# Approaches for Creation and Evaluation of Computationally Efficient Thermofluid System Models

Brian P. Baillie<sup>\*</sup> Harish Ravichandar<sup>\*\*</sup> Iman Salehi<sup>\*\*</sup> Ashwin P. Dani<sup>\*\*</sup> George M. Bollas<sup>\*</sup>

 \* Department of Chemical & Biomolecular Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: george.bollas@uconn.edu).
 \*\* Department of Electrical & Computer Engineering, University of Connecticut, Storrs, CT 06269 USA

**Abstract:** Useful simulation of complex systems for real-time or otherwise computationally constrained applications often requires the creation of simplified system models from existing analytical models and data. In this work, the creation of candidate models through machine learning and heuristic-informed model simplification methods are explored, and the resulting candidate models are evaluated and compared through an optimal experiment design process. The dynamic system of interest is a counterflow air-to-water heat exchanger.

Keywords: Model reduction, Model approximation, System identification, Optimal experiment design, Heat exchangers

## 1. INTRODUCTION

Modeling of complex physical systems for real-time or otherwise computationally constrained applications often demands the approximation of system behavior with simplified models. Numerous methods exist for deriving approximate representations of nonlinear systems, including projection- and truncation-based model reduction (Moore, 1981; Chatterjee, 2000), heuristic simplification (Nilsson et al., 2006), surrogate modeling (Cozad et al., 2014), and statistical machine learning (Ghahramani and Roweis, 1999). Each approach for the creation of approximate models carries its own advantages and disadvantages. Heuristic and surrogate approaches require a relatively thorough understanding of the principal dynamics and behavior of the system of interest in order to ensure that key physics are properly represented in the resulting approximations. Approaches based on statistical modeling and machine learning can be effective for identification of systems without significant prior knowledge, but tend to be relatively computationally intensive and present challenges in algorithm selection and training data requirements.

Selection of a proper approximate model for a particular system and application requires a means for evaluating the performance of candidate models within the state space of interest. Such evaluation is typically performed through experiment by comparing the outputs of the candidate model to those of the true system or a more detailed original model for the same input conditions. These experiments are generally performed for arbitrary sets of nominal conditions, conditions at the outer limits of the input space, or for sets of randomly generated inputs. These experiments can be improved by the determination of input conditions that will readily identify the weaknesses of the candidate model. In this work, an optimal experimental design method is applied in order to select input conditions which maximize the error between the candidate and original models.

The methods applied in this work are generally applicable to nonlinear dynamic system models of the form (1), composed of differential and algebraic equations:

$$M \begin{cases}
F(\dot{X}(t), X(t), U(t), \hat{\Theta}, t) = 0, \\
F_0(\dot{X}_0, X_0, U_0, \hat{\Theta}, t_0) = 0, \\
Y(t) = H(X(t)),
\end{cases} (1)$$

wherein the full system model M consists of a set of equations F defining the state variables X, with time invariant parameters  $\hat{\Theta}$  and time varying inputs U. The additional equations F<sub>0</sub> and inputs U<sub>0</sub> define the initial conditions. The system output measurements Y are functions of the state variables. Simulation of the original model is subject to constraints on the parameters, inputs, and measurements as follows:

$$\hat{\Theta}_{i,min} \le \hat{\Theta}_i \le \hat{\Theta}_{i,max} \qquad i = 1, \dots, N_{\Theta} \qquad (2)$$

$$U_{min} \le U(t) \le U_{max} \qquad \forall t \in [0, \tau], \quad (3)$$

$$\mathbf{Y}_{min} \le \mathbf{Y}(t) \le \mathbf{Y}_{max} \qquad \forall t \in [0, \tau].$$
(4)

Model structures compatible with the form of (1) are frequently encountered in thermofluid systems applications. In this work, the particular system of interest is based on a model of a counter-flow air-to-water heat exchanger with vapor condensation, from the LBNL Modelica Buildings Library (Wetter, 2010; Baillie and Bollas, 2017a). The

<sup>\*</sup> This project was sponsored by the UTC Institute for Advanced Systems Engineering (UTC-IASE) of the University of Connecticut and the United Technologies Corporation. Any opinions expressed herein are those of the authors and do not represent those of the sponsor.



