2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Feasible Parameter Space Characterization with Adaptive Sparse Grids for Nonlinear Systems Biology Models

Sarah L. Noble, Gregery T. Buzzard, and Ann E. Rundell

Abstract—**Mathematical models are commonly used to interrogate and control biological systems. However, such models are often uncertain and sloppy, with multiple parameter sets equally capable of reproducing the experimental data. These features make systems biology models unreliable when used to support a model-based control strategy. Multi-scenario control can help account for this uncertainty, but a computationally feasible method for characterizing all data-consistent regions of the global parameter space is necessary. Herein, we propose a tool for multi-scenario control in which sparse grid-based optimization is paired with a grid focusing algorithm to characterize acceptable regions of the uncertain parameter space. The grid focusing algorithm is first demonstrated on a test function before being applied within a multi-scenario control framework to an uncertain model of cell differentiation. The results show the algorithm's ability to identify disparate low-cost regions of the parameter space and selectively increase the grid resolution in these areas to help determine appropriate model scenarios for the multi-scenario controller. While particularly relevant to biological systems, this approach is broadly applicable to the control of any uncertain system.**

I. Introduction

The use of nonlinear mathematical models to provide insight into biological processes is well-established [1]. Increasingly, these models are being used to optimize or control the complex dynamics associated with these systems; however, unique challenges exist for the creation, refinement and utilization of systems biology models [1]. The ability of a model to accurately and reliably describe the underlying process dynamics depends on the model structure and the values of the model parameters. However, mathematical models of biological processes represent only simple abstractions of highly complex biochemical networks. As directly measuring model parameters can be difficult or impossible, parameters are generally fit to experimental data, which is usually sparsely sampled, corrupted with noise, and obtained only from a limited number of measurable model species. In addition, systems biology models are often "sloppy" as they contain sets of correlated parameters and can exhibit parameter sensitivities that span several orders of magnitude [2]. As a result, the model characterization is non-unique, with several sets of parameters equally capable of reproducing the available data.

This intrinsic uncertainty is particularly problematic when these models are relied upon to support a model-based control strategy. The resulting plant-model mismatch error limits the utility of the model predictions and undermines the success of the controller. Even within an adaptive control framework, where the model parameters are refined based on the plant feedback data to help alleviate mismatch error, the problem of non-uniqueness still exists. Relying on a single set of model parameters ignores the fact that multiple model characterizations exist given the available data and, when utilized in the controller, each characterization may predict a different set of control inputs. A multi-scenario control framework can ensure that the implemented control sequence is feasible for all data-consistent model characterizations; however, a parameter identification scheme is needed that can characterize all acceptable regions of the parameter space in a computationally feasible manner.

Adaptive sparse grids can be used to screen the global parameter space while limiting computational expense [3]. With this approach, the model is selectively evaluated at points in the parameter space, building a grid of model-evaluated support nodes. An error-controlled interpolated approximation for the cost function is created by combining basis functions at the support nodes [3-4]. This interpolating polynomial can be used as a surrogate for the model output to estimate the value of the cost function at parameter sets not evaluated by the model. Thus, unlike traditional stochastic optimization approaches, the sparse grid-based approach quantifies the behavior of the cost function over the entire uncertain parameter space so that disparate regions of acceptable parameter sets can be identified and considered. However, one drawback to utilizing grid-based techniques is that to increase the accuracy of the grid, additional points are added symmetrically across the global parameter space [4]. This approach wastes computational effort by placing support nodes in regions of the parameter space known to be unacceptable. Previous work developed a sparse grid focusing technique where smaller sparse grids were placed about some acceptable regions of the parameter space [5], but the determination of focused grid placement and size were *ad hoc* and must be formalized.

Herein, we present a tool for multi-scenario control in which sparse grid-based optimization is combined with an improved grid focusing algorithm to increase the grid resolution in all regions of low cost. The algorithm is described in Section II, and in Section III, we demonstrate

Manuscript received March 30, 2011.

S.L. Noble is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906

G.T. Buzzard is with the Mathematics department, Purdue University, West Lafayette, IN 47906

A.E. Rundell is with the Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN 47906 (e-mail: rundell@purdue.edu)

the algorithm on a test function as well as on an uncertain biological model. Conclusions and future work are described in Section IV.

