A Framework for Validation of Computer Models*

M.J. Bayarri, J.O. Berger, D. Higdon, M.C. Kennedy, A. Kottas, R. Paulo, J. Sacks
National Institute of Statistical Sciences

J.A. Cafeo, J. Cavendish, C.H. Lin, J. Tu
General Motors

October 27, 2002

Abstract

In this paper, we present a framework that enables computer model evaluation oriented towards answering the question:

Does the computer model adequately represent reality?

The proposed validation framework is a six-step procedure based upon Bayesian statistical methodology. The Bayesian methodology is particularly suited to treating the major issues associated with the validation process: quantifying multiple sources of error and uncertainty in computer models; combining multiple sources of information; and updating validation assessments as new information is acquired. Moreover, it allows inferential statements to be made about predictive error associated with model predictions in untested situations.

The framework is implemented in two test bed models (a vehicle crash model and a resistance spot weld model) that provide context for each of the six steps in the proposed validation process.

*This research was supported by grants from General Motors and the National Science Foundation (Grant DMS-0073952) to the National Institute of Statistical Sciences.

Contents

1 Introduction

1.1 Motivation and overview

We view the most important question in evaluation of a computer model to be

Does the computer model adequately represent reality?

In practice, the processes of computer model development and validation often occur in concert; aspects of validation interact with and feed back to development (e.g., a shortcoming in the model uncovered during the validation process may require change in the mathematical implementation). In this paper, however, we address the process of computer model development only to the extent that it interacts with the framework we envision for evaluation; the bulk of the paper focuses instead on answering the above basic question. In particular, we do not address the issue of code verification. General discussions of the entire V&V process, with discussion of many other pertinent issues, can be found in Roache (1998), Oberkampf and Trucano (2000), Cafeo and Cavendish (2001), Easterling (2001), Pilch *et al.* (2001), and Trucano *et al.* (2002).

Tolerance bounds: To motivate the approach we take to model evaluation, it is useful to begin at the end, and consider the type of outputs that will result from the methodology. We do not focus on answering the yes/no question "Is the model correct?"[1] In the vast majority of the cases, the relevant question is instead "Does the model provide predictions that are accurate enough for the intended use of the model?" While there are several concepts within this question that deserve – and will be given – careful definition, the central issue is simply that of assessing the accuracy of model predictions. This will be done by presenting *tolerance bounds,* such as 5.17 ± 0.44, for a model prediction 5.17, with the interpretation that there is a specified chance (e.g., 80%) that the corresponding true process value would lie within the specified range. Such tolerance bounds should be given *whenever predictions are made*, i.e., they should routinely be included along with any predictions arising from use of the model.

This focus on giving tolerance bounds, rather than stating a yes/no answer as to model validity, arises for three reasons:

1. It is often difficult to characterize regions of input variables over which the model achieves sufficient accuracy.

2. The degree of accuracy that is needed can vary from one application of the computer model to another.

3. Tolerance bounds incorporate *model bias*; accuracy of the model cannot simply be represented by a variance or standard error.

All these difficulties are obviated by the simple device of routinely presenting tolerance bounds along with model predictions. Thus, at a different input value, the model prediction and tolerance

[1] It is possible to ask and answer this question within the proposed framework – see Section 9 – but the question is often not a relevant question.

bound might be 6.28 ± 1.6, and it is immediately apparent that the model is considerably less accurate at this input value. Either of the bounds, 0.44 or 1.6, might be acceptable or unacceptable predictive accuracies, depending on the intended use of the model.

Bayesian analysis: Producing tolerance bounds is not easy. Here is a partial list of the hurdles one faces.

- There are uncertainties in model inputs or parameters, and these uncertainties can be of a variety of types: based on data, expert opinion, or simply an 'uncertainty range.'
- Only limited model-run data may be available.
- Field data of the actual process under consideration may be limited and noisy.
- Data may be of a variety of types, including functional data.
- Model-run data and field data may be observed at different input values.
- One may desire to 'tune' unknown parameters of the computer model based on field data, and at the same time (because of sparse data) apply the validation methodology.
- There may be more tuning parameters than data, so that the tuning parameters are not even identifiable.
- The computer model itself will typically be highly non-linear.
- Accounting for possible model bias is challenging.
- Validation should be viewed as an accumulation of evidence to support confidence in the model outputs and their use, and the methodology needs to be able to update its current conclusions as additional information arrives.

Overcoming these hurdles requires a powerful and flexible methodology. The Bayesian approach to assessment and analysis of uncertainty, which we adopt here, is one such methodology. This approach is discussed in Section 5, together with its modern computational implementation via Markov Chain Monte Carlo analysis (see, e.g. Robert and Casella, 1999).

Bridging two philosophies: At the risk of considerable oversimplification, it is useful to categorize the approaches to model evaluation as being in one of two camps. In one camp, evaluation is performed primarily by comparing model output to field data from the real process being modeled. The common rationale for this philosophy is the viewpoint that the only way to see if a model actually works is to see if its predictions are correct. We will call this the *predictive* approach to evaluation.

The second camp primarily focuses on the model itself, and tries to assess the accuracy or uncertainty corresponding to each constructed element of the model. The common rationale for this philosophy is that, if all the elements of the model (including computational elements) can be shown to be correct, then logically the model must give accurate predictions. We will call this the *physical* approach to model evaluation.

Our own view lies primarily in the predictive camp, in that a modeler faces considerable difficulty in convincing others that all elements of the model have been correctly constructed, without demonstration of validity on actual field data.

That said, it is worth noting that Bayesian methodology bridges both these philosophies. First, one can specify a prior probability that the computer model is correct and update this probability based on any available data. Thus someone in the physical camp might declare that their prior probability is 0.96 that the model is correct. If field data is then obtained, a Bayesian computation (see Section 9) might yield a *posterior* probability of 0.99 (in the case of supporting data) or 0.009 (in the case of non-supporting data) that the model is correct. Those in the predictive camp (including ourselves) believe that such extreme prior specification is excessively informative and only rarely justifiable.

Even in the predictive approach, however, Bayesian analysis allows utilization of prior information about elements of the model from the physical approach (either expert opinion or partial scientific knowledge), together with field data, in the construction of the tolerance bounds for model predictions; it incorporates whatever information is available to produce defensible quantification of the adequacy of the model's representation of reality. Furthermore, such physical knowledge can significantly reduce the amount of field data that is needed for predictive validation.

Side benefits of the methodology: Because the investment in understanding and using this methodology is admittedly significant, we mention some of the side benefits that arise from the implementation as done in the body of this paper.

1. When a bias in the model is detected by comparison with field data, the methodology automatically allows one to adjust the prediction by the estimated bias, and provides tolerance bounds for this adjusted prediction. This can result in considerably more accurate predictions than use of the model alone (or use of the field data alone).

2. A fast approximation to the computer model is available for use in situations, such as optimization, where it may be too expensive to use the computer model itself.

3. Predictions and tolerance bounds can be given for applications of the computer model to new situations in which there is little – or no – field data, assuming information about 'related' scenarios is available.

A Caveat: The process of model validation is inherently highly statistical, and is inherently a hard statistical problem. This is not to say that the scientific and mathematical sides of the V&V process are not also of central importance, but the basic problem cannot be solved without use of sophisticated statistical methodology. Indeed, the statistical problem is so hard that one rarely sees analyses that actually produce tolerance bounds for computer model predictions.

The intent of this paper is essentially to provide a 'proof of concept,' that it is possible to provide tolerance bounds for predictions of computer models, while taking into account all the uncertainties present in the problem. However, the computations required in the methodology we propose can be intensive, especially when there are large numbers of model inputs, large numbers of unknown parameters, or a large amount of data (model-run or field). The test bed examples we consider in this paper are relatively modest in these dimensions, and we have yet to see how the full methodology scales-up to more complex settings (although some components of the methodology are known to scale-up to considerably more complex situations). It is likely that a variety of simplifications and/or innovations will be needed in such settings in order to apply the methodology.

Overview: In this paper we will restrict consideration to computer models that are deterministic, as opposed to stochastic. Section 1.2 provides an outline of the framework we recommend for computer model evaluation. Two testbed models are introduced in Section 1.3, a *resistance spot welding model* and a *crash model.* Background details of the test bed models are in Appendices A and B.

The proposed methodology for model evaluation is presented in Sections 2 through 6, with illustrations on the two test bed models. Sections 7 through 10 introduce a variety of generalizations that are needed to deal with specific contexts.

To prevent notational overload, we introduce notation and concepts as they arise in the evaluation framework. Appendices C, D, E and F present some of the technical details needed for implementation of the methodology.

1.2 Sketch of the framework

Validation can be thought of as a series of activities or steps. These are roughly ordered by the sequence in which they are typically performed. The completion of some or all in the series of activities will typically lead to new issues and questions, requiring revision and revisiting of some or all of the activities, even if the model is unchanged. New demands placed on the model and changes in the model through new development make validation a continuing process. The framework must allow for such dynamics.

Step 1. Specify model inputs and parameters with associated uncertainties or ranges - the Input/Uncertainty (I/U) map. This step requires considerable expertise to help set priorities among a (possibly) vast number of inputs. As information is acquired through undertaking further steps of the validation process, the I/U map is revisited, revised and updated.

Step 2. Determine evaluation criteria. The defining criteria must account for the context in which the model is used, the feasibility of acquiring adequate computer-run and field data, and the methodology to permit an evaluation. In turn the data collection and analyses will be critically affected by the criteria. Moreover, initially stated criteria will typically be revisited in light of constraints and results from later analyses.

Step 3. Data collection and design of experiments. Both computer and field experiments are part of the validation (and development) processes; multiple stages of experimentation will be common. The need to design the computer runs along with field experiments can pose non-standard issues. As noted above, any stage of design must interact with the other parts of the framework, especially the evaluation criteria.

Step 4. Approximation of computer model output. Model approximations (fast surrogates) are usually key for enabling the analyses carried out in Step 5; fast surrogates are essential also when the model is used for optimization of e.g., a manufacturing product design.

Step 5. Analyses of model ouput; comparing computer model output with field data. Uncertainty in model inputs will propagate to uncertainty in model output and estimating the resulting output distribution is often required. The related 'sensitivity analysis' focuses on ascertaining which inputs most strongly affect outputs, a key tool in refining the I/U map.

Comparing model output with field data has several aspects.

- The relation of reality to the computer model ("reality = model + bias")
- Statistical modeling of the data (computer runs and field data where "field data = reality + measurement error")
- Tuning/calibrating model input parameters based on the field data
- Updating uncertainties in the parameters (given the data)
- Accuracy of prediction given the data

The methods used here rely on a Bayesian formulation; the details are in Section 5. The fundamental goal of assessing model accuracy is addressed there.

Step 6. Feedback information into current validation exercise and feed-forward information into future validation activities. Feedback refers to use of results from Step 5 to improve aspects of the model, as well as to refine aspects of the validation process. Feed-forward refers to the process of utilizing validations of current models to predict the validity of related future models, for which field data are lacking.

1.3 Testbeds

The test beds provide context for implementing each activity and also prompt consideration of a full variety of issues. The description of the validation framework, in Section 2, does not capture the details and nuances encountered in any implementation. This fleshing out of details for the test beds is done throughout Sections 2–8 where each activity/step of validation is accompanied by explicit application to the test bed models. The result is the addition of concreteness to the generalities of the methods.

Testbed 1. The Resistance Spot Welding Model (SPOT WELD): In resistance spot welding, two metal sheets are compressed by water-cooled copper electrodes, under an applied load, L. Figure 14 in Appendix A is a simplified representation of the spot weld process, illustrating some of the essential features for producing a weld. A direct current of magnitude C

is supplied to the sheets via the two electrodes to create concentrated and localized heating at the interface where the two sheets have been pressed together by the applied load (the so-called faying surface). The heat produced by the current flow across the faying surface leads to melting and, after cooling, a weld "nugget" is formed.

The resistance offered at the faying surface is particularly critical in determining the magnitude of heat generated. Because contact resistance at the faying surface, as a function of temperature, is poorly understood a nominal function is specified and "tuned" to field data. The effect of this tuning on the behavior of the model is the focus of the test bed example.

The physical properties of the materials will change locally as a consequence of local increase in temperature. Young's modulus and the yield stress of the sheet will fall (that is, the metal will "soften") resulting in more deformation and increase in the size of the faying contact surface, further affecting the formation of the weld. At the same time, the electrical and thermal conductivities will decrease as the temperature rises; all of which will affect the rate of heat generation and removal by conduction away from the faying surface.

The thermal/electrical/mechanical physics of the spot weld process is modeled by a coupling of partial differential equations that govern heat and electrical conduction with those that govern temperature-dependent, elastic/plastic mechanical deformation (Wang and Hayden, 1999).

Finite element implementations are used to provide a computer model of the electro-thermal conceptual model. Similarly, a finite element implementation is made for the equilibrium and constitutive equations that comprise the conceptual model of mechanical/thermal deformation. These two computer models are implemented using a commercial code (ANSYS).

Details of the inputs and outputs of the models are in Appendix A and are summarized in Table 1. The particular issues faced are spelled out as we proceed through the exposition in the following sections.

Testbed 2. The Crash Model (CRASH): The effect of a collision of a vehicle with a barrier is routinely done through a computer model implemented as a non-linear dynamic analysis code using a finite element representation of the vehicle. Proving ground tests with prototype vehicles are ultimately made to meet mandated standards for crashworthiness. But, the computer models play an integral part in the design of the vehicle to assure crashworthiness before manufacturing the prototypes. How well the models perform is therefore crucial to the manufacturing process.

CRASH is implemented via a commercial code, LS-DYNA. Our focus is on the velocity changes after impact at key positions on the vehicle, such as the driver seat and radiator. Details of the model and a typical set of inputs are in Appendix B. Geometric representation of the vehicle and the material properties play critical roles in the behaviour of the vehicle after impact and the necessary detailing of these inputs leads to very costly (in time) computer runs. Field data involve crashing of prototype vehicles and therefore costly in dollars. CRASH is then inherently data-limited, presenting a basic challenge to assessing the validity of the model.

Variables and sources of uncertainty in the vehicle manufacturing process and proving ground procedures induce uncertainties in the test results. The acceleration and velocity histories of two production vehicles of the same type, subjected to 30mph zero degree rigid barrier frontal impact tests, as shown in Figure 15 demonstrate the differences in "replicate" crashes. There are a variety of materials used in components of the vehicle and, consequently, a variety of material properties to deal with, not all of which may be satisfactorily specified.

2 Understanding the Model and Its Uses (Steps 1 and 2)

The beginning of the validation process is understanding the uncertainties associated with the computer model itself, and determining how the model is to be used.

2.1 Step 1. Specify model inputs and parameters with associated uncertainties or ranges - the Input/Uncertainty (I/U) Map

Understanding what is known and not known about a computer model can be important in its evaluation. A convenient way to organize such information is through what we call the Input/Uncertainty map. (This is related to the idea of a PIRT - see Pilch *et al.*, 2001.) The map has four attributes:

a) A list of model features or inputs of potential importance

b) A ranking of the importance of each input

c) Uncertainties, either distributions or ranges of possible values, for each input

d) Current status of each input describing how the input is currently treated in the model.

The I/U map is dynamic: as information is acquired and the validation process proceeds, the attributes, especially b)-d), will change or be updated. This will become more evident following Steps 4-6.

The inputs in the map are drawn from the development process. They will include parameters inherent to the scientific and engineering assumptions and mathematical implementation, and numerical parameters associated with the implementing code; in short, all the ingredients necessary to make the model run. Because this list can be enormous, the more important parameters must be singled out to help structure the validation process by providing a sense, albeit imperfect, of priorities. We adopt a scale of 1-5 for ranking the inputs with 1 indicating only minor likely impact on prediction error and 5 indicating significant potential impact.

SPOT WELD: The purpose of the spot weld model is to investigate the process parameters for welding aluminum. The I/U map of the model is in Table 1. The list of inputs in Table 1 is more fully described in Appendix A. Initially, only three inputs have rank 5 based on the model developer's assessment. These three parameters (and gauge) are the focus of the initial validation experiments; earlier experiments by the model developer led to the impact assessments appearing in the table. The controllable parameters, current, load, and gauge, will be given ranges when the experiments are designed. In the current context, validation is with laboratory data and "no uncertainty" is appropriate when current and load levels are set in the laboratory. If, however, validation is required at the production level then uncertainties in current and load may be significant. In brief, the I/U map is context dependent.

There are several specific items connected with the I/U map in Table 1 that are worth noting. First, the most significant specified uncertainty (impact factor 5) in the model elements is that of the contact resistance. The model incorporates contact resistance through an equation that, for the faying surface, has a multiplicative constant u about which it is only known that

u lies in the interval $[0.8, 8.0]$. It will be necessary to *tune* this parameter of the model with field data. The second most significant uncertainty in the model (impact factor 4) is the linear approximation for stress/strain. The modeler was unable to specify the uncertainty regarding this input, and so error in this element will simply have to enter into the overall unknown (and to be estimated) bias of the model.

INPUT		IMPACT	UNCERTAINTY	CURRENT STATUS
Geometry	electrode symmetry-2d	3	unspecified	fixed
	cooling channel	1	unspecified	fixed
	gauge	unclear	unspecified	1, 2mm
materials		unclear	Aluminum (2 types $\times$ 2 surfaces)	fixed
Stress/ strain	piecewise linear	4	unspecified (worse at high T)	fixed
	C_0, C_1, σ_s	3	unspecified	fixed
contact resistance	$1/\sigma = u \cdot f$; f fixed	3	unspecified	fixed by modeler
	$u = 0$ (electrode/sheet) u =tuning (faying)	5	$u \in [0.8, 8.0]$	tuned to data for 1 metal
thermal conductivity κ		2	unspecified	fixed
current		5	no uncertainty	controllable
load		5	no uncertainty	controllable
mass density (ρ)		1	unspecified	fixed
specific heat (c)		1	unspecified	fixed
numerical parameters	mesh	1	unspecified	convergence/speed compromise
	M/E coupling time	1	unspecified	
	boundary conditions	1	unspecified	fixed
	initial conditions	1	unspecified	fixed

Table 1: The I/U map for the spot weld model

Table 7 in Appendix B gives the corresponding I/U map for the crash model.

Initial impact assessments will be based on experience to reflect a combined judgment of the inherent sensitivity of the input (the extent to which small changes in the input would affect the output) and the range of uncertainty in the input. These will be revised through sensitivity analyses and 'tuning with data' that occur later in the process. Inputs about which we are "clueless" might be singled out for attention at some point along the validation path but the effect of "missing" inputs (i.e., non-modeled features) may never be quantifiable or only emerge after all effects of "present" inputs are accounted for.

In model validation, considerable attention is often paid to the issue of numerical accuracy of the implemented model – for instance, in assessing if numerical solvers and finite element (FEM) codes have ‘converged’ to the solution of the driving differential equations. This is an important consideration and, as detailed in Cafeo and Cavendish (2001), is an issue of model and code verification. It should ideally be addressed early in the model development process and prior to the validation activity emphasized in this paper.

It is often the case, however, that convergence will not have been obtained; e.g., modelers may simply use the finest mesh size that is computationally feasible, recognizing that this mesh size is not sufficient to have achieved convergence. The method we are describing for validation still works. The error introduced by a lack of convergence becomes part of the ‘bias’ of the model that is to be assessed (see Section 3). The I/U map should, of course, clearly indicate the situation involving such convergence. This means that parameters such as grid size may be confounded with other assumptions about the model making it more difficult to improve the model. Ideally, this could be done through designed experiments, varying values of the numerical parameters in order to assess numerical accuracy.

2.2 Step 2. Determine evaluation criteria

Evaluation of a model depends on the context in which it is used. Two key elements of evaluation criteria are:

- Specification of an evaluation criterion (or criteria) defined on model output
- Specification of the domain of input variables over which evaluation is sought.

Even if only one evaluation criterion is initially considered, other evaluation criteria inevitably emerge during the validation process. In fact, it is often desirable to have multiple outputs to compare with reality to help assess the usefulness of the model. The overall performance of the model may then depend on the outcomes of the validation process for several evaluation criteria – the model may fail for some and pass for others – leading ultimately to follow-on analyses about when and how the model should be used in prediction.

Informal evaluations are typical during the development process – does the computer model produce results that appear consistent with scientific and engineering intuition? Later in the validation process these informal evaluations may need to be quantified and incorporated in the “formal” process. Sensitivity analyses may, in some respects, be considered part of evaluation if, for example, the sensitivities confirm (or conflict with) scientific judgment. We defer discussion of sensitivity to Section 10.1.