Fig. 1. Heat exchanger model, as seen in Dymola

principal temperature dynamics of the heat exchanger are defined by the following partial differential equations:

$$\mathbf{F} = \begin{cases} \frac{\partial T_a}{\partial t} = \frac{\dot{m}_a C_{pa} \frac{\partial T_a}{\partial z} - h_a A (T_a - T_w)}{V_a \rho_a C_{pa}} \\ \frac{\partial T_b}{\partial t} = \frac{\dot{m}_b C_{pb} \frac{\partial T_b}{\partial z} + h_b A (T_w - T_b)}{V_b \rho_b C_{pb}} \\ \frac{\partial T_w}{\partial t} = \frac{h_a A (T_a - T_w) - h_b A (T_w - T_b)}{M_w C_w} \end{cases}$$
(5)

where  $T_a$ ,  $T_b$ , and  $T_w$  are the air, water, and wall temperatures;  $h_a$  and  $h_b$  are the heat transfer coefficients for air and water sides;  $\dot{m}_a$  and  $\dot{m}_b$  are the mass flow rates of air and water;  $C_{pa}$ ,  $C_{pb}$ , and  $C_w$  are the specific heat capacities of air, water, and the wall;  $V_a$  and  $V_b$  are the volumes of the air and water flow paths,;  $\rho_a$  and  $\rho_b$  are the densities of air and water; A is the interface area of the wall; and  $M_w$  is the mass of the wall.

The partial differential equations (5) are axially discretized into ordinary differential equations in the original model. The heat transfer coefficients for both the water and air sides  $(h_a \text{ and } h_b)$  are algebraically determined from the inlet temperatures and flow rates. The default air and water models of the Modelica Buildings Library are used to determine the fluid properties, and the remaining parameters are user-defined. The full discretized model consists of 1127 total equations.

The inlet temperatures  $(T_{a_{in}} \text{ and } T_{b_{in}})$ , air mass flow rate  $(\dot{m}_a)$ , and inlet air relative humidity  $(\phi)$  are selected as the dynamic inputs to the system, while the outputs of interest are the outlet temperatures  $(T_{a_{out}} \text{ and } T_{b_{out}})$ , air outlet moisture fraction X, rate of energy accumulation  $E_{acc}$ , and the log mean temperature difference (LMTD).

#### 2. METHODS

Two methods for the creation of computationally simple system models are presented here. The first is heuristicdirected elimination of components from an original model. The second is a constrained machine learning method which guarantees convergence when applied to stable systems.

#### 2.1 Heuristic-directed Elimination

The objective of the elimination approach is to remove extraneous terms, variables, and equations from an original model (1) in order to produce a model of lower dimensionality or complexity in the form of

$$m \begin{cases} f(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{U}(t), \hat{\theta}, t) = 0, \\ f_0(\dot{\mathbf{x}}_0, \mathbf{x}_0, \mathbf{U}_0, \hat{\theta}, t_0) = 0, \\ \hat{\mathbf{y}}(t) = h(\mathbf{x}(t)), \end{cases}$$
(6)

wherein the simplified model takes the same inputs and retains the same output structure as the full model, with reduced sets of state variables, parameters, and equations. The simplified model is subject to constraints similar to those of the original:

$$\hat{\theta}_{i,min} \le \hat{\theta}_i \le \hat{\theta}_{i,max} \qquad i = 1, \dots, N_{\theta} \qquad (7)$$

$$U_{min} \le U(t) \le U_{max} \qquad \forall t \in [0, \tau], \quad (8)$$

$$\mathbf{y}_{min} \le \mathbf{y}(t) \le \mathbf{y}_{max} \qquad \forall t \in [0, \tau].$$
(9)

A suitable simplified model should satisfy the condition:

$$\|\mathbf{Y} - \mathbf{y}\| < \varepsilon \qquad \forall t \in [0, \tau], \mathbf{U} \in [\mathbf{U}_{min}, \mathbf{U}_{max}]$$
(10)

wherein the outputs produced by the reduced model match those produced by the full model within some acceptable error bound  $\varepsilon$ .

The heuristic algorithm applied herein is similar in principle to selective node elimination and other symbolic reduction methods developed for application to electronic circuits, as summarized in Haffmann and Wichmann (2003). The effectiveness of this approach and the quality of the resulting simplified model are highly dependent on the method applied for ranking and selecting the terms and components of the original model to be eliminated. This approach builds on the exhaustive term elimination method described previously (Baillie and Bollas, 2017b), with the inclusion of a timed output sensitivity based ranking method for the selection of components to eliminate. Variables, parameters, or equations which define key physical or empirical features of the system can be identified as ineliminable by the user. The original Modelica model is translated into a symbolic form using the JModelica platform (Åkesson et al., 2010) in order to allow the use of the CasADi (Andersson et al., In Press, 2018) tool to generate expressions for output sensitivities for each eliminable variable. The original model is augmented with these sensitivity expressions and simulated for a set of nominal experiments. Relative values of the output sensitivities are used to rank variables for elimination. Additional ranking heuristics based on component complexity, variable/output correlation, and variable incidence classification can also be applied to rank model components.

