Flexible Approximation Model Approach for Bi-Level Integrated System Synthesis

Hongman Kim[*], Scott Ragon[*], Grant Soremekun[*], Brett Malone[*]
Phoenix Integration, Inc.
and
Jaroslaw Sobieszczanski-Sobieski[†]
NASA Langley Research Center

Abstract

Bi-Level Integrated System Synthesis (BLISS) is an approach that allows design problems to be naturally decomposed into a set of subsystem optimizations and a single system optimization. In the BLISS approach, approximate mathematical models are used to transfer information from the subsystem optimizations to the system optimization. Accurate approximation models are therefore critical to the success of the BLISS procedure. In this paper, new capabilities that are being developed to generate accurate approximation models for BLISS procedure are described. The benefits of using flexible approximation models such as Kriging are demonstrated in terms of convergence characteristics and computational cost. An approach of dealing with cases where subsystem optimization cannot find a feasible design is introduced by using the new flexible approximation models for the violated local constraints.

Introduction

Approximation models have become an essential element in many multidisciplinary design optimization (MDO) techniques. For example, polynomial models known as response surface (RS) models have been widely used[1,2]. Approximation models replace expensive simulation codes in the optimization process to reduce total computational cost. By performing the simulation a priori, the simulation model may be separated from the optimizer or other coupled simulation codes so that the tool integration efforts can be reduced.

Bi-Level Integrated System Synthesis (BLISS)[3] is an approach that allows design problems to be naturally decomposed into a set of subsystem optimizations and a single system optimization. The BLISS approach overcomes the difficulties of optimizing complex systems such as aircraft or automobiles where multidisciplinary interactions are prominent. BLISS performs a set of subsystem optimizations to create approximation models of the optimal subsystem, which are then used in the system optimization. The approximation model approach of BLISS effectively separates coupled subsystems and plays an important role in transferring information from subsystem level to system level. Therefore the success of the BLISS process depends on generating accurate approximation models.

A parallel and collaborative environment is being developed for BLISS MDO[4]. Its goal is three-fold: First, development of parallel trade study support to perform hundreds of subsystem optimizations; second, development of collaborative design support to maximize utilization of project data and to provide easy access to the data from anywhere; third, development of advanced MDO support to allow building advanced MDO scenarios quickly. This work describes efforts towards the third goal. An example design case for a supersonic business jet (SBJ) is used to demonstrate that ModelCenter[5] MDO framework can successfully solve BLISS problems. In this paper, the use of new approximate modeling capabilities will be demonstrated and the results will be compared to those of polynomial based models.

Bi-Level Integrated System Synthesis

The design optimization of complex systems is a challenge because, on the one hand, contributing engineering groups need autonomy to perform specialized analysis while, on the other hand, their domains are coupled by data exchange. The BLISS MDO procedure addresses this problem via decomposition. Disciplinary groups create response surface models of their domain with disciplinary constraints, and system group performs system optimization with compatibility variables. To explain the BLISS MDO process, we begin with the MDF

---

[*] member AIAA
[†] Senior Research Scientist, Fellow AIAA

(Multidisciplinary Feasible) formulation as the baseline.

MDF Formulation

Find:	$X = \{X_{loc}, X_{sh}\}$	1.a
Minimize:	$f(X)$	1.b
Satisfy:	$g_k(X) \leq 0$	1.c
	$X_L \leq X \leq X_U$	1.d

The MDF formulation has only a single level optimization. Each disciplinary module that can be isolated as a unit with its input and output variables can be considered as a black box (BB). The design variables of MDF formulation include both local design variables (X_{loc}) and shared design variables (X_{sh}) common to more than one BB modules considered. For coupled multidisciplinary design problem, MDF formulation often requires iterative runs of BB modules to get converged values of the objective, $f(X)$, and constraints, $g_k(X)$. But this iterative multidisciplinary analysis (MDA) process can be expensive and sometimes it may be impossible to solve due to convergence problems involving various tools of varying accuracy.

In BLISS MDO, two levels of optimization are performed in sequence. First, sub-systems are optimized with respect to local design variables specific to each disciplinary BB module. This step is called BB optimization (BBOPT). Second, system optimization (SOPT) is performed with respect to system variables shared by multiple BB modules and coupled response variables.

Black Box Optimization (BBOPT) and Surrogate Model Generation

Given:	$Q = \{X_{sh}, Y^*, w\}$	2.a
Find:	$U = \{X_{loc}\}$	2.b
Minimize:	$f_k(U) = \Sigma \, w_i \hat{Y}_i$	2.c
Satisfy:	$g_k(U) \leq 0$	2.d
	$U_L \leq U \leq U_U$	2.e

Using the BBOPT runs create surrogate models: $\hat{Y_a}$ (for coupled response $\hat{Y}$)

System Optimization (SOPT)

Find:	$Q = \{X_{sh}, Y^*, w\}$	3.a
Minimize:	$f_k(Q) = f_k(\hat{Y_a})$	3.b
Satisfy:	$Y_a^* - \hat{Y_a} = 0$	3.c
	$Q_L \leq Q \leq Q_U$	3.d

The influence of a BB on the system objective is captured by formulating the sub-system objective as a weighted sum of the coupled output responses ($\hat{Y}$) of the BB. Shared design variables, coupled input variables coming from another BB (Y^*), and the weighting factors are fixed for each BB optimization run. The disciplinary constraints are satisfied at BB optimization (2.d). In BLISS, the BB optimizations are repeated as required by the design of experiments (DOE) technique to generate a large number of BB designs in the design space Q of the system optimization. Approximate surrogate models ($\hat{Y_a}$) are constructed using the BB optimization data for each of the coupled response variables, which will be used in the system optimization.

