A Surrogate Accelerated Bayesian Inverse Analysis of the HyShot II Supersonic Combustion Data

By Paul G. Constantine †, Alireza Doostan ‡, Qiqi Wang ¶ AND Gianluca Iaccarino ‖

We implement and assess a Markov chain Monte Carlo method with a surrogate forward computational model for determining the flight conditions of the HyShot II supersonic re-entry vehicle given pressure measurements. The surrogate models we examine are non-intrusive, i.e., sampling-based, and include (1) a dimension-reduced global polynomial surrogate, (2) ordinary Kriging, and (3) a low-rank separated representation for high-dimensional functions. We find that the errors in the specific surrogates are overwhelmed by errors due to finite sample size and the ill-posedness of the inverse problem.

1. Introduction

Among other experimental efforts in the area of scramjet propulsion and supersonic combustion are the ballistic re-entry HyShot flight experiments conducted by the University of Queensland, Australia (Hass et al. (2005), Smart et al. (2006)). The post-flight data obtained from the second trial of the experiment – named "HyShot II" – confirmed the presence of a three second supersonic combustion. The available HyShot II flight measurements provide a unique model validation test bed for supersonic combustion predictions. However, a radar tracking failure left the flight trajectory data incomplete. This implies that the flight conditions – e. g. mach number, angle-of-attack, and flight altitude – are not directly available and thus have to be inferred from the onboard pressure, accelerometer, magnetometer and horizon sensor measurements. To this end, Cain et al. (2004) developed a semi-analytical deterministic approach that yields consistent pressure measurements at the vehicle nose. However, further CFD analyses conducted using these flight conditions have not been able to accurately predict the pressure measurements inside the vehicle s combustors. This could partially be due to the fact that the uncertainties associated with the sensor measurements and the ill-posedness of the inference problem have been ignored in that work. To overcome these shortcomings, we explore alternative statistical inversion techniques that not only estimate the flight conditions from the entire onboard pressure measurement but also quantify the uncertainties in these estimates. In particular, we develop a Bayesian inference technique with accelerated Markov Chain Monte Carlo (MCMC) sampler based on accurate reduced order forward models. Our effort centers around construction of efficient surrogate models that

† John von Neumann Research Fellow, Uncertainty Quantification and Optimization *Sandia National Labs, Albuquerque, NM 87185*

‡ Assistant Professor, Aeronautics and Astronautics *University of Colorado, Boulder, CO 80309*

¶ Assistant Professor, Aeronautics and Astronautics *Massachussettes Institute of Technology, Cambridge, MA 02139*

‖ Assistant Professor, Mechanical Engineering *Stanford University, Stanford, CA 94305*

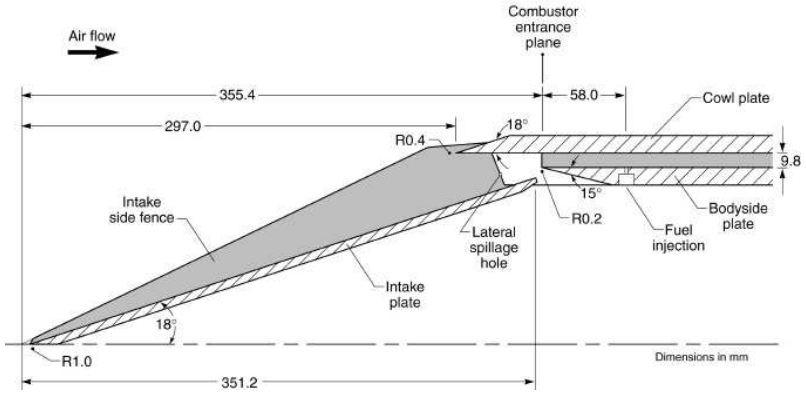

FIGURE 1. The geometry of the nose and the combustor inlet of the HyShot hypersonic vehicle (Smart et al. (2006)). Note that the vehicle is symmetric with two identical combustion chambers, but the fuel was injected only in one.

accurately map potentially large number of unknown parameters (flight conditions) to the data (pressure measurements).