#### 2.2 Contracting Dynamical System Primitive

Numerous dynamical systems have important properties, such as convergence to equilibrium points (stability), that need to be retained by the models used to approximate them. While classical statistical approaches attempt to minimize modeling errors, they do not exploit this important piece of information in the learning process (Cohn et al., 1996; Quinonero-Candela et al., 2007). Algorithms using Lyapunov analysis to ensure the stability of nonlinear dynamical systems represented by GMMs are presented in (Khansari-Zadeh and Billard, 2011, 2014; Umlauft et al., 2017). In our earlier works (Ravichandar and Dani, 2015; Ravichandar et al., 2017), we developed an algorithm called contracting dynamical system primitive (CDSP) to learn the dynamics of stable nonlinear systems from data using statistical models. However, the



Fig. 2. Block diagram representation of the CDSP algorithm for learning stable nonlinear dynamical systems.

methods introduced in (Khansari-Zadeh and Billard, 2011, 2014; Umlauft et al., 2017; Ravichandar and Dani, 2015; Ravichandar et al., 2017), consider class of nonlinear dynamical systems without time-varying external inputs. Indeed, many physical systems, including thermofluid systems like the counter-flow heat exchanger, are modeled as a nonlinear dynamical system with external control inputs that exhibit multiple equilibria for different control inputs.

In this effort, we extend the CDSP method presented in (Ravichandar et al., 2017) to learn the dynamics of stable nonlinear systems with time-varying inputs and multiple equilibria. The state trajectories are modeled using an autonomous dynamical system of the form (1). The model function  $F(\cdot)$  is approximated using a Gaussian mixture model (GMM) and the problem of learning nonlinear dynamics is formulated as a parameter learning problem with constraints derived from partial contraction analysis of nonlinear systems (Wang and Slotine, 2005; Dani et al., 2015). The constraints developed using partial contraction analysis guarantee that the trajectories generated by the learned model accurately reproduce the available data, and converge to the appropriate steady-state from any initial condition. Since the steady-state value might not be readily known or cannot be identified analytically, a separate GMM is trained to predict the steady-state value given the initial conditions and the steady-state inputs. A block diagram, describing the overall workflow of the CDSP algorithm, is shown in Fig. 2.

A training dataset is produced from the original system, consisting of a range of output and input trajectories which converge to steady-state values. The steady-state value of the system states X(T) is assumed to be based on the initial state X(0) and the steady-state control input U(T). If the steady-state values do not converge to zero, they are translated to zero by creating new variables and writing the original dynamics in the new variables.

Estimating Steady-State Values using GMMs It is assumed that there exists a function that estimates the steady-state value  $X_{SS} \in \mathbb{R}^n$  given the initial output state X (0) and the steady-state control input U (T) as follows

$$\mathbf{X}_{SS} = \mathbf{F}_{SS} \left( \mathbf{X} \left( 0 \right), \mathbf{U} \left( T \right) \right) \tag{11}$$

where  $\mathcal{F}_{SS} \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$  is a continuously differentiable function. A statistical model, represented using a GMM, is used to approximate  $\mathcal{F}_{SS}(\cdot)$  and estimate the steady-state values

$$\hat{\mathbf{X}}_{SS} = \sum_{k=1}^{K_{SS}} h_{SS}^k \left( \boldsymbol{U}_{SS} \right) \left( \boldsymbol{A}_{SS}^k \boldsymbol{U}_{SS} + \boldsymbol{b}_{SS}^k \right)$$
(12)

