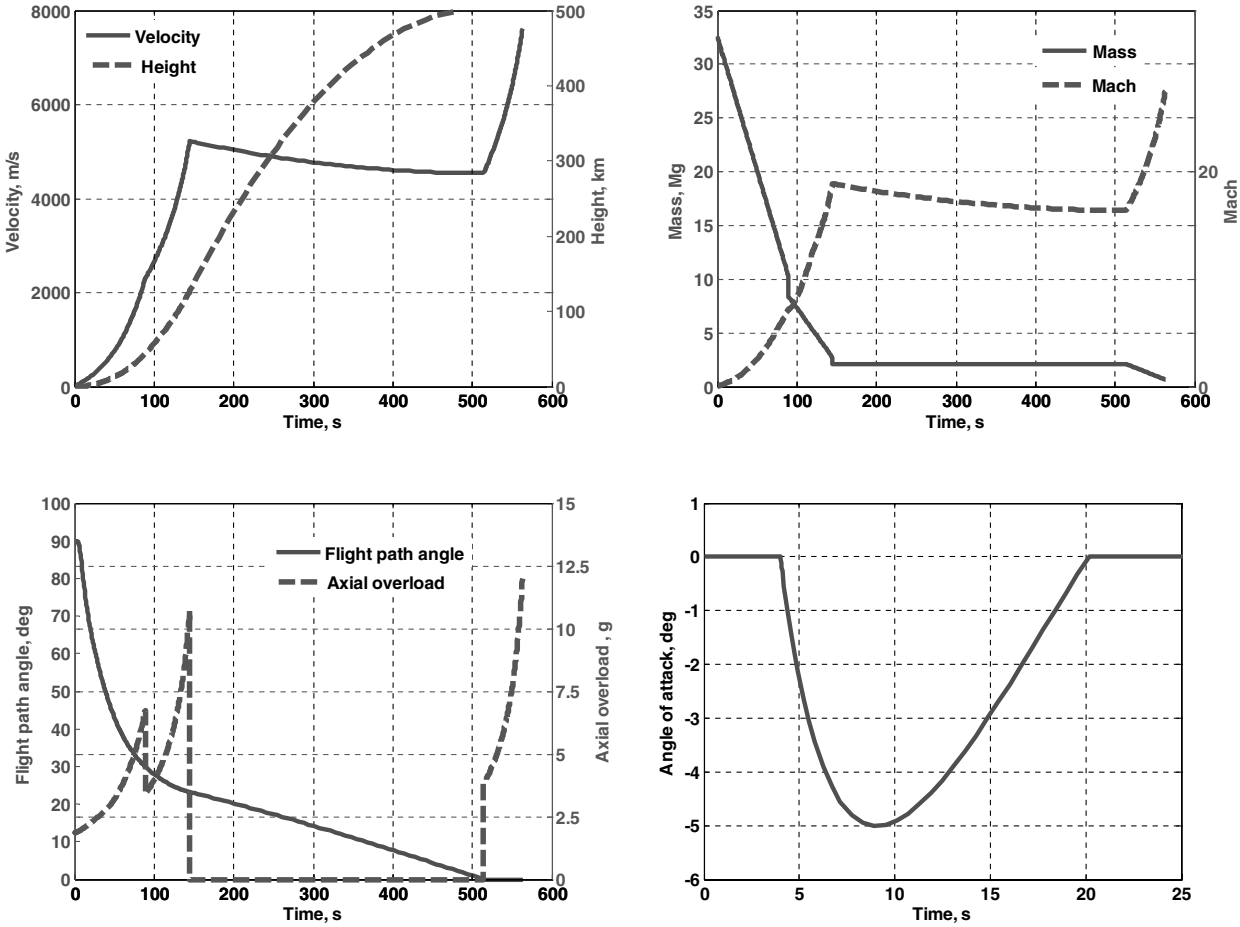

Fig. 8 Flight performance and trajectory.

Optimal Solution

The reported results in this section are obtained using the proposed optimization approach using LNOHSS training sets, SVR-based trajectory metamodels, and hybrid optimization employing GA and SQP. Table 9 shows the 19 optimized design variables. Most of the optimized design variables lie between their upper and lower bounds. Table 10 gives the specifications of the optimal SLV, including the propulsion and trajectory parameters. The stagewise weight distribution is also shown here. Required velocity and height are perfectly achieved by GA, and final-stage axial overload and angle of attack are at their maximum constraint limits. Figure 8 shows the flight performance and optimal trajectory. The optimal-angle-of-attack profile during the launch maneuver is also shown in the same figure.

Conclusions

This study reveals that a combination of a novel design of experiment based on Latinized nearly orthogonal Hammersley sequence sampling and support-vector-regression-based metamodeling is well suited for integrated design of multistage space launch vehicle systems consisting of flight dynamics, propulsion, aerodynamics, and structure. The proposed scheme provides LNOHSS designs that sample across a representation of the entire experimental region in a reasonable number of runs, even for high-dimensional space, and the designs are both nearly orthogonal and have good space-filling properties. Support vector regression proves to be computationally economical due to optimal training methodology, even for high-dimensional space, and requires very few parameters to be tuned. This work showed that an all-at-once design process, controlled by computational intelligence tools, is not only possible, but much more efficient than iteratively designing each subsystem component into a workable package. Through this study, it was confirmed that the proposed sampling methodology is efficient and that the metamodeling-driven integrated design and optimization method is effective.