II. METHODS

Typically, only a small region of the global parameter space will have acceptably low cost function values. We aim to identify these disparate regions of acceptability and increase the grid resolution in these areas only. To achieve this, we propose a grid focusing algorithm in which new, smaller sparse grids are placed in areas of low cost. This selective grid refinement will increase grid resolution, and therefore grid accuracy, in regions of interest while limiting "wasted" model evaluations in the unacceptable space.

The following sections will discuss the details of grid placement including the choice of center and range in each parameter direction. All sparse grids presented herein were generated using the Sparse Grid Toolbox for Matlab, version 5.1.1 [6] and utilized a Chebyshev grid construction.

A. Creation of Screening Sparse Grid

A screening sparse grid is used initially to identify acceptable regions of the parameter space. An acceptable parameter set is one whose cost function value is below a defined threshold. This threshold is problem- and cost function-dependent and quantifies how well a parameter set is expected to reproduce the experimental data.

The screening grid samples the entire uncertain parameter space, $\Omega \subset \Re^d$. We specify a maximum of $100d$ points initially. Grid points are iteratively added in sets of $50d$ until the estimated relative grid error (as calculated by the sparse grid toolbox) is less than 1% to ensure sufficient interpolating polynomial accuracy.

B. Determination of Focused Sparse Grid Center

Based on the results of the screening grid, areas of low cost must be identified. These represent local minima and are the regions in which we want to selectively increase the grid resolution; however, the identification of local minima on a multi-dimensional surface is not trivial. We propose two strategies: one which is very effective in low dimensions (~5 or less, depending on the number of grid points used), and one which is scalable to higher dimensions.

1) Low dimensions: Critical point method

In low dimensions, the local minima can be found through identification of all critical points of the interpolated cost function surface. As described in [7], the Lagrange form of the interpolating polynomial for the cost function values at the sparse grid nodes is converted to a generalized polynomial chaos (gPC) representation, which is a sum of products of one-dimensional orthogonal polynomials. In this case, we expanded the polynomial in terms of the Legendre polynomials, with a resulting polynomial:

$$f(x_1,...,x_n) = \sum_{\alpha} c_\alpha L_{\alpha_1}(x_1)...L_{\alpha_n}(x_n) \quad (1)$$

where $\alpha = (\alpha_1,...,\alpha_n)$ is a multi-index and $L_j(x)$ is the Legendre polynomial of degree j. Since the derivatives of the Legendre polynomials are known explicitly, we may take the gradient and Hessian of f directly from the formula above. Expanding the gradient in terms of the usual monomials, we apply the package HOM4PS-2.0 [8] to find the critical points of f.

To begin the focusing algorithm, the lowest cost critical point, $\mathbf{p}_{min}^{crit}$, is selected as a candidate center. Using the Hessian calculated previously, the concavity at this candidate point is evaluated and the point is classified. If the candidate point is a minimum, it is selected as the focused grid center. If the candidate point is not a minimum, it is thrown out, and the Hessian of the next lowest cost critical point is tested. This continues until the center of the first focused grid is placed. Once the size of this grid is determined (see Section IIC), all remaining critical points, $\mathbf{p}^{crit}$ (those which have not been thrown out), are checked to see if they lie within the grid. Any critical points that lie in the region spanned by the focused grid are thrown out. All remaining points are retained for subsequent iterations.

To place the subsequent focused grids, the lowest cost critical point remaining is selected as a candidate center. The process repeats as described above until no critical points remain for consideration. Thus, the number of focused grids placed in the global parameter space depends only on the topography of the cost function.

2) High dimensions: Acceptable point method

In higher dimensions, calculating the critical points becomes a memory-intensive task. In this case, we propose a semi-heuristic strategy in which sampled grid points are used to identify the local minima. To ensure a more thorough characterization of the uncertain parameter space, the screening grid interpolant is sampled with an additional $100d$ points using a Latin hypercube. All acceptable points are combined (those from the screening grid and those from the hypercube); this subset of points, $\mathbf{p}^{accept}$, will be used to locate the local minima in a manner similar to how the critical points were used previously. If there exist more than 100 acceptable points, the algorithm chooses the 100 lowest cost points to reduce computational time, but this point limit could easily be increased or removed.