The system optimization manipulates the system variables, the coupling variables, and weighting coefficients to improve the system objective and to restore couplings, i.e., the output-input equalities in the data exchanges among the BBs. Because surrogate models are used, the BB modules can be completely separated from the system optimizer. A step-by-step procedure of BLISS is described below (Ref. 6).

1. Initialize system variables, weighting coefficients, and coupled output variables, along with initial anticipation of their lower and upper bounds.

2. Use design of experiment (DOE) patterns to create a dispersion of inputs to a BB to provide a wide and unbiased coverage of the design space defined by the system design variables, the weighting coefficients, and the coupling variables

3. Perform sub-system optimization at each design point from the DOE. Apply local disciplinary constraints, and use the weighted sum of the BB output variables as the objective.

4. Create a surrogate approximation model for each output item. Polynomial-based response surface (RS) is one of commonly used approximation models.

5. Repeat steps 2-4 for all the BBs.

6. Perform system optimization in the design space defined by Q, using the RS models from the BB optimization. Apply equality constraints to restore couplings among the BBs, and use a system performance measure as the objective.

7. Check if the design history has converged. If convergence has been achieved, terminate the process. Otherwise, update variable bounds to adjust the search area for the system optimization and continue steps 2 to 6 until convergence (i.e., BLISS cycles).

Choice of Surrogate Model for BLISS

Note that approximation models play a key role in the BLISS process because they transfer subsystem optimization results to the system optimization. The accuracy of the approximations will directly affect the quality of the optimum found from the BLISS procedure. Because of approximate nature of the surrogate models, it is required to repeat the sequence of surrogate model generation of BBOPT and SOPT. One sequence of BBOPT surrogate model generation and SOPT is called a BLISS cycle. As BLISS cycles progress, the search region defined by lower and upper bounds in Q space is moved toward the optimum point and gradually reduced, so that the search region becomes small enough to approximate responses accurately around the optimum.

However, when surrogate models are not accurate during BLISS cycles, it may cause convergence problem or even convergence to spurious non-physical optimum. Accurate surrogate models will accelerate convergence of the BLISS cycle. If the surrogate models are perfectly accurate, it is possible that BLISS converges with only a single cycle. Since the construction steps for the approximation models are independent for each BB module, one is free to select the most suitable model for each BB module. For example, a polynomial model can be used for BB1, while Kriging model could be used for BB2. Disciplinary groups may use a surrogate model that would fit their BB modules the best based on their experience. In cases where the BB module is very cheap and easy to handle, one may choose to skip surrogate model building and use the actual analysis at the system level.

In previous BLISS studies[3,8,9], polynomial RS models have been used. The response surface methodology[10] was originally developed in experimental sciences, but it has become popular to fit computer simulation results. RS techniques, assuming a polynomial model for underlying input and output relationship creates a global approximation to cover the design space under examination. RS technique can be effective to approximate uni-modal response with its popular choice of quadratic model. Least squares method is used to fit the model to the observed data. RS techniques are based on well established statistical theories such as analysis of variance (ANOVA), which is useful to analyze relative importance of input and output relationship. The polynomial model results in a compact representation of the fit, which is inexpensive to use. However, the number of regressions coefficients increases very rapidly as the number of design variables, d, increases. For example, the number of computer simulations required for a quadratic model is $(d+1)(d+2)/2$, which would be prohibitively large for about more than 20 variables, unless these runs can be computed in parallel using massively parallel computers.

In this work, Kriging model is compared to polynomial model in a BLISS application. Kriging model, originally developed in the field of geostatistics, was proposed to fit deterministic computer simulations by Sack, et al[11]. In Kriging, the response Y as a function of d-dimentaional input $\mathbf{x}$ is modeled as

$$Y(\mathbf{x}) = \beta + Z(\mathbf{x}) \qquad 4.a$$

where β is a constant, Z is a Gaussian process with mean 0 and variance σ^2 at each $\mathbf{x}$. Correlation of Z of two different locations $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, is modeled as a positive definite function. For example an exponential function is the most common choice:

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left[-\sum_{k=1}^{d} \theta_k \mid x_k^{(i)} - x_k^{(j)} \mid^2 \right], \qquad 4.b$$

where θ_k are scale parameters that need to be estimated. If there are N observations (or sites) of Y, define

$$\mathbf{S} = \{ \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)} \}, \qquad 4.c$$

$$\mathbf{Y_s} = \{ Y(\mathbf{x}^{(1)}), Y(\mathbf{x}^{(2)}), \ldots, Y(\mathbf{x}^{(N)}) \}, \qquad 4.d$$

$$\text{Cov}[Z(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)})] = \sigma^2 \mathbf{R}[R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})], \qquad 4.e$$

$$\mathbf{r}(\mathbf{x},\mathbf{S}) = \{ R(\mathbf{x}, \mathbf{x}^{(1)}), R(\mathbf{x}, \mathbf{x}^{(2)}), \ldots, R(\mathbf{x}, \mathbf{x}^{(N)}) \}, \ 4.f$$

where $\mathbf{R}$ is the correlation matrix of Z between observed sites $\mathbf{S}$, and $\mathbf{r}$ is the correlation vector of Z between a location $\mathbf{x}$ and $\mathbf{S}$. Kriging model seeks expected value of Y given the N observations:

$$\hat{Y}(\mathbf{x}) = E(Y(\mathbf{x})) \mid \mathbf{Y_s}), \qquad 4.g$$

which can be shown to be

$$\hat{Y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}(\mathbf{x},\mathbf{S})^T \mathbf{R}^{-1}(\mathbf{Y_s} - \vec{1} \cdot \hat{\beta}) \qquad 4.h$$

and

$$\hat{\beta} = (\vec{1}^T \mathbf{R}^{-1} \vec{1})^{-1} \vec{1}^T \mathbf{R}^{-1} \mathbf{Y_s}. \qquad 4.i$$

Fitting Kriging model requires finding parameter θ_j for d independent design variables via maximum likelihood estimate. This is typically solved as a d-dimensional unconstrained optimization problem, which can be computational expensive for high dimensional problems.

Kriging model interpolates the responses so that the prediction is exact at the observation sites. Since it does not make any assumption for underlying model of the response, Kriging is able to approximate multi-

American Institute of Aeronautics and Astronautics

modal responses whereas the quadratic polynomial model is inherently uni-modal. Due to its interpolation nature, Kriging can be used to approximate the design space globally along with local refinement by adding data into area of interests. The Kriging model program[12,13] that was developed from researches of the Boeing Company was repackaged so that it can be used within ModelCenter framework environment.

Giunta and Watson[14] compared accuracy of quadratic polynomial and Kriging model on two types of test functions: sinusoidal and quasi-quadratic functions in 1, 5, and 10 dimensions. It was reported that polynomial model was more accurate than Kriging model except the sinusoidal function in one dimension. However, its authors noted that the results should not be taken as a general conclusion because the accuracy of surrogate models will vary depending on the nature of the response being approximated. In fact, the Kriging model used in ref. 14 was a simplified model, using the same θ_k for all dimensions, which may have reduced modeling capability. Simpson et al.[15] reported comparison of Kriging and polynomial model on a nozzle design problem. The two showed comparable prediction accuracy. Jin et al.[16], compared four different surrogate models including polynomial, Kriging, multivariate adaptive regression splines (MARS), and radial basis function (RBF). It was reported that RBF performed better in many test functions, and Kriging and polynomial model showed comparable prediction accuracy. This paper adds one more case to make such comparison from a perspective of an MDO application.

BLISS Example Model

A supersonic business jet (SBJ) design example has been selected for this study. The example was taken from a previous BLISS research work[7]. There are four disciplinary modules in this problem: structures, aerodynamics, propulsion, and performance. Each module uses simple algebraic equations written in MATLAB® scripts, suitable for simplified conceptual design. There are 6 system variables, and the structures module has two local variables, while the aerodynamics and propulsion modules have one local variable each.

Table 1 summarized design variables and response variables of the SBJ problem.

Figure 1 shows a design structure matrix of the SBJ example. It is seen that structures, aerodynamics and propulsion BBs are coupled by the feedback loops. The performance module simply calculates the system performance from the outputs from the upstream modules. The system objective is to maximize the range.

Table 1: Variable table of supersonic business jet problem.

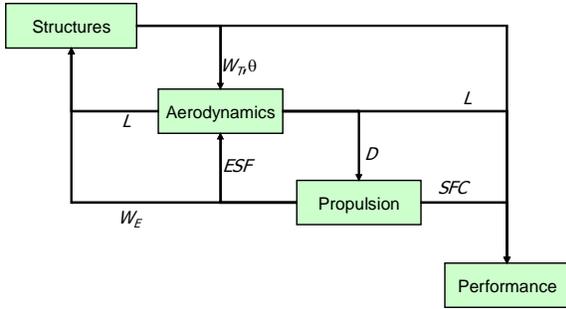| BB module	Output variables	Input variables X_{sh}	Input variables X_{loc}	Input variables Y^*	Local constraints
Structures	W_T: total weight W_F: fuel weight θ: wing twist	t/c: thickness ratio AR: aspect ratio Λ: sweep angle S_{ref}: reference area	λ: taper ratio X: wingbox cross section	L W_E	g_{1-5}:wing stress g_6: wing twist
Aerodynamics	L: lift D: drag L/D: lift to drag ratio	t/c: thickness ratio h: altitude M: mach number AR: aspect ratio Λ: sweep angle S_{ref}: reference area	C_f: skin friction coefficient	W_T θ ESF	g_1: pressure gradient
Propulsion	SFC: specific fuel consumption W_E: engine weight ESF: engine scale factor	h: altitude M: mach number	T: thrust	D	g_1:engine scale factor g_2: engine temperature g_3: throttle setting
Range	R: range	h: altitude M: mach number		W_T W_F L/D SFC	

Figure 1: Data dependency of the SBJ example.