where  $\boldsymbol{U}_{SS} = \left[ \mathbf{X} \left( 0 \right)^{T}, \ \mathbf{U} \left( T \right)^{T} \right]^{T}$  is the augmented input vector to the GMM, and  $h_{SS}^{k} \left( \boldsymbol{U}_{SS} \right) = \frac{\mathcal{P}(k)\mathcal{P}(\boldsymbol{U}_{SS}|k)}{\sum_{i=1}^{K_{SS}} \mathcal{P}(i)\mathcal{P}(\boldsymbol{U}_{SS}|i)}$ is the scalar weight associated with the kth Gaussian, such that  $0 \leq h_{SS}^{k} \left( \boldsymbol{U}_{SS} \right) \leq 1$  and  $\sum_{k=1}^{K_{SS}} h_{SS}^{k} \left( \boldsymbol{U}_{SS} \right) =$  $1, \ \mathcal{P}(k) = \pi_{SS}^{k}$  is the prior probability,  $\boldsymbol{A}_{SS}^{k}(x) =$  $\boldsymbol{\Sigma}_{\hat{X}_{SS} \boldsymbol{U}_{SS}}^{k} \left( \boldsymbol{\Sigma}_{SS}^{k} \right)^{-1}, \ \boldsymbol{b}_{USS}^{k} = \boldsymbol{\mu}_{\hat{X}_{SS}}^{k} - \boldsymbol{A}_{SS}^{k} \boldsymbol{\mu}_{USS}^{k}, \ \boldsymbol{\mu}_{SS}^{k} =$  $\left[ \left( \boldsymbol{\mu}_{U_{SS}}^{k} \right)^{T}, \left( \boldsymbol{\mu}_{\hat{X}_{SS}}^{k} \right)^{T} \right]^{T}$  and  $\boldsymbol{\Sigma}_{SS}^{k} = \left[ \begin{array}{c} \boldsymbol{\Sigma}_{\boldsymbol{y}_{SS}}^{k} \ \boldsymbol{\Sigma}_{\boldsymbol{U}_{SS}}^{k} \hat{\boldsymbol{\chi}}_{SS} \\ \boldsymbol{\Sigma}_{\hat{X}_{S} \boldsymbol{U}_{SS}}^{k} \ \boldsymbol{\Sigma}_{\hat{X}_{SS}}^{k} \end{array} \right]$ are the mean and the covariance of the kth Gaussian, respectively. The parameters of the CMM in (12) are learned

spectively. The parameters of the GMM in (12) are learned by solving a maximum likelihood problem using the expectation maximization (E-M) algorithm as in Dempster et al. (1977).

Learning Partially Contracting Dynamics using GMMs For designing the learning algorithm, the system is first written in terms of new variables such that the steadystate values of the state and the control input shift to zero, this reformulation is termed  $\overline{F}(\cdot)$ , with adjusted states  $\overline{X}(t)$  and adjusted inputs  $\overline{U}(t)$ . The nonlinear function  $\overline{F}(\cdot)$  is modeled using a separate GMM, and the resulting autonomous dynamical system is given by

$$\dot{\bar{\mathbf{X}}}(t) = \sum_{k=1}^{K} h^{k} \left( \boldsymbol{z}(t) \right) \left( \boldsymbol{A}^{k} \boldsymbol{z}(t) + \boldsymbol{b}^{k} \right)$$
(13)

 $\begin{aligned} & k=1 \\ \text{where } \boldsymbol{z}\left(t\right) = [\bar{\mathbf{X}}\left(t\right)^{T}, \ \bar{\mathbf{U}}\left(t\right)^{T}]^{T}, \ h^{k}\left(\boldsymbol{z}\right) = \frac{\mathcal{P}(k)\mathcal{P}(\boldsymbol{z}|k)}{\sum_{i=1}^{K}\mathcal{P}(i)\mathcal{P}(\boldsymbol{z}|i)} \\ \text{is the scalar weight associated with the kth Gaussian, such that } 0 \leq h^{k}\left(\boldsymbol{z}\right) \leq 1 \text{ and } \sum_{k=1}^{K}h^{k}\left(\boldsymbol{z}\right) = 1, \ \mathcal{P}\left(k\right) = \pi^{k} \text{ is the prior probability, } \boldsymbol{A}^{k} = \boldsymbol{\Sigma}_{\bar{\mathbf{X}}\boldsymbol{z}}^{k}\left(\boldsymbol{\Sigma}_{\boldsymbol{z}}^{k}\right)^{-1}, \ \boldsymbol{b}^{k} = \boldsymbol{\mu}_{\bar{\mathbf{X}}}^{k} - \boldsymbol{A}^{k}\boldsymbol{\mu}_{\boldsymbol{z}}^{k}, \\ \boldsymbol{\mu}^{k} = [\boldsymbol{\mu}_{\boldsymbol{z}}^{k}, \ \boldsymbol{\mu}_{\bar{\mathbf{X}}}^{k}]^{T} \text{ and } \boldsymbol{\Sigma}^{k} = \begin{pmatrix} \boldsymbol{\Sigma}_{\boldsymbol{z}}^{k} \ \boldsymbol{\Sigma}_{\boldsymbol{z}}^{k} \\ \boldsymbol{\Sigma}_{\bar{\mathbf{X}}\boldsymbol{z}}^{k} \ \boldsymbol{\Sigma}_{\bar{\mathbf{X}}}^{k} \end{pmatrix} \text{ are the mean and the covariance of the kth Gaussian, respectively. The second second$ 

and the covariance of the kth Gaussian, respectively. The constrained optimization problem to be solved in order to train the GMM model in (13) can be written as