To begin the focusing algorithm, the lowest cost acceptable point, $\mathbf{p}_{min}^{accept}$, is selected as a candidate center. Matlab's constrained optimization solver (*fmincon*) is applied to this point to find the nearest minimum, $\mathbf{q}_{min}^{accept}$. This extra minimization step is added because, unlike previously, the acceptable points are not necessarily critical points. The concavity of $\mathbf{q}_{min}^{accept}$ is then evaluated using the Hessian and the point is classified. If the point is truly a minimum, it is selected as the focused grid center. If the candidate point is not a minimum, it is thrown out, and the Hessian of the next lowest cost acceptable point is tested. This process continues until the center of the first focused grid is placed. Similar to the previous section, the remaining

points in $\mathbf{p}^{accept}$ are checked with the new focused grid, and those that are contained in the grid are thrown out.

To place subsequent grids, the lowest cost acceptable point remaining is selected as a candidate center. The local search is applied and the point is classified. If the point is a minimum, an additional check is made to see if it is contained in an existing focused grid. This check is necessary because the local search has modified the candidate center. While we previously verified that no acceptable points were contained in existing focused grids, the local search can tweak the candidate center such that this is no longer true. The algorithm continues, selecting the first distinct minimum as the focused grid center and eliminating any acceptable points that fall within this grid. As before, the process repeats until no acceptable points remain.

C. Determination of Focused Sparse Grid Range

The size of each focused grid is determined based on a quadratic approximation about the grid's center point, $\mathbf{p}_0$. We calculate the change in each parameter, Δp_i, that results in a 10% increase in the cost function:

$$1.1f(\mathbf{p}_0) = f(\mathbf{p}_0) + J_i(\mathbf{p}_0)\Delta p_i + \frac{1}{2}H_{ii}(\mathbf{p}_0)\Delta p_i^2 \qquad (2)$$

where J_i is the i^{th} element of the Jacobian and H_{ii} is the i^{th} diagonal element of the Hessian for $i = 1{:}d$. The range of the grid in the i^{th} parameter direction is:

$$range_i = \left\lfloor p_{0,i} - \Delta p_i^+ , p_{0,i} + \Delta p_i^+ \right\rfloor$$

where $p_{0,i}$ is the i^{th} element of the center point, and Δp_i^+ is the positive root of (2). The range of the focused grid is cutoff if it stretches outside Ω.

D. Creation of Focused Sparse Grid

As with the screening grid, we specify a maximum of $100d$ points for focused grid initially. Grid points are iteratively added in sets of $50d$ until the estimated relative grid error is less than 1%.

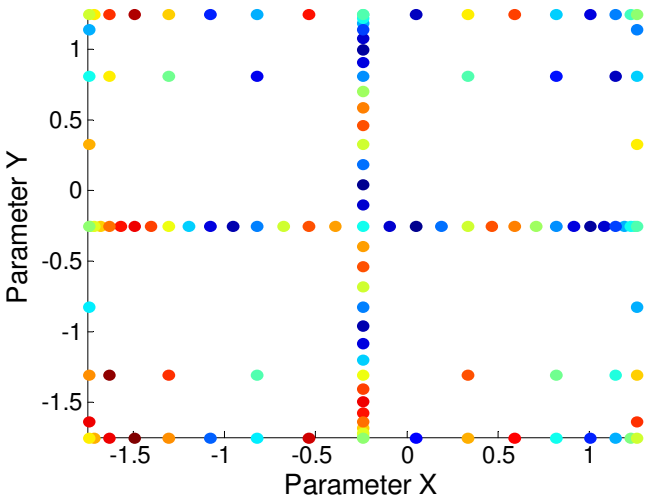

Fig. 1. Screening grid points for two dimensional modified Rastrigin's function. The grid placed 121 points to achieve an estimated relative grid error of 2.06e-4. The points are color coded by cost (red: high, blue: low).