2. HyShot Simulations and Problem Set-up

The HyShot experiment was designed to demonstrate supersonic combustion in flight, and for this reason the configuration was very simple, consisting of a double wedge intake and two straight combustors (Fig. 1). The vehicle flew in a parabolic trajectory with an apogee in excess of 300km. During the descent part of the flight, a Mach number in excess of 7.5 (between 35 and 25 km altitude) was reached, thus supplying a useful range of flight conditions for the testing of a scramjet. One of the combustors remained unfueled (cold) for the entire duration of the flight, and wall pressure was measured in several locations.

Although the goal of the experiment was to supply uniform flow into the two combustors at conditions ranging between Mach 7.2 and 8.0, the vehicle trajectory was complicated by the presence of rolling, pitching and nutation (Smart et al. (2006)). In addition, the failure of the radar tracking system lead to considerable uncertainties in the operating conditions (altitude, speed, angle of attack, etc.) as a function of time. This creates an inherent difficulty in comparing numerical predictions to the flight data.

In a previous effort, we built a computational model that represents the physical processes in scramjet and performed a validation study which included the HyShot geometry (Iaccarino et al. (2010)). The fluid flow computations were performed using a novel Reynolds-averaged Navier-Stokes (RANS) solver, developed at the PSAAP (Predictive Science Academic Alliance Program) Center at Stanford University. The flow solver is a parallel solver for the solution of the compressible flow equations on unstructured meshes based on a finite volume formulation and implicit time-integration on arbitrary polyhedral mesh elements (Pecnik et al. (2008)). The predictions reported in Iaccarino et al. (2010) illustrate that the model is accurately representing the overall pressure distribution in the scramjet and the interaction of the shock-train with the turbulent boundary layers, but a residual discrepancy can be observed. This *error* introduces an additional element of difficulty - and uncertainty - in the inference process. Since the objective of the present effort is to assess the computational tools that enable the inference analysis, we

decided to replace the actual flight measurements with synthetic data generated by the computational model. The computed pressure is modified by introducing *uncertainty* in the form of Gaussian noise to replicate the presence of measurement errors. An additional advantage of the present test is the ability to inject a controlled amount of uncertainty and observe its effect on the final estimates. We will assume to have eleven pressure measurements along the combustion chamber, and the goal is to estimate six parameters: the first three corresponding to the flight conditions - Mach number M, altitude h and angle of attack α - the others representing the wall temperature (T_{bottom} and T_{top}) in the combustor and one of the parameters (C_{sa}) in the turbulence model. It is useful to point out that we expect the pressure distribution in the combustor to be strongly affected by the first three parameters mentioned above, but fairly insensitive to the last three which mainly affect the development of the flow within the boundary layers.

In summary, the problem is defined as follows. We assume to have well defined ranges (priors) for six parameters: M, h, α, T_{bottom}, T_{top}, C_{sa}. We select one value for each, perform a simulation and extract wall pressures at eleven locations within the combustor (sensors). These values are modified by adding white noise and considered as the input data for the inference process. The goal is to determine the parameters and to estimate the remaining uncertainty (posteriors). One key aspect of the inference algorithm used in this work is the need to repeatedly compute the wall pressure given different choices of the parameters: in this context, this step requires a full RANS computation and, in spite of the efficiency of the present computational tools, this remains a intractable task. Therefore, as a preliminary step of the inversion process, we construct surrogate models of the combustor wall pressure. In the following sections we illustrate three different approaches. The common element is the use of a large number of realizations of the true flow field. Specifically, 1,800 HyShot simulations were carried out, corresponding to a Monte Carlo sampling in the space of the six parameters defined above.