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{M}} = \arg\min_{\boldsymbol{\theta}, \boldsymbol{M}} \frac{1}{2\mathcal{T}} \sum_{n=1}^{N} \sum_{t=0}^{T_n} \|\hat{\bar{\mathbf{X}}}_n(t) - \dot{\bar{\mathbf{X}}}_n(t)\|^2$$
(14)

s.t. 
$$\left(\boldsymbol{A}_{\bar{\mathbf{X}}}^{k}\right)^{T}\boldsymbol{M} + \boldsymbol{M}\boldsymbol{A}_{\bar{\mathbf{X}}}^{k} + \gamma\boldsymbol{M} \leq 0, \ k = 1, ..., K,$$
 (15)

$$b^{\kappa} = 0, \ k = 1, ..., K,$$
 (16)  
 $M \succ 0$  (17)

$$l \succ 0$$
 (17)

$$\Sigma^{n} \succ 0, \ k = 1, \dots, K, \tag{18}$$

$$0 \le \pi^k \le 1, \ k = 1, ..., K, \tag{19}$$

$$\sum_{k} \pi^{k} = 1, \tag{20}$$

where  $\boldsymbol{\theta} = \{\boldsymbol{\mu}^1...\boldsymbol{\mu}^K, \boldsymbol{\Sigma}^1...\boldsymbol{\Sigma}^K, \pi^1...\pi^K\}$  is a vector containing the parameters of the GMM model,  $\mathcal{T} = \sum_{n=1}^N T_n$ is the total number of data points in the training data and  $\hat{\vec{X}}_n(t) = \sum_{k=1}^K h^k(\boldsymbol{z}(t)) (\boldsymbol{A}^k \boldsymbol{z}(t) + \boldsymbol{b}^k)$  is the predicted state derivative computed based on (13). The constraints (15)-(17) ensure the global attraction to the steady-state  $0_{n\times 1}$  and the constraints in (18)-(20) are a result of using a GMM to model the dynamics. Proof of the convergence to steady state can be found in (Ravichandar et al., 2018).

The CDSP and other statistical frameworks do not require modeling every state variable of a particular system in order to sufficiently learn the behavior of the outputs of interest. In this case, the GMM models trained for the heat exchanger model only consider the outputs themselves, resulting in a significantly reduced number of states.

#### 2.3 Model discrimination and selection

Optimal experimental design methods are intended to maximize the information that can be extracted from an experiment or series of experiments by minimizing the variance of estimators according to a design criterion. In the case of discriminating between two or more models of similar structure, the T-design criterion can be used to find the optimal conditions for differentiating between models while considering the entire state space (Atkinson and Fedorov, 1975a,b). Similar optimal design approaches have been applied to determine optimal experiments for selection between competing candidate chemical kinetics models (Han et al., 2016a,b; Atkinson et al., 1998) and to design test procedures to identify fault conditions in thermofluid systems (Palmer et al., 2016).

The T-design criterion used in this work is as defined in (21), subject to the same simulation constraints as the models of 1 and 6 .

$$\Phi_T = \max_{U(t)\in\mathbb{R}} \min_{\theta\in\mathbb{R}} \int_0^\tau \left( \hat{\mathbf{Y}}(t) - \hat{\mathbf{y}}(t) \right)^\mathsf{T} \mathbf{W}_{\mathbf{y}} \left( \hat{\mathbf{Y}}(t) - \hat{\mathbf{y}}(t) \right) dt$$
(21)

Subject to:

J

$$\mathbf{F}(\dot{\mathbf{X}}(t), \mathbf{X}(t), \mathbf{U}(t), \hat{\Theta}, t) = 0, \qquad (22)$$

$$F_0(\dot{X}_0, X_0, U_0, \hat{\Theta}, t_0) = 0, \qquad (23)$$

$$\hat{\mathbf{Y}}(t) = \mathbf{H}\big(\mathbf{X}(t)\big),\tag{24}$$

$$f(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{U}(t), \hat{\boldsymbol{\theta}}, t) = 0,$$
(25)

$$f_0(\dot{x}_0, x_0, U_0, \hat{\theta}, t_0) = 0, \qquad (26)$$

$$\hat{\mathbf{y}}(t) = \mathbf{h}\big(\mathbf{x}(t)\big),\tag{27}$$

$$\hat{\theta}_{i,min} \le \hat{\theta}_i \le \hat{\theta}_{i,max} \qquad i = 1, \dots, N_{\theta} \qquad (28)$$

$$\mathbf{J}_{min} \le \mathbf{U}(t) \le \mathbf{U}_{max} \qquad \forall t \in [0, \tau].$$
(29)