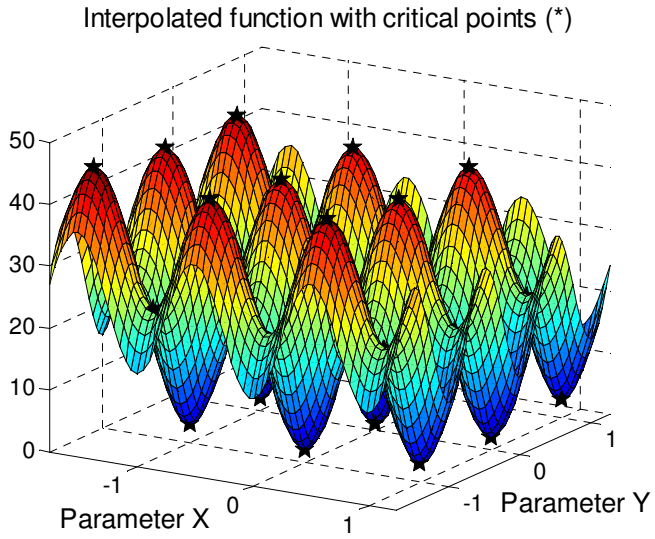

Fig. 2. Interpolated test function (modified Rastrigin's function) with the 36 critical points indicated by black stars. The critical points were found by converting the interpolating polynomial to a generalized polynomial chaos representation of Legendre polynomials. From these polynomials, the gradient and Hessian were directly computed and the critical points identified.

III. Results

A. Modified Rastrigin's Function in Two Dimensions – Critical Point Method

As a visual example, we use a modified Rastrigin's function:

$$h(\mathbf{x}) = 10d + \sum_{i=1}^{d}\left(x_i^2 - 10\cos(2\pi x_i)\right) + 1, \qquad -5.12 \le x_i \le 5.12 \qquad (3)$$

where d is the dimension, and the global minimum is $h(\mathbf{x}) = 1$ at $x_i = 0$ for $i = 1{:}d$. This nonlinear function exhibits cosine modulation and is highly multi-modal [9] (see Fig. 2).

In this example, the focusing algorithm was applied using the critical point method (Section IIB1) to place the focused grid centers. We applied the algorithm in two dimensions so the results could be visualized (in (3), $d = 2$). The screening grid is shown in Fig. 1, where the support nodes are color coded by cost function value. Note that the uncertain parameter space:

$$\Omega = \{x_i \,|\, x_i \in [-1.75, 1.25], i = 1,2\}$$

is not centered around (0,0). As the global minimum of Rastrigin's function is known to occur at this point, we did not want to bias the algorithm with the placement of the screening grid.

Fig. 2 shows the interpolated test function with the 36 identified critical points indicated by black stars. The focusing algorithm chose to place nine focused grids. Fig. 3 shows the points of both the screening grid and the nine focused grids superimposed over a contour plot of the test function for comparison. Table I compares the centers of the focused grids (in the order they were placed) with the known values of the local minima, as well as the calculated versus true cost of these points.

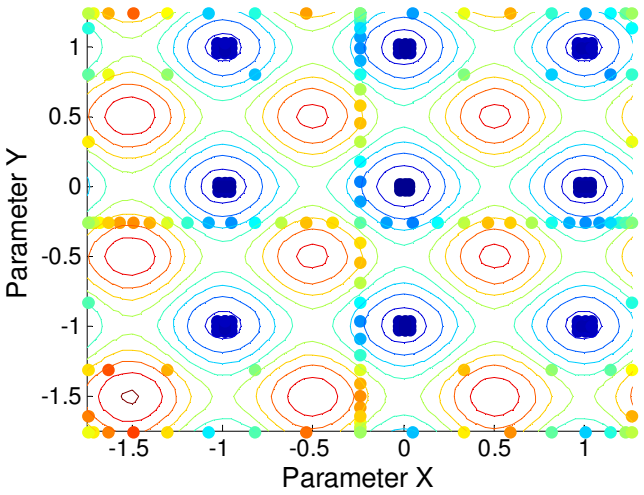

Fig. 3. Demonstration of placement of focused grids using the **critical point method**. The screening grid and focused grid points for the two dimensional modified Rastrigin's function have been plotted on top of the contour plot for the function. The focusing algorithm placed nine focused grids corresponding to the nine local minima of the test function.

From these results, the algorithm correctly identified the regions of low cost and placed focused grids about these regions. The algorithm identified the global minimum first followed by the next lowest cost minima. While shown on a test function to allow for comparison to true values, this approach is very effective (in low dimensions) at identifying regions of low cost in more complicated functions (results not shown) to enable selective grid refinement.

B. Modified Rastrigin's Function in Two Dimensions – Acceptable Point Method

We also applied the grid focusing algorithm to the modified Rastrigin's function using the acceptable point method (Section IIB2) to place the focused grid centers. We have applied it here in two dimensions for visualization, but the method scales to higher dimensions as well. For the test function, the acceptable threshold was chosen to be 15. In general, the choice of this threshold is not obvious and trial-and-error may be necessary to find a threshold large enough to produce a set of acceptable points while small enough to prevent considering the entire parameter space acceptable.

