Model-Based Diagnosis using Variable-Fidelity Modeling*

Gregory Provan
University College Cork, Cork, Ireland
e-mail: g.provan@cs.ucc.ie

Abstract

Due to the intractability of multiple-fault inference, MBD needs to use the most tractable models possible. There has been limited research into how to create models with a fidelity that trades inference complexity for accuracy. In this article we examine methods for diagnostics inference that use computationally efficient low-fidelity models (LFMs) whose performance is tuned using a high-fidelity model (HFM). More precisely, we use create an LFM augmented by an auxiliary function learned by comparing the performance of an HFM and LFM. This creates an inference system that has the reduced computational expense of the LFM but approximates the inference accuracy of the HFM.

1 Introduction

Model-Based Diagnosis (MBD) entails the use of a model, typically based on the physics of the underlying system, to diagnose faults given anomalous data from the system. Many types of model have been used, ranging from logic-based models [Reiter, 1987] to qualitative models [Struss, 1997] to ODE-based models [Ding, 2008]. These different model types make a range of assumptions about the system and the underlying faults. However, there has been no comprehensive comparison of the different approaches. Some simple comparisons have been done, e.g., [Sterling *et al.*, 2014], but there is no clear understanding of how the trade-offs of the different models, in terms of modeling language or fidelity, affect the resulting diagnostics accuracy.

Due to the intractability of multiple-fault inference, MBD needs to use the most tractable models possible. There has been limited research into how to trade model fidelity for inference complexity and accuracy. This need to explicitly trade model fidelity for computational tractability has been addressed in several other fields, e.g., statistical model-selection [Vehtari *et al.*, 2012], and engineering design optimization [Viana *et al.*, 2014]. This related work focuses on simulation models, with accuracy metrics based on simulation accuracy, e.g., mean-squared error (MSE). In contrast, MBD models have other requirements, e.g., capturing nominal and fault behaviors, and they have different metrics, e.g., fault-isolation accuracy, etc. [Sweet *et al.*, 2013]. Nonetheless, the techniques for using low-fidelity models that make limited accuracy trade-offs [Zhou *et al.*, 2016] can be useful for MBD.

One widely-used approach within design is the use of meta-models, or variable-fidelity models (VFMs). Variable-fidelity meta-modeling uses a high-fidelity model (HFM) to "tune" a low-fidelity model (LFM) [Viana *et al.*, 2014]. An HFM accurately describes the underlying physics of the system, but typically HFM inference entails high computational expense; examples of such models include ODE, finite element, or computational fluid dynamics representations. An LFM captures the most prominent characteristics of the system, but with lower computationally demands than that of an HFM. Our goal is to use the LFM with appropriate tuning to obtain adequate accuracy with low computational demands.

This article examines the accuracy with which we can use LFMs to approximate the diagnostics of a system by developing an "appropriate" tuning function. Our approximation approach should work for arbitrary system conditions, i.e., arbitrary initial conditions (or system observations) and failure scenarios. Given that it is computationally infeasible to guarantee this, we use Gaussian-Process (or Kriging) methods to develop a statistically unbiased estimator for the tuning function based on a finite set of system conditions [Kleijnen, 2009; O'Hagan and Kingman, 1978; Williams and Rasmussen, 2006].

Our contributions are:

1. We develop a framework for diagnostics inference that uses low-fidelity models (LFMs) tuned by a finite set of outputs of a high-fidelity model (HFM).
2. We empirically study the performance of this framework for a well-known tank benchmark system.
3. We empirically show that using an HFM to parameterize a LFM leads to an algorithm for which the tuned LFM provides "reasonable" inference accuracy but with lower computational costs than using the HFM alone.

2 Related Work

There has been limited work on using variable-fidelity models in MBD. One key paper, [Sachenbacher and Struss, 2005], differs with our work in framework and outcomes. Our definition of task concerns the nature of the diagnosis (e.g., it may be all diagnoses, a most-probable explanation, etc.) whereas [Sachenbacher and Struss, 2005] spec-

*The paper has been supported by SFI grants 12/RC/2289 and 13/RC/2094.

ify a task in terms of a behavior model (as composed from a component library), and specified granularity of the possible observations and of the diagnosis. In terms of outcomes, our approach is a strict generalization of [Sachenbacher and Struss, 2005], who vary fidelity only of the variable domains, whereas we can vary fidelity of all aspects of a model, including the (sub)set of variables, the variable domains, the class of relation, etc. In addition, our approach works with any type of model, whereas the approach of [Sachenbacher and Struss, 2005] addresses only qualitative models.

Related work in FDI using multiple models includes [Hanlon and Maybeck, 2000; Singh *et al.*, 2013]. These papers use one model for each fault, i.e., each model defines a single fault and tracks just that fault. However, this approach has scaling limitations, as computing all multiple-fault diagnoses requires an exponential number of "k-fault" models, $k = 1, ..., 2^n$ for n possible faults.

One approach to diagnosis model approximation has been to learn VFMs from a database of variable-fidelity component models [Feldman *et al.*, 2015]. This approach learns a model of a fidelity that optimizes diagnostic isolation accuracy. In contrast, this article uses model "linking" to make use of the computational advantages of the LFM, but acquires reasonable accuracy through tuning the LFM based on the HFM.