The underlying assumption in the present formulation of BLISS is that a BB is capable of generating a design feasible with respect to local constraints for any settings of system design variables, weighting coefficients, and the coupling variables. In some cases, this assumption may not be correct. It may happen that for some combinations of system variables, a BB module would be unable to produce a feasible design by manipulating only the local variables. The inability to conclude a BB optimization with a feasible design may cause trouble in the BLISS process. For example, the flow separation constraint (e.g., pressure gradient) of the aerodynamic BB of the supersonic business jet example could not be satisfied for certain combinations of the system variables. In fact, the pressure gradient depends only on t/c, a shared variable as seen in Figure 2. Therefore it is not possible to get a feasible design for BB2 if t/c is greater than 0.06.
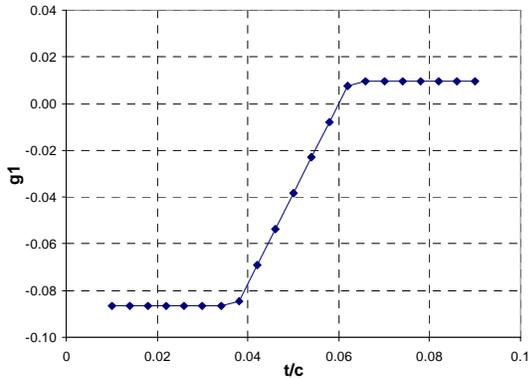

Figure 2: Pressure gradient constraint of the aerodynamic module (BB2).

We are considering two variations of the SBJ problem: a simplified version with the pressure gradient constraint excluded (SBJ-1) and the one counting for the constraint (SBJ-2). An approach is presented later in this paper to handle the infeasible BB optimization cases in a general way.

ModelCenter MDO framework was used and this exercise served to measure the current capabilities of ModelCenter against the requirements to deploy BLISS methodology. BB optimizations, surrogate model generation, and system optimizations are implemented in a fully automated manner in ModelCenter. Figure 3 shows a screenshot of ModelCenter that reflects the workflow of BLISS MDO.
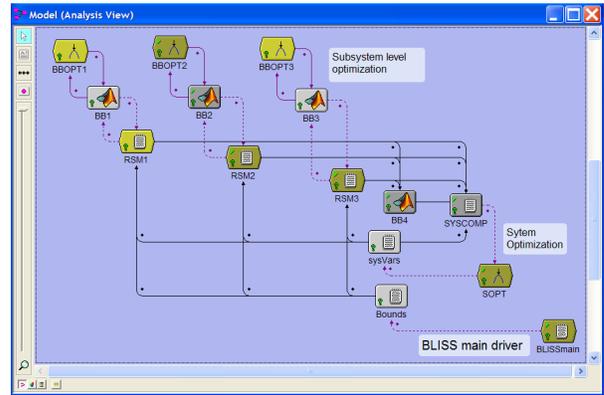

Figure 3: BLISS implementation of SBJ example in ModelCenter.

Comparison of Polynomial Model and Kriging Model in BLISS Application

Quadratic and linear polynomial model were applied to the simplified example, SBJ-1. Latin hypercube sampling (LHS)[17] design was used throughout this study. LHS has a good space filling characteristics that is suited to generating sample sites in a high dimensional space. Users are free to select the number of sites as desired. The minimum number of sites required for least squares fit increases quickly as the number of design variables increases for polynomial model. In this study twice of the minimum requirement was used to get reliable least squares fits. Table 2 summarizes the number of sites of the three BB modules.

Table 2: Number of runs used to fit polynomial response surface models.

	Number of variables in Q space	Number of sites for linear model	Number of sites for quadratic model
Structures	9	20	110
Aerodynamics	12	26	182
Propulsion	6	14	56

5

However, the BLISS process using the polynomial models did not converge well. The results were sensitive to initial starting points and BLISS did not find the known best solution of MDF formulation. Figure 4 and Figure 5 shows the convergence history of the system objective when linear or quadratic polynomial model was used, respectively. The predicted range via surrogate model was compared to actual range value from the computer simulation. The predicted range was not in satisfactory accuracy during the BLISS cycles. The quadratic model result shows faster improvement of the objective initially than linear model but shows larger oscillation later on.
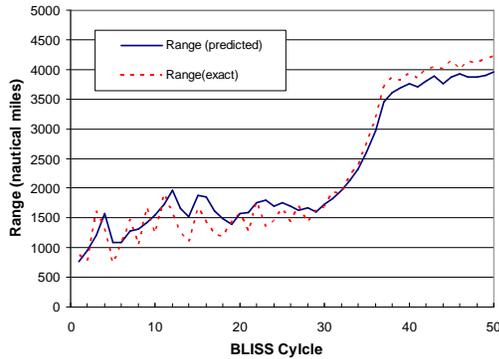
Figure 4: Convergence history of the system objective for SBJ-1 problem (linear polynomial model).
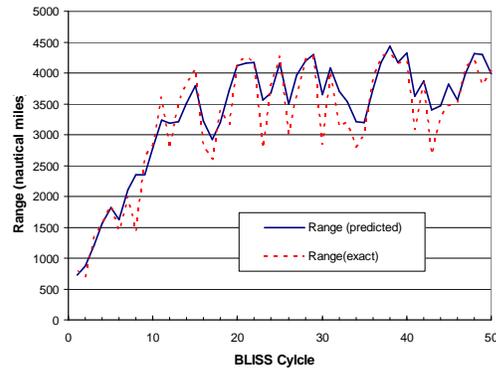

Figure 5: Convergence history of the system objective for SBJ-1 problem (quadratic polynomial model).