In this formulation,  $\arg \Phi_T$  is the design vector of the experiment, consisting of the set of input trajectories U(t)and adjusted parameters for the candidate reduced model  $\theta$  determined by the optimization. The experiment duration  $\tau$ , sampling rate, and original model parameters  $\Theta$ are specified by the user. An optimal design vector is one which satisfies the objective of (21) by maximizing (w.r.t. input trajectories) the minimum (w.r.t. candidate model parameters) of the integral sum of squares of the weighted or relative error in the outputs of the candidate model  $\hat{y}$  as compared to those of the original  $\hat{Y}$ . Minimizing the error with regard to the parameters of the candidate model allows for the adjustment of a subset the reduced model parameters to compensate for the missing information between the two models. Maximizing the error with regard to the input trajectories seeks to identify and evaluate

the worst-case performance of the candidate model in representing the behavior of the original model within the specified input space. For this work, this method has been implemented in Matlab/Simulink to allow for analogous comparison of candidate models produced by either the heuristic elimination or CDSP algorithms. Solution behavior of the design problem for the T-optimal experiment is highly dependent on the size and complexity of the system, as well as the number of inputs, adjustable parameters, and outputs considered. Solution uniqueness depends on the convexity properties of (21) and is targeted by use of multi-start algorithms.

Table 1. Input conditions for each experiment

Heuristic, Nominal							
Input	Low Bd	Up Bd	Initial	Final			
$T_{a_{in}}, (^{\circ}C)$			40	35			
$T_{b_{in}}, (^{\circ}C)$			20	30			
$\dot{m}_a$ , (kg/s)			0.3	0.4			
$\phi_{in}$			0.3	0.5			
Heur	istic, Optin	nal, Low H	Iumidity				
Input	Low Bd	Up Bd	Initial	Final			
$T_{a_{in}}, (^{\circ}C)$	5	50	5	50			
$T_{b_{in}}, (^{\circ}C)$	5	40	40	5			
$\dot{m}_a$ , (kg/s)	0.1	0.55	0.1	0.55			
$\phi_{in}$	0.01	0.3	0.01	0.3			
Heuristic, Optimal, Full Range							
Input	Low Bd	Up Bd	Initial	Final			
$T_{a_{in}}, (^{\circ}C)$	5	50	50	50			
$T_{b_{in}}, (^{\circ}C)$	5	40	5	5			
$\dot{m}_a$ , (kg/s)	0.1	0.55	0.1	0.1			
$\phi_{in}$	0.01	0.6	0.6	0.6			
CDSP GMM, Nominal							
Input	Low Bd	Up Bd	Initial	Final			
$T_{a_{in}}, (^{\circ}C)$			40	35			
$T_{b_{in}}, (^{\circ}C)$			20	30			
$\dot{m}_a$ , (kg/s)			0.3	0.4			
$\phi_{in}$			0.3	0.5			
CDSP GMM, Optimal, Constrained Inputs							
Input	Low Bd	Up Bd	Initial	Final			
$T_{a_{in}}, (^{\circ}C)$	10	25	10.5	10			
$T_{b_{in}}, (^{\circ}C)$	10	25	10	25			
$\dot{m}_a$ , (kg/s)	0.2	0.4	0.2	0.2			
$\phi_{in}$	0.2	0.4	0.265	0.2			
CDSP GMM, Optimal, Full Range							
Input	Low Bd	Up Bd	Initial	Final			
$T_{a_{in}}, (^{\circ}C)$	5	50	5	5			
$T_{b_{in}}, (^{\circ}C)$	5	40	5	40			
$\dot{m}$ (leg /g)	0.1	0 55	0.1	0.044			
$m_a$ , (kg/s)	0.1	0.55	0.1	0.244			

### 3. RESULTS

The first candidate reduced model evaluated was one produced by the heuristic elimination algorithm. Sensitivity analysis of the original model indicated that several of its states and equations have very low output sensitivities near the nominal operating conditions of the model. Most of these model functions were related to the mixed media, condensation, and latent heat related behavior of the wet air stream. A candidate reduced model was created with 986 equations to the original model's 1127. With the mixing and condensation terms removed, this simplified model effectively simulates a dry-coil heat exchanger.