This work extends prior research on the use of meta-models (or surrogate models), which are designed to mimic the simulation behaviour of a system, but at a considerably reduced computational cost. Meta-models have been designed using many different approaches, including techniques such as Polynomial Response Surface (PRS) [Paduart *et al.*, 2010], Kriging [Kleijnen, 2009], Artificial Neural Networks (ANN), Radial Basis Function (RBF) [Bliznyuk *et al.*, 2012], and Support Vector Regression (SVR) [Kromanis and Kripakaran, 2013]. [Frangos *et al.*, 2010; Wang and Shan, 2007] presents a more detailed overview on several meta-modeling techniques. The same mathematical functions can be used as linking functions; see, e.g., [Razavi *et al.*, 2012; Wang and Shan, 2007; Zhao and Xue, 2010] for details of these techniques together with analyses of their advantages and disadvantages.

The fidelity (accuracy) of meta-models governs the computational cost and convergence characteristics of the meta-model-based inference. There is a clear trade-off between high accuracy and low computational expense. The fidelity of the meta-models directly depends on the number of scenarios simulated by the HFM, which are then used to tune the LFM. Generally, more scenarios (sample points) offer increased fidelity, but with higher cost [Shan and Wang, 2010]. Fewer sample points decreases computational costs but also inference accuracy.

This article adopts a global scaling method, i.e., it tunes the LF model using outputs from the HF model over all operating conditions for the model. There are two types of scaling method: (1) local VF meta-modeling approaches, which use local meta-models, e.g., linear regression [Chang *et al.*, 1993] or Taylor series [Eldred and Dunlavy, 2006]; and (2) global VF meta-modeling approaches, which use global meta-models, e.g., Gaussian processes [Williams and Rasmussen, 2006] or Kriging [Kleijnen, 2009].

3 Example: Tank Benchmark

This section introduces a tank benchmark system that we use for our experimental analysis.

3.1 High-Fidelity Model: Non-Linear Tank System

We illustrate our approach using the three-tank system shown in Fig. 1. Tank T_i has area A_i and inflow q_{i-1}, for $i = 1, 2, 3$. This system connects together three tanks, with a valve V_i regulating the flow q_i out of tank i, for $i = 1, 2, 3$.

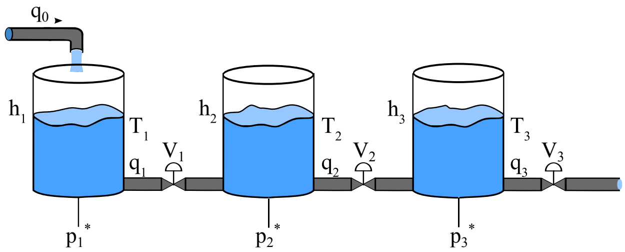

Figure 1: Diagram of the three-tank system.

Tank T_1 gets filled from a pipe, with measured flow q_0. Hence, our control input is $u = \{q_0\}$.

We assume that we don't directly measure any flows other than the inlet flow q_0. As a consequence, we use the tank heights as a proxy for deriving flows through the multi-tank system. We can control the valve settings $\boldsymbol{V} = \{V_1, V_2, V_3\}$, where we assume a continuous-valued setting ranging over $[0, 1]$, where 0 denotes closed and 1 denotes open.

According to Torricelli's Law, flow q_i out of tank i, with liquid level h_i, into tank j, is given by:

$$q_i = \gamma \; sign(h_i - h_j)\sqrt{2g(h_i - h_j)}, \tag{1}$$

where the coefficient γ is used to model the area of the drainage hole and its friction factor through the hole.

We can use equation 1 to derive the following equations:

$$\begin{aligned} \dot{h}_1 &= q_0 - c_1 \cdot V_1\sqrt{h_1 - h_2} \\ \dot{h}_2 &= c_1 \cdot V_1\sqrt{h_1 - h_2} - c_2 \cdot V_2\sqrt{h_2 - h_3}, \\ \dot{h}_3 &= c_2 \cdot V_2\sqrt{h_2 - h_3} - c_3 \cdot V_3\sqrt{h_3}, \end{aligned} \tag{2}$$

where the constants c_1, c_2, c_3 summarize the system parameters representing cross-sectional areas, friction factors, gravity, etc., and V_1, V_2, V_3 represent the valve settings (assuming a constant relationship between the setting and the flow). Consequently, the parameter set is given by $\Theta = \{c_1, c_2, c_3, g\}$. We assume that we can measure the height of liquid in each tank. The set of state variables is $\{h_1, \ h_2, \ h_3, q_0\}$, and the set of controllable variables is $\{q_0, V_1, V_2, V_3\}$.

3.2 Low-Fidelity Model: Linear Tank System

This section defines an LFM for the tank benchmark, using a linear version of the benchmark. To formally transform a non-linear system into a linear one, we need to use techniques like small signal linearization or perturbation theory [Sun, 2009; Taylor and Antoniotti, 1993]. For example, in small signal linearization, an equilibrium point x_0 of a fault-free non-linear function is first identified, about which the perturbed non-linear function is expanded: $x = x_0 + \delta x$. Then we can use a Taylor series expansion, neglecting the higher order terms, to obtain the linear function.

In this article, our objective is to use simplified low-fidelity models in order to see the trade-off of model fidelity (and associated computational simplicity) for diagnostics accuracy. As a consequence, we create a linear model simpler than that using small signal linearization, by ignoring the equilibrium points necessary for formal linearization and just replacing a non-linear function with a linear variant. In particular, we replace the non-linear flow equation (equation 1) derived using Torricelli's Law, with a linear version:

$$q_i = \gamma \; sign(h_i - h_j) 2g(h_i - h_j), \tag{3}$$

We can use equation 3 to derive the following equations:

$$\begin{aligned} \dot{h}_1 &= q_0 - c_1 \cdot V_1(h_1 - h_2) \\ \dot{h}_2 &= c_1 \cdot V_1(h_1 - h_2) - c_2 \cdot V_2(h_2 - h_3), \\ \dot{h}_3 &= c_2 \cdot V_2(h_2 - h_3) - c_3 \cdot V_3 h_3. \end{aligned} \tag{4}$$