Because accuracy of the surrogate model is crucial to the success of the BLISS, quality of the fit of the polynomial model was checked. For each of the response surface model, 30 validation sites were generated randomly and the coefficient of the variance (COV) was calculated:

$$COV = RMSE / \bar{y}$$

$$= \sqrt{\frac{\sum_{i=1}^{nv}(y_i - \hat{y}_i)^2}{nv}} \Bigg/ \frac{\sum_{i=1}^{nv} y_i}{nv}, \qquad 5.a$$

where nv is the number of validation sites. COV of the polynomial models is plotted in Figure 6 and Figure 7 for three response variables: W_T (Y1 of BB1), D (Y2 of BB2), and SFC (Y1 of BB3), where the polynomial model had large errors compared to other responses. In both linear (Figure 6) and quadratic model (Figure 7), W_T had the largest COV level.
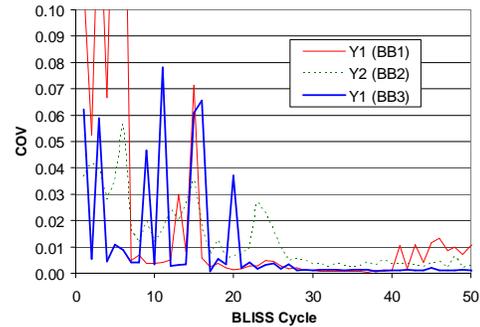

Figure 6: Coefficient of variance of linear polynomial model for SBJ-1 problem.
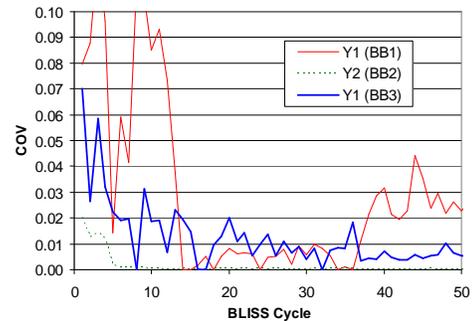

Figure 7: Coefficient of variance of quadratic polynomial model for SBJ-1 problem.

We are interested in whether using Kriging model would help improve the convergence of the BLISS cycles. Figure 8 shows that the BLISS process converged after 22 cycles when a Kriging model was used. In fact, the number of data points for the Kriging model was the same as that of the linear response surface model as shown in Table 2. It is remarkable that Kriging model produced success with reasonably small sample size, while polynomial model did not produce a converged solution with much larger sample size. The COV plot in Figure 9

shows that the error level of the Kriging model is consistently lower than that of polynomial results from Figure 6 and Figure 7. The maximum COV of the Kriging models was less than 0.003 after convergence. This explains why Kriging model was more successful. Table 3 summarized SBJ-1 optimization results. It shows that the BLISS process with Kriging model found essentially the same design as that of MDF formulation.

Table 3: Summary of BLISS results on the supersonic business jet example (SBJ-1).

Variable	Initial	Lower bound	Upper bound	MDF optimum	BLISS Optimum Linear RSM*	BLISS Optimum Quadratic RSM*	BLISS Optimum Kriging
t/c	0.05	0.01	0.09	0.09	0.07848	0.07975	0.08977
h	45000	30000	60000	60000	60000	59976	60000
$Mach$	1.6	1.4	1.8	1.4	1.4	1.402	1.4
AR	5.5	2.5	8.5	2.5	2.5	2.67	2.5
Λ	55	40	70	70	70	70	70
S_{ref}	1000	500	1500	1500	1401	1379	1500
λ	0.25	0.1	0.4	0.1122	0.1521	0.1227	0.1123
X	1.0	0.75	1.25	0.75	0.75	0.75	0.75
C_f	1.0	0.75	1.25	0.75	0.75	0.75	0.75
T	0.5	0.1	1.0	0.1562	0.1562	0.1566	0.1562
Range	535.8			4505	3957 (predicted) 4220 (exact)	3992 (predicted) 4043 (exact)	4484 (predicted) 4503 (exact)

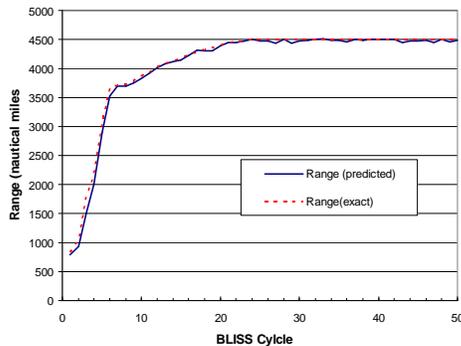(*: values at cycle 50, convergence was not achieved)