References

[1] Hammond, W. E., *Design Methodologies for Space Transportation Systems*, AIAA Education Series, AIAA, Reston, VA, 2001, Chap. 10.

[2] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, No. 1, 1997, pp. 1–23.

[3] Wolf, D. M., "TRANSYS—Space Transportation System Preliminary Design Software," *Journal of Spacecraft and Rockets*, Vol. 31, No. 6, 1994, pp. 1067–1071.

[4] Rahn, M., Schöttle, U. M., and Messerschmid, E., "Multidisciplinary Design Tool for System and Mission Optimization of Launch Vehicles," 6th NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, AIAA Paper 96-4130, Sept. 1996.

[5] Simpson, T. W., Lin, D. K. J., and Chen, W., "Sampling Strategies for Computer Experiments: Design and Analysis," *International Journal of Reliability and Applications*, Vol. 2, No. 3, 2001, pp. 209–240.

[6] Simpson, T. W., "A Concept Exploration Method for Product Family Design," Ph.D. Dissertation, Dept. of Mechanical Engineering, Georgia Inst. of Technology, Atlanta, GA, Aug. 1998.

[7] Ye, Q., "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association*, Vol. 93, No. 444, 1998, pp. 1430–1439.

[8] Kalagnanam, J. R., and Diwekar, U. M., "An Efficient Sampling Technique for Off-Line Quality Control," *Technometrics*, Vol. 39, No. 3, 1997, pp. 308–319.

[9] Cresp, L. G., and Kenny, S. P., "Robust Control Design for Systems with Probabilistic Uncertainty," NASA TP-213531, Mar. 2005.

[10] Ponce-Ortega, J. M., Ramirez, V. R., Castro, S. H., Diwekar, U. M., "Improving Convergence of the Stochastic Decomposition Algorithm by Using an Efficient Sampling Technique," *Computer and Chemical*

Engineering, Vol. 28, No. 5, 2004, pp. 767–773.

[11] Johnson, D., and Bogle, D., "Handling Uncertainty in the Development and Design of Chemical Processes," *Reliable Computing*, Vol. 12, No. 6, Dec. 2006, pp. 409–426.

[12] Romero, V., Burkardt, J., Gunzburger, M., Peterson, J., and Krishnamurthy, T., "Initial Application and Evaluation of a Promising New Sampling Method for Response Surface Generation: Centroidal Voronoi Tessellation," 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Norfolk, VA, AIAA Paper 2003-2008, Apr. 2003.

[13] Qazi, M., He, L., and Qin, S., "Shape Optimization of Star Grain Geometry For Solid Motors Using Computational Intelligence," *Proceedings of the 16th International Conference on Computer Communications*, Publishing House of Electronics Industry, Beijing, 2004, pp. 1718–1724.

[14] Qazi, M., and He, L., "Artificial Neural Network Based Trajectory Simulation for Conceptual Design and Guidance of Multistage Space Launch Vehicle," *12th Saint Petersburg International Conference on Integrated Navigation Systems*, Electropribor, St. Petersburg, Russia, 2005, pp. 60–69.

[15] Qazi, M., He, L., and Elhabian, T., "Rapid Trajectory Optimization using Computational Intelligence for Guidance and Conceptual Design of Multistage Space Launch Vehicles," AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, AIAA Paper 2005-6062, Aug. 2005.

[16] Haykin, S., *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 1999.

[17] Cioppa, T. M., "Efficient Nealry Orthogonal and Space Filling Experimental Designs for High-Dimensional Complex Models," Ph.D. Dissertation, Dept. of Operations Research, Naval Postgraduate School, Monterey, CA, Sept. 2002.

[18] Giunta, A., Wojtkiewicz, S., and Eldred, M., "Overview of Modern Design of Experiments Methods for Computational Simulations," 41st Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2003-649, Jan. 2003.

[19] Fang, K. T., Ma, C., and Winker, P., "Centered L_2-Discrepancy of Random Sampling and Latin Hypercube Design, and Construction of Uniform Designs," *Mathematics of Computation*, Vol. 71, No. 237, 2000, pp. 275–296.

[20] He, L., *Solid Ballistic Missile Design*, Beijing Univ. of Aeronautics and Astronautics, Beijing, 2004.

[21] Tsuchiya, T., and Mori. T., "Optimal Conceptual Design of Two-Stage Reusable Rocket Vehicles Including Trajectory Optimization," *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, 2004, pp. 770–778.

[22] Zipfel, P. H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA Education Series, AIAA, Reston, VA, 2000, Chap. 8.

[23] Tsuchiya, T., and Suzuki, S., "Spaceplane Trajectory Optimization with Vehicle Size Analysis," *Automatic Control in Aerospace 1998*, Pergamon, New York, 1998, pp. 444–449.

[24] Clarke, S. M., Griebsch, J. H., and Simpson, T. W., "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," *Journal of Mechanical Design*, Vol. 127, No. 6, 2005, pp. 1077–1087.

[25] Smola, A. J., Scholkopf, B., "A Tutorial on Support Vector Regression," Royal Holloway College, Univ. of London, NC2-TR-1998-030, London, 1998.

J. Korte
Associate Editor