2.1. *Bayesian Inverse Analysis*

Bayesian analysis has been successfully applied in a multitude of contexts; we refer the reader to a comprehensive and well written tutorial for details (Silva and Skilling (2006)). The general formulation of the inverse problem follows from Bayes theorem, which tell us that the conditional probability of the parameters given the measurements (i.e. the *posterior*) is proportional to the product of the conditional probability of the measurements given the parameters (i.e. the *likelihood*) times the probability of the parameters (i.e. the *prior*). Given measurement data d and input parameters $s := (M,\ h,\ \alpha,\ T_{bottom},\ T_{top},\ C_{sa})$†, this is written

$$\mathbb{P}(s \,|\, d) \propto \mathbb{P}(d \,|\, s)\mathbb{P}(s). \tag{2.1}$$

If we assume an uninformative uniform prior on the parameters, then $\mathbb{P}(s)$ is included in the proportionality constant. We assume that the measurements are corrupted by independent errors, so the true pressure $p(s)$ is described by

$$p(s) = d(s) + \eta \tag{2.2}$$

† The input parameters correspond to the parameters defining the CFD model parameters given in Section 2.

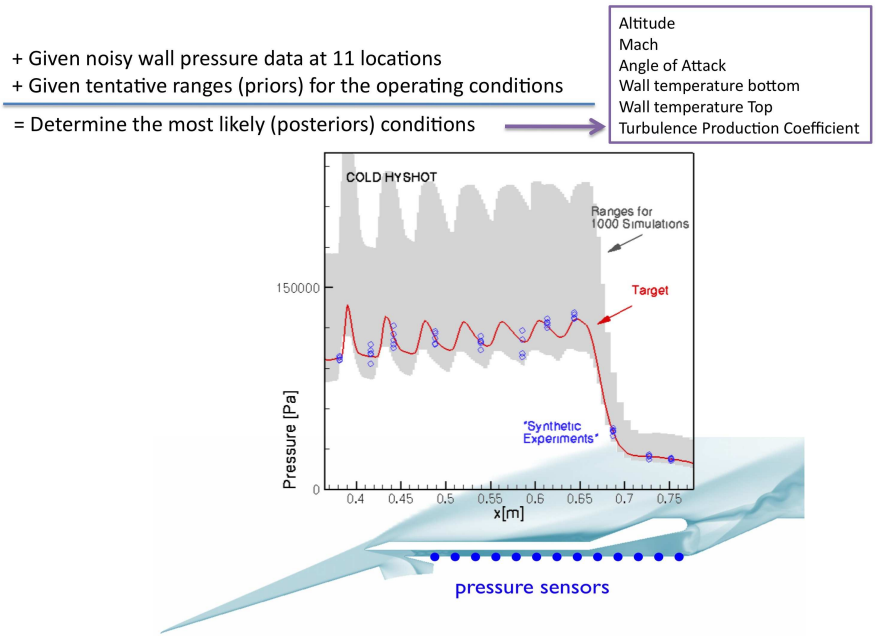

FIGURE 2. The location of the *sensors* in the HyShot combustion chamber and the range of the wall pressure corresponding to the parameters' prior.

where η is an m-vector of independent Gaussian random variables with zero mean and a given variance. This implies that the joint likelihood function separates,

$$\mathbb{P}(d \,|\, s) = \prod_{j=1}^{m} \mathbb{P}(d_j \,|\, s), \tag{2.3}$$

and the densities $\mathbb{P}(d_j \,|\, s)$ are Gaussian density functions. The Markov Chain Monte Carlo (MCMC) algorithm is a method for drawing samples from the posterior $\mathbb{P}(s \,|\, d)$ by constructing a Markov chain with an equivalent stationary distribution. Given sufficient samples from the posterior, one can compute the expectation over the parameter space of the posterior to estimate the parameters that are consistent with the measurements. We next briefly introduce the MCMC algorithm in the context of Bayesian inference.