The screening grid is identical to that shown in Fig. 1. Following the procedure outlined previously, the algorithm found 64 acceptable grid points; local searches were applied to the acceptable points to place the focused grid centers. The focusing algorithm placed ten focused grids. Fig. 4 shows the points of the screening grid and the ten focused grids over a contour plot of the test function. Table II compares the centers of the focused grids (in the order they were placed) with the known values of the local minima, as well as the calculated versus true cost of these points.

TABLE I
CRITICAL POINT METHOD

Focused Grid Number	Focused Grid Center (x_1, x_2)	Calculated Cost	True Local Minimum (x_1, x_2)	True Cost
1	(0, 0)	1	(0, 0)	1
2	(0.995, 0)	1.995	(1, 0)	2
3	(0, 0.995)	1.995	(0, 1)	2
4	(0, -0.995)	1.995	(0, -1)	2
5	(-0.995, 0)	1.995	(-1, 0)	2
6	(0.995, -0.995)	2.989	(1, -1)	3
7	(0.995, 0.995)	2.989	(1, 1)	3
8	(-0.995, -0.995)	2.989	(-1, -1)	3
9	(-0.995, 0.995)	2.989	(-1, 1)	3

Comparison of focused grid centers with true local minima, as well as the calculated versus true cost of these points for the **critical point method**.

TABLE II
ACCEPTABLE POINT METHOD

Focused Grid Number	Focused Grid Center (x_1, x_2)	Calculated Cost	True Local Minimum (x_1, x_2)	True Cost
1	(-0.995, 0)	1.995	(-1, 0)	2
2	(0, 0.995)	1.995	(0, 1)	2
3	(-0.995, 0.995)	2.989	(-1, 1)	3
4	(0.995, -0.995)	2.989	(1, -1)	3
5	(-0.995, -0.995)	2.989	(-1, -1)	3
6	(0.995, 0)	1.995	(1, 0)	2
7	(0.995, 0.995)	2.989	(1, 1)	3
8	(0, -0.995)	1.995	(0, -1)	2
9	(0, 0)	1	(0, 0)	1
10	(0, -1.75)	14.062	N/A	N/A

Comparison of focused grid centers with true local minima, as well as the calculated versus true cost of these points for the **acceptable point method**.

From Fig. 4 and Table II, nine grids were placed about the nine local minima. A tenth grid was placed along the boundary at (0, -1.75). This highlights the tradeoff that comes with this focusing method. While scalable to higher dimensions, the method relies on sampling and local searches to find the areas of low cost. Depending on how the sampled acceptable points are arrayed in the global parameter space, the local search may lead to the placement of an extraneous grid. This can be prevented if the acceptable threshold is placed below the cost function value of these boundary points, but this type of insight is generally not available. Additionally, the minima are not found in order of increasing cost function value. The algorithm begins from the lowest cost acceptable point which may not fall in the trough of the global minimum.

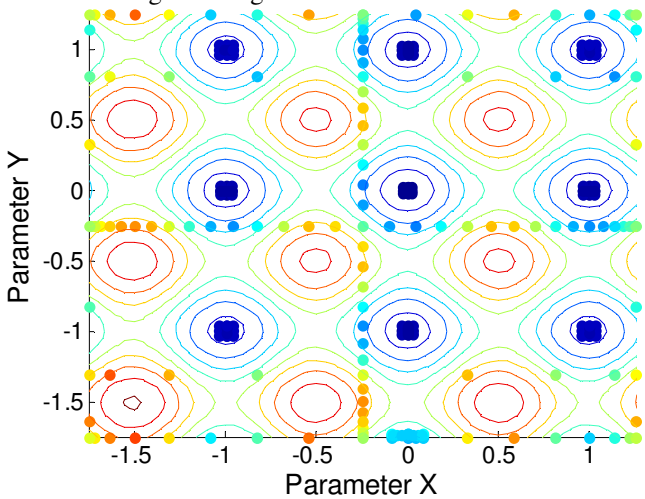

Fig. 4. Demonstration of placement of focused grids using the **acceptable point method**. The screening grid and focused grid points for the two dimensional modified Rastrigin's function have been plotted on top of the contour plot for the function. The focusing algorithm placed ten focused grids, with the tenth grid at a boundary point (0, -1.75).