Figure 8: Convergence history of the system objective for SBJ-1 problem (Kriging model).
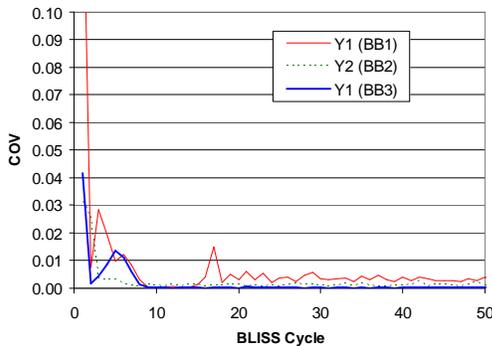

Figure 9: Coefficient of variance of quadratic polynomial model for SBJ-1 problem.

BLISS Strategy for Infeasible Subsystem Optimization

For the SBJ-1 problem, the pressure gradient constraint of BB2 was excluded because the BB optimization was not feasible when a shared variable, t/c, is greater than 0.06. If this constraint is included in the BLISS process, all of DOE runs of BB optimization for BB2 would fail when bounds of t/c fall on the infeasible domain. This prevents creation of surrogate model of the BB and BLISS process cannot progress.

A remedy is proposed that uses approximation models for constraint violation. When a BB optimization fails to generate a feasible design, mark the design as infeasible in the domain of the system optimization, and record the value of the violated constraint and continue performing the BB optimizations as usual. An approximation model is fitted to the constraint violation, which may be used to find the constraint feasibility boundary. An approximation model (g_a) will need to be built for each BB constraint that may be infeasible. The original formulation of BLISS uses only the compatibility constraints of coupled variables. Now, the feasibility constraint needs to be considered by the system optimization (Eq. 7.*e*).

Given: $Q = \{X_{sh}, Y^*, w\}$ 6.a
Find: $U = \{X_{loc}\}$ 6.b
Minimize: $f_k(U) = \Sigma\, w_i \hat{Y}_i$ 6.c
Satisfy: $g_k(U) \le 0$ 6.d
           $U_L \le U \le U_U$ 6.e

Using the BBOPT runs create surrogate models: $\hat{Y_a}$ (for coupled response $\hat{Y}$) and g_a (for each of infeasible local constraint g_k).

System Optimization (SOPT)

Find: $Q = \{X_{sh}, Y^*, w\}$ 7.a
Minimize: $f_k(Q) = f_k(\hat{Y_a})$ 7.b
Satisfy: $Y_a^* - \hat{Y_a} = 0$ 7.c
           $Q_L \le Q \le Q_U$ 7.d
           $g_a(Q) \le 0$ 7.e

It is important to note that the system optimization is still separated from the BB optimizations because surrogate models are used for the local constraints. Disciplinary groups are still in charge of performing BB optimizations independent of other groups or system optimization. Therefore, modus operandi of BLISS MDO is still valid with this modification; Disciplinary autonomy is maintained while approximation models are used to transfer information from the BB optimizations. Because the constraint violation may be irregular and non-smooth, there will be benefits in using flexible models such as Kriging over polynomial model for such violated constraints. This infeasible BBOPT handling approach was applied to the SBJ problem (SBJ-2). Only one local constraint, the pressure gradient of BB2, was considered for surrogate modeling. Because the pressure constraint depends only on the shared variable, it is natural to fit the constraint as a function of Q, including all cases of successful and unsuccessful runs of BBOPT. In other words, the constraint value at the end of unsuccessful BBOPT is reliable. However, if the constraint were a function of local variables, care should be taken in fitting surrogate models because the final constraint value returned at the end of unsuccessful optimization may be erratic.

Considering that polynomial model was not successful even when the problematic pressure gradient was excluded, only the Kriging model was tried to apply the infeasible BBOPT handling strategy. Again the same size DOE as the linear model was used. Figure 10 shows the convergence history of the system objective for SBJ-2 problem. It is observed a convergence was achieved after around 21 cycles. As

seen from Table 4, the BLISS solution found the same solution of the MDF approach. Note that the t/c value is exactly 0.06 where the constraint boundary is located.
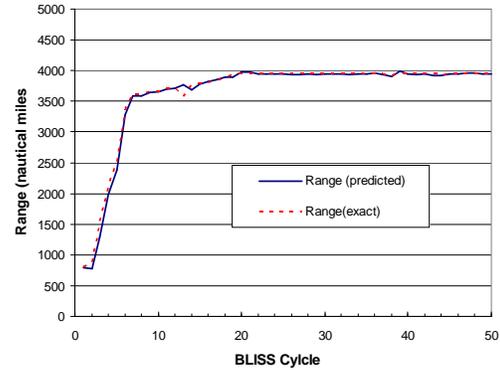

Figure 10: Convergence history of the system objective for SBJ-2 problem (Kriging model).

Table 4: Summary of optimization results for SBJ-2 problem.