2.1.1. *Markov Chain Monte Carlo (MCMC)*

In the MCMC-based Bayesian method (Tarantola (2005)), a Markov chain with a special transition probability is constructed so that its stationary distribution coincides with the posterior distribution $\mathbb{P}(s \,|\, d)$. After a sufficiently large number of steps, the states of the chain are samples of the parameters s following the posterior distribution $\mathbb{P}(s \,|\, d)$. The Bayesian estimates of the parameters s are then obtained by computing the mean of the posterior samples, i.e., Minimum Mean-Squares Error (MMSE) estimate, or by computing the mode of the posterior samples, i.e., Maximum A-posteriori Probability (MAP) estimate.

The Metropolis-Hastings sampler, among a number of other algorithms, provides an elementary construction of the Markov chain whose stationary distribution is guaranteed to converge to $\mathbb{P}(s \,|\, d)$. Let $q(\tilde{s}|s_i)$ be the *proposal distribution* to generate a candidate

state $\tilde{s}$ given that the underlying Markov chain is currently at state s. The probability of moving to the candidate state $\tilde{s}$ is

$$\rho(\tilde{s}, s_i) = \min\left\{\frac{\mathbb{P}(\tilde{s}\,|\,d)q(s_i|\tilde{s})}{\mathbb{P}(s_i\,|\,d)q(\tilde{s}|s_i)}, 1\right\} \tag{2.4}$$

where $\mathbb{P}(\tilde{s}\,|\,d)$ is computed from (2.1) once the forward model $\boldsymbol{p}(\cdot)$ is evaluated for the parameter values s. When the proposal distribution $q(\cdot)$ is strictly positive on the domain of parameters s, the chain is *irreducible* and *aperiodic*. Therefore, the stationary distribution of the Markov chain will be $\mathbb{P}(s\,|\,d)$. The following algorithm outlines the implementation of the Metropolis-Hastings sampler:

(1) Choose the length of the burn-in period $t_b \in \mathbb{N}$ and an initial state s_1. Set $j = 1$.
(2) Evaluate the forward model $p(s)$ and compute $\mathbb{P}(s\,|\,d) \propto \mathbb{P}(d\,|\,s)\mathbb{P}(s)$.
(3) Generate candidate state $\tilde{s}$ according to $q(\cdot)$.
(4) Generate U from $U(0,1)$ distribution. Set $s_{j+1} = \tilde{s}$ if $U \leq \rho(\tilde{s}, s_i)$. Otherwise, set $s_{j+1} = s_j$.
(5) Repeat steps (2) to (4) sufficiently large, say $r \in \mathbb{N}$, number of times with $r > t_b$.

Notice that the burn-in period, t_b, ensures the dissipation of the initial condition effects. The MMSE Bayesian estimate of the parameters s are the samples $\{s_j\}_{j>t_b}$. In the present study, we choose an independent Gaussian proposal distribution $q(\cdot)$ with the mean s_j (the state of the chain at step j) and a coefficient of variation equal to 10% of the range of the prior of s_i.

2.2. *Reduced Order Modeling*

The primary computational expense of the Bayesian inverse analysis for complex models occurs in the forward simulation used in the MCMC algorithm, which must perform an optimization over the parameter space and thus evaluate the forward model many times. Thus, one can achieve tremendous savings by employing inexpensive surrogates in place of the true model evaluation. This approach is not unprecedented; some have used global polynomial approximation (Marzouk and Najm (2009)) in place of the forward model with striking results for relatively low dimensional parameters spaces – roughly six to ten input parameters. In this work we examine three distinct surrogate models: (1) a global regression polynomial, (2) a Kriging surrogate model, and (3) a low-rank separated representation with a polynomial basis. We construct these surrogates from the database of solutions to approximate the pressure output as a function of the six input parameters; therefore, our surrogate constructions are non-intrusive.