The evaluation criteria can introduce complexities that would need to be addressed at Steps 4 - 6, but may also affect the choices made here. For example, an evaluation criterion that leads to comparisons of curves or surfaces or images places greater demands on the analyst than simpler scalar comparisons.

Of necessity, the specifications must take into account the feasibility of collecting data, particularly field data, to carry out the validation. This can be further complicated by the need to calibrate or tune the model using the collected data; the tuning itself being driven by the specifications.

SPOT WELD: Two evaluation criteria were initially posed:

I. Size of the nugget after 8-cycles

II. Size of the nugget as a function of the number of cycles

The first criterion is of interest because of the primary production use of the model; the second as a possible aid in reducing the number of cycles to achieve a desired nugget size. Ideally the evaluation would be based directly on the strength of the weld, but weld diameter is taken as a surrogate because of the feasibility of collecting laboratory data on the latter. (Of course, if nugget size is not strongly correlated with weld strength, these criteria would probably be inappropriate.) In production, the spot welding process results in a multiple set of welds, but the evaluation criterion considered here involves only a single weld. Criterion (II) was later discarded as a result of the difficulty during data collection of getting reliable computer runs producing output at earlier times than 8-cycles.

Specification of the feasible domains of the input variables is another aspect of formulating the evaluation criteria. For the spot weld model, these domains are:

- Material: Aluminum 5182-O and Aluminum 6111-T4
- Surface: treated or untreated
- Gauge (mm): 1 or 2
- Current (kA): 21 to 26 for 1mm aluminum; 24 to 29 for 2mm aluminum
- Load (kN): 4.0 to 5.3

Material and surface might enter the model through other input variables relating to properties of materials. Our initial specification in Table 1 considers material, surface and gauge as fixed. The tuning parameter, u, has the range indicated and is the only other input that is not fixed.

CRASH: For the first experiment the input consists solely of the impact velocity v. The specific output data to be analyzed is the velocity of the "Sensing and Diagnostic Module", SDM, situated under the driver's seat, relative to a free-flight dummy. This relative velocity is obtained by subtracting the impact velocity v from the actual SDM velocity (it being assumed that the dummy maintains velocity v over the time interval of interest). The resulting functions vary (at least theoretically) between 0 at the time of impact $t = 0$ and $-v$ at the time the vehicle is stationary.

The evaluation criterion we consider is the SDM velocity calculated 30ms before the time the SDM displacement (relative to the free-flight dummy), DISP, reaches 125mm. Call this quantity CRITV. The airbag takes around 30ms to fully deploy, which is why this particular evaluation criterion, CRITV, is important. Our analysis takes account of the dependence between displacement and velocity (displacement is the integral of velocity) by working with the probability distribution of the velocity and then finding the implied distribution of the displacement.

The process we follow can be adapted to treat other evaluation criteria such as,

- Time at which SDM displacement reaches 125mm
- SDM velocity when SDM displacement reaches 250mm and 350mm

The evaluation criterion

- Velocity at the center of the radiator, RDC, 30ms before SDM displacement reaches 125mm poses different issues because it requires a combined analysis of the functional data from two sensors, one located at the radiator center, the other under the driver's seat.

3 Data Collection (Step 3)

Both computer and field (laboratory or production) experiments are part of the validation and development processes and produce data that are essential for

- Developing needed approximations to (expensive) numerical models
- Assessing bias and uncertainty in model predictions
- Studying sensitivity of a model to inputs
- Identifying suspect components of models
- Designing and collecting data that build on, and augment, existing, or historical, data.

The iterative and interactive nature of the validation and development processes will result in multiple stages of computer experiments and even field experiments.

Typically, an effort is made to construct experiments that yield data over the ranges of what are considered to be the key input values. For low-dimensional input spaces, this can be done rather informally. For instance, in CRASH, the key inputs are the impact speed of the vehicle and the collision barrier type. Table 2 exhibits the entire set of model inputs and measured field inputs for the available data. The type of data resulting from each experiment is indicated in Figure 15 of Appendix B.

When the input space is of larger dimension, it is preferable to use formal "space-filling" strategies of choosing the input values at which to experiment. For instance, in the spot weld test bed there is one discrete and three continuous input variables of major importance, and covering the 3-dimensional space with a limited number of runs (field or model) requires careful experimental design. Among the most useful designs in such contexts is the Latin Hypercube Design (McKay, Conover and Beckman(1979)). We utilize code from W. Welch to produce such designs.

SPOT WELD: For the spot weld model there was limited model data about the tuning parameter u. The initial computer experiment was therefore aimed at assessing the effect of u. The inputs to be varied are C = current, L = load, G = gauge, and u. The other inputs were held fixed. The cost – thirty minutes per computer run – is high so a limited number, 26, of runs were planned for each of the two gauge sizes. The 26 runs for 1 mm metal covered the 3-dimensional rectangle, [20,27]x[3.8,5.5]x[1]x[1.0,7.0], in C, L, G, u space, while those for the 2mm metal covered the 3-dimensional rectangle, [23,30]x[3.8,5.5]x[2]x[0.8,8.0]. The explicit design values obtained from the Welch code are in Table 3, along with the model output and the corresponding laboratory data for the nugget diameter.

The computer runs exhibited some aberrant behavior. Many (17) runs failed to produce a meaningful outcome at cycle 8; these runs were eliminated. For reasons that are not yet clear

Impact velocity (km/h) used in model	barrier type	Impact velocity (km/h) of field tests
19.3	straight frontal	19.3
25.5	straight frontal	25.5
28.9	straight frontal	28.9
32.1	straight frontal	32.1
35.3	straight frontal	35.3
38.4	straight frontal	38.4
41.3	straight frontal	41.3, 41.3
49.3	straight frontal	49.4, 49.2, 49.4, 49.3, 49.3, 49.4
56.4	straight frontal	56.4
22.5	left angle	22.5
32.2	left angle	32.2
40.2	left angle	40.2, 41.4, 41.5
41.9	left angle	41.9
49.3	left angle	49.5, 49.2
56.2	left angle	56.2
57.3	left angle	57.3
28.9	right angle	28.9
31.9	right angle	31.9
41.7	right angle	41.7, 41.8
48.3	right angle	48.3
19.3	center pole	19.3
25.5	center pole	25.5
32.0	center pole	32.0
36.8	center pole	36.8
40.3	center pole	40.3
48.6	center pole	48.6

Table 2: Available input data

many runs were unable to produce reliable data for earlier cycle times; as a result evaluation criteria depending on early cycle times were abandoned. The data retained (35 runs) are used in the subsequent analyses.

To provide insight into the space-filling nature of the Latin Hypercube Design used for gauge=1 in Table 3, the 2-dimensional projections of this design are shown in Figure 1. An important feature of such designs is that they exercise the code over a wide range of inputs and often unearth code difficulties (for example, in Table 3 there were many failed runs for reasons not yet determined). Such designs are effective for a wide variety of purposes (sensitivity analyses, response surface approximations to model output, predicting outcomes of the specified evaluation criteria). In contexts where initial computer experimentation points to narrowing, or altering, the region for exploration specified in Step 2, new designs or augmentation of an initial design must be found. For extremely expensive model runs (or field runs), sequential designs might be considered, where each additional design point is chosen 'optimally' based on the information from previous runs.

Field data will usually be harder to obtain than computer experimental data. As a result, designing the field data will depend more crucially on the specifications in Section 2.2 and specific

Gauge (mm)	u (-)	Load (kN)	Current (kA)	Nugget Dia. (mm)		Gauge (mm)	u (-)	Load (kN)	Current (kA)	Nugget Dia. (mm)
1	6.52	4.072	26.44	–		2	4.544	3.936	27.76	7.15
1	4.60	4.684	21.68	5.64		2	5.696	4.14	25.52	6.39
1	3.64	5.024	23.64	–		2	1.088	4.684	28.32	6.38
1	7.00	4.412	23.36	–		2	0.8	4.276	24.40	4.87
1	6.76	4.888	25.04	–		2	3.68	4.412	26.08	6.47
1	1.00	4.82	22.52	4.36		2	4.832	4.616	23.00	6.68
1	3.40	4.616	27.00	–		2	7.136	4.344	27.20	6.71
1	5.32	4.48	20.84	6.12		2	4.256	5.228	24.68	6.54
1	2.92	5.092	20.56	5.00		2	3.392	4.004	23.28	5.97
1	1.48	5.364	21.12	4.53		2	1.952	4.48	23.84	5.72
1	2.20	4.004	21.40	5.20		2	2.528	3.8	24.96	6.23
1	2.68	4.344	25.88	–		2	2.24	4.208	29.72	–
1	2.44	5.50	23.08	–		2	1.376	5.024	25.80	5.46
1	4.36	3.80	25.32	–		2	7.424	4.072	28.88	–
1	1.24	4.208	24.76	6.06		2	6.272	4.548	29.16	7.36
1	6.04	4.752	20.00	–		2	6.848	5.364	23.56	–
1	5.56	5.432	25.60	–		2	3.968	4.888	29.44	7.16
1	1.96	4.956	26.16	6.69		2	3.104	5.432	28.60	6.61
1	5.80	3.936	23.92	7.17		2	5.12	5.5	26.64	5.98
1	4.84	4.14	22.80	–		2	6.56	3.868	26.36	6.74
1	3.16	3.868	22.24	5.71		2	5.984	4.956	24.12	5.32
1	6.28	5.228	21.96	5.38		2	8	5.092	28.04	–
1	1.72	4.548	24.20	5.85		2	2.816	4.82	26.92	6.70
1	5.08	5.16	26.72	–		2	5.408	5.16	30.00	–
1	4.12	5.296	24.48	6.87		2	1.664	5.296	27.48	6.02
1	3.88	4.276	20.28	4.91		2	7.712	4.752	25.24	5.50

Table 3: Spot weld data from 52 model runs. Run failures indicated by –

methods cannot be stated in advance.

In Sections 4, 5 and 7 we set down an informal description and assumptions for the computer model data and field data. This includes consideration of calibration parameters, function output and the treatment of the arguments of such functions. A more formal description can be found in Bayarri *et al.* (2002).

4 Model Approximation (Step 4)

It is often of interest to see the effect of uncertainty in model inputs on model outputs. When the code is cheap to run, then straight simulation (i.e., randomly generate input variables from their distributions and compute the corresponding model outputs) is a practical option for determining the output distribution. More refined methods relying on pseudorandom (e.g., Latin hypercube) generation of inputs can also be employed – at least when the number of input variables is modest – and somewhat extend the range of applicability of straight simulation. None of these techniques

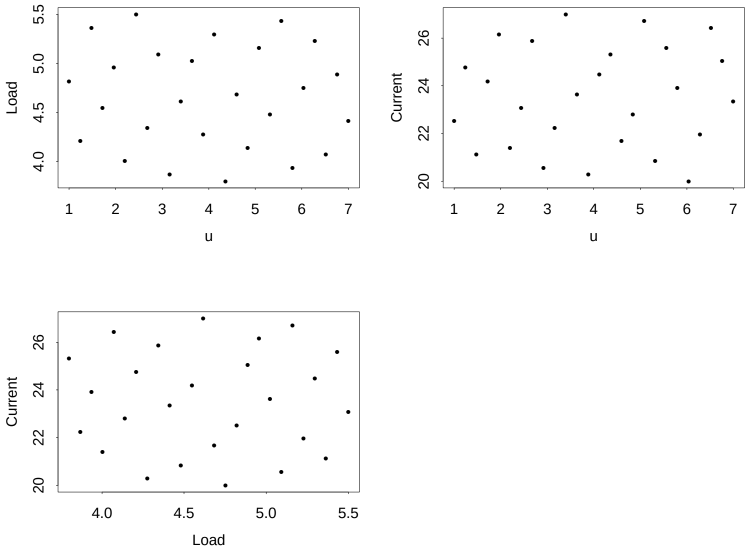

Figure 1: Latin Hypercube Design used for spot weld

are feasible, however, for expensive codes, and one must then resort to model approximations to obtain output distributions. Such approximations can also be useful in their own right, for at least the following reasons.

- It might not be feasible to directly employ the model ‘in the field’, whereas a fast approximation to the model could be directly employed.
- It is often desired to perform an optimization over inputs. Common optimization algorithms can be too expensive to implement with the computer model, but can be implemented with the approximation (or at least the approximation can be used to significantly narrow the range of input values over which optimization with the computer model needs to be done).
- Finding optimal designs for additional model-development or validation experiments can require a fast approximation to the computer model.
- In Step 5, we will make crucial use of model approximations in implementing the calibration and validation methodology.

There are four basic techniques that can be useful in model approximation: (i) use of models having lower resolution (e.g., larger mesh size) or including only significant basis elements (based on, e.g., Proper Orthogonal Decomposition or Principal Components methods); (ii) linearization/Gaussian error accumulation; (iii) response surface methodology, including Gaussian processes and neural networks; (iv) Bayesian networks, which allow uncertainty transference between sub-models from which the model is constructed. The first technique is always an option, and can be combined with the other methods; of course, evaluation of the error introduced by using a model of lower resolution (or with a smaller basis) can be difficult. The second technique, which essentially linearizes the model so that (Gaussian) input distributions can be passed through the model using linear Gaussian updating, is useful if it is feasible to work with the underlying code of the model and if linearization does not introduce severe bias. The use of Bayesian networks is not addressed here.

A very useful general tool, for models whose output depends smoothly on inputs (very common in engineering and scientific processes), is the response surface technique. (It should be noted that, even when the underlying process is not a smooth function of the inputs, one is often primarily interested in features of the output that are smooth.) The approach we recommend has been successfully used when the number of input variables is less than 20 (typically requiring less than 10 runs per input) and even as high as 40 (although then several hundreds of model runs may be needed for accurate fitting). Below we briefly describe this technique. The particular technique we recommend meshes well with the validation analysis proposed in Step 5.

Notation: Denote model output by $y^M(\boldsymbol{x}, \boldsymbol{u})$, where $\boldsymbol{x}$ is a vector of controllable inputs and $\boldsymbol{u}$ is a vector of unknown calibration and/or tuning parameters in the model. Sometimes we write $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$.

In specific examples $\boldsymbol{u}$ may be absent. The goal is to approximate $y^M(\boldsymbol{x}, \boldsymbol{u})$ by a function $\hat{y}^M(\boldsymbol{x}, \boldsymbol{u})$, to be called the *model approximation*, which is much easier to compute. In addition, it is desirable to have a variance function $V^M(\boldsymbol{x}, \boldsymbol{u})$ that measures the accuracy of $\hat{y}^M(\boldsymbol{x}, \boldsymbol{u})$ as an estimate of $y^M(\boldsymbol{x}, \boldsymbol{u})$. A response surface approach that achieves both these goals is the Gaussian process response surface approximation (GASP), described in Sacks et al. (1989) and Kennedy and O'Hagan (2001); the approach is outlined below.

SPOT WELD: The vector of controllable inputs is $\boldsymbol{x} = (C, L, G)$, the tuning parameter is u. Use of GASP with the data from Table 3 leads to the response surface approximation to $y^M(C, L, G, u)$ that is exhibited in Figure 2. We do not explicitly show the variance function, but it is available. For instance, at $(C, L, G, u) = (26, 5, 2, 4)$, the response surface approximation to $y^M(26, 5, 2, 4)$ is $\hat{y}^M(26, 5, 2, 4) = 6.12$, and the variance of the approximation is $V^M(26, 5, 2, 4) = 0.0046$. At the values of the actual model data of Table 3, i.e. the solid dots in the figures below, the response surface approximation is exact; it 'passes through' these points. The slight up-curve at the edges, for extreme values of u, occurs because the model data in those regions is very sparse and an overall mean level was used in the GASP analysis (as opposed to, say, a linear function). This has essentially no effect on ultimate predictions, since we will see that the central values of u are those that are most relevant.

CRASH: The controllable inputs are $\boldsymbol{x} = (v, B)$, where v is the impact velocity and B is the barrier type. There is no tuning parameter.

Let $\boldsymbol{y}^M = (y^M(\boldsymbol{x}^1, \boldsymbol{u}^1), \ldots, y^M(\boldsymbol{x}^m, \boldsymbol{u}^m))$ denote the vector of m evaluations of the model at the inputs $D^M = \{(\boldsymbol{x}^i, \boldsymbol{u}^i) : i = 1, \ldots, m\}$. The computer model is exercised only at the inputs D^M, so that $y^M(\boldsymbol{z}) = y^M(\boldsymbol{x}, \boldsymbol{u})$ is effectively unknown for other inputs $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$. Thus, in

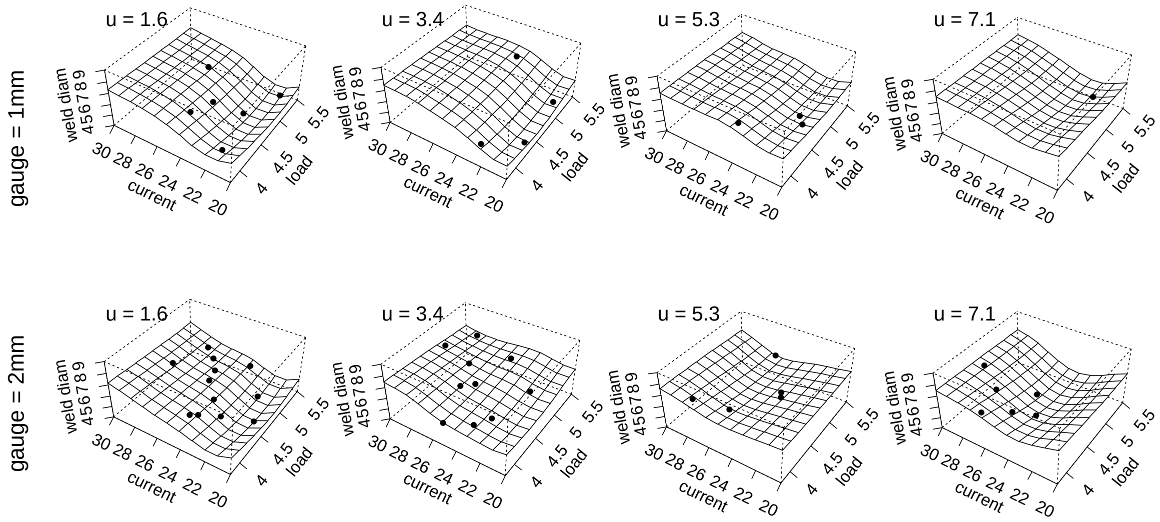

Figure 2: GASP response surface approximation $\hat{y}^M$ to y^M for the spot weld model, constructed from the data in Table 3. The surfaces show estimated weld diameter, for the two gauge values, as a function of load and current for various values of the tuning parameter u. The solid dots denote model data $y^M(C, L, G, u)$ and are plotted on the surface corresponding to the closest value of u.

the Bayesian framework, we assign $y^M(\boldsymbol{z})$ a prior distribution, specifically, a stationary Gaussian process with mean and covariance functions governed by unknown parameters. (In application, we always only deal with a finite set of $\boldsymbol{z}_i$, in which case the Gaussian process at these points reduces to a multivariate normal distribution.) Choice of the mean function is discussed below, but discussion of the covariance function is delayed until Appendix C.1.

The mean function of the Gaussian process will be assumed to be of the form $\boldsymbol{\Psi}(\cdot)\boldsymbol{\theta}^L$ where $\boldsymbol{\Psi}(\boldsymbol{z})$ is a specified $1 \times k$ vector function of the input $\boldsymbol{z}$ and $\boldsymbol{\theta}^L$ is a $k \times 1$ vector of unknown parameters. A constant mean ($k = 1$, $\Psi(\boldsymbol{z}) = 1$, and $\boldsymbol{\theta}^L = \theta$) is often satisfactory, if one plans only to use the model approximation within the range of the available model-run data, but a more complicated mean function can be useful if the model approximation is to be used outside the range of the data. (When outside the range of the model-run data, the Gaussian process approximation to the model will gradually tend towards its estimated mean function, so that an accurate estimated mean function will provide more accurate model approximations.) This can be especially important when features such as seasonal trends are present. Formally, the mean function above does not allow the presence of a known constant (e.g., $c + \boldsymbol{\Psi}(\cdot)\boldsymbol{\theta}^L$), but this can be easily accommodated by carrying out the analysis with the Gaussian process defined by subtracting c from the original process.