Variable	MDF optimum	BLISS optimum	
		Kriging	Kriging (with data reuse)
t/c	0.06	0.06	0.06
H	60000	60000	60000
$Mach$	1.4	1.4	1.4
AR	2.5	2.5	2.5
Λ	70	70	70
S_{ref}	1500	1500	1500
λ	0.112	0.1119	0.1123
X	0.75	0.75	0.75
C_f	0.75	0.75	0.75
T	0.1562	0.1562	0.1562
Range	3958	3946 (predicted) 3958 (exact)	3951 (predicted) 3960 (exact)

Data Reuse Strategy to Reduce Computational Cost of Surrogate Model Building

As BLISS process progress, many BBOPT runs are available from previous cycles. So far, a brand new set of runs were generated for each cycle on the new search region of BLISS. But it is not difficult to include previous BBOPT runs when creating surrogate models on the new BLISS search region. If the previous database of BBOPT runs can be reused, it is expected to increase accuracy of surrogate models and reduce computational cost required performing BBOPT runs. Toward this approach, Kriging model has a capability

to refine its approximation by adding more data points into area of interests due to its interpolating nature.

This section presents a technique to systematically reuse previous BB optimization runs for Kriging in BLISS MDO. There are a few characteristics of Kriging we need to consider to devise such technique. First, computational cost of fitting Kriging model increases as more data points are included. The d-dimensional optimization problem to find Kriging coefficients, θ_k, may become prohibitively expensive if too many data points are used. Second, Kriging model may also experience ill-conditioning of the correlation matrix, **R**, when data points are located close to each other. Use of space filling DOE such as orthogonal array or LHS avoids this problem. However, when data points are reused, it is possible to have clustering of data points, as BLISS cycle progresses. Therefore, it is not always good to reuse all data points available. Balance should be taken between computational saving and quality of DOE samples when data points are reused.
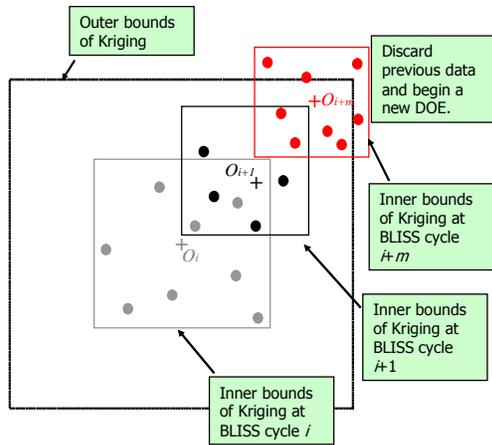

Figure 11: Data reuse strategy for Kriging.

Figure 11 illustrates a data reuse strategy for Kriging. Here we introduce two levels of bounds in Q space, inner bounds and outer bounds. Inner bounds of Kriging match the region defined by upper and lower bounds of BBOPT. The outer bounds define a limit where the inner region can move along BLISS cycles accumulating data points. For example, at cycle i, we have eight DOE design points within the inner bounds defined around the current optimum, O_i. Now in the next cycle, the BLISS solution moved to O_{i+1}, and new inner bounds are defined. We carry over all data points from cycle i, and because of that we can use less number of new data points. For example, we can use only half number of data points, i.e., four. Now we create Kriging model using the twelve accumulated data points from cycles i and $i+1$. Data points are

accumulated in this way until the inner bounds goes outside the outer bounds. If this happens at cycle $i+m$ (m is the number of cycles the data accumulation was done consecutively), we discard previous data points and generate a full size DOE with eight runs. The outer bounds are redefined around the new inner bounds at BLISS cycle $i+m$. The size of the outer bounds cannot be too big because it will cause ill-conditioning of correlation matrix, **R**, due to data clustering. In this study the size of the outer bounds was three times greater than the inner bounds for each dimension.
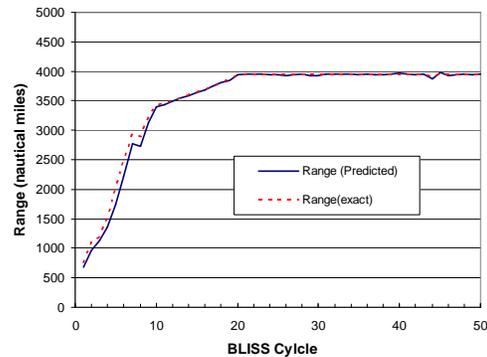

Figure 12: Convergence history of the system objective for SBJ-2 problem (Kriging model with data reuse strategy).
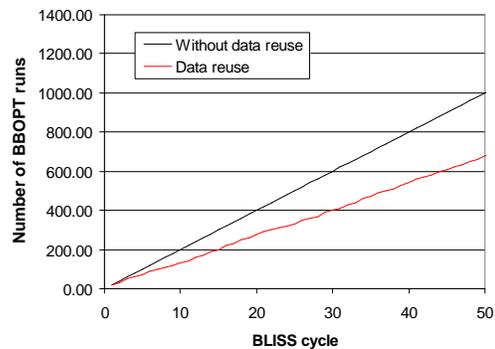

Figure 13: Number of BBOPT runs for BB1.