2.2.1. *Regression Polynomial with Sensitivity based Dimensionality Reduction*

Polynomial approximations are widely used in constructing surrogate surfaces. We construct a multivariate approximation surface by first identifying the nullspace of the nonlinear input-output relation and then constructing a multivariate regression polynomial on the reduced space. Dimensionality reduction of the parameter space is achieved by analyzing the Jacobian matrix of the randomly selected points. Let

$$J(s) = \frac{\partial p}{\partial s} \tag{2.5}$$

be the Jacobian matrix at the parameter point s. The rows of J correspond to the pressure measurements, and the columns of J correspond to the input parameters. We

aim to compute the eigenvalues and eigenvectors of the covariance matrix

$$\mathbb{E}_s\Big(J^{\mathrm{T}}J\Big) = U\Lambda U^{\mathrm{T}}, \tag{2.6}$$

and identify the nullspace of this matrix, which is spanned by the eigenvectors in U corresponding to the zero eigenvalues. The nullspace can be reduced in constructing the surrogate surface. If u is in the nullspace, then

$$0 \;=\; u^{\mathrm{T}}\mathbb{E}_s\Big(J^{\mathrm{T}}J\Big)u \;=\; \mathbb{E}_s\Big(\|Ju\|_2^2\Big) \tag{2.7}$$

Therefore, $Ju = 0$ almost surely. By the mean value theorem,

$$p(s+\alpha q) - p(s) \;=\; \alpha J(\xi)u \;=\; 0 \tag{2.8}$$

almost surely. In other words, an arbitrary perturbation of the parameter s in the direction of p does not change the pressure at the measurement points.

In practice, we approximate he covariance matrix by Monte Carlo approximation

$$\mathbb{E}_s\Big(J^{\mathrm{T}}J\Big) \approx \frac{1}{N}\sum_{i=1}^{N} J(s_i)^{\mathrm{T}}J(s_i) \tag{2.9}$$

where s_i are randomly sampled points in the parameter space. The eigen decomposition of the Monte Carlo covariance matrix can be computed by SVD of the stacked Jacobian

$$\begin{bmatrix} J(s_1) \\ \vdots \\ J(s_N) \end{bmatrix} \tag{2.10}$$

without the expensive evaluation of the Monte Carlo covariance matrix itself. The Jacobian matrix at each s_i is in turn approximated by selecting the nearest 20 sample points whose function value p is already computed, and performing a linear least squares to approximate the Jacobian.

Through this dimensionality reduction procedure, we found that three of the six eigenvalues are close to zero. Denote U_r as the first three columns and U_n as the last three columns of the matrix U computed through the SVD procedure described above. For any s in the parameter space,

$$u \;=\; s - U_r U_r^{\mathrm{T}} s \;=\; U_n U_n^{\mathrm{T}} \tag{2.11}$$

is in the nullspace (spanned by columns of U_n), and

$$p(s) = p(U_r U_r^{\mathrm{T}} s) \tag{2.12}$$

for any s. Therefore, we can represent the surface we are trying to approximate as

$$p(s) = p_r(U_r^{\mathrm{T}} s) \quad \text{where} \quad p_r(t) = p(U_r t). \tag{2.13}$$

Because p_r is depends on three variables – as opposed to the original six – the number of terms in our polynomial approximation is significantly reduced.

A total order polynomial in three variables is fit to the sampled data points with a least squares procedure. Small coefficients in the polynomial are manually removed to further reduce the degrees of freedom. The resulting polynomial surface is used to approximate p_r, which combines with U_r to form an approximation for p.

2.2.2. *Kriging*

Kriging methods (Cressie (1993)) have been used in geostatistics for decades to interpolate spatially dependent data. In our case, we wish to interpolate the parameter dependent pressure from the CFD ensemble at the parameter values requested by the MCMC algorithm; in other words, we treat the parameter space as spatial coordinates for interpolation. The CFD ensemble of pressure acts as the measurements that the Kriging surface will interpolate.