A secondary benefit of introducing a mean function that is a reasonable approximation to the model is that it will often result in smaller variances for the model approximations. If, however, a more complicated mean function is used but is not a more reasonable approximation to the model, it will result in larger variances, since it will contain more parameters that must be estimated. No firm guidelines are available as to whether a simple mean function or a carefully developed mean function are best. Our recommendation is to try to incorporate into the mean function any obvious trends that exist in the model output but, again, even a constant mean function is often satisfactory.

> **CRASH**: The computer model output corresponding to velocity, in a typical case, is indicated in Figure 15 (the curve on the right). Such curves are clearly better modeled by a linear function of time than a constant in time. Furthermore, we know the initial velocity v of the vehicle, so use of the mean function $v(1-\theta^L t)$ for the Gaussian process will clearly do a better job of approximating the computer model than would a constant mean. (It should be emphasized, however, that the methodology will typically provide accurate within-sample approximation to the model output no matter what mean function is chosen for the Gaussian process.) We actually follow common practice in this area and first transform the data by subtracting the initial velocity, leading to what are called 'relative velocity' curves; for relative velocity curves, the natural mean function would be $-\theta^L vt$, corresponding to choosing $k=1$ and $\Psi(v,t) = -\,v\,t$ in the above notation. Note that since the theoretical range of the relative velocity is from 0 (at time $t=0$) to $-v$ (at time t_s, when the vehicle reaches stationarity), θ^L can here be interpreted as $1/t_s$.

For specified values of the parameters (such as $\boldsymbol{\theta}^L$) of the Gaussian process, the GASP behaves as a Kalman Filter, yielding a posterior mean function that can be used as the fast approximation to $y^M(\cdot)$ together with a variance measuring the uncertainty in the approximation. (Details are given in Appendix C.1.) Note that this variance is zero at the design points at which the function was actually evaluated. The model approximation obtained through the GASP theory can thus roughly be thought of as an interpolator of the data, unless there is numerical instability in the computer model, as mentioned in footnote 2, in which case the approximation smoothes the data.

Unfortunately, the parameters (such as $\boldsymbol{\theta}^L$) of the Gaussian process are rarely, if ever, known. Two possibilities then arise:

a) Plug-in some estimates, for instance maximum likelihood estimates (as in the GASP software of W. Welch – see also Bayarri *et al.* (2002)), pretending they are the 'true' values.

b) Average over the posterior distribution of the parameters, in a full Bayesian analysis (as described in Section 5).

The full Bayesian analysis is typically superior, in the sense that the resulting variance of the model approximation will more accurately reflect reality, since the parameters are unknown. In terms of the actual model approximation $\hat{y}^M(\boldsymbol{x},\boldsymbol{u})$, however, use of maximum likelihood estimates of the parameters typically yield much the same answers as the full Bayesian analysis, and so may be preferable in computationally intensive situations.

The particular GASP approach that we use has the added bonus that certain types of stochastic inputs, $\boldsymbol{z}$, can easily be handled within the same framework; see Appendix C.2 for details.

5 Analysis of Model Output (Step 5)

In this section, we describe the basics of the statistical modeling and analysis that are used for model evaluation. For illustration in this section we use only SPOT WELD, since CRASH has a functional data structure that we do not introduce until Section 7.

5.1 Notation and statistical modeling

The model is an approximation to reality. Another way of saying this is that the model is a biased representation of reality, and accounting for this bias is the central issue for model validation. There are (at least) three sources for this bias:

1. The science or engineering used to construct the model may be be incomplete.
2. The numerical implementation may introduce errors (e.g., may not have converged).
3. Any tuned parameters may be in error.

Furthermore, the model alone cannot provide evidence of bias. Either expert opinion or field data is necessary to assess bias – we focus on the latter.

Recall that we denote by $y^M(\boldsymbol{x}, \boldsymbol{u})$ the model output when $(\boldsymbol{x}, \boldsymbol{u})$ is input. When $\boldsymbol{u}$ is not present, as in CRASH, we can statistically model "reality = model + bias" as

$$y^R(\boldsymbol{x}) = y^M(\boldsymbol{x}) + b(\boldsymbol{x})\,, \tag{5.1}$$

where $y^R(\boldsymbol{x})$ is the value of the 'real' process at input $\boldsymbol{x}$ and $b(\boldsymbol{x})$ is the unknown bias function, arising from the sources discussed above. When $\boldsymbol{u}$ is present we call its true (but unknown) value $\boldsymbol{u}^*$ and then model the bias via

$$y^R(\boldsymbol{x}) = y^M(\boldsymbol{x}, \boldsymbol{u}^*) + b(\boldsymbol{x}). \tag{5.2}$$

Field data at inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ are obtained, and modeled as

$$y^F(\boldsymbol{x}_i) = y^R(\boldsymbol{x}_i) + \epsilon_i^F\,, \tag{5.3}$$

where the ϵ_i^F are independent Normal random errors with mean zero and variance $1/\lambda^F$. Note that $\boldsymbol{u}$ is not an input in determining the field data. (We could have included $\boldsymbol{u}^*$ in the definition of y^R and b, but that would have simply been extra notational burden.) These assumptions may only be reasonable after suitable transformations of the data and, in any case, more complicated error structures can be easily accommodated. For example, the ϵ_i^F can have a correlated error structure; indeed, this will be seen to be the case in dealing with CRASH.

The assumption that ϵ^F has mean zero is formally the assumption that the field observations have no bias. If the field observations do have bias, the situation is quite problematic, in that presumably the field experiments were designed so as to eliminate bias, yet failed to completely do so. If bias does exist in the field observations, there is no purely data-based way to separate the field bias from the model bias; expert opinion would typically be needed to make any such separation. Estimates of bias that arise from our methodology could still be interpreted as the systematic difference between the computer model and field observations, but this is of little interest, in that prediction of reality (not possibly biased field data) is the primary goal. Note that it is quite common for 'existing field data' to itself be biased (see, e.g. Roache, 1998), and obtaining unbiased field data is perhaps the most crucial aspect of model validation. See Trucano *et al.* (2002) for extensive discussion.

Assuming computation of y^M is fast, Bayesian analysis now proceeds by specifying prior distributions for unknown elements of the model,

- the probability density $p(\boldsymbol{u})$ for $\boldsymbol{u}$, which we take to be that specified in the I/U map;
- a prior density $p(\lambda^F)$ for the precision (the inverse of the variance) of the field measurement error – see Bayarri *et al.* (2002) for description of the prior we use;
- a prior density for the bias function $b(\boldsymbol{x})$ (see Appendix D.1),

and utilizing Bayes theorem. (For full details on these priors see Bayarri *et al.* (2002)). Typically, however, y^M is a slow computer model and we will then need to also incorporate the model approximation from Section 4 into the Bayesian analysis, so that the model output y^M is then viewed as the Gaussian process discussed therein. (This will be necessary in both the SPOT WELD and CRASH test beds, since the corresponding computer models are too expensive to run directly within the Bayesian computation.)

5.2 Bayesian inferences

Section 5.3 discusses implementation of the Bayesian analysis. Here we focus on discussion of the possible outputs of the analysis. The basic output from the Bayesian analysis is the posterior probability distribution of all unknown quantities, given the models and the data (model-runs and field). The key feature of the Bayesian approach is that this distribution incorporates all uncertainties in the problem, including uncertainties as specified in the I/U map and measurement errors in the data. From this probability distribution, a variety of quantities of interest can be computed and analyses made.

5.2.1 Calibration/tuning

Using field data to bring the model closer to reality, *tuning*, is often confused with calibration, the process by which unknown model parameters are estimated from data. The distinction is that, in calibration, one tries to find the true – but unknown – physical value of a parameter, while in tuning one simply tries to find the best fitting value. Calibration and tuning parameters are mathematically the same and are therefore treated identically in the analysis, but conceptually there is a potentially significant difference. Tuning will tend to give a better model for prediction with inputs in the range of the field data, but may well give worse predictions outside this range. For this reason, it is not uncommon for modelers to limit the extent of tuning. This can be done, if desired, by simply restricting the allowed range of variation in the tuning parameter (or the spread in the prior distribution of the tuning parameter) in the I/U map.

One often hears that data used for calibration/tuning cannot simultaneously be used for model validation. However, Bayesian methodology does formally allow such simultaneous use of data. In part, this is because Bayesian analysis does not simply replace the parameter by some optimal 'tuned' parameter value $\hat{\boldsymbol{u}}$, but rather utilizes its entire posterior distribution, which reflects the uncertainty that exists in the value of the parameter.

SPOT WELD: The vector of controllable inputs is $\boldsymbol{x} = (C, L, G)$ and the tuning parameter is u. (This could also be viewed as a calibration parameter, since it corresponds to an unknown feature of the contact resistance which is, in essence, being estimated from the field data.) As mentioned above, the Bayesian analysis produces complete posterior distributions for the unknowns in the model. For instance, Figure 3 gives the posterior density of u. The optimal tuned value of u is the mean of this distribution, which is $\hat{u} = 3.96$. Note that there is considerable uncertainty in this value, and the Bayesian analysis will take this uncertainty into account in all assessments of variance and accuracy. Doing so also helps alleviate the type of over-tuning that can result if one were to simply pick and use the best-fitting parameter value. In this regard it is also interesting to note that the two main modes of the posterior correspond to tuning on the gauge=1mm and 2mm data separately; use of either data-set alone would likely have worsened the situation in regards to over-tuning.

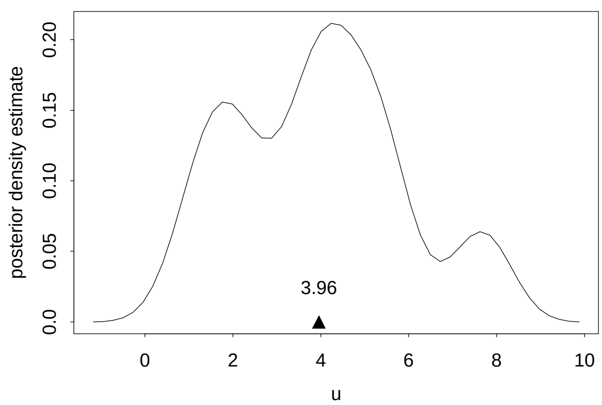

Figure 3: The posterior distribution of the tuning parameter u.

5.2.2 Predictions and bias estimates

Assume we want to predict the real process $y^R(\boldsymbol{x}, \boldsymbol{u}^*)$ at some (new) input $\boldsymbol{x}$. The preferred approach is to base this prediction on the output from a new model run at input $\boldsymbol{x}$ (**Case 1**). Sometimes this is not feasible (as when it is desired to produce a grid of predictions, as in Figure 4), and then predictions must be based on use of the model approximation (**Case 2**). We describe the analysis separately for these two cases.

Case 1. Predictions utilizing a new model run: When using a new model run (a new piece of data) for predicting the underlying process $y^R(\boldsymbol{x}, \boldsymbol{u}^*)$, we have at least two options. First, we can simply obtain an estimate $\hat{\boldsymbol{u}}$ and run the model at inputs $(\boldsymbol{x}, \hat{\boldsymbol{u}})$ to obtain a prediction; this will be called *model prediction*. The second and preferred approach is to use *bias-corrected prediction*, in which the model prediction is corrected by an estimate of the bias. The predictors, their bias, and their associated variances, are specified below (with full details given in Subsection 5.3).

Model prediction: The most commonly used predictor of $y^R(\boldsymbol{x}, \boldsymbol{u}^*)$ is $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$, for some estimate $\hat{\boldsymbol{u}}$ of the tuning parameter. (We recommend use of the posterior mean of $\boldsymbol{u}$, but the argument applies to any estimate). The accuracy of this estimate is determined by its variance, $V_{\hat{\boldsymbol{u}}}(\boldsymbol{x})$, which is one of the outputs of the Bayesian analysis.

It is often of separate interest to estimate the bias of the prediction $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$. This is given by

$$b_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) = y^R(\boldsymbol{x}) - y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})\,. \tag{5.4}$$

The variance of this estimated bias is also available from the Bayesian analysis.

SPOT WELD: For G=2, L=4.888 and C=29.44, and using the posterior mean $\hat{u} = 3.96$, the pure model prediction, resulting from running the computer model at these inputs, is $\hat{y}^M(4.888, 29.44, 2, 3.96) = 7.16$. The variance of this prediction is $V_{3.96}(4.888, 29.44, 2) = 0.628$, and the estimated bias of the prediction is $\hat{b}_{3.96}(4.888, 29.44, 2) = 0.342$.

Bias-corrected prediction: An important observation is that one can improve upon the pure model prediction $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$. Indeed, since an estimate of the bias is available, it is clear that

$$\hat{y}^R(\boldsymbol{x}) = y^M(\boldsymbol{x}, \hat{\boldsymbol{u}}) + \hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) \tag{5.5}$$

would be the optimal predictor of the actual process value $y^R(\boldsymbol{x})$. Furthermore, the variance of this improved prediction can be shown to be $(V_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) - [\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x})]^2)$, which can be significantly smaller than the variance, $V_{\hat{\boldsymbol{u}}}(\boldsymbol{x})$, of the pure model prediction.

SPOT WELD: For G=2, L=4.888 and C=29.44, and using the posterior mean $\hat{u} = 3.96$, the bias-corrected prediction is $\hat{y}^R(4.888, 29.44, 2) = 7.16 + 0.342 = 7.50$, with a variance of $0.628 - 0.342^2 = 0.512$. Bias-correction here has not resulted in a significantly reduced variance (compare with 0.628 for the pure model prediction), because the amount of bias was rather modest at this input value. We will see in Figure 6 that the bias can be significantly greater (as high as 1.0) at other input values.

There are several important subtleties in the above analysis. The first is that, in principle, a superior model prediction could be obtained by 'averaging' $y^M(\boldsymbol{x}, \boldsymbol{u})$, at the new input $\boldsymbol{x}$, over the posterior density of $\boldsymbol{u}$. This cannot be done, however, if the model is expensive to run. The recommended analysis in (5.5) achieves a compromise by utilizing the information from the new model run, $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$, but also averaging $y^M(\boldsymbol{x}, \boldsymbol{u})$ over other values of $\boldsymbol{u}$ through the fast model approximation. A related point is that the bias defined in (5.4) is different than that defined earlier in (5.1); this earlier bias was defined relative to the true (but unknown) value $\boldsymbol{u}^*$, rather than the estimated value $\hat{\boldsymbol{u}}$. The Bayesian analysis properly accounts for this definitional difference in the analysis.

Case 2. Approximate prediction, based solely on previous model runs: If it is not feasible to evaluate $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$ at the new input value $\boldsymbol{x}$ (for instance, if prediction is desired at many new inputs), one can still proceed with prediction of $y^R(\boldsymbol{x}, \boldsymbol{u}^*)$, using the model approximation $\hat{y}^M(\boldsymbol{x}, \boldsymbol{u})$. Indeed, since the model approximation is fast, 'averaging' the model approximation $\hat{y}^M(\boldsymbol{x}, \boldsymbol{u})$ over the posterior density of $\boldsymbol{u}$, is now feasible, which would lead to bias-corrected prediction. We also consider pure model prediction, which is here given by $\hat{y}^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$, and the corresponding estimated bias function, defined analogously to (5.4).

SPOT WELD: Using solely the previous model runs (and field data), and the model approximation $\hat{y}^M$, Figure 4 gives the pure model predictions $\hat{y}^M(L, C, G, \hat{u})$, the estimated bias functions $\hat{b}_{\hat{u}}(L, C, G)$, and the bias-corrected predictions $\hat{y}^M(L, C, G, \hat{u}) + \hat{b}_{\hat{u}}(L, C, G)$, as discussed above, for the spot weld model. For each gauge, these are presented as surfaces (as a function of L and C), with the height again being the predicted weld nugget diameter.

Note that the information obtained by running the computer model to obtain $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$ can considerably improve the prediction (and reduce the variance of the prediction), so the Case 1 analysis should be done, when possible. This is particularly true when 'local' predictions are being made, such as predicting the effect of changing from input $\boldsymbol{x}$ to input $\boldsymbol{x}'$, where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close. It will often then be the case that the pure model prediction of the difference, $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}}) - y^M(\boldsymbol{x}', \hat{\boldsymbol{u}})$, is close to the optimal bias-corrected prediction, $\hat{y}^R(\boldsymbol{x}') - \hat{y}^R(\boldsymbol{x})$, and has much smaller variance than if the same prediction were made based on the fast model approximation alone. The reason is that the bias, being smooth, essentially cancels when one computes the difference of model predictions at close values of the input. The bias would also cancel in the analysis based on the fast model approximation, but the comparatively significant uncertainty in the fast model approximation (as an estimate of the actual computer model) will remain. On the other hand, for simply predicting the process at a new input, the size of the bias correction will often be more significant than the uncertainty in the fast model approximation.

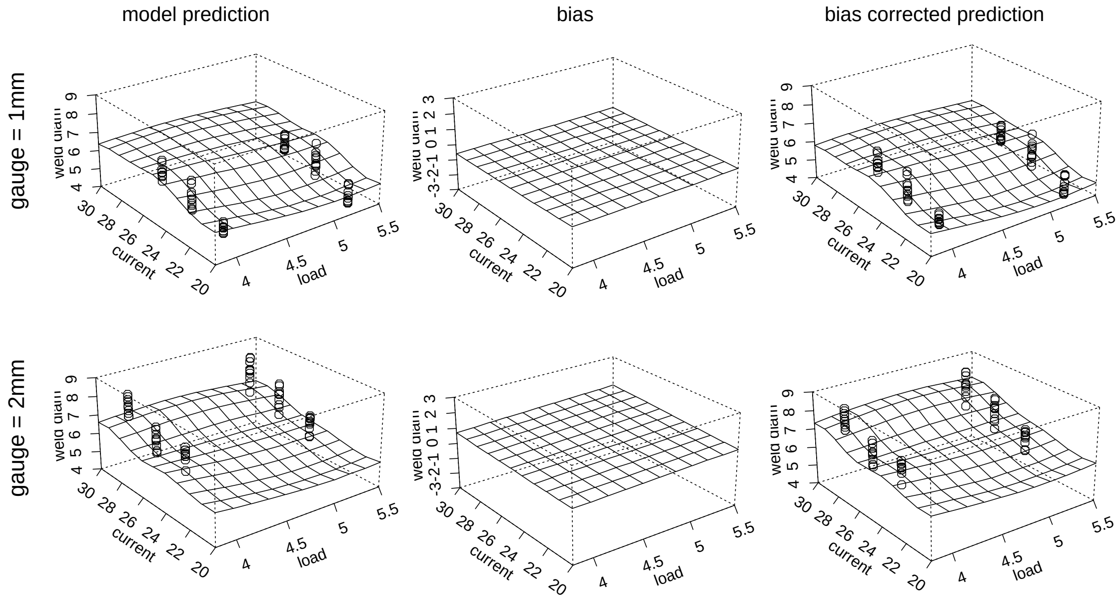

Figure 4: For a Case 2 analysis using only previous model runs, Left figures: the weld diameter predictions $\hat{y}^M(L, C, G, \hat{u})$ from the model approximation; Middle figures: the biases $\hat{b}_{\hat{u}}(L, C, G)$; Right figures: the bias-corrected predictions $\hat{y}^M(L, C, G, \hat{u}) + \hat{b}_{\hat{u}}(L, C, G)$. The circles represent the field data that were utilized in this analysis.

This helps to explain the often-heard comment by modelers that, even when the overall model predictions are not particularly accurate, predictions of process changes arising from small changes in inputs often seem to be quite accurate. It also partly explains why statistical analysis of the field data alone does not yield as useful predictions as analysis which incorporates the model information. For some global predictions, the statistical analysis alone might be nearly as good, but for exploring fine details of the process under study, the information from the computer model typically dominates. This discussion also underscores the fundamental importance of using a method of analysis that can accommodate, and properly weight, these different types of information.

5.2.3 Tolerance bounds

Predictive accuracy statements, such as "with probability 0.90, the prediction is within a specified tolerance τ of the true $y^R(\boldsymbol{x})$" are obtainable from the Bayesian analysis and provide a single simple measure of the effectiveness of the computer model. These can be obtained both with, or without, running the model at the (new) input $\boldsymbol{x}$ (Cases 1 and 2, respectively) and correcting or not for bias. Recall that bias correction results in smaller variances.

Case 1. Obtaining a new model run at input $\boldsymbol{x}$ improves prediction and results in tighter tolerance bounds.