Further, we can also define other LFMs that consist of mixed linear/non-linear models that contain both linear and non-linear representations of equation 1, e.g.,

$$\begin{aligned} \dot{h}_1 &= q_0 - c_1 \cdot V_1 \sqrt{h_1 - h_2} \\ \dot{h}_2 &= c_1 \cdot V_1(h_1 - h_2) - c_2 \cdot V_2 \sqrt{h_2 - h_3}, \\ \dot{h}_3 &= c_2 \cdot V_2(h_2 - h_3) - c_3 \cdot V_3 \sqrt{h_3}. \end{aligned} \tag{5}$$

We can specify an arbitrary tank model in terms of the exponents of the terms in the model, i.e., if we have a term denoted $(c_i - c_j)^\gamma$, this means that we have exponent γ, which if it is set to $\frac{1}{2}$ gives $\sqrt{(c_i - c_j)}$. Using this representation, we can define our model as

$$\begin{aligned} \dot{h}_1 &= q_0 - c_1 \cdot V_1(h_1 - h_2)^{\gamma_1} \\ \dot{h}_2 &= c_1 \cdot V_1(h_1 - h_2)^{\gamma_2} - c_2 \cdot V_2(h_2 - h_3)^{\gamma_3}, \\ \dot{h}_3 &= c_2 \cdot V_2(h_2 - h_3)^{\gamma_4} - c_3 \cdot V_3(h_3)^{\gamma_5}. \end{aligned} \tag{6}$$

In this way we can define a model in terms of $\boldsymbol{\gamma} = \{\gamma_1, ..., \gamma_5\}$. This representation allows us to define a model in terms of the parameter vector $\boldsymbol{\gamma}$ and generate alternative models with arbitrary values of $\boldsymbol{\gamma}$. In future work we plan to use this approach to generate "arbitrary" parameterized models rather than use models from a component library.

3.3 Approach

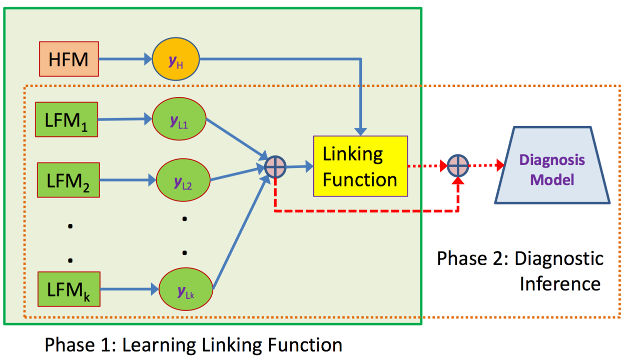

Figure 2: Architecture of variable fidelity approach.

Figure 2 shows the top-level architecture of our approach. This figure shows two phases: (1) learning the linking function and (2) computing the diagnostic model. We simulate data from each model $\boldsymbol{y}_.$, from which we compute the linking function ξ. The portion of the figure enclosed in a shaded box, Phase 1, shows taking simulation outputs from HFM model and the LFM models to compute the linking function. Phase 2 is shown enclosed in the dashed box. Here, we perform diagnostics inference using the linking function ξ and outputs from the LFM ensemble, so that the HFM is not used, leading to computationally more efficient inference than if we used the HFM.

Our objective is to compute a linking function that minimizes the error of using the LFM instead of the HFM, i.e., $\xi(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{y}_H - \boldsymbol{y}_L$, where $\xi(\boldsymbol{x}, \boldsymbol{\beta})$ is the linking function with parameters β. We estimate β using the outputs of several HFM and LFM simulations.

This approach can use any HFM and LFM; we can choose a wide variety of linking functions to learn, ranging from polynomials [Bianchi *et al.*, 2016] to Bayesian models [Bliznyuk *et al.*, 2012] and fuzzy membership functions [Bardossy *et al.*, 1990]. Once the linking function has been learned, inference uses only the computationally cheap LFM, so that the computationally expensive HFM does not have to used.

The computational complexity of the VFM approach, relative to that of the "standard" approach (using the HFM), depends on several factors. The first factor is the relative complexity of inference using the HFM and LFM models. The second factor is the computational expense of creating the linking function. Because the phase of creating the linking function (Phase 1) is done off-line, and typically once, the computational expense is amortized over the number of times diagnostics inference is done, which we term the on-line phase. We can trade off the computational expense of Phase 1 for the error ϵ induced by using the linking function. The error ϵ decreases with the amount of data (simulation runs of HFM and LFM) we generate for learning the linking function.

We now illustrate our approach with the tank benchmark. We use the benchmark model's output $\boldsymbol{y}$ as a means for developing a high-accuracy model approximation. We define a fault vector as the failure-state the set of three valves $\boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \zeta_3\}$, Our output measurement at any time step is denoted by $\boldsymbol{y} = \{h_1, h_2, h_3\}$, and the state vector is $\boldsymbol{x} = \{h_1, h_2, h_3, q_0, \boldsymbol{\zeta}\}$. Each simulation scenario is governed by the setting $\boldsymbol{x}(t)$ at simulation time t. For example, we may have a nominal scenario (S_1) with initial conditions of $h_1(0) = 1$, $q_0 = 1.5$, and all valve settings are 0.75 and non-faulty. Here, the valve health remains non-faulty. Scenario S_2 modifies S_1 by adding a fault in valve V_1 at time $t = 20$. For S_1 and S_2, we run the HFM and each LFM, and record $\boldsymbol{y}$ at selected time instants, which we denote as $\boldsymbol{y}^{0,q} = \boldsymbol{y}(0), ..., \boldsymbol{y}(q)$. We can then compute a linking function that minimizes an error function between $\boldsymbol{y}_H^{0,q}$ and $\boldsymbol{y}_L^{0,q}$ using S_1 and S_2. The on-line accuracy of VFM will be relatively high for scenarios similar to S_1 and S_2, but probably less high for different scenarios. As a consequence, it is important to train the linking function using scenarios similar to those we expect to encounter in real situations.