Given the pressure $p_i = p(s_i)$ at a parameter points s_i, the ordinary Kriging interpolant is constructed as

$$p(s) \approx \hat{p}(s) = \sum_{i=1}^{n} p_i \lambda_i(s) \tag{2.14}$$

where the parameter dependent coefficients of the linear approximation $\lambda_i(s)$ are constrained so that

$$\sum_{i=1}^{n} \lambda_i(s) = 1, \tag{2.15}$$

which ensures that the estimate is statistically unbiased. We assume a Gaussian covariance model of the pressure field. In other words, given two parameter values s_i and s_j, the covariance of pressure is given by

$$\gamma(s_i, s_j) = ce^{-s_i^T W s_j} \tag{2.16}$$

where c is a normalization constant and W is a diagonal matrix of correlation lengths – one for each component of s. Define λ to be the n-vector of coefficients λ_i; define e to be an n-vector of ones; define the $n \times n$ matrix G with elements $G_{ij} = \gamma(s_i, s_j)$; and define the n-vector b with elements $b_i = \gamma(s, s_i)$ for an arbitrary s. Then the coefficients of the Kriging approximation are solved by computing the solution to the matrix equation

$$\begin{bmatrix} G & e \\ e^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} b \\ 1 \end{bmatrix}, \tag{2.17}$$

where μ is the Lagrange multiplier. We use the Matlab toolbox DACE (Lophaven et al. (2002)) for all computational experiments.

2.2.3. *Low-rank Separated Representation*

Separated representations, also known as canonical decomposition (CANDECOMP) or parallel factor analysis (PARAFAC), were first introduced by Hitchcock (1927) to represent a multi-way tensor as a finite sum of rank-one tensors, and they have been successfully applied to several areas. The basic idea behind the separated representation is to decompose a multivariate function into a linear combination of products of univariate functions. More specifically, let p be the pressure at a particular sensor location and a function of unknown input parameters $s = (s^1, \ldots, s^n)$. (Note that the superscript denotes a component of the vector of inputs; a subscript denotes a sample from the inputs.) The approximation of p in the form

$$p(s^1, \cdots, s^n) = \sum_{l=1}^{r} \lambda_l p_1^l(s^1) \cdots p_n^l(s^n) + \mathcal{O}(\epsilon), \tag{2.18}$$

is called the separated representation with *separation rank* r. Here, the scalars $\{\lambda_l\}_{l=1}^r$ are selected such that the factors $p_j^l(s^i)$ have unit length. The representation (2.18) for

tensors may be considered a generalization of the standard Singular Value Decomposition (SVD). However we use (2.18) to split different directions whereas the standard use of SVD for matrices split input from the output. Given a target accuracy ϵ, the approximation (2.18) can be achieved by tailoring the unknown functions (factors) $p_j^l(s^i)$ and an optimal separation rank r, for instance, through a sequence of linear one-dimensional approximations leading to a cost that is linear with respect to d. The refinement is then done by increasing the separation rank r until the prescribed accuracy ϵ is achieved. In what follows, we extend the algorithm proposed by Beylkin et al. (2008) to compute the factors $p_j^l(s^j)$ and the separation rank r from N random sample pairs of s_i and $p(s_i)$. Assuming regularity for the unknown factors, we first expand each factor $p_j^l(s^i)$ into the standard Legendre polynomial basis $\psi_k(s^j)$ of maximum order $M-1 \geq 0$, i.e.,

$$p_j^l(s^j) = \sum_{k=1}^{M} c_k^{j,l} \psi_k(s^j), \quad l = 1, \cdots, r, \quad j = 1, \cdots, n. \tag{2.19}$$

Note that the particular selection of the Legendre basis is motivated by the fact that uniform priors are assumed for the unknown parameters s^j. For a fixed separation rank r, the total number of unknowns $c_k^{j,l}$ in separated representation (2.18) is $n \cdot r \cdot M$. Heuristically, the number of solution samples, N, required to accurately compute the coefficients $c_k^{j,l}$ is

$$N \sim \mathcal{O}(n \cdot r \cdot M) \tag{2.20}$$

which scales linearly with respect to the number of parameters. This has to be compared to exponential growth in many other multivariate data fitting techniques.