C. Biological Model and Multi-Scenario Control

Herein, we applied the acceptable point method to a model of HL60 cell differentiation [10] to support a multi-scenario model predictive controller. The model consists of four ordinary differential equations and eight parameters. For the *in silico* controller implementation, the parameters of the plant model were established by randomly perturbing each value from its nominal value by ±30% according to a Gaussian distribution. Mock feedback data was created by adding realistic levels of Gaussian noise to the state variables (±5% [11]) and cell counts (±10% [12]).

As in [10], we aimed to control the percentage of mature granulocytes in the population through periodic additions of the differentiation-inducing chemical dimethyl sulfoxide (DMSO). Here we applied a multi-scenario model predictive control strategy in which all eight model parameters were refit based on the noisy plant feedback data.

The nominal parameter values, $\mathbf{p}_{nom}$, were assumed for the first controller iteration as feedback data is not yet available. For subsequent iterations of the controller, the sparse grid focusing algorithm sampled the eight-dimensional uncertain parameter space, constrained according to:

$$\Omega = \left\{ p_i \middle| p_i \in \left[0.5\,p_{nom,i} \quad 2\,p_{nom,i} \right], i = 1,...,8 \right\}.$$

Focused grids were placed about the regions of low cost. In this example, the cost function was the mean square error between the simulated data and "mock" experimental data:

$$g\left(\hat{\mathbf{x}}\right) = \left(\frac{1}{n_i} \sum_{i=1}^{n_i} \left[\hat{\mathbf{x}}\left(t_i \middle| \mathbf{p}\right) - \mathbf{x}\left(t_i\right) \right]^2 \right),$$

of sampled time points at time t_i (where n_i increases at each iteration with the addition of a new sample).

The acceptable parameter sets from the screening grid and all focused grids were combined. (For this example, the acceptable threshold was set to 20 based on trial and error). To find the representative parameter sets for the multi-scenario controller, these acceptable points were clustered into disjoint sets according to Euclidian distance using Matlab's hierarchical clustering routine (*cluster*). The lowest cost parameter set in each cluster was taken to be a scenario representative. In this application, we specified a maximum of three scenarios, with the maximum distance between points in a cluster iteratively increased until only three disjoint clusters were formed. The resulting representative parameter sets, $\mathbf{p}^{rep}$, were used by the multi-scenario controller to solve the input optimization problem.

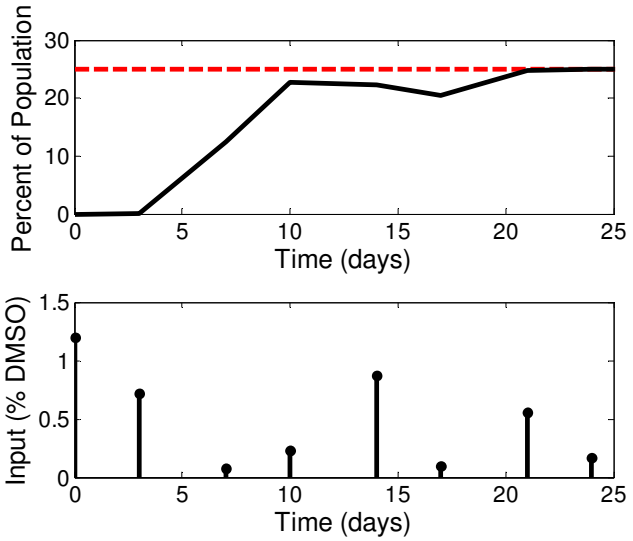

Fig. 5. Representative *in silico* controller results. The upper plot shows the time-course trajectory of the percentage of granulocytes (solid line). The lower plot shows the multi-scenario MPC-derived control inputs (%v/v).