The VFM approach is applicable to any type of diagnosis scenario, e.g., steady-state, transient or intermittent faults, subject to having data to train for the fault type in question.

Steady-state faults require the least data, under the assumption that once the fault occurs it is of indefinite duration and effect. Transient and intermittent faults are more complex, in that the effect of the fault changes over time. As a consequence, the VFM approach needs significantly more training data to create a linking function that accurately diagnoses transient or intermittent faults than steady-state faults. In this article we focus on steady-state faults.

4 Variable-Fidelity Modeling

This section describes the approach in more detail. We show that we can use multiple LFMs in place of an HFM, in a manner analogous to that done in ensemble learning [Dietterich, 2000].

4.1 Model Representation

We represent a system $\mathcal{S}$ using a deterministic high-fidelity model (HFM), ϕ_H. We represent the observed variables in ϕ_H using $\boldsymbol{y}_H = f(\boldsymbol{x}_H)$, where $\boldsymbol{y}_H$ is a vector of observable values, and $\boldsymbol{x}_H$is a vector of state variables $\boldsymbol{x}_H = \{x_1, \cdots, x_n\}$. We assume that the system $\mathcal{S}$ can also be simulated with lower-fidelity models (LFMs), each of which represents a simplification of the HFM. We represent a LFM using ϕ_L: $\boldsymbol{y}_L = g(\boldsymbol{x}_L)$. We also assume that the HFM takes into account a larger number of variables, parameters and processes describing the physical system $\mathcal{S}$ than does any LFM; hence, inference using the HFM is likely to be more computationally demanding than that with the LFMs. More precisely, our assumption entails that the state vector of a lower-fidelity model, $\boldsymbol{x}_L$, is a subset of the state vector of the HFM: $\boldsymbol{x}_L \subseteq \boldsymbol{x}_H$. We represent a collection of lower-fidelity models using $\mathcal{Y} = \{\phi_{L^1}, \cdots, \phi_{L^k}\}$.

We model the relationship between the outputs from the HFM and from the lower-fidelity models using a mathematical function ξ, such that the output from the HFM can be written as:

$$\boldsymbol{y}_H = \xi(\mathcal{Y}, \beta) + \epsilon \tag{7}$$

where ξ is a mathematical function (denoted as a "linking function"), $\boldsymbol{\beta}$ is a vector of unknown parameters of the linking function, and ϵ is an error term. Equation 7 uses the linking function as a surrogate model to "link" outputs from models with different levels of fidelity. If we treat $\xi(\mathcal{Y}, \beta)$ as a random process, then we can describe this task in terms of a Gaussian Process [Kleijnen, 2009].

4.2 Defining the Linking Function

We can define the linking function using many mathematical representations. For situations where we have a complete physical model (e.g., a set of ODEs), then we may define the linking function using the physics of the problem. However, for more complex problems where there is no obvious relationship, we must compute the linking function empirically. In these latter scenarios, the linking function represents a response surface relating the corresponding outputs of the LFM and the HFM.

In this article we focus on polynomials, which we can specify with a wide variety of functional forms. Assuming the linking function ξ in Eq. 7 is a polynomial of degree n, then, for $\boldsymbol{y}_{Li} \in \mathcal{Y}$, we can define it as:

$$\begin{aligned}\xi(\boldsymbol{y}_{Li}, \boldsymbol{\beta}) &= \beta_0 + \sum_i \beta_i \boldsymbol{y}_{Li} + \sum_i \sum_{j>i} \beta_{ij} \boldsymbol{y}_{Li} \boldsymbol{y}_{Lj} \\ &\quad + \sum_i \beta_{ii} \boldsymbol{y}_{Li}^2 + \sum_i \sum_{j>i} \sum_{k>j} \beta_{ijk} \boldsymbol{y}_{Li} \boldsymbol{y}_{Lj} \boldsymbol{y}_{Lk} \\ &\quad + \cdots + \sum_i \beta_{ii \cdots i} \boldsymbol{y}_{Li}^n\end{aligned}$$

We compute the coefficients $\boldsymbol{\beta}$ of the polynomial using a least-squares solution of the equation $G\boldsymbol{\beta} = \boldsymbol{y}_H$, where G is a matrix operator, and $\boldsymbol{y}_H$ is a vector of outputs determined from a number m of runs of the HFM. We can define the maximum likelihood estimates of the coefficients as $\boldsymbol{\beta} = (G^T G)^{-1} G^T \boldsymbol{y}_H$. If we consider for simplicity three levels of fidelity, one high-fidelity model and two lower-fidelity models (LFM-1 and LFM-2), and we assume that the linking function can be written in the form of a second order polynomial ($n = 2$), we can define the matrix operator $\boldsymbol{G}$ as:

$$\boldsymbol{G} = \begin{bmatrix} 1 & \boldsymbol{y}_{L-1_1} & \cdots & \boldsymbol{y}_{L-2_1}^2 \\ 1 & \boldsymbol{y}_{L-1_2} & \cdots & \boldsymbol{y}_{L-2_2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \boldsymbol{y}_{L-1_m} & \cdots & \boldsymbol{y}_{L-2_m}^2 \end{bmatrix} \tag{8}$$

where $\boldsymbol{y}_{L-1_k}$ and $\boldsymbol{y}_{L-2_k}$, with $k = (1, , m)$, are the outputs from the k^{th}-run of the first and second lower-fidelity model, respectively.