Fig. 3. Temperature results for heuristic elimination and CDSP GMM candidate models.

The second candidate model considered was produced by the CDSP algorithm and consisted of two GMMs, one trained to predict the steady state values of the model outputs for constant inputs, and one trained to approximate the nonlinear dynamics of the model by predicting the state derivatives given the current state and input values. Using the trained GMMs and first order Euler integration, state trajectories for the model can be produced.

Table 1 presents the input values used in the nominal and optimal experiments to test the accuracy of each model, as well as the bounds imposed on the input space for the optimization. The nominal input values were used as seed values for the optimization. The optimal trajectories determined, albeit local, were effective in identifying regions of the input space in which the models are significantly less accurate than at nominal conditions.

Figure 3 presents outlet air and water temperature trajectories for nominal and optimal experiments with each candidate model. The input trajectories for each experiment consist of initial and final values for each of the four inputs. Differences in the initialization behavior of the candidate models required different experiment designs for each case. In the heuristic elimination case, both the original and reduced models were initialized to a fixed set of states and allowed to settle to steady state before input changes were applied, starting at 200s into the experiment duration. For the CDSP case, both models were initialized directly to steady state and the input changes were applied at 10s into the experiment duration. The full range of permissible inputs for the optimal experiments were specified such that the lower bounds satisfy  $U_{min} = [5^{\circ}C, 5^{\circ}C, 0.1 \text{kg/s}, 0.01]^{T}$  and the upper bound  $U_{max} = [40^{\circ}C, 50^{\circ}C, 0.55 \text{kg/s}, 0.6]^{T}$ .

The heuristic elimination candidate model was extremely accurate in reproducing the output trajectories of the original model for the nominal case and for all cases where the air inlet relative humidity was less than 0.3. At higher humidity conditions where the effect of condensation becomes significant, the simplified model was inaccurate. The simplified model consistently demonstrated an improvement in CPU time for simulation of approximately 10% over the original full model.

The GMM model evaluated here was trained within a constrained input range, wherein the lower bounds were  $U_{min,GMM} = [10^{\circ}C, 10^{\circ}C, 0.2 \text{kg/s}, 0.2]^{T}$  and the upper bound was  $U_{max,GMM} = [25^{\circ}C, 25^{\circ}C, 0.4 \text{kg/s}, 0.4]^{T}$ . The GMM model was extremely accurate in predicting the initial and final steady state values of the system outputs, but less effective in predicting their dynamic trajectories. At high sampling rates, simulation of the GMM model was significantly slower than the original model. Its performance was substantially improved with lower sampling rate and more efficient pre-processing, as shown in Table 2.

Table 2 shows performance characteristics for each of the simulation models. The simulation time reported is the average of ten simulations of the nominal experiment for each model in Matlab. For the original and heuristic elimination models, the sampling rate is 100hz, for the CDSP-GMM model, the sampling rate is 1hz. The simulation time for the CDSP-GMM model scales closely with the sampling rate. The CDSP-GMM model at a 100hz rate simulates in 12.68s without a significant difference in accuracy. The mean absolute percent error reported is the relative error in absolute temperatures over the nominal and full range optimal experiments. All simulations were carried out in Matlab R2016b on a Windows 7 workstation with a Xeon E3-1241 CPU and 32GB of memory. Mean absolute percentage error is reported for the outlet temperatures for each model at both nominal and optimal experiment conditions.

Table 2. Candidate Model Characteristics

Model	States	Sim. time	MAPE(nom)	MAPE(opt)
Original	1127	2.13s	0	0
Heuristic	986	1.94s	< 0.01%	4.38%
CDSP	5	0.156s	0.14%	1.04%
			1 . 4001	1

Simulation times reported correspond to 100hz sampling rate for the Original and Heuristic models and 1hz for the CDSP model.

### 4. CONCLUSIONS

Two approaches were studied for the production of computationally efficient thermofluid system models: (a) a heuristic-directed term elimination approach, reducing a high-fidelity model of the system; and (b) a contracting dynamical system primitive method, based on machine learning. Both methods demonstrated different advantages in fitting the original, high fidelity model of a heat exchanger. The heuristic elimination model generated was extremely accurate within a relatively large subset of the input space, and demonstrated minor improvements in computation time. The CDSP GMM statistical model consistently converged to the correct steady state values for the system outputs, but was less effective at matching system dynamics. The GMM model was also capable of simulating at a much lower sampling rate than the original model, with a significant reduction in required CPU time. Efficiency improvements and refinements of the Heuristic Elimination and CDSP algorithms are the focus of ongoing work, along with application of these methods to more complex systems and the exploration of hybrid model structures that may share the advantages of each approach.