TABLE III

		3-7 days	10-14 days	17-21 days
	Number of focused grids	0	2	1
Scenario 1	Parameter set	[0.38, 6.3e-3, **0.09**, 0.01, **6.06**, **1.14**, 1.72, **0.19**]	[**0.15**, **6.3e-3**, **0.21**, 0.01, **6.06**, **0.46**, **0.69**, **0.19**]	[**0.38**, **6.3e-3**, **0.21**, 0.01, **4.67**, **1.14**, **1.72**, **0.19**]
	Cost	1.52	11.53	18.40
	Control input	1.20	0	0.30
Scenario 2	Parameter set	[0.38, 6.3e-3, **0.34**, 0.01, **6.06**, **1.14**, 1.72, **0.19**]	[**0.38**, **6.3e-3**, **0.21**, 0.01, **5.35**, **1.14**, **1.72**, **0.30**]	[**0.33**, **6.3e-3**, **0.21**, 0.01, **6.06**, **1.14**, **1.72**, **0.19**]
	Cost	1.57	1.28	9.31
	Control input	0.66	0.63	0
Scenario 3	Parameter set	[0.38, 6.3e-3, **0.21**, 0.01, **4.04**, **0.46**, 1.72, **0.30**]	[**0.25**, **3e-3**, **0.18**, 0.01, **5.53**, **1.37**, **1.01**, **0.27**]	[**0.36**, **5.7e-3**, **0.27**, 0.01, **6.11**, **0.54**, **1.71**, **0.23**]
	Cost	0.11	3.79	2.94
	Control input	0.28	0.04	0
	Applied control input	0.71	0.22	0.10

Determination of multi-scenario control strategy in three time intervals for the *in silico* experiment shown in Fig. 5. In a given time interval, the parameter values that differ across the three scenarios are shown in bold. The applied input was found by averaging the inputs for each scenario.

For each representative parameter set, $\mathbf{p}_j^{rep}$, the input optimization minimized the least squares deviation of the predicted output trajectory for the percentage of granulocytes, $\hat{y}_j\left(k+i\middle|k\right)$, from the desired target, s, as well as the magnitude of the DMSO dose (control input, $u_j(\cdot)$):

$$\arg\min_{u_j} \left(Q \sum_{i=1}^{H_p} \left[\hat{y}_j\left(k+i\middle|k\right) - s \right]^2 + R \sum_{i=1}^{H_u} \left[u_j\left(k+i\middle|k\right) \right]^2 \right), \quad (4)$$

for $j = 1,...,n_j$ where H_p and H_u are the prediction and control horizon lengths, Q and R are weighting constants, and n_j is the number representative parameter sets. Here, H_p = 7 days, H_u = 7 days, Q = 10, R = 1, s = 25%, and n_j = 3. The controlled output was the percent of total cells that are mature granulocytes (state x_3), so the predicted output trajectory took the form:

$$\hat{y}_j\left(k+i\middle|k\right) = \frac{\left[0 \quad 0 \quad 1 \quad 0 \quad 0\right]\hat{\mathbf{x}}\left(t_{k+i}\middle|\mathbf{p}_j^{rep}\right)}{\left[1 \quad 1 \quad 1 \quad 1 \quad 0\right]\hat{\mathbf{x}}\left(t_{k+i}\middle|\mathbf{p}_j^{rep}\right)} \times 100$$

where $\hat{\mathbf{x}}\left(t_{k+i}\middle|\mathbf{p}_j^{rep}\right)$ represents the model-predicted state vector at each future sampling time for parameter set $\mathbf{p}_j^{rep}$. As in [10], Matlab's constrained optimization solver

(*fmincon*) was used to solve (4) subject to the control input constraint $u_j(k+i|k) \in [0,\ 1.2]$. To calculate the control input to be applied to the plant, we averaged the values of the first element of each control input vector, $u_j(1), \forall\ j$.

A sample *in silico* experiment is provided in Fig. 5. The upper plot shows the time-course trajectory of the percentage of granulocytes as directed by the multi-scenario model predictive controller. The lower plot shows the multi-scenario-derived control strategy. The controller drives the output to within 5% of the target trajectory by day 10 and subsequently sustains that level despite the uncertain parameters and the sparse and noisy feedback data.

Table III provides a glimpse into how the multi-scenario control strategy was derived for three different time intervals of the *in silico* experiment shown in Fig. 5 (all time intervals not shown). The number of focused grids placed during each interval is shown in the first row. The middle of the table shows the (eight-dimensional) representative parameter set, its cost, and the calculated control input for each scenario. Within a given time interval, the parameter values that differ across the three controller scenarios have been highlighted in bold. The final row shows the applied input, found by averaging the control inputs for each scenario. As shown in Table III, the lack of data early in the *in silico* experiment leads to significant variability in potential control inputs (spanning almost the entire feasible control input space). Averaging these inputs helps avoid biasing the applied input toward any particular model characterization. Later in the *in silico* experiment, when more data has been collected, the variation in the potential control inputs has decreased significantly with the averaged value more closely reflecting the preference of each scenario.