4.3 Algorithm

We now describe the algorithm we use for model approximation. We assume that we have an HFM ϕ_H and a collection $\mathcal{L}$ of LFMs, $\{\phi_{L1}, ..., \phi_{Lk}\}$.

Step 1: Generate a sample m of input parameters for the HFM. From the generated input parameters sets, extract the subsets corresponding to each LFM.

Step 2: Perform m runs of the HFM and the LFM models with the generated samples of input parameters to calculate the vectors of outputs $\boldsymbol{y}_H = \boldsymbol{y}_H^1, ..., \boldsymbol{y}_H^m$ and $\boldsymbol{y}_{Li} = \boldsymbol{y}_{Li}^1, ..., \boldsymbol{y}_{Li}^m$ for each of the implemented models.

Step 3: Use the generated numerical data $\boldsymbol{y}_H$ and $\boldsymbol{y}_{Li}$ to identify a mathematical function (i.e., the linking function), representing the best match between outputs from the different models.

5 Experimental Analysis

5.1 Design

In this article we focus on linear and quadratic polynomial linking functions, with LFMs consisting of linear and mixed linear/non-linear models. We represent the polynomials in terms of the state vector $\boldsymbol{x} = \{h_1, h_2, h_3, q_0\}$. To simplify our notation, we denote $\boldsymbol{x}$ as an arbitrary k-vector.

We first address an additive framework, i.e., where we use a special case of equations 7 and 8 that takes the form

$$\boldsymbol{y}_H = \boldsymbol{y}_L + \xi(\boldsymbol{x}, \boldsymbol{\beta}), \tag{9}$$

where $\xi(\boldsymbol{x}, \boldsymbol{\beta})$ is the additive polynomial linking function. We can rewrite this as

$$\xi(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{y}_H - \boldsymbol{y}_L = \eta(\boldsymbol{y}_H, \boldsymbol{y}_L), \tag{10}$$

where η denotes the error the LFM makes in approximating the HFM.

We will obtain the best linear unbiased estimator if we compute the mean-squared error (MSE) such that

$$\operatorname*{argmin}_{\boldsymbol{\beta}} \xi(\boldsymbol{x}, \boldsymbol{\beta}) \;=\; \operatorname*{argmin}_{\boldsymbol{\beta}} MSE(\boldsymbol{y}_H - \boldsymbol{y}_L). \qquad (11)$$

If we assume a single LFM ϕ, for which we can generate output values of the state variables given by $\boldsymbol{x}$, then we can numerically estimate the linear polynomial by

$$\hat{\xi}(\boldsymbol{x}, \boldsymbol{\beta}) \simeq \beta_0 \;+\; \sum_{i=1}^{k} \beta_i x_i, \qquad (12)$$

where the $\beta_i,\ i = 0, ..., k$ are the weights to be learned.

The analogous quadratic polynomial is given by

$$\hat{\xi}(\boldsymbol{x}, \boldsymbol{\beta}) \simeq \beta_0 \;+\; \sum_{i=1}^{k} \beta_i x_i \;+\; \sum_{i=1}^{k} \sum_{j \geq i}^{k} \beta_{ij} x_i x_j \qquad (13)$$

where the β_i and β_{ij} are the weights to be learned. We can generalize this to incorporate multiple LFMs in a straightforward manner.

We run two sets of experiments: (1) simulation-focused experiments, to compute fault-free linking function representations, and (2) diagnosis-focused experiments, to compute linking function representations under conditions of systems faults. In the second case, we must augment our state vector $\boldsymbol{x}$ with a fault vector, which we assume is the set of three valves $\boldsymbol{\zeta} = \{V_1, V_2, V_3\}$, where a valve can fail stuck to a position different than the commanded position. We assume that a valve can take a continuous position, in a range $[0, 100]$.

Our linking functions cannot be valid over all possible scenarios (system conditions), i.e., all possible settings of initial conditions and fault-combinations, as this is an infinite manifold. As a consequence, we must select a subset of scenarios to construct the linking function. At each measured point, we minimize the difference between HFM ϕ_H and LFM ϕ_L in creating the linking function ξ. We then hope that the difference on all non-measured points will be small enough, within the region of interest.

Identifying the "best" scenarios for building the linking function is a complex process; further, this set of scenarios must be computationally tractable in number and computational difficulty. This is beyond the scope of this article, but is addressed in articles on optimal design of experiments, e.g., [Kleijnen, 2009; Wang and Shan, 2007]. In this article, we focus on steady-state approximations. This approach can also approximate transient conditions, but at a high computational cost in (1) identifying the transients, and in (2) creating linking functions for transients, which requires many sample points in the region of the transient.

Our experiments make the assumption that every set of initial conditions is equally likely, and we randomly sample a finite subset of the infinite space of initial conditions.[1] For the diagnosis-focused experiments, we assume a prior $P(\boldsymbol{\zeta}_0)$, and by assuming conditional independence of faults, can compute a prior over the k-fault scenarios based on $P_0(\boldsymbol{\zeta})$. Given this, we use likelihood sampling to generate fault scenarios based on their likelihood. We use these scenarios to generate simulation data, which is used for tuning the linking function representation.

We use the following metrics for simulation and diagnostics.

[1]We could also have a prior distribution over the likely initial conditions.

Definition 1. *The simulation accuracy of an LFM ϕ_L with respect to an HFM ϕ_H is computed using $MSE(\boldsymbol{y}_H, \boldsymbol{y}_L)$.*

For diagnostics accuracy, we derive a metric as follows. At each time t, we denote the true fault state $(\boldsymbol{\zeta}_t)$, and we compute the distribution over the fault state of the system, denoted $P(\hat{\boldsymbol{\zeta}}_t)$.