SPOT WELD: For G=2, L=4.888 and C=29.44, and using the posterior mean $\hat{u} = 3.96$, the pure model prediction was $\hat{y}^M(4.888, 29.44, 2, 3.96) = 7.16$. The 5% and 95% percentiles of the posterior distribution give the 90% tolerance bounds, which, in this case, are $(6.02, 8.30)$. Similarly, the bias-corrected prediction is 7.50, with associated 90% tolerance bounds $(6.15, 8.30)$.

Case 2. If it is not feasible to obtain a new model run, or we have to give tolerance bounds for many new inputs (as when drawing a graph), then predictions, and tolerance bounds are based only on the previous model runs (and field data). Note that the resulting tolerance bounds will typically be wider.

SPOT WELD: Figure 5 provides 90% tolerance bands for two typical cases, one of low load ($L = 4.0$) and one of high load ($L = 5.3$), for each of the two gauges. In particular, the graphs present the pure model and bias-corrected predictions, and the error bands are 90th percentile bands for $y^M(\boldsymbol{x}, \hat{u})$ and $\hat{y}^R(\boldsymbol{x})$. Thus, for the top figures, there is a 90% probability for a specified current, load, and gauge that the real nugget size lies between the upper and lower dotted lines; the model approximation $\hat{y}^M(\boldsymbol{x}, \hat{u})$ at the optimal value of $\hat{u} = 3.96$ (see Figure 2) is indicated by the solid line. Note that the errors for the bias-corrected predictions (see the lower figures) are considerably smaller.

5.2.4 Uncertainty decomposition

Bayesian analysis not only allows for incorporation of all uncertainties into the accuracy statements, but also enables decomposition of the uncertainty into its component parts. For instance, in the overall model we use for SPOT WELD, there are three sources of error: uncertainty in the tuning parameter u, uncertainty in the bias function $b(\boldsymbol{x})$, and uncertainty in the residual error ϵ^F (which can arise from random error in the field data and/or randomness inherent in the actual process). One can separately assess, and report, the variation inherent in each of these sources, which can be important for determination of sensitivities and for improving the model. (Indeed, this aspect of the analysis can be considered to be a part of 'sensitivity analysis', as discussed in Section 10.1.)

Case 1. In this case, prediction is based on both the previous data (model runs and field data) as well as a new model run at the new input $\boldsymbol{x}$ at which prediction is desired.

SPOT WELD: For G=2, L=4.888 and C=29.44, and using the posterior mean $\hat{u} = 3.96$, the prediction was $\hat{y}^M(4.888, 29.44, 2, 3.96) = 7.16$. The three sources of error in this prediction and their relative importance can be judged by decomposing the 90% tolerance interval into intervals corresponding to each estimated quantity. The 90% interval for $y^M(4.888, 29.44, 2, u)$ (with u being considered as the unknown and random quantity) is $(6.50, 7.56)$; the interval for $b(4.888, 29.44, 2)$ is $(-1.00, 1.24)$; and the additional variability of the interval, induced by uncertainty in ϵ^F, is $(-0.71, 0.71)$.

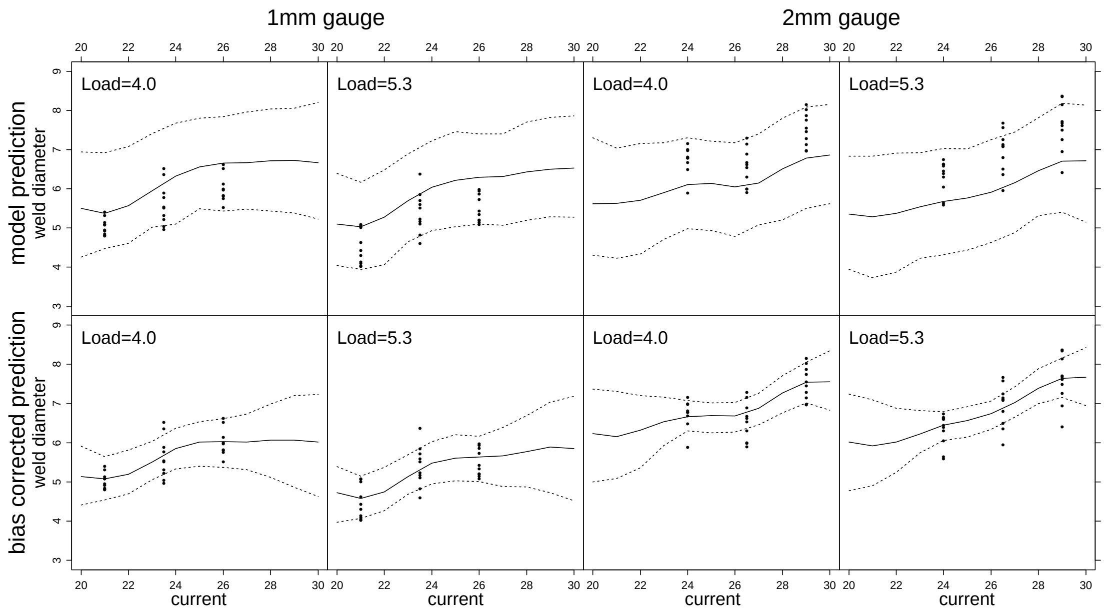

Figure 5: A posterior summary of the error associated with predictions in the Case 2 scenario (i.e., when only previous model runs are utilized). As a function of current for low ($L = 4.0$, left column) and high ($L = 5.3$, right column) loads, the top graphs show the model approximation $\hat{y}^M(\boldsymbol{x}, \hat{u})$ (solid line) and 90th percentile bands for the pure model predictions. The bottom row of the figure presents the same for the bias-corrected predictions $\hat{y}^R(\boldsymbol{x})$. The dots indicate the observed field data.

Case 2. If it is desired to graph the uncertainty due to each of the sources of error, as a function of the inputs, it will typically be necessary to use the analysis based on only previous model runs.

> **SPOT WELD**: The uncertainty associated with each of the unknown elements of the problem ($u, b(\boldsymbol{x})$, and ϵ^F) is presented in Figure 6. The top graphs present percentiles for $y^M(\boldsymbol{x}, u)$, and indicates the effect of the uncertainty in u. The second and third graphs indicate the percentiles for $b(L, C, G)$ and for ϵ^F, respectively. Interestingly, all three sources of uncertainty contribute comparable amounts (as measured by the width of the percentile bands) to the overall uncertainty. Clearly, ignoring any of these sources of uncertainty can lead to overconfidence in prediction. (Note that the constant lines corresponding to the residual error are a feature of the model used; it was assumed that the residual error does not depend on $\boldsymbol{x}$.)

5.3 Outline of the Bayesian methodology

We first consider the case in which the computer model is fast (so y^M is treated as a *known* function, and no model approximation is needed). We recall the modeling assumptions from Section 5.1:

$$
\begin{aligned}
y^F(\boldsymbol{x}) &= y^R(\boldsymbol{x}) + \epsilon^F \\
y^R(\boldsymbol{x}) &= y^M(\boldsymbol{x}, \boldsymbol{u}) + b(\boldsymbol{x})
\end{aligned}
$$

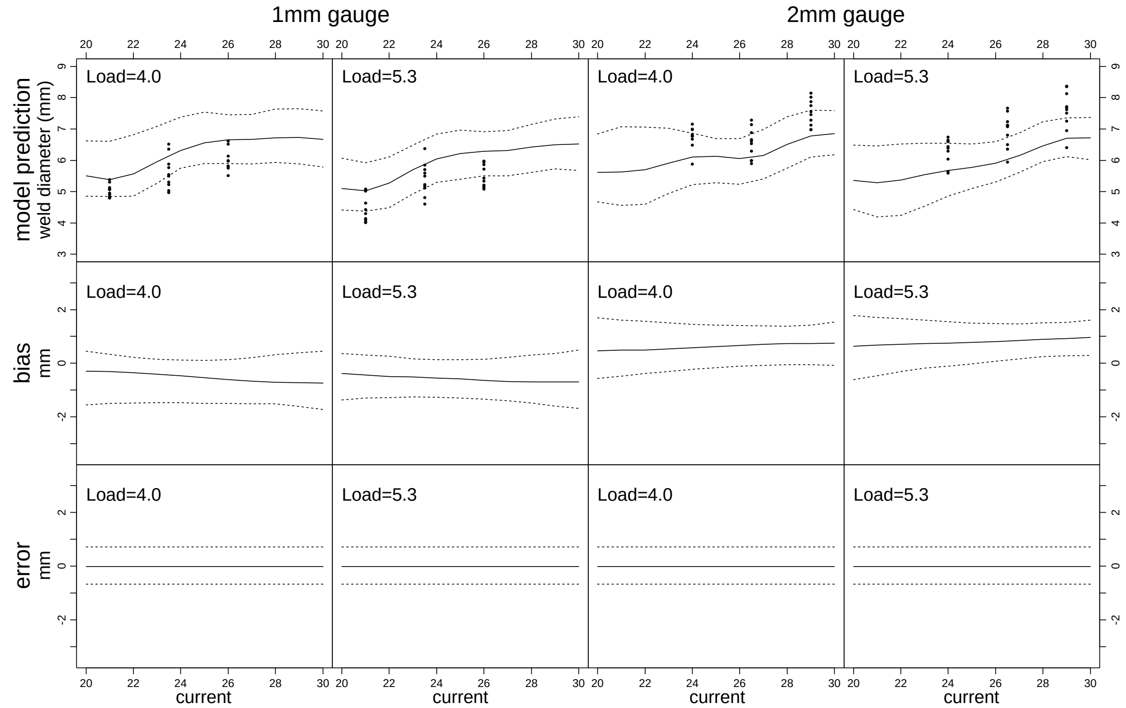

Figure 6: A posterior summary of the contributions of each source of uncertainty to the overall uncertainty of predictions under a Case 2 analysis. The top graphs show pointwise 90th percentile bands for $y^M(\boldsymbol{x}, u)$ (with u being considered as the unknown and random quantity) as a function of current for low ($L = 4.0$, left column) and high ($L = 5.3$, right column) loads. The middle row of graphs shows 90th percentile bands for $b(L, C, G)$. The bottom row shows 90th percentile bands for ϵ^F.

$$\epsilon^F \quad \sim \quad N(0, 1/\lambda^F)\,.$$

These produce a multivariate normal density for the collection of all field data, $\boldsymbol{y}^F$, which we shall denote by $f(\boldsymbol{y}^F \mid \boldsymbol{u}, \lambda^F, b)$. (Strictly, we should write $\boldsymbol{u}^*$ instead of $\boldsymbol{u}$ but, in the Bayesian approach, all unknowns are considered to be random and so we will drop the * superscript for notational simplicity.) The prior distribution of the unknown elements $\boldsymbol{u}, \lambda^F, b$ of the model will be denoted by $p(\boldsymbol{u}, \lambda^F, b)$ and is described in Appendix D.2. Bayes theorem then yields the posterior density of these unknowns, given the data $\boldsymbol{y}^F$, as

$$p(\boldsymbol{u}, \lambda^F, b \mid \boldsymbol{y}^F) \propto f(\boldsymbol{y}^F \mid \boldsymbol{u}, \lambda^F, b) p(\boldsymbol{u}, \lambda^F, b). \tag{5.6}$$

To actually compute the posterior density, one would need to determine the normalizing constant that makes the expression on the right hand side of (5.6) integrate to one. It will typically be necessary to deal with this posterior distribution by Markov chain Monte Carlo (MCMC) analysis (cf. Robert and Casella, 1999), however, and for this the normalizing constant is not needed. The result of the MCMC analysis will be, say, N draws from this posterior distribution of the unknowns $\boldsymbol{u}, \lambda^F$ and b. Call these samples $\boldsymbol{u}_i, \lambda_i^F$ and $b_i, i = 1, \ldots N$. From these samples, the posterior distribution of any quantities can be estimated. (Thus Figure 3 is just a smoothed histogram arising from the samples of the u_i generated from the SPOT WELD posterior distribution.)

The estimate of the unknown $\boldsymbol{u}$ is now simply $\hat{\boldsymbol{u}}$, the average of the $\boldsymbol{u}_i$. (This is the estimated posterior mean from the MCMC analysis.) Similarly, the estimated bias function is given by

$$\hat{b}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} b_i(\boldsymbol{x})$$

To predict the real process, $y^R(\boldsymbol{x})$, at any input $\boldsymbol{x}$ we have two options:

Model prediction: Here the prediction of the real process at input $\boldsymbol{x}$ is simply given by $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$. We recommend using the posterior mean as the estimate of $\boldsymbol{u}$, but analysis can be done for any estimate, such as the maximum likelihood estimate or any ad-hoc tuned estimate. The estimated bias of this prediction is given from the MCMC by

$$\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \left[y^M(\boldsymbol{x}, \boldsymbol{u}_i) + b_i(\boldsymbol{x}) \right] - y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})\,. \tag{5.7}$$

The variance, $V_{\hat{\boldsymbol{u}}}(\boldsymbol{x})$, associated with the model prediction $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$, is computed as

$$V_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) = [\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x})]^2 + \frac{1}{N} \sum_{i=1}^{N} \left[y^M(\boldsymbol{x}, \boldsymbol{u}_i) + b_i(\boldsymbol{x}) - \hat{y}^R(\boldsymbol{x}) \right]^2 , \tag{5.8}$$

where $\hat{y}^R(\boldsymbol{x})$ is the bias-corrected prediction, which is computed as in (5.9). Note that it is necessary to use the MCMC computational analysis to obtain the estimated bias and variance of the prediction, so that there is no gain in efficiency of computation in using the pure model predictor $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$.

The posterior probability that $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$ is within a specified tolerance τ of the true $y^R(\boldsymbol{x})$ is simply estimated by the fraction of the samples $(\boldsymbol{u}_i, b_i)$ for which $|y^M(\boldsymbol{x}, \hat{\boldsymbol{u}}) - [y^M(\boldsymbol{x}, \boldsymbol{u}_i) + b_i(\boldsymbol{x})]| < \tau$.

Bias-corrected prediction: It is optimal to use the bias-corrected predictor, given by the MCMC estimate of the posterior mean of the true process at $\boldsymbol{x}$, namely

$$\hat{y}^R(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \left[y^M(\boldsymbol{x}, \boldsymbol{u}_i) + b_i(\boldsymbol{x}) \right] . \tag{5.9}$$

An alternative expression for the estimate of the bias, $\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x})$, of the pure model prediction is

$$\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) = \hat{\boldsymbol{y}}^R(\boldsymbol{x}) - y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})\,, \tag{5.10}$$

thus making obvious its interpretation as the 'bias' of the predictor $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$. The bias, $\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x})$, of the pure model predictor is, in general, different from the prediction of the bias funtion $\hat{b}(\boldsymbol{x})$. The variance of the optimal predictor $\hat{y}^R(\boldsymbol{x})$ is simply computed as:

$$\frac{1}{N}\sum_{i=1}^{N}[y^M(\boldsymbol{x}, \boldsymbol{u}_i) + b_i(\boldsymbol{x}) - \hat{y}^R(\boldsymbol{x})]^2 = V_{\hat{\boldsymbol{u}}}(\boldsymbol{x}) - [\hat{b}_{\hat{\boldsymbol{u}}}(\boldsymbol{x})]^2 \,. \tag{5.11}$$

Note that for large bias, the reduction from the previous $V_{\hat{\boldsymbol{u}}}(\boldsymbol{x})$ can be substantial.

The posterior probability that $\hat{y}^R(\boldsymbol{x})$ is within a specified tolerance τ of the true $y^R(\boldsymbol{x})$ is simply estimated by the fraction of the samples $(\boldsymbol{u}_i, b_i)$ for which $|\,\hat{y}^R(\boldsymbol{x}) - [y^M(\boldsymbol{x}, \boldsymbol{u}_i) + b_i(\boldsymbol{x})]\,| < \tau$.

The difficulty with the above analysis is that it requires evaluation of $y^M(\boldsymbol{x}, \boldsymbol{u}_i)$ at each generated value of $\boldsymbol{u}_i$ (and also at each of the data inputs $\boldsymbol{x}_i$), which is infeasible when model runs are expensive. It is then necessary to use the Gaussian process approximation to y^M, described in Section 5, in order to carry out the computations. This (unavoidably) introduces additional uncertainty into the predictions. The analysis, however, is very similar to the one just presented; further details are given in Appendix D.2.

6 Feedback; Feed Forward (Step 6)

The analyses in Step 4 and Step 5 will contribute to the dynamic process of improving the model and updating the I/U map by identifying

- Model inputs whose uncertainties need to be reduced
- Needs (such as additional analyses and additional data) for closer examination of important regions or parts of the model
- Flaws that require changes in the model
- Revisions to the evaluation criteria.

In SPOT WELD, for instance, the posterior distribution of u (Figure 3) will now replace the uncertainty entry in the I/U map. Another aspect of feedback is use of the Step 4 and Step 5 analyses to further refine the validation process; e.g. to design additional validation experiments.

The feed-forward notion is to develop capability to predict the accuracy of new models that are related to models that have been studied, but for which no specific field data is available. This will be done through utilization of hierarchical Bayesian techniques introduced in Section 8. In CRASH, for example, the the data available for centerpole impacts can be augmented through hierarchical modeling.

7 Functional Data

Often, data arises in functional form. For instance, in CRASH, the data arises as functions of time (see Figure 15). In SPOT WELD, the model-run data was given as a function of the number of weld cycles, but we ended up using only the output at 8 cycles in the analysis (because of data-quality issues), so functional representation of the data was not needed.

We use $\boldsymbol{t}$ to denote the r-vector of arguments in functional data. In CRASH, $\boldsymbol{t}$ is time, a scalar quantity (i.e., $r = 1$). In the remainder of this section, we restrict attention to the scalar case, although the more general situation can be handled similarly. Also for simplicity of notation, we assume in this section that there are no $\boldsymbol{u}$ variables (true for CRASH, which will be the test bed application here).

We can now add t to the list of model inputs, and write the true process value at $(\boldsymbol{x}, t)$ as $y^R(\boldsymbol{x}, t)$, the model output at $(\boldsymbol{x}, t)$ as $y^M(\boldsymbol{x}, t)$, etc. As before, reality is linked to model output by

$$y^R(\boldsymbol{x}, t) = y^M(\boldsymbol{x}, t) + b(\boldsymbol{x}, t). \tag{7.1}$$

In practice we cannot work with complete function data, and it is necessary to discretize the data. One approach is simply to run separate analyses for each of a small set of t. This is not recommended, unless it is only a small set of t that are of interest. (For example, in SPOT WELD, interest primarily focused on evaluation of model predictions at $t = 8$ cycles.) A second approach would be to represent the functions that arise through a basis expansion (e.g., a polynomial expansion), taking only a finite number of terms of the expansion to represent the function. The coefficients of the terms in this expansion would then be additional input variables in the analysis. This approach might well be optimal in certain settings, but we turn instead to the most direct possibility.

The most direct approach is to lump t with $\boldsymbol{x}$ and model y^M and b with (single) Gaussian processes defined on the joint input space. Since we only consider discrete input values here, we further must pretend that we have only observed the function at a discrete set of points, $D^T = \{t_1, \ldots, t_T\}$. In essence, we are thus 'throwing away available data.' However, it is clear that, if T is chosen large enough and the points at which we record the function are chosen well (see Section 10.2), then the function values at these T points will very well represent the function. While this keeps the dimension of the Gaussian processes reasonable (only one new input is added), the number of observations becomes much larger; at each input value $\boldsymbol{x}$ in the data set, there are now T function evaluations that must be included as data. The total number of observations thus becomes $(m + l)T$, where m, and l are the number of $\boldsymbol{x}$ points in the model and field data, respectively. Computational complexity grows rapidly with the size of the data set so, at first sight, this approach is untenable.

Luckily, if we choose the same set, D^T, of t points for each of the $\boldsymbol{x}$ inputs in the model-run or field data, and we make a reasonable simplifying assumption as to the nature of the Gaussian process correlations involving t, a considerable simplification is effected that reduces the computational burden to something like the sum of the burdens for $(m + l)$ and T data points. This is discussed in Appendix E.1.

This analysis now proceeds as in Sections 4 and 5, and produces all the estimates and tolerance bounds discussed therein. For instance, the GASP approximation, $\hat{y}^M(\boldsymbol{x}, t)$, of the computer model,

$y^M(\boldsymbol{x}, t)$, can be computed, and pointwise error bands given. Thus, for fixed $\boldsymbol{x}$, the 80% pointwise posterior error bands are calculated by choosing $L_1(t), L_2(t)$ so that 80% of the MCMC samples lie in $I^M = [\hat{y}^M(\boldsymbol{x}, t) - L_1(t), \hat{y}^M(\boldsymbol{x}, t) + L_2(t)]$; the interpretation is then that "the probability is .80 that the computer model output (if run) would lie within the interval I^M."