A multi-variate regression approach is adopted to compute the unknown coefficients $c_k^{j,l}$ from the random samples of s_i and $p(s_i)$. Specifically, for a fixed r,

$$\hat{c}_{j,k,l} = \arg\min_{c_{i,j,k}} \sum_{i=1}^{N} \left(p(s_i) - \sum_{l=1}^{r} \lambda_l \prod_{j=1}^{n} \left(\sum_{k=1}^{M} c_k^{j,l} \psi_k(s_i^j) \right) \right)^2, \tag{2.21}$$

which is a non-linear least-squares problem. Following Beylkin et al. (2008), the optimization problem (2.21) may be solved by a sequence of linear least-squares problems via the Alternating Least-Squares (ALS) algorithm. In the ALS method, a sequence of one dimensional least-squares problems are constructed to solve for unknowns $c_k^{j,l}$, along direction j while freezing variables along all other directions at their current approximation. Once the algorithm converges, the separation rank r is increased if the norm of the residual is still above the target accuracy ϵ. At any iteration of the ALS algorithm, the computational cost scales linearly with respect to n.

3. Results

We first used a cross-validation procedure to assess each surrogate model. We constructed each surrogate on the first N runs – with $N = 200$, 500, and 1000 – from the ensemble of CFD solutions, and we tested their predictions at the parameter values of the last 300 runs; the results are given in table 3. We see that the surrogates constructed with 1000 samples all achieve a relative error of less than% – no method has significant advantage over the others in this cross validation test.

Similar to the cross validation results, the estimates of the posteriors on the input parameters produced by MCMC vary little between the surrogates. Therefore we present

N	Regression	Kriging	Low-rank
200	0.0081	0.0108	0.0038
500	0.0068	0.0083	0.0047
1000	0.0067	0.0077	0.0043

TABLE 1. Cross-validation comparison of the three surrogate models constructed on 200, 500, and 1000 samples. The numbers reported correspond to the relative error.