We compute the diagnostics accuracy of an LFM ϕ_L with respect to an HFM ϕ_H using the Kullback-Leibler (KL) divergence.

Definition 2. *If $P_H(\hat{\boldsymbol{\zeta}}_t)$ and $P_L(\hat{\boldsymbol{\zeta}}_t)$ denote the fault-state distributions for the high- and low-fidelity models, respectively, then the relative diagnostics accuracy of the HFM to the LFM is*

$$D_{\mathrm{KL}}(P_H \| P_L) = \sum_i P_H(i) \, \log \frac{P_H(i)}{P_L(i)}. \qquad (14)$$

5.2 Results

Simulation-focused Experiments

We have run experiments on the three-tank system where our measurement vector is the set of tank heights, $\tilde{\boldsymbol{h}} = \{\tilde{h}_1, \tilde{h}_2, \tilde{h}_3\}$. The experiments vary the inflow rate q_0, the initial tank heights, the outflow cross-section area a, and the valve settings $\boldsymbol{V} = \{V_1, V_2, V_3\}$.

We use as our measure of simulation accuracy the mean squared error (MSE) between the measured heights in the non-linear (HFM) and LFM models, i.e., $MSE(\tilde{\boldsymbol{h}}(\phi_H), \tilde{\boldsymbol{h}}(\phi_L))$.

We used OpenModelica v1.9.6 for running simulations and for computing the linking function representation parameters.

Figure 3 shows an example of a simulation in which $h_1(0) = 1$, $q_0 = 1.5$, and all valve settings are 0.75. This figure shows that the level of tank 1 increases linearly over time, as do the levels in tanks 2 and 3, but with successively slower rates. The mixed model in this example linearized the tank 1 equations. The figure also depicts the error in the linear model, which is dominated by the error in tank 1.

We fitted linear and polynomial linking functions to the LFM, and Figure 4 shows an example of a simulation depicting the MSE of the "corrected" LFMs with linear correction. This figure shows how the corrected LFMs can simulate the system with a fair degree of accuracy ($R^2 > 0.9$),with the correction improving the accuracy over time.

Diagnosis-focused Experiments

Our next set of experiments focussed on the fidelity of diagnostics. Two types of experiment were run: (1) using the "corrected" LFMs as the model for diagnosing injected faults (grey-box correction), and (2) using the output of a diagnosis engine running with the HFM as a "black box" that generates high-fidelity (correct) data, and correcting the diagnosis outputs of a black box diagnoser running with the LFM (black-box correction). We used Lydia-NG [Feldman *et al.*, 2013] as the diagnostics engine for our experiments.

Computing diagnoses using LFMs can lead to significant inaccuracies in diagnostics outputs. For example, Figure 5 shows an example of diagnosing the system using a non-linear and a linear model, where we introduce a fault in V_1 at time $t = 20$. The non-linear model diagnoses the fault perfectly, whereas for the linear model we get spurious diagnoses occurring at the start of the simulation, as well as throughout the simulation. Figure 5 shows a set of linear

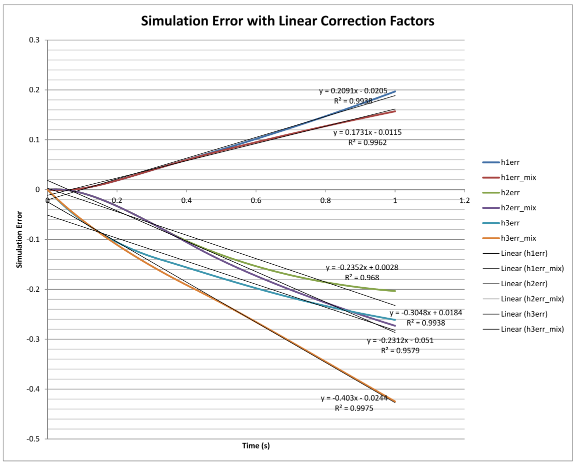

Figure 3: Relative error of simulation of tank benchmark with initial condition of $h_1(0) = 1$, $q_0 = 1.5$. Data shows results for linear model plus a mixed model containing a partial linearlization: for example, h1-err and h1-error-mix shows the error is height of tank 1, for the linear and mixed models respectively. Linear-· denotes the form of the linear correction factor, together with its R^2 fit.

correction equations, and Figure 6 shows a set of polynomial correction equations.

Figure 7 shows the corrected diagnosis probabilities over time, using the linear correction equations. The linear correction works relatively well for steady-state errors in diagnostics, but for the transient errors, it fails to correct the problems.

Our results indicate that polynomial correction works well for steady-state conditions, but not for transient issues. It is likely that the highly non-linear nature of ODE diagnostics inference with non-linear models (necessitating a non-linear inverse mapping) produces spurious diagnoses when using linear models. These spurious diagnoses will need techniques such as filtering (to smooth the spurious transients) or other types of linking functions beyond polynomial linking functions. These additional methods are the subject of ongoing research.

5.3 Discussion

This approach shows promise, as we obtain orders of magnitude speedups for steady-state diagnostics. This approach provides a principled methodology for making trade-offs in MBD, as we can now tailor "model quality" to constraints of real-world applications, e.g., inference time.