Inference for a specific evaluation criterion can also be made (see Appendix E.2). In the examples of this section, the CRITV values arising from use of the GASP model approximation and bias-corrected prediction will be given.

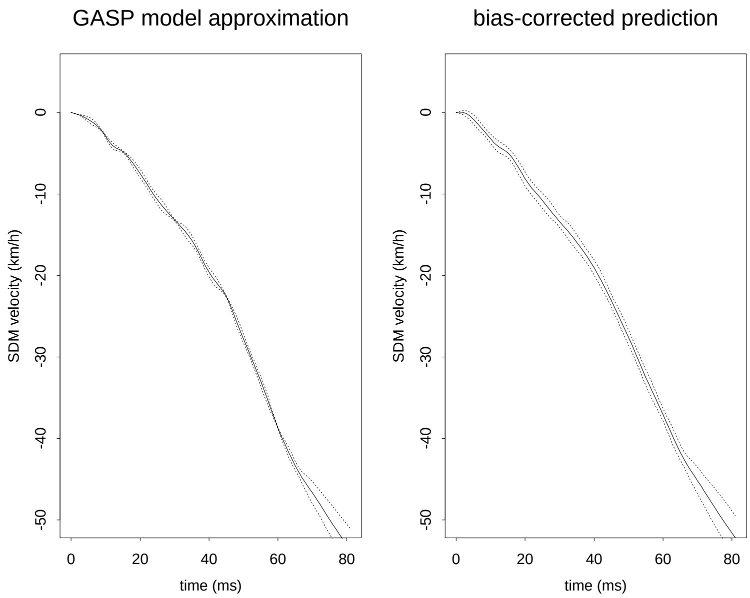

Figure 7: 80% posterior intervals of SDM velocity for y^M, arising from the GASP model approximation, and for predicting y^R, using bias-corrected prediction (in the Case 2 scenario in which only previous model runs are used in the analysis).

CRASH: We restrict attention to evaluation when the impact velocity is 56.3 km/h. The evaluation criterion, CRITV, is "SDM velocity calculated 30ms before SDM displacement, DISP, reaches 125mm." We take D^T to be the set of 19 time points $t = 1, 3, \ldots, 15, 17, 20, 25, \ldots, 65$ms. More points are chosen in the region $t < 20$ms since information from this region is more important in estimating CRITV. More t points could be used at some computational expense but this selection is adequate because SDM velocity is comparatively smooth and information at times greater than 65ms is irrelevant for the context at hand.

Since, for any v, the relative velocity is 0 at the time of impact $t = 0$, and since the slopes are roughly proportional to v, we assume that the prior mean of y^M has the form $\theta^L vt$. We carry out a MCMC analysis to approximate the posterior distribution of the parameters (see Bayarri *et al.* (2002).) Pointwise posterior intervals are computed. Figure 7 shows 80% pointwise posterior intervals for $y^M(56.3, \cdot)$ and $y^R(56.3, \cdot)$, corresponding to relative SDM velocity in the range $t < 80$ms. The greatest uncertainty in these distributions occurs for $t > 65$ms, the region where no data were observed. Note that the error bands about $\hat{y}^M(56.3, \cdot)$ only reflect the uncertainty in the GASP model approximation to the model, while the error bands about $\hat{y}^R(56.3, \cdot)$ incorporate all the uncertainty in prediction of reality.

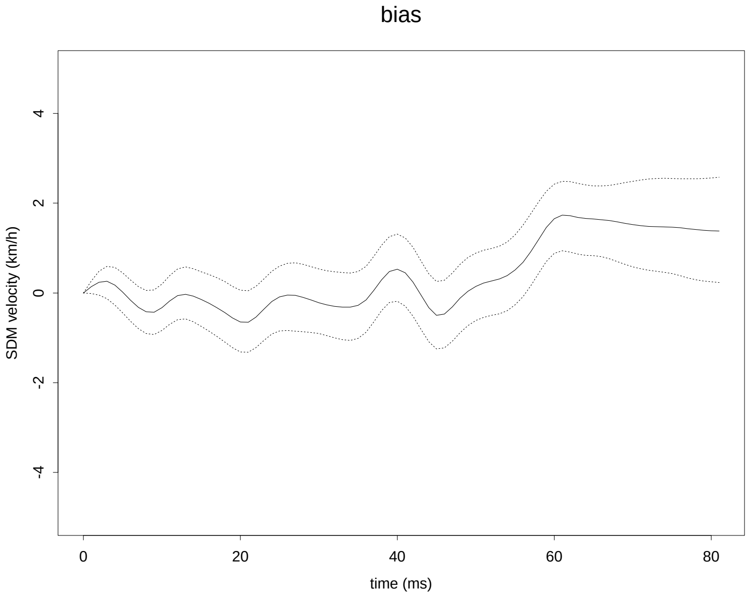

Figure 8: 80% posterior intervals for SDM velocity bias, at 56.3km/h impact velocity.

A simple graphical way of judging the validity of the model is to study tolerance bands for the estimated bias, as given in Figure 8. This shows a small predicted bias (ranging from 0 to 2km/h, with mild uncertainty), slowly increasing as a function of t.

Figure 9 illustrates the uncertainty in the criterion of interest, $CRITV$. The lower figure shows the uncertainty of prediction of real $CRITV$, using the optimal bias-corrected estimate (Case 2). This would be the result of primary interest to the engineer. The mean and standard deviation of this posterior distribution of $CRITV$ are -5.21 and 0.33, respectively (so that -5.21 would be the bias-corrected estimate).

To see how much of this uncertainty is due to the use of the GASP model approximation to the computer model, the upper figure presents the distribution of $CRITV$ that arises from the uncertainty in the GASP approximation. The mean of -5.11 is similar to that for the real prediction (suggesting that there is minimal bias), and the standard deviation is 0.13, indicating that most of the uncertainty in the real $CRITV$ prediction is due to sources other than the GASP model approximation. (The standard deviation 0.13 in part reflects the fact that only previous model runs were used - i.e., the model was not re-run at the desired input $v = 56.3$, and in part reflects the uncertainty that arises from discretizing t; note that this part of the uncertainty could be eliminated with more computational effort.)

Since bias is a function of the impact velocity v, the bias should be examined at different values of v. Figure 10 shows the bias for a 30km/h impact. The bias is clearly larger in the 20-59ms interval than it was for the 56.3km/h impact. The mean and standard deviation of $CRITV$ are now $(-6.53, 0.38)$ for y^M (GASP estimation of the model) and $(-6.56, 0.49)$ for y^R (bias-corrected prediction of real $CRITV$). It is interesting to note that the bias seen in

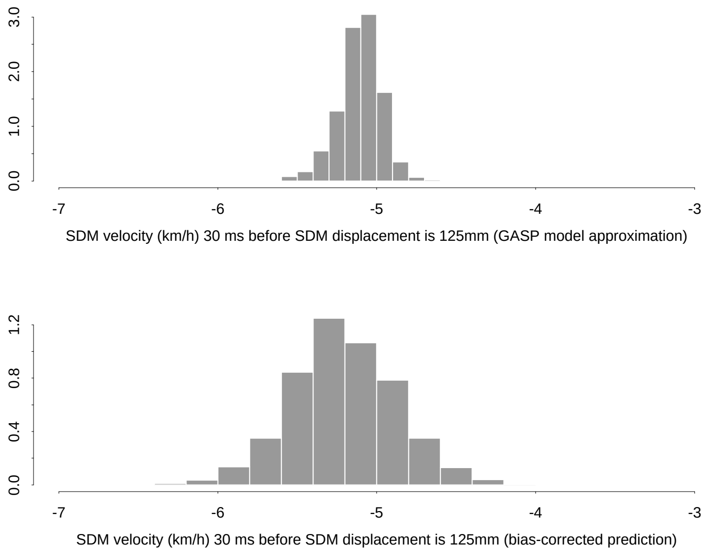

Figure 9: Posterior distributions for CRITV.

Figure 10 does not have a serious effect on the evaluation criterion (since both the GASP model approximation and the prediction of reality are very close). This serves as a potent reminder that validity of a computer model can depend strongly on the evaluation criterion of interest; while there is a clear indication that the computer model does have bias for SDM velocity at lower impact velocities and larger times, this bias disappears if only the CRITV criterion is of interest.

8 Extrapolation Past the Range of the Data

One of main motivations for using computer models is the hope that they can adequately predict reality in regions outside the range of the available data. We have advocated use of bias-correction, based on field data to improve (typically biased) computer model predictions. The difficulty is that the estimates of bias may not extrapolate well outside the range of the actual field data. When this is the case, the Bayesian methodology will tend to return very large tolerance bands; while one is at least not then making misleading claims of accuracy, the large bands may make assertion of predictive validity of the model impossible. (On a technical note, the best way to minimize the size of the tolerance bands in extrapolation is to choose the mean, $\boldsymbol{\Psi}(\cdot)\boldsymbol{\theta}^L$, of the model Gaussian process and the mean of the bias Gaussian process to be as accurate representations of the real process as possible.)

One 'solution' to this difficulty is to simply make the scientific judgement that the bias estimates do extrapolate. For instance, in CRASH, the entire analysis was performed for a fixed vehicle configuration. If, say, an element of the vehicle frame were increased in thickness by 5%, one might

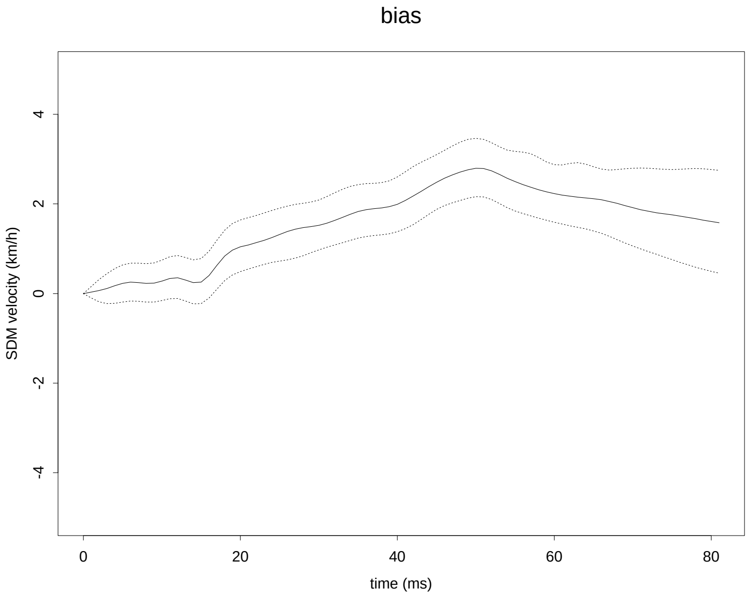

Figure 10: 80% posterior intervals for SDM velocity bias, at 30km/h impact velocity.

reasonably judge that the bias estimates would extend to this domain, even though no field data was obtained for varying thicknesses of the element. Of course, any such assumption should be reported, along with the conclusions of predictive accuracy of the model.

Bayesian methodology allows a weaker (and typically much more palatable) way of extrapolating past the range of the data. The idea is to model the new scenario as being related to that (or those) for which data is available, but not to insist that the situations are identical in terms of bias and predictive accuracy. There are many variants on this theme; here we consider one that applies to the CRASH model and is typically called *hierarchical modeling*.

Hierarchical modeling applies most directly to scenarios in which there are K different function outputs, each coming from different inputs to a computer model (or even from different computer models). Each of these functions can be modeled as was done in Section 7, through Gaussian process priors. We will be particularly concerned with settings where the Gaussian processes for y^M and b can be assumed to share common features, typically where the parameters governing the priors are drawn from a common distribution. This induces connections among the individual models and enables us to combine information from the separate models, sharpen analyses and reduce uncertainties. Clearly, disparate computer models are unlikely to be usefully treated this way but, for CRASH, such a hierarchical approach will be seen to be useful.

Implementation of these ideas will depend heavily on what data, both computer and field, are available as well as the legitimacy of the assumptions imposed. Here we informally state and comment on these assumptions for the simplest structure we will impose. (Full details can be found in Appendix F.)

Assumption 1. The smoothness of the model approximation processes are identical across the K models being considered. This is a very reasonable assumption in the contexts for which hierarchical modeling would typically be employed.

Assumption 2. The variances of the model approximation processes are equal, across the various cases. Similarly, we assume that the variances, $1/\lambda^F$, of the field data for all K cases are equal. Again, this is typically reasonable.

Assumption 3. The relation among the means of the Gaussian processes for the K computer models is quantified by assuming a common prior distribution for the $\boldsymbol{\theta}_i^L$ as specified in Appendix F. This prior distribution will allow the $\boldsymbol{\theta}_i^L$ to vary considerably between the K situations, but still ensures that information is appropriately pooled in their estimation.

Assumption 4. The biases for the K situations are assumed to be related in a fashion described by a parameter q, whose value must be specified. This parameter describes the believed degree of similarity in the biases for the K different computer models; indeed, $1 + q$ can be interpreted as an upper bound on the believed ratio of the standard deviations of the biases, or, stated another way, the proportional variation in the bias is q. (See Appendix F for details.) Specifying $q = 0.1$ is stating that the biases are expected to vary by about 10% among the various cases being considered.

Note that specification of q is a judgement as to the *comparative* accuracy of the K different computer models, as opposed to their *absolute* accuracy (which need not be specified). The reason we require specification of this parameter by the engineer/scientist is that there is typically very little information about this parameter in the data (unless K is large). Note that specifying q to be zero could be reasonable, if one is unsure as to the accuracy of the computer models but is quite sure that the accuracies are the same across the various K.

CRASH: The analysis performed earlier was on data and model for rigid barrier, straight frontal impact. By use of hierarchical modeling we can simultaneously treat rigid barrier, left angle and right angle impacts as well as center pole impact. The analyses and predictions are made for a 56.3km/h impact (this is at the high end of the data). For illustration, we choose $q = 0.1$.

Figure 11 shows the differing posterior predicted SDM velocity curves and pointwise uncertainty bands for each of the four barrier types. The straight frontal and left angle posterior intervals in Figure 11 are tight because there are data close to 56.3km/h for these barrier types (so the analysis is effectively like a Case 1 analysis – i.e., based on a new model run at the desired input – than a Case 2 analysis). In contrast, the intervals are not tight for the other barriers because data near 56.3km/h are lacking. This thus reinforces the value of making a model run at a new desired input. Figure 7 and the straight frontal pictures in Figure 11 are very similar.

Figure 12 gives the estimates of the four bias functions and the associated pointwise uncertainties. Because of the large uncertainties in the bias estimates, the only case in which the bias seems clearly different from zero is for left angle impacts, after 43ms. (While we cannot clearly assert that there is bias in the other cases, the tolerance bounds for predictions will be quite large, reflecting the uncertainty in the bias estimates.)

We again consider the criterion $CRITV$ = SDM velocity 30ms before SDM displacement is 125mm. Table 4 presents the mean and standard deviations of $CRITV$ for each barrier type for the GASP model approximation and for the bias-adjusted prediction of $CRITV$. The corresponding posterior distributions for $CRITV$ are available, but omitted here.

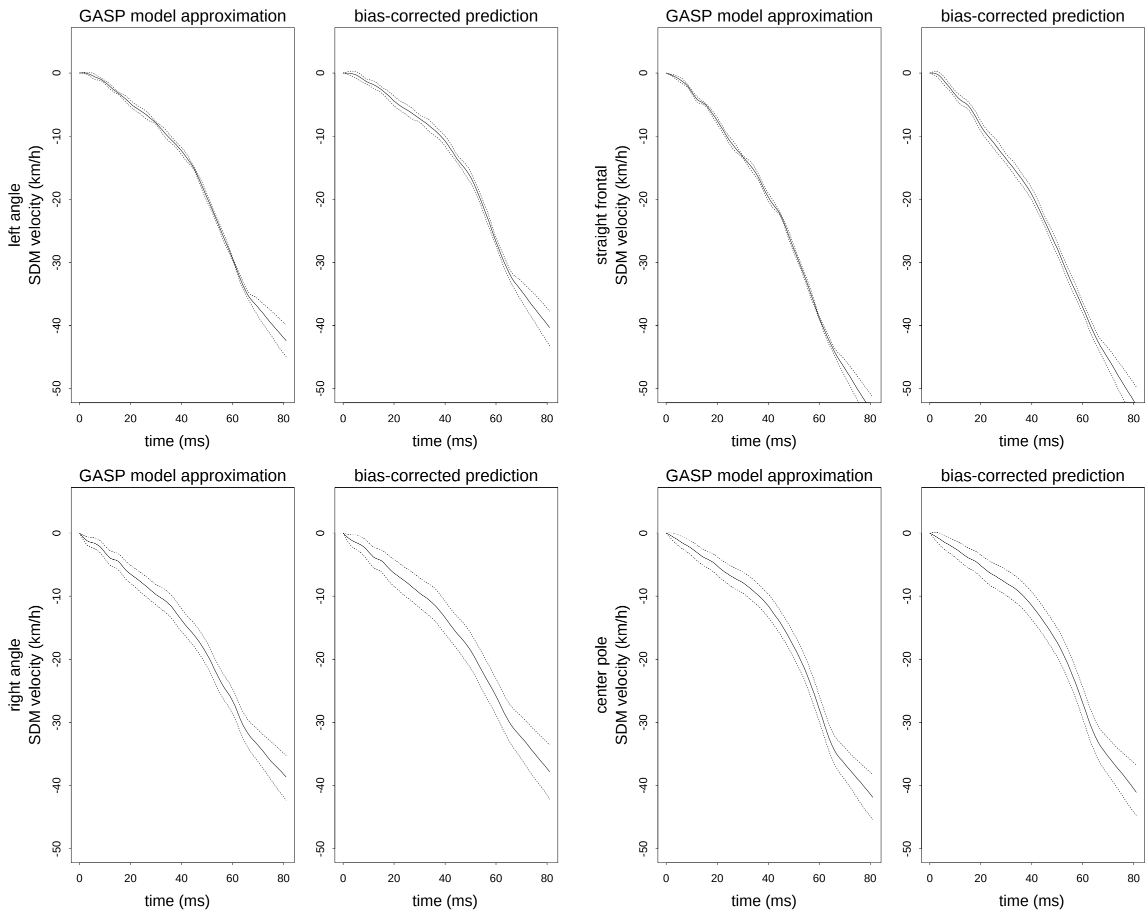

Figure 11: Pointwise 80% posterior intervals for 4 barrier types, based on the hierarchical model, in the Case 2 scenario in which only previous model runs are utilized.

<ins>*Angle as an additional input*</ins>: If we only consider the three rigid barrier impacts (frontal, right angle and left angle) and ignore the center pole impact and data, we could proceed without use of hierarchical modeling by incorporating the angle of impact, $\boldsymbol{x}_A$, as an input to the model. The smoothness assumption required for the Gaussian process analysis is plausible: it is reasonable to assume that small changes in the angle will result in small changes in the velocity-time curve so that y^M is a smooth function of $\boldsymbol{x}_A$.

Combining the data from left angle ($\boldsymbol{x}_A = 0.0$), right angle ($\boldsymbol{x}_A = 1.0$), and straight frontal ($\boldsymbol{x}_A = 0.5$) barrier impacts led to computations that were considerably more expensive than in the hierarchical model because we now need to invert 26×26 matrices instead of the smaller

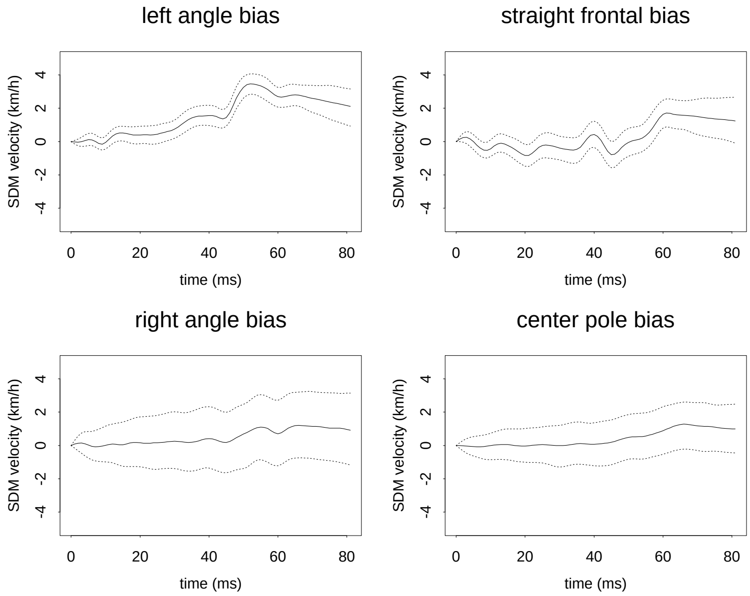

Figure 12: Pointwise 80% posterior intervals for bias, based on the hierarchical model, in the Case 2 scenario where only previous model runs are utilized.