To test the sparse grid-based multi-scenario control strategy on a larger scale, we performed 100 *in silico* experiments (each with different randomly perturbed plant parameters). The controller performance was evaluated based on two metrics: steady state error and response time. The first metric quantifies achievement of the control objective by calculating the difference between the actual and desired steady-state granulocyte levels. Ninety-nine percent of *in silico* experiments achieved the control objective with no detectable steady-state offset (not shown). The second metric calculates the rise time, which is defined as the time needed to rise to within 5% of the desired steady state. Ninety-nine percent of *in silico* experiments exhibited a rise time of 3 weeks or less, while 94% had a rise time of 2 weeks or less (not shown).

From these results we see that the sparse grid-based multi-scenario control strategy effectively controlled this uncertain and highly abstracted model of cell differentiation. The grid focusing algorithm was applied in eight dimensions using the acceptable point method, helping to provide finer grid resolution in acceptable regions of the parameter space to aid in the selection of scenario-specific parameter sets.

IV. Conclusions and Future Work

This work presents a tool for multi-scenario control in which a sparse grid-based focusing algorithm screens the global parameter space and identifies disparate regions of acceptable parameter sets. The algorithm places smaller sparse grids in these regions to increase the grid resolution about potential local minima without wasting computational effort in the unacceptable regions of the parameter space.

We presented two versions of the grid focusing algorithm: one that is applicable only in low dimensions (critical point method) and one that is scalable to higher dimensions (acceptable point method). Each was demonstrated on a modified Rastrigin's function in two dimensions for ease of visualization. The grid focusing algorithm was also applied to a simple model of cell differentiation with eight unknown parameters to support a multi-scenario model predictive control strategy. The *in silico* controller implementation achieved the target level of cell differentiation despite the uncertainty in the model characterization.

Future work will continue to combine sparse grid-based optimization with the grid focusing algorithm for robust control strategy development. The scope will be expanded beyond model parameter identification to include characterization of the control input space with future applications including the robust control of multiple models.

V. Acknowledgements

This research was sponsored in part by the National Science Foundation (NSF) CAREER Award and funded under the American Recovery and Reinvestment Act of 2009 (ARRA) under the grant ECCS-0846572.

VI. References

[1] E. D. Sontag, "Molecular systems biology and control," *European Journal of Control,* vol. 11, pp. 396-435, 2005.

[2] R. N. Gutenkunst, *et al.*, "Universally sloppy parameter sensitivities in systems biology models," *PLoS Computational Biology,* vol. 3, pp. 1871-1878, Oct 2007.

[3] T. Gerstner and M. Griebel, "Dimension-adaptive tensor-product quadrature," *Computing,* vol. 71, pp. 65-87, Sep 2003.

[4] A. Klimke and B. Wohlmuth, "Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB," *ACM Transactions on Mathematical Software,* vol. 31, pp. 561-579, Dec 2005.

[5] M. M. Donahue, *et al.*, "Experiment design through dynamical characterisation of non-linear systems biology models utilising sparse grids," *IET Systems Biology,* vol. 4, pp. 249-262, 2010.

[6] A. Klimke. 2008, *Sparse Grid Interpolation Toolbox*.

[7] G. Buzzard, "Efficient basis change for sparse-grid interpolating polynomials," *submitted*.

[8] T. L. Lee, *et al.*, "HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method," *Computing,* vol. 83, pp. 109-133, Nov 2008.

[9] H. Pohlheim. (2006, *GEATbx: Example Functions (single and multi-objective functions) 2 Parametric Optimization*. Available: http://www.geatbx.com/docu/fcnindex-01.html#P140_6155

[10] S. L. Noble and A. E. Rundell, "Targeting a fixed percentage of granulocyte differentiation using experiments designed via nonlinear model predictive control," *2009 American Control Conference, Vols 1-9,* pp. 313-318, 2009.

[11] M. Buttarello and M. Plebani, "Automated blood cell counts - State of the art," *American Journal of Clinical Pathology,* vol. 130, pp. 104-116, Jul 2008.

[12] A. Richardson and S. Fedoroff, "Quantification of Cells in Culture," in *Protocols for Neural Cell Culture*, S. Fedoroff and A. Richardson, Eds., 3 ed: Humana Press, Inc., 2001.