This work has limitations that require further investigation to relax. First, the approach should work for all model types in theory, but additional work is necessary on a wide collection of models in order to fully validate this claim. Second, we have focused on steady-state diagnostics, and this leads to large transient errors in diagnostics. Using higher-order polynomial correction reduces the speedups, but enables better accuracy. Removing the issues of spurious diagnostics will require better correction or filtering algorithms, but entails higher inference costs. In general, we can use a regularization framework [Neumaier, 1998], where one can penalize both diagnostics errors and inference costs, in order to make a proper trade-off. However, this would require a clear (domain-specific) knowledge of

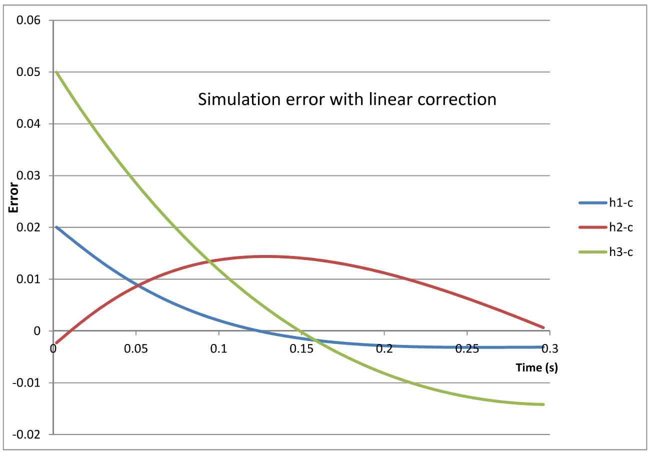

Figure 4: Corrected LFM simulation of tank benchmark, using linear linking function, with initial condition of $h_1(0) = 1$, $q_0 = 1.5$.

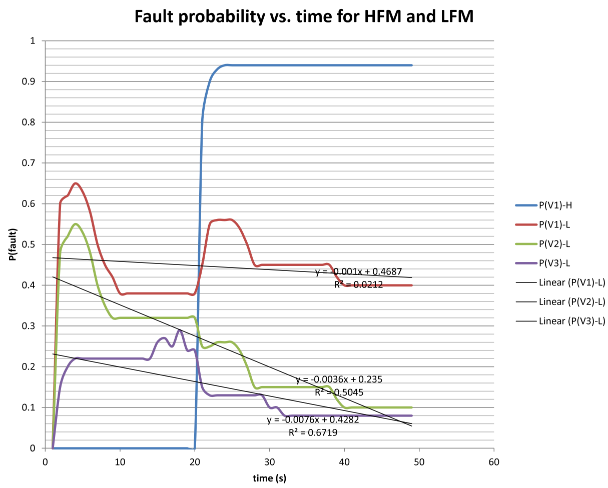

Figure 5: Diagnosis probabilities computed with fault injection at time $t = 20$, with linear correction factors. $P(V_1)$-H denotes the HFM data, and $P(\cdot) - L$ denotes the LFM data.

what trade-offs are desired.

6 Summary

This article has described the use of a novel variable-fidelity MBD approach that trades off inference complexity for accuracy. We have shown that a simple linear correction factor can obtain steady-state faults with high accuracy, but fails to remove spurious faults that occur when using a linearization of a non-linear system. This approach can reduce inference time by orders of magnitude for diagnosing non-linear systems. Additional research is necessary to define higher-accuracy LFMs or using filters in conjunction with LFMs to improve accuracy.

References

[Bardossy *et al.*, 1990] A Bardossy, I Bogardi, and WE Kelly. Kriging with imprecise (fuzzy) variograms. i:

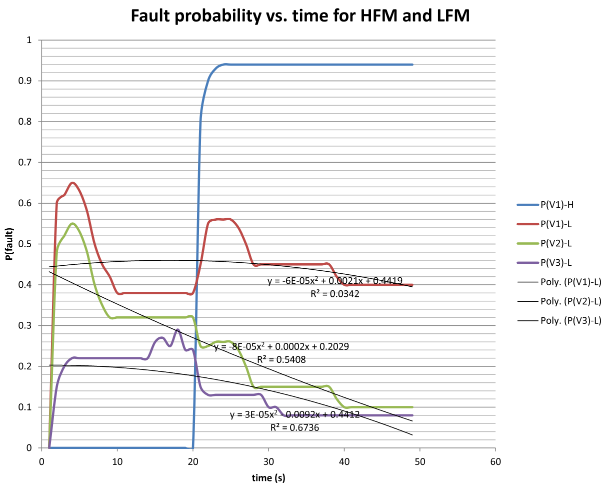

Figure 6: Diagnosis probabilities computed for an LFM, with fault injection at time $t = 20$, with 2^{nd} order polynomial correction factors

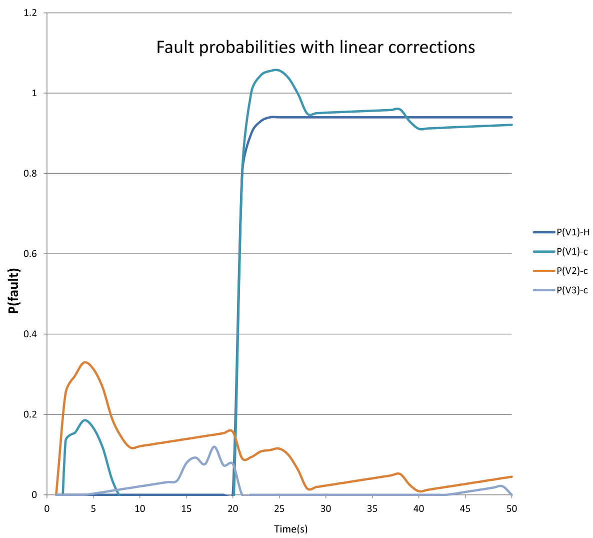

Figure 7: Corrected LFM Diagnosis probabilities, with fault injection at time $t = 20$

Theory. *Mathematical Geology*, 22(1):63–79, 1990.