	Hierarchical model		Using frontal data only	
Barrier type	$CRITV$ for y^M	$CRITV$ for y^R	$CRITV$ for y^M	$CRITV$ for y^R
left angle	-6.08 (0.34)	-6.34 (0.49)		
straight frontal	-5.13 (0.13)	-5.22 (0.30)	-5.11 (0.13)	-5.21 (0.33)
right angle	-6.89 (0.65)	-6.80 (0.96)		
center pole	-6.55 (0.74)	-6.54 (0.91)		

Table 4: Posterior mean and standard deviation of $CRITV$, arising from the GASP model approximation estimate of y^M, and arising as the bias-corrected prediction ($\hat{y}^R$) of real $CRITV$ (in the Case 2 scenario where only previous model runs are utilized).

matrices encountered in dealing with the individual barrier types.

Comparison of the results in Tables 5 and 4 generally shows close agreement between using angle as input and the hierarchical model. There are differences associated with the right angle CRITV, reflecting the paucity of data for that angle. (The hierarchical model makes weaker assumptions about the relationship between the various cases than does incorporation of angle as an input variable, and hence is more affected by the shortage of data for right angle.)

The tolerances in Table 6 refer to relative velocities, but ranges for other evaluation criteria can be easily computed. For example, the tolerance range for “time at which SDM displacement is 125 mm” corresponding to the frontal analysis is 15.79 ± 1.03ms.

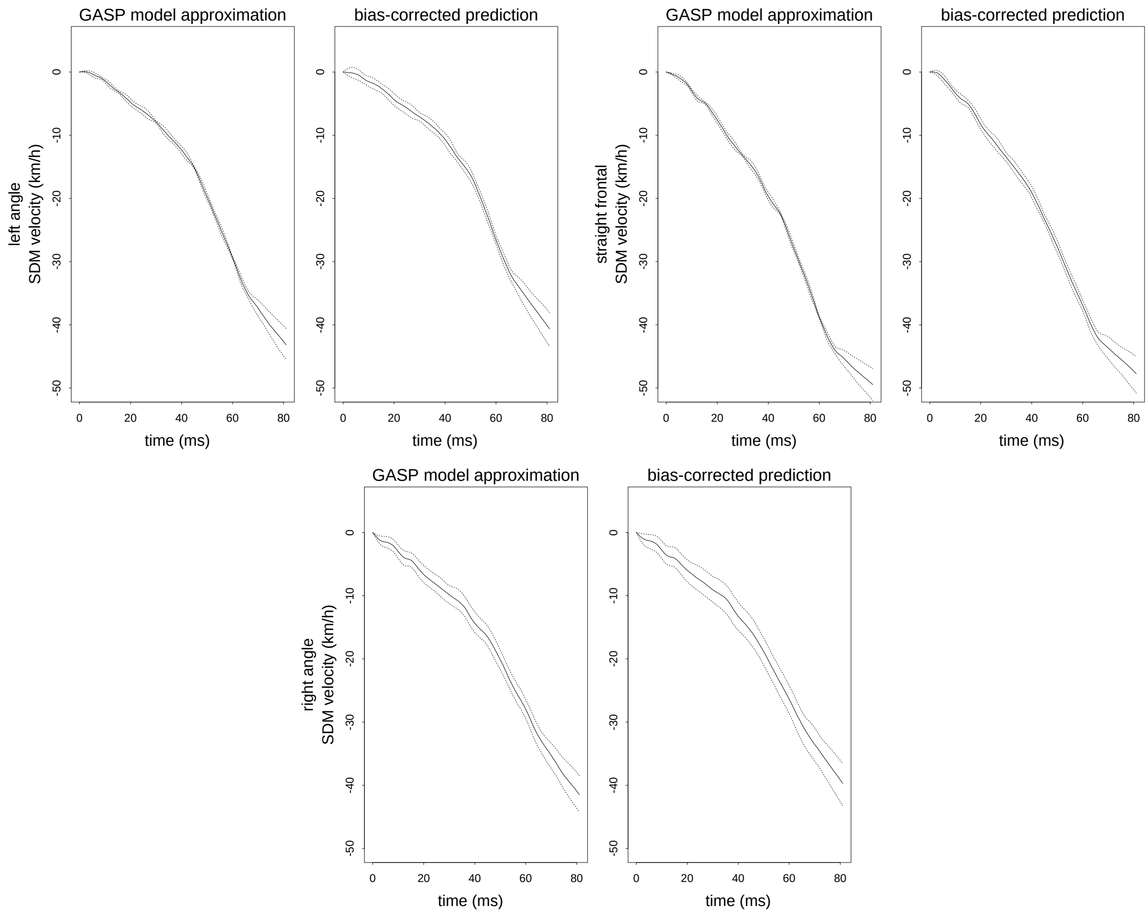

Figure 13: 80% pointwise posterior intervals for SDM velocity, using angle as additional input, in the Case 2 scenario of using only previous model runs.

9 Merging Predictive and Physical Approaches to Validation

9.1 The probability that the computer model is correct

In the introduction, the two philosophies towards validation of a computer model were discussed. We have focused on the predictive approach, that of determining the accuracy of the predictions of the computer model, assuming that some bias exists. In the physical school, a modeler that has carefully constructed and exhaustively tested each component of a model (including component interfaces) might argue that the model is correct by construction, i.e., that there can be no bias. Such claims should be supported by at least some confirming data, but how much confirming data is needed?

This same question arises in the pure view of the scientific process. A scientist proposes a new theory, which makes precise predictions of a process, say $y^M(\boldsymbol{x})\pm 0.0001$ at input $\boldsymbol{x}$. Other scientists then try to devise an experiment that can test this theory, i.e., an experiment that, for some input $\boldsymbol{x}^*$, will provide a field observation $y^F(\boldsymbol{x}^*)$ that is within, say, 0.00001 of the true process value

Barrier type	$CRITV$ for y^M	$CRITV$ for y^R
left angle	-6.04 (0.36)	-6.29 (0.53)
straight frontal	-5.12 (0.15)	-5.24 (0.30)
right angle	-6.88 (0.64)	-6.69 (0.90)

Table 5: Posterior mean and standard deviation of $CRITV$, arising from the GASP model approximation estimate of y^M, and arising as the bias-corrected prediction ($\hat{y}^R$) of real $CRITV$ (in the Case 2 scenario where only previous model runs are utilized).

Barrier type	Hierarchical model	Angle input $\boldsymbol{x}_A$	Frontal data only
left angle	-6.08 $\pm$ 0.70	-6.04 $\pm$ 0.76	
straight frontal	-5.13 $\pm$ 0.40	-5.12 $\pm$ 0.41	-5.11 $\pm$ 0.43
right angle	-6.89 $\pm$ 1.27	-6.88 $\pm$ 1.17	
center pole	-6.55 $\pm$ 1.18		

Table 6: Posterior mean and 80% tolerance range for $CRITV$, arising from the GASP model approximation estimate of y^M.

$y^R(\boldsymbol{x}^*)$. If the experiment is conducted and $y^F(\boldsymbol{x}^*)$ is within ± 0.0001 of $y^M(\boldsymbol{x}^*)$, then the scientific theory is viewed as being validated. The key to this scientific process is that, if the scientist makes even one very precise prediction, and the prediction turns out to be true, then that would seem to be considerable evidence in favor of the hypothesis. If the prediction of the scientist were not very precise, then a single observation could disprove, but not really confirm the theory.

The natural language in which to discuss and implement these ideas is the Bayesian language. The proposed new theory (or proposed computer model) is $\mathcal{M}_0$, and one asks the question (after seeing one or more field observations) "What is the probability, given the data, that $\mathcal{M}_0$ is correct?" This question can be asked – and answered – through the Bayesian approach, and the result behaves as the scientific intuition from the previous paragraph would suggest. In particular, this probability can be quite high in the scientific context of a precise theory, after even one confirming precise field observation, while it will not be high in the case of an imprecise theory (or an imprecise field observation).

This is a problem of hypothesis testing or model selection. In the classical approach to hypothesis testing, one can essentially show that $\mathcal{M}_0$ is false, if the data so suggest, but it is much harder to show that $\mathcal{M}_0$ is true. The Bayesian approach does allow direct answer of this primary question of interest.

Several ingredients are needed to implement the Bayesian approach.

1. A prior probability, π_0, that $\mathcal{M}_0$ is true. This can, of course, vary from one individual to another. The modeler might feel π_0 to be quite high, while a skeptic might judge it to be low. Often, however, the default choice $\pi_0 = 1/2$ is made, in order to 'see what the data has to say.'

2. An alternative model $\mathcal{M}_1$.

3. Suitable prior probability distributions on unknown parameters of $\mathcal{M}_0$ and $\mathcal{M}_1$.

In the context of evaluation of computer models, we have already constructed these needed ingredients. In particular,

- The prior distribution on the parameters of the computer model, $\mathcal{M}_0$, is that provided by the I/U map.
- The alternative model, $\mathcal{M}_1$, is the model we constructed in Sections 5.1 and 7 , which includes the bias term $b(\boldsymbol{x})$. Full details can be found in Bayarri *et al.* (2002).
- The prior distribution on the parameters of the alternative model (including the unknown bias) are as constructed for the predictive validation.

The result of the analysis is called the *posterior probability that $\mathcal{M}_0$ is true*, and will be denoted by $P(\mathcal{M}_0 \mid \boldsymbol{y})$, where here we generically let $\boldsymbol{y}$ refer to all the data.

> **CRASH**: Analysis in this test-bed resulted in a posterior probability of near 0 that the computer model is true (assuming an initial prior probability of $\pi_0 = 1/2$. This was actually also apparent from earlier graphs of the estimated bias, and illustrates an important point: if a computer model has statistically significant bias over any part of the domain under study, the model will have essentially zero posterior probability of being correct. This, of course, is as it should be, but does point out the reason that looking at predictive accuracy of the model (which can vary over the input domain) is greatly superior to simply asking yes/no questions.

Before discussing the details of the analysis, another feature of the Bayesian approach deserves highlighting, namely that the conclusions regarding accuracy of predictions will now be a weighted average of the accuracy statements arising from $\mathcal{M}_0$ and $\mathcal{M}_1$. For instance, if it is desired to know the probability that $y^R(\boldsymbol{x}^*)$, at specified input $\boldsymbol{x}^*$, lies within the interval $(8, 10)$, the answer would be

$$P(\mathcal{M}_0 \mid \boldsymbol{y})\, P(8 < y^R(\boldsymbol{x}^*) < 10 \mid \mathcal{M}_0) + (1 - P(\mathcal{M}_0 \mid \boldsymbol{y}))\, P(8 < y^R(\boldsymbol{x}^*) < 10 \mid \mathcal{M}_1)\,,$$

with the $P(8 < y^R(\boldsymbol{x}^*) < 10 \mid \mathcal{M}_i)$ being computable from our previous analyses. In this expression, we see a complete merging of the physical and predictive approaches to model validation. The physical approach would produce the accuracy statement $P(8 < y^R(\boldsymbol{x}^*) < 10 \mid \mathcal{M}_0)$, while the predictive approach would produce the accuracy statement $P(8 < y^R(\boldsymbol{x}^*) < 10 \mid \mathcal{M}_1)$. The overall correct answer is their weighted average, with the weights being the posterior probability that each of the models is true.

9.2 Implementation

In carrying out the Bayesian computation of $P(\mathcal{M}_0 \mid \boldsymbol{y})$ for a slow computer model, the approximation introduced in Section 4 will be required. In this case, $\mathcal{M}_0$ is like the overall model $\mathcal{M}_1$, but with the bias function $b(\cdot) = 0$.

Let $\boldsymbol{\phi}_i$ be the full parameter vector for model $\mathcal{M}_i$, $i = 0, 1$ (including all parameters of the mean functions and the Gaussian processes involved). In addition, for model $\mathcal{M}_i$, $i = 0, 1$, denote by $f_i(\boldsymbol{y} \mid \boldsymbol{\phi}_i)$, $p_i(\boldsymbol{\phi}_i)$ and $p_i(\boldsymbol{\phi}_i \mid \boldsymbol{y})$ the likelihood function of the full data vector $\boldsymbol{y}$ (both computer model and field data), the prior density and the posterior density for the parameter vector, respectively. The form of the likelihood function and the approaches for prior specification and posterior inference, using MCMC methods, for model $\mathcal{M}_0$ are similar to the corresponding ones for model $\mathcal{M}_1$, described earlier and detailed in Bayarri *et al.* (2002).

Letting $\pi_1 = 1 - \pi_0$ denote the prior probability of $\mathcal{M}_1$, Bayes theorem gives that the posterior probability of $\mathcal{M}_0$ is given by

$$P(\mathcal{M}_0 \mid \boldsymbol{y}) = \frac{\pi_0 m_0(\boldsymbol{y})}{\pi_0 m_0(\boldsymbol{y}) + \pi_1 m_1(\boldsymbol{y})}, \tag{9.1}$$

where

$$m_i(\boldsymbol{y}) = \int f_i(\boldsymbol{y} \mid \boldsymbol{\phi}_i) p_i(\boldsymbol{\phi}_i) d\boldsymbol{\phi}_i \tag{9.2}$$

is the marginal likelihood for model $\mathcal{M}_i$, $i = 0, 1$.

Although we are typically able to integrate over a part of the parameter vector $\boldsymbol{\phi}_i$, analytic expressions for the integrals in (9.2) are not available. However, numerical evaluation, based on Monte Carlo estimates, is feasible using the posterior samples from $p_i(\boldsymbol{\phi}_i \mid \boldsymbol{y})$, $i = 0, 1$, and the existing approaches to this problem (see, e.g., Chib and Jeliazkov, 2001, and references therein).

Note that use of improper priors is typically not possible when interest lies in computation of marginal likelihoods of models. However, given specific structure of the models, improper priors for some of the parameters can be employed. (See Berger, De Oliveira and Sansó, 2001, for discussion of this issue and additional references.) In our setting, it is, in general, possible to use the improper prior, given in Bayarri *et al.* (2002), for $\boldsymbol{\theta}^L$, the vector of parameters associated with the mean function of $y^M(\cdot)$.

9.3 Merging numerical and statistical modeling

When there is a significant amount of field data available, a statistician might consider simply modeling the field data by statistical methods, forgoing utilization of the computer model of the process. It is natural to ask if such pure statistical modeling can be merged with the computer model to produce improved predictions. The answer is yes, and involves incorporation of the statistical modeling into both the mean function of the bias and error structure of the data. We do not consider this further here, as the focus of the paper is on validation of the computer model.

10 Additional Issues

10.1 Computer model simplification

Sensitivity analysis

Sensitivity analyses focus on ascertaining which inputs most strongly affect outputs, a key tool in refining the I/U map. There are 'local' and 'global' approaches to sensitivity analysis. The local approach is based on derivatives of model outputs with respect to model inputs. This can sometimes be done by automatic differentiation (actual line-by-line differentiation within the computer code), but is almost always a difficult process. The global approach is based on statistical analysis comparing the output and input distributions. There are many versions of such global analyses, but the most commonly used are variants of 'analysis of variance' (ANOVA) decomposition, to assess which input variables have the greatest effect on the variance of the output distributions. Models based on the most important variables can then be studied, with the less important variables fixed (at, say, their prior means). Methods of model simplification based on principal component analysis (or POD in the applied mathematics literature) also fall in this domain. (See Saltelli et al., 2000, for discussion and examples of output and sensitivity analyses.)

Again, model approximations are needed for expensive codes and these, in turn, require special methods of global sensitivity analysis. Elaboration of these methods will not be treated here, although there is code by W. Welch which provides an ANOVA decomposition for the model approximation mentioned in Step 4 based on maximum likelihood estimation of the parameters.

10.2 Utilization of transformations

- Often a transformation of the data is helpful in the statistical modeling. For instance, in CRASH, it was helpful to consider relative velocity (subtracting the initial impact velocity from all data) instead of raw velocity.

- One can perform a change in time scale to deal with nonstationarity in time. For instance, define a new time scale by taking a functional output $y^M(x_0, t)$, at a typical x_0, and defining

$$t^* = \int_0^t \left| \frac{\partial}{\partial v} y^M(x_0, v) \right| dv.$$

  A similar rescaling could be done for any continuous variable, if desired.

- Transformations of the Gaussian process $y(\boldsymbol{z})$ can be made, such as $y^*(\boldsymbol{z}) = g(\boldsymbol{z}) y(\boldsymbol{z})$. This is a new Gaussian process with mean multiplied by $g(\cdot)$ and a covariance function that is often of suitable form.

  CRASH: The process is 'tied down' at time 0, and a smaller variance is expected there. Choosing, say, $g(t) = t/(10 + t)$ will result in a Gaussian process with near zero variance initially, yet a process that behaves essentially like the previous stationary process once one is significantly far from 0.

10.3 Modularization

The basic idea is to first do the Bayesian analysis of all the model data, ignoring the contribution of the field data in estimating GASP model approximation parameters (including the $\boldsymbol{\theta}^L$ and those relating to the functional parameters $\boldsymbol{t}$). Then, treating the model parameters (other than tuning parameters) as specified by the resulting posterior distribution (or, possibly, by their maximum likelihood estimates), one incorporates the field data by a separate Bayesian analysis. The motivation and advantages of the modular approach are as follows.

1. This is a Bayesian version of 'partial likelihoods': if $f(data|\theta,\nu) = f(data|\theta)g(data|\nu,\theta)$, partial likelihood uses only f to estimate θ which then gets plugged into g for further inference about ν.

2. Field data can affect the GASP model approximation parameters in undesirable ways, resulting in pushing $\boldsymbol{u}$ to 'bad' regions of its space. The modular approach can prevent this from happening.

3. This easily generalizes to systems with several model components, M_i, each of which has separate model-run data. Dealing first with the separate model-run data, in setting up the GASP model approximations, and incorporating the field data only at the tuning/validation stage, makes for an easier-to-understand and computationally more efficient process.

Details concerning the modular approach can be found in Bayarri *et al.* (2002).

10.4 Multivariate output functions

There are a variety of possible ways to extend the analysis to multivariate output, important for situations such as the following, but we do not consider them here.

> **CRASH**: The evaluation criterion "velocity at the center of the radiator, RDC, 30ms before SDM displacement reaches 125mm" involves a combination of two sensors, one located at the radiator center and the other under the driver's seat. This evaluation criterion thus requires an analysis of bivariate model output.

10.5 Updating

The model approximation is exact only at the observed model-run data points. Sometimes the values of the model output are also constrained at other points, and it can be important to include such constraints in the analysis. This is best done 'after the fact' by conditioning the unconstrained answer on the known constraints. (Trying to incorporate the constraints at the beginning often fatally disrupts the computations.)

> **CRASH**: A problem arises if we wish to predict the velocity-time curve when the initial velocity v_0 is between two data curves: at time $t = 0$ we know that the relative velocity is 0, but the

Gaussian process approximation only assumes that this is true in mean. If one tried to add the constraint that all initial relative velocities were zero, the Kronecker product computational simplification no longer applies, resulting in an impractical MCMC algorithm. A compromise is to introduce the information that the relative velocity is initially 0 only at the prediction stage, which is straightforward to do.

Another crucial instance of the conditioning idea is when an additional model data point, $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$, becomes available. Indeed, this is how the actual model can be utilized in future predictions, according to the recommended 'Case 1' approach to validation. The difficulty is that one can then rarely go back and re-run the entire MCMC computation with this new data point. The solution is simply to condition the existing posterior on this additional data point, using it in the Kalman filter part of the analysis, but not to obtain the posterior for tuning parameters or parameters in the Gaussian processes. Details can be found in Bayarri *et al.* (2002).