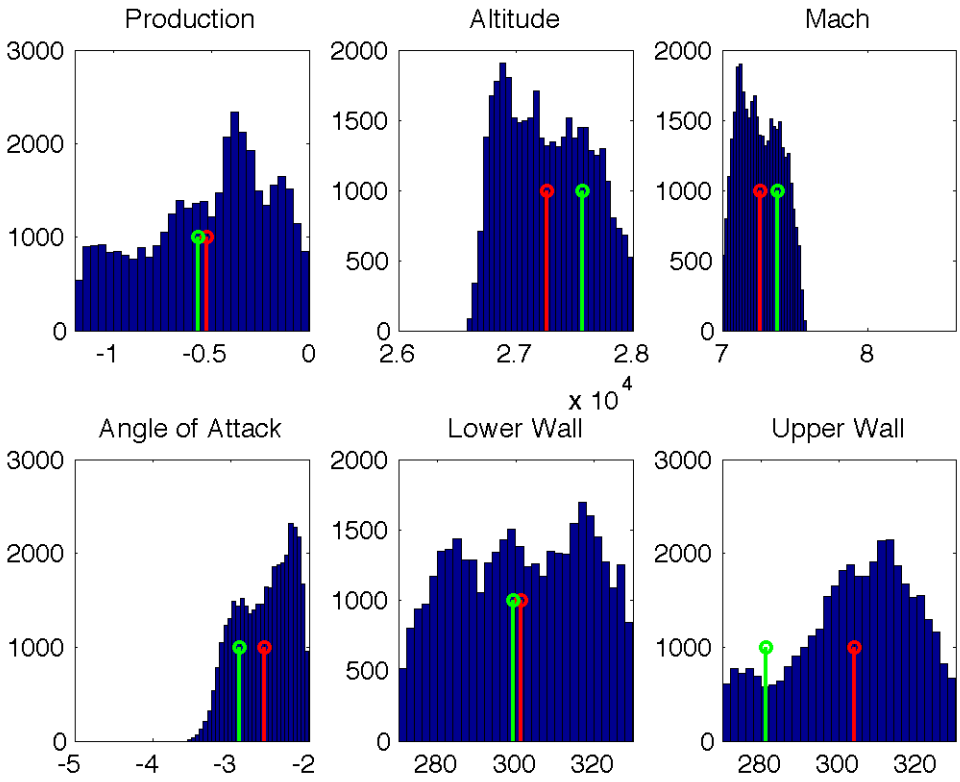

FIGURE 3. Posterior histograms of the input parameters given the measurements. The green staffs correspond to the true input parameter values. The red staffs correspond to the average of the samples from the posterior.

only the posteriors using the Kriging surrogate model. We apply a 1% noise in the synthetic data and run the MCMC; the resuls are shown in Figure 3. The wide posteriors identify the parameters with relatively small effect on the pressure.

4. Conclusions

Due to space limitations, we were not able to present the results for all the computational experiments performed in this study; Section 3 contains only representative

highlights. We found that the no surrogate model had any decisively clear advantage in terms of accuracy or convergence on cross validation tests. Also, any relatively small errors in the surrogate model were washed out by the sampling errors of the MCMC and the ill-posedness of the specific inverse problem – especially when the noise in the measurement data was increased. A subsequent study revealed significant redundancies in the measurement data due to the sensor placement, i.e. eleven sensors did not correspond to eleven independent sources of information. By synthetically adjusting the pressure sensor locations and including heat flux information, we were able to dramatically improve the results of the inversion. The analysis of this study will appear in future work.

We gratefully acknowledges support from DOE PSAAP program.

REFERENCES

N. Hass, M. K. Smart, A. Paull 2005 Flight data analysis of HyShot 2 *AIAA/CIRA 13th International Space Planes and Hypersonics Systems and Technologies*

M. K. Smart, N. Hass, A. Paull 2006 Flight data analysis of the HyShot 2 scramjet flight experiment *AIAA Journal* **44**(10), 2366 – 2375.

T. Cain, R. Owen, C. Walton 2004 Reconstruction of the HyShot 2 flight from onboard sensors *Fifth Symposium on Aerothermodynamics for Space Vehicles*

G. Iaccarino, R. Pecnik, V. Terrapon, A. Doostan 2010 Numerical predictions of the performance in in flight of an air-breathing hypersonic vehicle: HyShot II *IMECE 2010-40186, Proceedings of the ASME 2010 International Mechanical Engineering Congress & Exposition*

R. Pecnik, P. Constantine, F. Ham, G. Iaccarino 2008 A probabilistic framework for high-speed flow simulations *Annual Research Briefs, Center for Turbulence Research, Stanford*, 3 – 17.

D. S. Silva, J. Skilling 2006 Data Analysis: A Bayesian Tutorial *Oxford University Press*

A. Tarantola 2005 Inverse Problem Theory and Methods for Model Parameter Estimation *SIAM: Society for Industrial and Applied Mathematics*

Y. M. Marzouk, H. N. Najm 2009 Dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems *Journal of Computational Physics* **228**, 1862–1902.

N. A. C. Cressie 1993 Statistics for Spatial Data *John Wiley & Sons*

S. N. Lophaven, H.B. Nielsen, J. Sondegaard 2002 DACE: A Matlab Kriging Toolbox, Version 2.0 *IMM Technical University of Denmark, Lyngby*

F. L. Hitchcock 1927 The expression of a tensor or a polyadic as a sum of products *Journal of Mathematics and Physics* **6**, 164–189.

G. Beylkin, J. Garcke, M. J. Mohlenkamp 2008 Multivariate Regression and Machine Learning with Sums of Separable Functions *SIAM Journal on Scientific Computing* **31**(3), 1840–1857.