[Bianchi *et al.*, 2016] Marco Bianchi, Liange Zheng, and Jens T Birkholzer. Combining multiple lower-fidelity models for emulating complex model responses for CCS environmental risk assessment. *International Journal of Greenhouse Gas Control*, 46:248–258, 2016.

[Bliznyuk *et al.*, 2012] Nikolay Bliznyuk, David Ruppert, Christine Shoemaker, Rommel Regis, Stefan Wild, and Pradeep Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 2012.

[Chang *et al.*, 1993] Kwan J Chang, Raphael T Haftka, Gary L Giles, and I-Jen Kao. Sensitivity-based scaling for approximating structural response. *Journal of Aircraft*, 30(2):283–288, 1993.

[Dietterich, 2000] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[Ding, 2008] Steven Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[Eldred and Dunlavy, 2006] MS Eldred and DM Dunlavy. Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, number AIAA-2006-7117, Portsmouth, VA*, volume 199, 2006.

[Feldman *et al.*, 2013] Alexander Feldman, Helena Vicente De Castro, Arjan Van Gemund, and Gregory Provan. Model-based diagnostic decision-support system for satellites. In *Aerospace Conference, 2013 IEEE*, pages 1–14. IEEE, 2013.

[Feldman *et al.*, 2015] Alexander Feldman, Gregory Provan, Rui Abreu, and Johan de Kleer. Learning diagnosis models using variable-fidelity component model libraries. In *Proc. Safeprocess Conference*, volume 48, pages 428–433. Elsevier, 2015.

[Frangos *et al.*, 2010] M Frangos, Y Marzouk, K Willcox, and B van Bloemen Waanders. Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems. *Large-Scale Inverse Problems and Quantification of Uncertainty*, 123149, 2010.

[Hanlon and Maybeck, 2000] Peter D Hanlon and Peter S Maybeck. Multiple-model adaptive estimation using a residual correlation Kalman filter bank. *Aerospace and Electronic Systems, IEEE Transactions on*, 36(2):393–406, 2000.

[Kleijnen, 2009] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[Kromanis and Kripakaran, 2013] Rolands Kromanis and Prakash Kripakaran. Support vector regression for anomaly detection from measurement histories. *Advanced Engineering Informatics*, 27(4):486–495, 2013.

[Neumaier, 1998] Arnold Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM review*, 40(3):636–666, 1998.

[O'Hagan and Kingman, 1978] Anthony O'Hagan and JFC Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

[Paduart *et al.*, 2010] Johan Paduart, Lieve Lauwers, Jan Swevers, Kris Smolders, Johan Schoukens, and Rik Pintelon. Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, 46(4):647–656, 2010.

[Razavi *et al.*, 2012] Saman Razavi, Bryan A Tolson, and Donald H Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012.

[Reiter, 1987] Raymond Reiter. A theory of diagnosis from first principles. *Artificial intelligence*, 32(1):57–95, 1987.

[Sachenbacher and Struss, 2005] Martin Sachenbacher and Peter Struss. Task-dependent qualitative domain abstraction. *Artificial Intelligence*, 162(1):121–143, 2005.

[Shan and Wang, 2010] Songqing Shan and G Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010.

[Singh *et al.*, 2013] Ashutosh Singh, Afshin Izadian, and Sohel Anwar. Fault diagnosis of Li-Ion batteries using multiple-model adaptive estimation. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 3524–3529. IEEE, 2013.

[Sterling *et al.*, 2014] Raymond Sterling, Gregory Provan, Jesús Febres, Dominic O'Sullivan, Peter Struss, and Marcus M Keane. Model-based fault detection and diagnosis of air handling units: A comparison of methodologies. *Energy Procedia*, 62:686–693, 2014.

[Struss, 1997] Peter Struss. Model-based and qualitative reasoning: An introduction. *Annals of Mathematics and Artificial Intelligence*, 19(3-4):355–381, 1997.

[Sun, 2009] Jian Sun. Small-signal methods for AC distributed power systems–a review. *Power Electronics, IEEE Transactions on*, 24(11):2545–2554, 2009.

[Sweet *et al.*, 2013] Adam Sweet, Alexander Feldman, Sriram Narasimhan, Matthew Daigle, and Scott Poll. Fourth international diagnostic competition–dxcâĂŹ13. In *Proc. of the 24th international workshop on principles of diagnosis*, pages 224–229, 2013.

[Taylor and Antoniotti, 1993] James H Taylor and Alfred J Antoniotti. Linearization algorithms for computer-aided control engineering. *Control Systems, IEEE*, 13(2):58–64, 1993.

[Vehtari *et al.*, 2012] Aki Vehtari, Janne Ojanen, et al. A survey of bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

[Viana *et al.*, 2014] Felipe AC Viana, Timothy W Simpson, Vladimir Balabanov, and Vasilli Toropov. Special section on multidisciplinary design optimization: Metamodeling in multidisciplinary design optimization: How far have we really come? *AIAA Journal*, 52(4):670–690, 2014.

[Wang and Shan, 2007] G Gary Wang and Songqing Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, 129(4):370–380, 2007.

[Williams and Rasmussen, 2006] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

[Zhao and Xue, 2010] Dong Zhao and Deyi Xue. A comparative study of metamodeling methods considering sample quality merits. *Structural and Multidisciplinary Optimization*, 42(6):923–938, 2010.

[Zhou *et al.*, 2016] Qi Zhou, Xinyu Shao, Ping Jiang, Zhongmei Gao, Chaochao Wang, and Leshi Shu. An active learning metamodeling approach by sequentially exploiting difference information from variable-fidelity models. *Advanced Engineering Informatics*, 30(3):283–297, 2016.