### REFERENCES

- Åkesson, J., Årzén, K.E., Gäfvert, M., Bergdahl, T., and Tummescheit, H. (2010). Modeling and optimization with optimica and jmodelica. orglanguages and tools for solving large-scale dynamic optimization problems. *Computers & Chemical Engineering*, 34(11), 1737–1749.
- Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (In Press, 2018). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.
- Atkinson, A.C. and Fedorov, V.V. (1975a). The design of experiments for discriminating between two rival models. *Biometrika*, 62(1), 57–70.
- Atkinson, A.C. and Fedorov, V.V. (1975b). Optimal design : Experiments for discriminating between several models. *Biometrika*, 62(2), 289–303.
- Atkinson, A.C., Bogacka, B., and Bogacki, M.B. (1998). D- and t-optimum designs for the kinetics of a reversible chemical reaction. *Chemometrics and Intelligent Labo*ratory Systems, 43(12), 185 – 198.
- Baillie, B.P. and Bollas, G.M. (2017a). Development, validation, and assessment of a high fidelity chilled water plant model. *Applied Thermal Engineering*, 111, 477– 488.
- Baillie, B.P. and Bollas, G.M. (2017b). Model reduction by term elimination and optimal selection. In 27th European Symposium on Computer Aided Process Engineering, volume 40, 277–283.
- Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. *Current science*, 78(7), 808– 817.
- Cohn, D.A., Ghahramani, Z., and Jordan, M.I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1), 129–145.
- Cozad, A., Sahinidis, N.V., and Miller, D.C. (2014). Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6), 2211–2227.
- Dani, A.P., Chung, S.J., and Hutchinson, S. (2015). Observer design for stochastic nonlinear systems via contraction-based incremental stability. *IEEE Transactions on Automatic Control*, 60(3), 700–714.
- Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the E-

M algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1–38.

- Ghahramani, Z. and Roweis, S.T. (1999). Learning nonlinear dynamical systems using an E-M algorithm. Advances in neural information processing systems, 431– 437.
- Haffmann, T. and Wichmann, T. (2003). Overview of symbolic methods in industrial analog circuit design.
- Han, L., Zhou, Z., and Bollas, G.M. (2016a). Model-based analysis of chemical-looping combustion experiments. part i: Structural identifiability of kinetic models for nio reduction. AIChE Journal, 62(7), 2419–2431.
- Han, L., Zhou, Z., and Bollas, G.M. (2016b). Model-based analysis of chemical-looping combustion experiments. part ii: Optimal design of ch4-nio reduction experiments. *AIChE Journal*, 62(7), 2432–2446.
- Khansari-Zadeh, S.M. and Billard, A. (2014). Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6), 752–765.
- Khansari-Zadeh, S.M. and Billard, A. (2011). Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics*, 27(5), 943–957.
- Moore, B. (1981). Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1), 17–32.
- Nilsson, O., Rantzer, A., and Chauvin, J. (2006). A model reduction case study: Automotive engine air path. In 2006 IEEE Conference on Computer Aided Control System Design, 1398–1403.
- Palmer, K.A., Hale, W.T., Such, K.D., Shea, B.R., and Bollas, G.M. (2016). Optimal design of tests for heat exchanger fouling identification. *Applied Thermal Engineering*, 95, 382 – 393.
- Quinonero-Candela, J., Rasmussen, C.E., and Williams, C.K. (2007). Approximation methods for gaussian process regression. *Large-Scale Kernel Machines*, 203– 224.
- Ravichandar, H. and Dani, A.P. (2015). Learning contracting nonlinear dynamics from human demonstration for robot motion planning. In ASME Dynamic Systems and Control Conference (DSCC).
- Ravichandar, H., Salehi, I., and Dani, A.P. (2017). Learning partially contracting dynamical systems from demonstrations. In *First Annual Conference on Robot Learning (CoRL)*, volume 78, 369–378.
- Ravichandar, H.C., Salehi, I., Baillie, B.P., Bollas, G.M., and Dani, A.P. (2018). Learning stable nonlinear dynamical systems with external inputs using gaussian mixture models. In *American Control Conference* (ACC), to appear.
- Umlauft, J., Lederer, A., and Hirche, S. (2017). Learning stable gaussian process state space models. In American Control Conference (ACC), 1499–1504. IEEE.
- Wang, W. and Slotine, J.J.E. (2005). On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics*, 92(1).
- Wetter, M. (2010). A modelica-based model library for building energy and control systems. *Lawrence Berkeley National Laboratory*.