Figure 12 shows that the convergence history of SBJ-2 example when the data reuse strategy was used. Convergence was achieved about after 22 cycles and the MDF optimum was successfully found as shown in Table 4. Figure 13 shows the effect of the data reuse strategy on the number of BBOPT runs for BB1. The data reuse strategy reduced the number of BBOPT runs to 68% (=680/1000) after 50 cycles. For BB2 and BB3, the numbers have been reduced to 66% (=858/1300), and 63% (=441/700), respectively. Overall the data reuse strategy saved more than 30% of BBOPT runs.

Concluding Remarks

Building accurate approximation models is a key to the success of surrogate model based MDO techniques such as BLISS. BLISS decomposes a coupled system into sub-systems and recombines them in a systematic way using surrogate models of optimized sub-systems. In this work, a Kriging model was compared to polynomial RS models in the application of supersonic business jet (SBJ). It was seen the Kriging model was more accurate than polynomial model during BLISS process. Polynomial models experienced difficulties in convergence of BLISS cycles but Kriging model was able to find the known best solution with much less number of data points. However, this comparison between Kriging and polynomial model may not be interpreted as a general conclusion because accuracy of surrogate models is expected to depend on the nature of the BB modules being approximated. Therefore, it is important that BLISS offers complete autonomy in selection of surrogate models for each disciplinary module. Other alternatives such as radial basis function or neural network can be easily incorporated into BLISS as disciplinary experts find them suitable.

A modification of BLISS method was developed to handle cases where subsystem optimizations cannot find feasible designs. The approach uses surrogate models for the violated constraint and includes the constraint in the system optimization level. This technique was successfully demonstrated on the SBJ example where a violated constraint of the aerodynamics module depended only on a shared variable. Kriging model was very effective to model the non-linear constraint.

A strategy to reuse data points generated during the BLISS cycles was developed. It was possible to use less number of new design points by carrying over the data from previous cycles. Kriging model suited well to this technique because new data points can be added to area of interests to refine the approximation locally. It was shown that the data reuse strategy saved more than 30% of the total BB optimizations. Ref. 13 describes using a second Gaussian process to refine Kriging model locally. This technique has advantages that it can avoid ill-conditioning of correlation matrix and it is computational cheap, as well. For future work this technique will be compared to the simple data reuse strategy introduced here in BLISS applications.

The SBJ example was implemented within ModelCenter MDO framework. This work is a part of efforts to enhance capabilities to create and perform advanced MDO scenarios within ModelCenter. The Kriging model will be added to the surrogate model toolkit of ModelCenter. The enhancement will allow creating sophisticated MDO scenarios rapidly and perform BLISS design cycles in an effective manner.

References

1. Giunta, A. A., Balabanov, V., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T., "Multidisciplinary Optimisation of a Supersonic Transport Using Design of Experiments Theory and Response Surface Modelling," *Aeronautical Journal*, Vol. 101, 1997, pp. 347-356.

2. Chen, W., Allen, J. K., Schrage, D. P., and Mistree, F., "Statistical Experimentation Methods for Achieving Affordable Concurrent Systems Design," *AIAA Journal*, Vol. 35, No. 5, 1997, pp. 893-900.

3. Sobieszczanski-Sobieski, J., Altus, T. D., Phillips, M., and Sandusky, R., "Bi-Level Integrated System Synthesis for Concurrent and Distributed Processing," *AIAA Journal*, Vol. 41, No. 10, 2003, pp. 1996-2002.

4. Kim, H., Malone, B., and Sobieszczanski-Sobieski, J., "A Distributed, Parallel, and Collaborative Environment for Design of Complex Systems," AIAA Paper 2004-1848.

5. Phoenix Integration, Inc., "Improving the Engineering Process with Software Integration," a white paper, 2002. (See also www.phoenix-int.com).

6. Altus, T. D., "A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS)," NASA CR 2002-211652, 2002.

7. Sobieszczanski-Sobieski, J., Agte, J. S., and Sandusky, R. R., Jr., "Bi-Level Integrated System Synthesis (BLISS)," NASA TM 1998-208715, 1998.

8. De Baets, P., *A Methodology for Aeroelastic Constraint Analysis in a Conceptual Design Environment*, PhD Thesis, Georgia Institute of Technology, GA, 2004.

9. Brown, N., *Evaluation of Multidisciplinary Optimization (MDO) Techniques Applied to a Reusable Launch Vehicle*, Special Project Report, Georgia Institute of Technology, GA, 2004.

10. Myers, R. H., and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, Inc., New York, NY, 1995.

11. Sacks, J., Welch W. J., Mitchell, T. J., and Wynn, H.P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp.409-435, 1989.

12. Booker, A., "Design and Analysis of Computer Experiments," AIAA Paper 98-4757.

13. Audet, C., Dennis, J. E., Jr., Moore, D. W., Booker, A., and Frank, P. D., "A Surrogate-Model-Based Method for Constrained Optimization," AIAA Paper 2000-4891.

14. Giunta, A. A., and Watson, L. T., "A Comparison of Approximation Modeling Techniques: Polynomial versus Interpolating Models," AIAA Paper 98-4758.

15. Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization," AIAA Paper 98-4755.

16. Jin, R., Chen, W., and Simpson, T., "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria," AIAA Paper 2000-4801.

17. McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No.2, 1979, pp.239-245.