10.6 Accounting for numerical instability and stochastic inputs

Sometimes numerical instability in the computer model can be modeled by adding an additional random noise term to the GASP model approximation (often called a 'nugget' in the Gaussian process literature). Likewise, stochastic inputs can often be handled by simply enriching the stochastic structure of the GASP model approximation. We do not consider these generalizations here.

REFERENCES

Bayarri, M. J., Berger, J. O., Higdon, D., Kennedy, M. C., Kottas, A.,Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2002). A Framework for Validation of Computer Models. Technical Report # 128, National Institute of Statistical Sciences.

Berger, J.O, De Oliveira, V. and Sansó, B. (2001). Objective Bayesian Analysis of Spatially Correlated Data. *Journal of the American Statistical Association*, **396**, 1361–1374.

Cafeo, J.A. and Cavendish, J.C. (2001). A Framework For Verification And Validation Of Computer Models and Simulations. Internal General Motors document; to be published.

Chib, S. and Jeliazkov, I. (2001). Marginal Likelihood From the Metropolis-Hastings Output. *Journal of the American Statistical Association*, **96**, 270–281.

Easterling, R. G. (2001). Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations. Sandia National Laboratories Report SAND2001-0243, February, 2001.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian Calibration of Computer Models (with discussion). *JRSS B* **63**, 425–464.

McKay, M. D., Conover, W. J. and Beckman, R. J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code. *Technometrics*, **21**, 239–245.

Oberkampf, W.L., and Trucano, T. (2000). Validation Methodology in Computational Fluid Dynamics. AIAA, 2000-2549.

Pilch, M., Trucano, T., Moya, J. L., Froehlich, G. Hodges, A., and Peercy, D. (2001). Guidelines for Sandia ASCI Verification and Validation Plans - Content and Format: Version 2.0. Sandia National Laboratories Report SAND2000-3101, January, 2001.

Roache, P.J. (1998). *Verification and Validation in Computational Science and Engineering.* Hermosa Publishers, Albuquerque.

Robert, C.P. and Casella, G. (1999). *Monte Carlo Statistical Methods.* Springer- Verlag, New York.

Sacks, J., Welch, W., Mitchell, T. J., and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, **4**, 409-435.

Saltelli, A., Scott, M., and Chan, K., eds. (2000). *Sensitivity Analysis.* Wiley, Chichester.

Searle, S. R. (1982). *Matrix Algebra Useful for Statistics.* Wiley, New York.

Trucano, T., Pilch, M., and Oberkampf, W. O. (2002). General Concepts for Experimental Validation of ASCI Code Applications. Sandia National Laboratories Report SAND 2002-0341, March, 2002.

Wang, P.C. and Hayden, D.B. (1999). Computational Modeling of Resistance Spot Welding of Aluminum. Research Report R&D - 9152, GM Research & Development Center.

A Resistance Spot Weld Process Model

A.1 Introduction

In resistance spot welding, two metal sheets are compressed by water-cooled copper electrodes, under an applied load, L. Figure 14 is a simplified representation of the spot weld process, illustrating some of the essential features for producing a weld. A direct current of magnitude C is supplied to the sheets via the two electrodes to create concentrated and localized heating at the interface where the two sheets have been pressed together by the applied load (the so-called faying surface).

A.2 The welding process

The welding process is comprised of six steps:

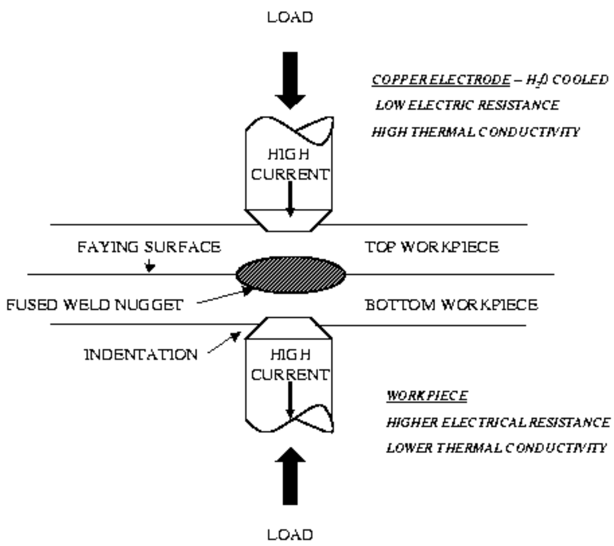

Figure 14: Resistance spot welding process

1. A load, L, is first applied to the electrodes producing an elastic and plastic deformation of the sheets. The resulting deformation brings into contact different areas at the electrode- sheet and faying surface, the size of which depends on the magnitude of the applied load, the *yield stress*, σ_S, of the sheet metal and *Young's modulus*, E, of the sheet and electrodes.

2. After the compression step, imposing a voltage drop across the electrodes generates a current of magnitude C. The current passes through the electrodes, sheets and faying surface. Both the electrodes and the sheets have well-defined temperature-dependent values of electrical and thermal conductivity, σ and κ, respectively.

3. Because of the current flow, heat will be generated and temperatures in the system will start to increase at a rate that depends on the local value of the resistance. The resistance offered at the faying surface is particularly critical in determining the magnitude of heat generated. If the applied load is too high, the two sheets will be pressed tightly together at the faying surface, producing little resistance and low generation of internal heat. This inhibits melting and nugget formation and growth. If the load is too low, then the air gap between the two sheets at the faying surface will be high producing a high resistance, high heating and possible uncontrolled nugget growth (expulsion and electrode degradation).

4. The physical properties of the materials will change locally as a consequence of the local increase in temperature. Young's modulus and the yield stress of the sheet will fall (that is, the metal will "soften") resulting in more deformation and increase in the size of the faying contact surface. At the same time, the electrical and thermal conductivities will decrease

as the temperature rises; all of which will affect the rate of heat generation and removal by conduction away from the faying surface.

5. If the applied current is high enough, sufficient heat will be generated to result in melting, first at the faying surface and then in an increasing volume of material about the faying surface. If the melt zone becomes too large, weld metal expulsion will occur.

6. When the current is turned off and the melt zone allowed to cool and quench, a nugget is formed and a spot weld results.

The thermal/electrical/mechanical physics of the spot weld process outlined above is modeled by a coupling of the continuum partial differential equations (PDE's) that govern heat and electrical conduction with those that govern temperature-dependent, elastic/plastic mechanical deformation (Wang and Hayden, 1999).

A.3 The computer models

Finite element implementations are used to provide a computerized model of the electro-thermal conceptual model. Similarly, a finite element implementation is made for the equilibrium and constitutive equations that comprise the conceptual model of mechanical/thermal deformation. These two computer models are implemented using two distinct modules of a commercial code (ANSYS).

Although the commercial finite element modules are distinct, they are coupled because the mechanical deformation affects the electro-thermal conduction process through its effect on the areas of the contacting surfaces. This is simulated in the computer model by passing the calculated temperature field to the deformation module, called as an external procedure, at intervals of a quarter of a cycle (1/240 seconds of simulated time). The updated contact areas are then passed back to the electro-thermal module from the deformation module.

B Modeling for Vehicle Crashworthiness

Modeling the effects of a collision of a vehicle with a barrier is routinely done by implementing a dynamic analysis code using a finite element representation of a vehicle. A finite element model includes the following components: complete body "in white" including windshield, cradle, bumper system, doors, engine/transmission, suspension, exhaust system, rear axle, drive shaft, radiator, steering system, instrument panel beam, and brake booster. Additional mass is often used to represent nonstructural components and inessential objects, while maintaining the actual vehicle test weight and its center of gravity. The element size is generally between 10 mm to 15 mm. Holes smaller than 15 mm diameter are not modeled unless located in critical areas; fillets, rounds, and radii less than 10 mm are ignored.

A finite element vehicle model consists mostly of shell and solid elements. Shell elements are used to model the rail, frame, and stamped/deep-drawn sheet panels; solid elements are used to model the bumper foam, radiator, battery, and the suspension system. An engine is usually modeled with shell elements on the exterior surface with properly assigned mass and moments of inertia to represent the massive engine block. Since the crash behavior of the vehicle is the primary concern, there is greater detail in the crush zone structure/components and the connections between them, to create greater accuracy. The number of elements range in size from 50,000 to 300,000. The duration of the simulations is between 100 and 150 milliseconds (msec) for most frontal impact conditions.

The computer model is run using a non-linear dynamic analysis (commercial) code, LS-DYNA. Computational turn around time can be great - from 1 to 5 days on a standard work station.

Variables and sources of uncertainty in the vehicle manufacturing process and proving ground test procedures induce uncertainties in the test results. The acceleration and velocity histories of two production vehicles of the same type, subjected to 30mph zero degree rigid barrier frontal impact tests, as shown in Figure 15 demonstrate the differences in "replicate" crashes. There are a variety of materials used in components of the vehicle and, consequently, a variety of material properties to deal with, not all of which my be well specified.

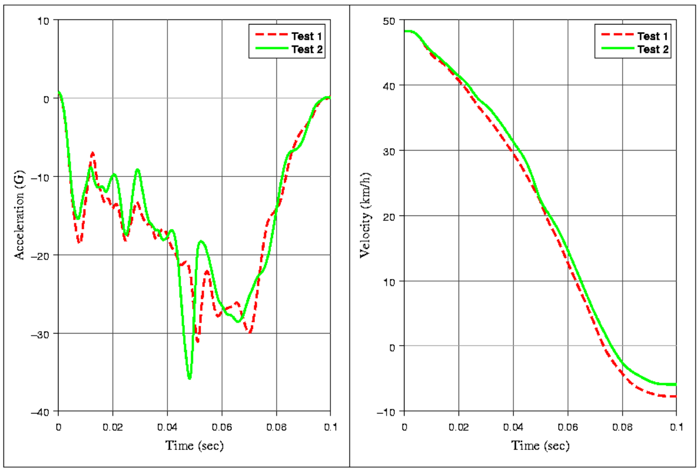

Figure 15: Acceleration and velocity pulses in the occupant compartment from 30mph zero degree rigid barrier frontal impact tests for two production vehicles of the same type.

An Input/Uncertainty map for the crash model is given in Table 7.

C Technical details for Section 4

C.1 The GASP response-surface methodology

Recall that $\boldsymbol{y}^M = (y^M(\boldsymbol{x}^1, \boldsymbol{u}^1), \dots, y^M(\boldsymbol{x}^m, \boldsymbol{u}^m))$ denotes the vector of m evaluations of the model at the inputs $D^M = \{(\boldsymbol{x}^i, \boldsymbol{u}^i) : i = 1, \dots, m\}$ and we write $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$. The computer model is exercised only at the inputs D^M, so that $y^M(\boldsymbol{z}) = y^M(\boldsymbol{x}, \boldsymbol{u})$ is effectively unknown for other inputs $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$. Thus, in the Bayesian framework, we assign $y^M(\boldsymbol{z})$ a prior distribution, specifically a stationary Gaussian process with mean and covariance functions governed by parameters $\boldsymbol{\theta}^L$ and

INPUT		IMP ACT	UNCERTAINTY	CURRENT STATUS
Geometry	Element size about 10mm	5	unspecified	fixed
	Holes $<$ 10mm, fillets, and rounds not meshed	4	unspecified	fixed
	Use of design surfaces, not surfaces after stamping	4	unspecified	fixed
	Spot weld locations are approximated	5	unspecified	fixed
	Thickness variation from location to location	3	Can be specified with c.v. of 2% for whole components	Controllable in some degree
Material Properties	Dynamic stress/strain curves	3	May be approximated with c.v. of 5% for major components	Controllable in some degree
	Spot weld failure force	3	unspecified	fixed
	Joints separation	2	Approximated with 5% c.v.	Controllable
	Damping factor	5	Controllable	fixed
	Friction coefficients between part	4	unspecified	fixed
	Material density	5	unspecified	fixed
Boundary/ Initial Conditions	Vehicle mass/speed	5	Can be matched with the test	fixed
	Barrier variation (plywood condition and barrier angle)	3	unspecified	fixed
	Vehicle test attitude	5	unspecified	fixed
	Testing environment (humidity & temperature)	5	unspecified	fixed
Restraint System	Steering column stroking force	2	5% of c.v.	controllable
	Airbag deployment time	4	5% of c.v.	controllable
	Seatbelt retractor force	2	5% of c.v.	controllable
	Airbag mass flow rate	2	5% of c.v.	controllable
	Occupant position	3	$\pm\frac{1}{2}$ inch on horizontal plan and $\pm\frac{1}{4}$ inch on vertical	controllable

Table 7: The input uncertainty map for a math vehicle.

$\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$, respectively. The mean function of the Gaussian process was discussed in Section 4, and so we turn to discussion of the covariance function.

The parameter λ^M is the precision (that is, the inverse of the variance) of the Gaussian process and the other parameters $(\boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$ control the correlation function of the Gaussian process, which we assume to be of the form

$$c^M(\boldsymbol{z}, \boldsymbol{z^*}) = \exp\left(-\sum_{j=1}^{d} \beta_j^M |z_j - z_j{}^*|^{\alpha_j^M}\right). \tag{C-1}$$

Here, d is the number of coordinates in $\boldsymbol{z}$, the α_j^M are numbers between 0 and 2, and the β_j^M are positive scale parameters. The product form of the correlation function (each factor is itself a correlation function in one-dimension) helps the computations made later. Prior beliefs about the smoothness properties of the function will affect the choice of $\boldsymbol{\alpha}^M$. The choice $\alpha_j^M = 2$ for all j reflects the belief that the function is infinitely differentiable, which is plausible for many engineering and scientific models.

This can be summarized by saying that, given the hyper-parameters $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$, the prior distribution of y^M is $GP(\boldsymbol{\Psi}(\cdot)\,\boldsymbol{\theta}^L, \frac{1}{\lambda^M}\, c^M(\cdot,\cdot)\,)$, i.e., a Gaussian process with the given mean and covariance function.

As before, let $\boldsymbol{y}^M$ denote the vector of model evaluations, at the set of inputs D^M. Then, before observing the y^M's, and conditionally on the hyperpartameters, $\boldsymbol{y}^M$ has a multivariate normal distribution with covariance matrix $\boldsymbol{\Gamma}^M = \boldsymbol{C}^M(D^M, D^M)/\lambda^M$, where $\boldsymbol{C}^M(D^M, D^M)$ is the matrix with (i,j) entry $c^M(\boldsymbol{z}_i, \boldsymbol{z}_j)$, for $\boldsymbol{z}_i, \boldsymbol{z}_j$ in D^M. Once $\boldsymbol{y}^M$ is observed, it is a likelihood function for the parameters $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M$ (based solely on the observed $\boldsymbol{y}^M$).

If $\boldsymbol{z}$ is a new input value, then the conditional distribution of $y^M(\boldsymbol{z})$, given $\boldsymbol{y}^M$, $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M$ is normal. Formally, the posterior density, $p(y^M(\cdot)|\,\boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$, is a Gaussian process with mean and covariance function given by

$$E[\,y^M(\boldsymbol{z})|\,\boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M] = \boldsymbol{\Psi}(\boldsymbol{z})\,\boldsymbol{\theta}^L + \boldsymbol{r}_z{}'(\boldsymbol{\Gamma}^M)^{-1}(\boldsymbol{y}^M - \boldsymbol{X}\boldsymbol{\theta}^L) \tag{C-2}$$

$$Cov[\,y^M(\boldsymbol{z}), y^M(\boldsymbol{z}^*)|\,\boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M] = \frac{1}{\lambda^M}\, c^M(\boldsymbol{z}, \boldsymbol{z}^*) - \boldsymbol{r}_z{}'(\boldsymbol{\Gamma}^M)^{-1}\boldsymbol{r}_{z^*}, \tag{C-3}$$

where $\boldsymbol{r}_z{}' = \frac{1}{\lambda^M}\,(c^M(\boldsymbol{z}, \boldsymbol{z}_1), \ldots, c^M(\boldsymbol{z}, \boldsymbol{z}_m))$, $\boldsymbol{\Gamma}^M$ is given above, $\mathbf{1} = (1, \ldots, 1)$ and $\boldsymbol{X}$ is the matrix with rows $\boldsymbol{\Psi}(\boldsymbol{z}_1), \ldots, \boldsymbol{\Psi}(\boldsymbol{z}_m)$.

With specifications for $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M$, the mean function, (C-2), can be used as an inexpensive emulator for $y^M(\cdot)$. Indeed, the response surface approximation to $y^M(\boldsymbol{x}, \boldsymbol{u})$, given $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M$ is simply $E[\,y^M(\boldsymbol{x}, \boldsymbol{u})|\,\boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M]$ and the variance of this approximation is $c^M((\boldsymbol{x}, \boldsymbol{u}), (\boldsymbol{x}, \boldsymbol{u}))/\lambda^M - \boldsymbol{r}'_{(\boldsymbol{x},\boldsymbol{u})}(\boldsymbol{\Gamma}^M)^{-1}\boldsymbol{r}_{(\boldsymbol{x},\boldsymbol{u})}$. Note that this variance is zero at the design points at which the function was actually evaluated.

However, the hyper-parameters $\boldsymbol{\theta}^M, \boldsymbol{\theta}^L$ are rarely, if ever, known. Two possibilities then arise:

a) Plug-in some estimates in the above formulae, for instance maximum likelihood estimates (as in the GASP software of W. Welch – see also Bayarri *et al.* (2002)), pretending they are the 'true' values. For MLE estimates $\hat{\boldsymbol{\theta}}^M, \hat{\boldsymbol{\theta}}^L$ this produces the following model approximation for inputs $(\boldsymbol{x}, \boldsymbol{u})$

$$\hat{y}^{MLE}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{u})\,\hat{\boldsymbol{\theta}}^L + \hat{\boldsymbol{r}}'_{(\boldsymbol{x},\boldsymbol{u})}(\hat{\boldsymbol{\Gamma}}^M)^{-1}(\boldsymbol{y}^M - \boldsymbol{X}\hat{\boldsymbol{\theta}}^L)\,,$$

where $\hat{\boldsymbol{\theta}}^M = (\hat{\lambda}^M, \hat{\boldsymbol{\alpha}}^M, \hat{\boldsymbol{\beta}}^M)$ is used to compute $\hat{\boldsymbol{\Gamma}}^M$ and $\hat{\boldsymbol{r}}_{(\boldsymbol{x},\boldsymbol{u})}$. Similarly, $\hat{\boldsymbol{\theta}}^M$ and $\hat{\boldsymbol{\theta}}^L$ are often 'plug-in' in $Cov[\,y^M(\boldsymbol{x}, \boldsymbol{u}), y^M(\boldsymbol{x}, \boldsymbol{u})|\,\boldsymbol{y}^M, \hat{\boldsymbol{\theta}}^L, \hat{\boldsymbol{\theta}}^M]$ (see equation (C-3)) when computing an estimate of 'error'. Notice that this can result in an underestimation of the true variability, since the uncertainty in the estimates $\hat{\boldsymbol{\theta}}^M$ and $\hat{\boldsymbol{\theta}}^L$ is not taken into account.

b) Integrating the parameters with respect to the posterior distribution in a full Bayesian analysis (as described in Section 5 and in Bayarri *et al.*, 2002) leads to a more appropriate emulator (model approximation), namely the integral of $E[\,y^M(\boldsymbol{x}, \boldsymbol{u})|\,\boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M]$ with respect to the posterior distribution of $\boldsymbol{\theta}^M, \boldsymbol{\theta}^L$. Likewise, the variance of the emulator is obtained by averaging $c^M((\boldsymbol{x}, \boldsymbol{u}), (\boldsymbol{x}, \boldsymbol{u}))/\lambda^M - \boldsymbol{r}'_{(\boldsymbol{x},\boldsymbol{u})}(\boldsymbol{\Gamma}^M)^{-1}\boldsymbol{r}_{(\boldsymbol{x},\boldsymbol{u})}$ over the posterior distribution of $\boldsymbol{\theta}^M, \boldsymbol{\theta}^L$. This, in practice (see Bayarri *et al.* (2002)) amounts to generating N (large) values $(\boldsymbol{\theta}_i^L, \boldsymbol{\theta}_i^M)$ from its posterior distribution, and then simply evaluate the previous quantities at these generated values and take the average. Hence the proposed Bayesian model approximation to $y^M(\cdot)$ is

$$\hat{y}^M(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{N} \sum_{i=1}^{N} \left[\boldsymbol{\Psi}(\boldsymbol{z})\, \boldsymbol{\theta}_i^L + \boldsymbol{r}_i{}'_{(\boldsymbol{x},\boldsymbol{u})}(\boldsymbol{\Gamma}_i^M)^{-1}(\boldsymbol{y}^M - \boldsymbol{X}\boldsymbol{\theta}_i^L) \right],$$

where $\boldsymbol{r}_i{}'_{(\boldsymbol{x},\boldsymbol{u})}$ and $\boldsymbol{\Gamma}_i^M$ are computed using the generated values $\boldsymbol{\theta}_i^M$ for the parameter ($i = 1, \ldots, N$). Likewise, the proposed variance function is

$$V^M(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{\lambda_i^M}\, c_i^M((\boldsymbol{x}, \boldsymbol{u}), (\boldsymbol{x}, \boldsymbol{u})) - \boldsymbol{r}_i{}'_{(\boldsymbol{x},\boldsymbol{u})}(\boldsymbol{\Gamma}_i^M)^{-1}\boldsymbol{r}_i{}'_{(\boldsymbol{x},\boldsymbol{u})} \right],$$

where again the generated values of $\boldsymbol{\theta}_i^M$ are used to evaluate the functions c_i^M.

Note that, while the proposed (Bayesian) model approximation $\hat{y}^M(\boldsymbol{x}, \boldsymbol{u})$ will often be similar to its MLE counterpart, $\hat{y}^{MLE}(\boldsymbol{x}, \boldsymbol{u})$, the proposed variance function, $V^M(\boldsymbol{x}, \boldsymbol{u})$, will typically be larger than the corresponding plug-in variance function, because it appropriately takes into account uncertainty in the parameters.

The procedure for obtaining a sample from the posterior distribution of y^M, for computing the posterior mean and variance above can be summarized as follows.

1. Start with (i) the likelihood function of the model-run data, which from Bayarri *et al.* (2002) is proportional to a multivariate normal $MVN(\boldsymbol{X}\boldsymbol{\theta}^L, \boldsymbol{\Gamma}^M)$ distribution; (ii) prior distributions for $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$, as given in Bayarri *et al.* (2002).

2. The posterior distribution of $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$ is then approximated by an MCMC analysis that is a simplified version of that described in Bayarri *et al.* (2002).

3. The posterior distribution of $y^M(\boldsymbol{z}_{\text{new}})$ is then obtained by first sampling the posterior distribution of $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$, then sampling the multivariate normal with mean and covariance given by (C-2) and (C-3) with the sampled hyper-parameters. This is repeated many times to produce the required sample from the posterior distribution of y^M.

This emulator can roughly be thought of as an interpolator of the data, unless there is numerical instability in the computer model, as mentioned in footnote 2, in which case the emulator smoothes the data.

C.2 Processing stochastic inputs with GASP

One of the attractions of the particular form of the covariance function that we use for GASP is that it can greatly simplify the handling of certain types of stochastic inputs. Writing

$$\boldsymbol{a} = (a_1, \ldots, a_m)' = \frac{1}{\lambda^M} (\boldsymbol{\Gamma}^M)^{-1} (\boldsymbol{y}^M - \boldsymbol{X}\boldsymbol{\theta}^L)\,,$$

expressions (C-2) and (C-1) yield

$$\begin{aligned} E[\, y^M(\boldsymbol{z}) |\, \boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M] &= \boldsymbol{\Psi}(\boldsymbol{z})\, \boldsymbol{\theta}^L + \sum_{i=1}^m a_i\, c^M(\boldsymbol{z}, \boldsymbol{z}_i) \\ &= \boldsymbol{\Psi}(\boldsymbol{z})\, \boldsymbol{\theta}^L + \sum_{i=1}^m a_i \prod_{j=1}^d \exp\{-\beta_j^M |z_j - z_{ij}|^{\alpha_j^M}\}. \end{aligned} \tag{C-4}$$

Suppose now that inputs $z_b, \ldots, z_d$ are stochastic, with (for simplicity) independent densities $p_j(z_j)$. Then taking the expectation of (C-4) over these random inputs yields

$$E[E[\, y^M(\boldsymbol{z}) |\, \boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M]] = E[\boldsymbol{\Psi}(\boldsymbol{z})]\, \boldsymbol{\theta}^L + \sum_{i=1}^m a_i \prod_{j=1}^{b-1} e^{-\beta_j^M |z_j - z_{ij}|^{\alpha_j^M}} \prod_{j=b}^d \int e^{-\beta_j^M |z_j - z_{ij}|^{\alpha_j^M}}\, p_j(z_j)\, dz_j\,. \tag{C-5}$$

Assuming the underlying basis functions $\boldsymbol{\Psi}(\boldsymbol{z})$ are chosen so that their expectation with respect to the $p_j(z_j)$ is easily computable (trivial, for instance, if the mean function is linear), (C-5) shows that the expectation reduces to computation of a collection of one-dimensional integrals.

This can be an enormous computational simplification, especially when, say, optimization over the nonrandom inputs $z_1, \ldots, z_{b-1}$ is desired. The one-dimensional integrals in (C-5) can be carried out in a pre-processing step, and the optimization then easily implemented.

Even greater simplifications are possible if the α_j^M equal 1 or 2 and the $p_j(z_j)$ are normal or exponential densities; the one-dimensional integrals can then be carried out in closed form. Furthermore, if the $\alpha_j^M = 2$ (a possible choice if, for instance, the computer model is expected to be very smooth), then even a multivariate normal density for $z_b, \ldots, z_d$ will lead to closed from integrals.

D Technical details for Section 5

D.1 Prior distribution for the bias function

The prior density of the bias is taken to be another Gaussian process with correlation function as in (C-1), but with its own set of hyper-parameters. However, we restrict attention to smooth bias functions by fixing all components of the vector $\boldsymbol{\alpha}^b$ to be two. In part, this is done for technical reasons; since the bias cannot be observed directly, there is very little information available about $\boldsymbol{\alpha}^b$, and numerical computations are more stable with $\boldsymbol{\alpha}^b$ specified. There is also the notion that the bias process might typically be smoother than the model process; for instance, the model process might only be ‘off’ by a level-shift, because of something forgotten or inappropriately specified in the model. Indeed, there is both empirical and ‘folklore’ evidence of this. Empirically, in the examples we have looked at, the maximum likelihood estimates of $\boldsymbol{\alpha}^b$ have mostly been near 2. As to folklore, it is often claimed that even biased models are typically accurate for predicting small changes, which would not be true if bias were not smoother than the model outputs. Finally, note that the bias can still assume the form of any infinitely differentiable function.

The mean function of the Gaussian process used to model the bias is typically chosen to be either zero or an unknown constant. (In the test beds, we used a zero mean for SPOT WELD and allowed a non-zero mean for CRASH.) More complicated linear structures, such as $\boldsymbol{\Psi}(\boldsymbol{z})\boldsymbol{\theta}^L$, are possible, but we have not yet ascertained the extent to which they are helpful.

D.2 Analysis with model approximation

An outline of the Bayesian analysis when the model y^M is inexpensive to run was given in Section 5.3. Typically, however, models runs are very expensive. The actual analysis, in this case, is similar to that described in Section 5.3, except that now y^M is also viewed as unknown, with the Gaussian process prior. Indeed, we recommend that one directly use $p(y^M|\boldsymbol{y}^M, \boldsymbol{\theta}^M)$ from Appendix C.1 as the posterior distribution of y^M. Then the only modification needed in the analysis in Section 5.3 is to draw $y^M(\boldsymbol{x}, \boldsymbol{u}_i)$ directly from this posterior (i.e., draw $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$ from its posterior, based on the model data, and then compute $y^M(\boldsymbol{x}, \boldsymbol{u}_i)$ from the GASP posterior using these parameter values) whenever it is needed to compute the likelihood $f(\boldsymbol{y}^F|\boldsymbol{u}, \lambda^F, b)$.

If one is going to predict the process at some new input vector $\boldsymbol{x}$, one can

Case 1. Compute $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$, (preferable, if possible). It is then important to update the posterior distribution $p(y^M|\boldsymbol{y}^M, \boldsymbol{\theta}^M)$ by including the data point $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$ in the data $\boldsymbol{y}^M$, but keeping the other aspects of the posterior distribution unchanged. This can be done by an updating formula that is given in Bayarri *et al.* (2002).

Case 2. Use the prediction arising directly from the above posterior, avoiding computation of $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$.

The above analysis is really only an approximate Bayesian analysis, in two respects. First, the recommendation to include the data point $y^M(\boldsymbol{x}, \hat{\boldsymbol{u}})$ in the data $\boldsymbol{y}^M$, but keep the other aspects of the posterior distribution unchanged, is not the full Bayesian analysis; but a full analysis would require rerunning the entire MCMC with this data point added, which will rarely be feasible. The second approximate aspect of the analysis is that the formal posterior distribution of all unknowns is actually

$$p(y^M, \boldsymbol{u}, \lambda^F, b, \boldsymbol{\theta}|\boldsymbol{y}^F, \boldsymbol{y}^M) \propto f(\boldsymbol{y}^F|y^M, \boldsymbol{u}, \lambda^F, b)p(y^M|\boldsymbol{y}^M, \boldsymbol{\theta}^M)f(\boldsymbol{y}^M|\boldsymbol{u}, \boldsymbol{\theta}^M)p(\boldsymbol{u}, \lambda^F, b)p(\boldsymbol{\theta}^M)\,,$$

where $p(\boldsymbol{\theta}^M)$ is the prior density of $\boldsymbol{\theta}^M$ and we now recognize that y^M is also unknown in the likelihood arising from $\boldsymbol{y}^F$. The posterior distribution $p(y^M|\boldsymbol{y}^M, \boldsymbol{\theta}^M)$ is readily available from the

GASP theory. The main reasons not to utilize this posterior is that it significantly increases the difficulty of performing the needed updating when $y^M(\boldsymbol{x}, \hat{u})$ is computed.

E Technical details for Section 7

E.1 Kronecker product

Since we assume that the functions are discretized at the same points in D_t^F for all $\boldsymbol{x}$ in the data, the overall design spaces (the sets of $(\boldsymbol{x}, t)$ points at which model-run and field data are obtained) can be written as the products $D_x^F \times D_t^F$ and $D_x^M \times D_t^M$. The product form of the correlation functions for the model approximation and bias processes then induces a simple algebraic structure. Specifically, the correlation *matrices* induced by the correlation functions (the (i, j)th entries of the matrices are the c's evaluated at the i and j data points) have a form that can be manipulated to simplify computation. The basic idea lies in recognizing that the matrices have the so-termed form of a Kronecker product defined as: $\boldsymbol{A} \otimes \boldsymbol{B}$ of matrices $\boldsymbol{A}_{m\times n}, \boldsymbol{B}_{p\times q}$ is the $mp \times nq$ matrix whose i, j block is $a_{ij}\boldsymbol{B}$.

Indeed, if we denote the correlation matrices by $\boldsymbol{C}$ (there will be appropriate superscripts corresponding to the particular correlation functions generating the matrices), then each element of $\boldsymbol{C}$ is a product of an "$\boldsymbol{x}$" term and a "t" term . Since the computer model design is of the form $D^M \times D^T$, we can write $\boldsymbol{C}^{M,T}$ as a Kronecker product:

$$\boldsymbol{C}^{M,T} = \boldsymbol{C}^M \otimes \boldsymbol{C}^T, \tag{E-1}$$

where $\boldsymbol{C}^M$ and $\boldsymbol{C}^T$ are correlation matrices corresponding to the $\boldsymbol{x}$ and t components of the correlation functions. The same can be done with $\boldsymbol{C}^{b,T}$ and $\boldsymbol{C}^{F,T}$. We could have different $\boldsymbol{C}^T$ for M, b and F in (E-1) but we take these C^T to be the same in each case for simplicity and to make computations feasible. The assumption that C^T is the same for M, b and F is not unreasonable - any choice of C^T results in the posterior means of y^M and y^F to be interpolators as functions of t for fixed x, and we can always select enough t points to ensure enough accuracy in the predictions along t.

The advantage of the Kronecker product structure lies in resulting simplifications for calculating inverses of the correlation matrices. This is crucial because these inverses must be calculated many times in the MCMC process that produces the posterior distributions of the model parameters. Specifically, the inverse of, say, $\boldsymbol{C}^M$ is the Kronecker product of the component inverses (Searle, 1982):

$$(\boldsymbol{C}^{M,T})^{-1} = (\boldsymbol{C}^M)^{-1} \otimes (\boldsymbol{C}^T)^{-1}. \tag{E-2}$$

Because the component matrices are $m \times m$ and $T \times T$ while $\boldsymbol{C}^{M,T}$ is $m\,T \times m\,T$ the computational savings in computing the inverse are obvious.

CRASH: Using data only for the straight frontal barrier there are 9 different impact velocities and if we use 19 time points we get a total of 171 data points. The correlation matrix, $\boldsymbol{C}^M$, corresponding to the computer runs, is then a 171×171 matrix but is a Kronecker product of 9×9 and 19×19 matrices.

E.2 Analysis of function output

We have two sets of data: computer model data and field data. The measurement error term in the model for the field data must incorporate "time" dependence of the field observations. We give here only an sketch of the approach followed. Full details can be found in Bayarri *et al.* (2002). Let $\bar{\boldsymbol{y}}^F$ denote the sufficient statistic for field data Given the hyper-parameters of the Gaussian process priors the distribution of the complete data vector $\boldsymbol{y} = (\boldsymbol{y}^M, \bar{\boldsymbol{y}}^F)$ is

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}^L, \boldsymbol{\beta}^M, \lambda^M, \boldsymbol{\beta}^{\boldsymbol{b}}, \lambda^b, \lambda^F, \beta_t^\epsilon, \alpha_t^\epsilon, \boldsymbol{\alpha}^M, \boldsymbol{\alpha}^b) = N\left(\begin{pmatrix} \boldsymbol{\psi}(\mathbf{z}_1) \\ \vdots \\ \boldsymbol{\psi}(\mathbf{z}_n) \end{pmatrix} \boldsymbol{\theta}^L, \boldsymbol{\Sigma} \otimes \boldsymbol{C}^T \right), \tag{E-3}$$

where $\boldsymbol{z}_i$ is the $(\boldsymbol{x}, t)$ input associated with $\boldsymbol{y}_i$, and $\boldsymbol{\Sigma}$ is a covariance matrix whose specific form is given in Bayarri *et al.* (2002).

The MCMC analysis (see Bayarri *et al.*, 2002 for full details) can be carried out for function output. The necessary inversions of $\boldsymbol{\Sigma} \otimes \boldsymbol{C}^T$ are simplified because of the Kronecker product structure and (E-2). The posterior distribution of $y^M(\boldsymbol{x}, t)$, for each selected point in $D^P = D_x^P \times D_t^P$ (D_x^P contains the $\boldsymbol{x}$ points at which we want function realizations; D_t^P is dense enough to get a good image of the function $y^M(\boldsymbol{x}, \cdot)$ and is not the same as D^T), can be obtained (simulated). This produces a prediction $\hat{y}^M(\boldsymbol{x}, t)$ of $y^M(\boldsymbol{x}, t)$ and accounts for uncertainties in the unknown parameters. By doing so for each t we get a prediction of $y^M(\boldsymbol{x}, \cdot)$ with *pointwise* uncertainties.

Inference for a specific evaluation criterion proceeds as follows. For each realization of the parameters the output function is simulated (at least at a reasonably dense set of t— see Bayarri *et al.*, 2002). Then the evaluation criterion is calculated. Repeating this yields a sample from its posterior distribution. This procedure can be applied to simulations from the posterior distribution of the model y^M and from the posterior distribution of reality y^R.

F Technical details for Section 8

Suppose we have K related models. We assume that the models are related as follows (see Bayarri *et al.*, 2000, for discussion):

1. All models and field data have common α's, common β's, common λ^M's and common λ^F. The choice of priors for these is explained in Bayarri *et al.* (2002).

2. $\boldsymbol{\theta}_i^L \sim N_k(\boldsymbol{\mu}, \mathrm{Diag}(\tau_1^{-1}, \ldots, \tau_k^{-1}))$, where $p(\boldsymbol{\mu}, \tau_1, \ldots, \tau_k \mid \lambda^M) = \prod_{i=1}^k [\tau_i^{-1}(\tau_i + \frac{\nu}{\lambda^M})^{-1}]$, with ν equalling the average number of model runs.

3. $\log(\lambda_i^b) \sim N(\xi, 4q^2)$, where q is the proportional variation in the bias that is expected among models (e.g., $q = 0.1$). A constant prior density is assigned to ξ.

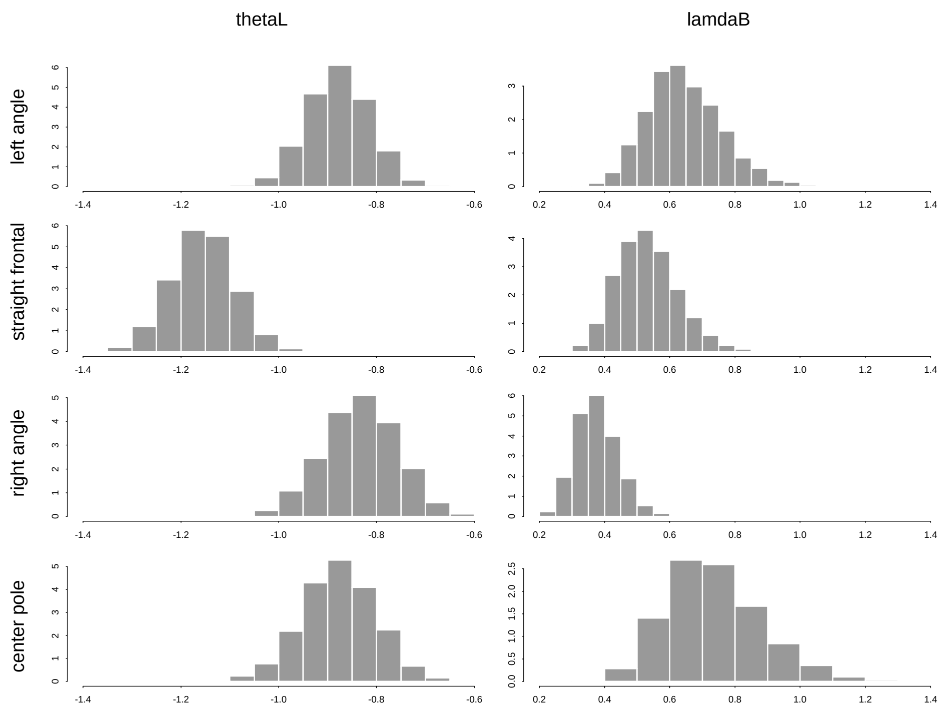

Figure 16: Posterior distributions for 4 barrier types, based on the hierarchical model.

The crucial input needed here is q in Assumption 3. The condition in Assumption 3 implies that the variance of $\log(\lambda^b_j/\lambda^b_i)$ is $8q^2$ so that the standard deviation is $\sqrt{8}q$. Therefore $\log(\lambda^b_j/\lambda^b_i)$ is likely to be less than $2q$ or equivalently $2\log\sqrt{\lambda^b_j/\lambda^b_i} < 2q$ and then $\sqrt{\lambda^b_j/\lambda^b_i} < e^q \sim 1+q$. So Assumption 3 is roughly equivalent to saying that the ratio of the standard deviation (SD) of the biases is less than $1+q$ or, stated another way, the proportional variation in the bias is q. Specifying $q = 0.1$ is stating that the biases are expected to vary by about 10% among the various cases being considered. Then, also roughly, $SD(\log\sqrt{\lambda^b_j/\lambda^b_i}) \sim \log SD(\sqrt{\lambda^b_j/\lambda^b_i}) \sim q$ or $SD(\log(\lambda^b_j/\lambda^b_i)) \sim 2q$, which is a consequence of Assumption 3.

Note that, because of these assumptions, our earlier notation does not need to be changed to deal with the hierarchical situation (i.e., we simply add the index i corresponding to different models to the parameters $\boldsymbol{\theta}^L$, λ^M and λ^b that we allow to vary between models).

CRASH: The hierarchical model is as given with $q = 0.1$, and for the prior on $\boldsymbol{\theta}^L$, we take $\nu = 6.5$. The prior distributions are the same as used for the straight frontal analysis, which are described in Bayarri *et al.*(2002); this is reasonable, since the priors are relatively non-informative and the straight frontal dataset is the largest of the 4 categories.

Figure 16 shows the posterior distributions of $\log\lambda^b$ and θ^L for individual barrier types. Note that, while the assumed similarity between the models allows information to be passed from 'large data' to 'small data' models, the models are still allowed to vary significantly.