

# Smoothing Techniques in Dynamic Building System Simulation

Zhelun Chen

*Dept. of Civil, Architectural & Environmental Engineering  
Drexel University  
Philadelphia, PA, United States  
zc98@drexel.edu*

Anthony Kearsley

*Applied and Computational Mathematics Division  
National Institute of Standards and Technology  
Gaithersburg, MD, United States  
anthony.kearsley@nist.gov*

Jin Wen

*Dept. of Civil, Architectural & Environmental Engineering  
Drexel University  
Philadelphia, PA, United States  
jw325@drexel.edu*

Amanda Pertzborn

*Energy and Environment Division  
National Institute of Standards and Technology  
Gaithersburg, MD, United States  
amanda.pertzborn@nist.gov*

**Abstract**—Efficiently, robustly, and accurately solving systems of nonlinear differential algebraic equations (DAE) for dynamic building system simulation is becoming more important due to the increasing demand to simulate large-scale problems including the integration of multiple buildings. However, in some building system simulations, the formulation of DAEs can include discontinuous variables and equations, and thus, derivatives. Even if these occurrences are infrequent, the presence of a discontinuity in a building system simulation can result in loss of efficiency and robustness in calculating solutions. This issue becomes more burdensome as problem sizes grow. In this paper, a smoothing technique that attempts to remove function discontinuity in dynamic building system simulation, enabling desirable numerical behaviour while preserving physical accuracy, is investigated. The impact of the smoothing technique on the numerical simulation is demonstrated in an example employing the HVACSIM+ environment: smoothing of a discontinuous coil component model. This example shows that the smoothing technique can greatly improve the efficiency and robustness of the numerical simulation with similar solution accuracy to the original, discontinuous model, while it is argued that the method does not deviate far from physical reality.

**Keywords**—HVAC simulation, function discontinuity, cooling coil

## I. INTRODUCTION

The building sector (including residential and commercial buildings) represents the largest energy-consuming sector in the United States; in 2019 it was responsible for 39 % of the country's energy consumption, in comparison to 28 % for the transportation sector and 33 % for the industrial sector [1]. Moreover, buildings consume 74 % of the electricity in the United States [1], which makes the building sector significant to the overall smart grid infrastructure. Given the rapid development of the smart grid and the potential of buildings to store and generate electricity [2] through demand shifting and transactive control, there is a need to improve the dynamic interactions between buildings and the smart grid, which further calls for robust, efficient, and accurate dynamic building energy system modelling and simulation.

Functions employed by these simulations, which give rise to function discontinuities, can negatively affect the efficiency and robustness of the simulation. Function discontinuity is

commonly observed in building system component models, such as coils [3][4][5][6], damper/valve control, and flow resistance models [5], and is known to be a cause of numerical difficulty [4][5]. These component models also generate discontinuous outputs [4] that introduce oscillation to the simultaneous solution when integrated with a global heating, ventilating, and air conditioning (HVAC) system, increasing the difficulty for control parameter tuning. One way to remove discontinuities is to replace the discontinuous model with a more detailed, continuous model. However, such a model usually requires more inputs, geometrical details, or computational effort [4][7][8][9][10]. In addition, developing a new component model to replace the discontinuous component model may require considerable knowledge about the component (such as the complicated heat transfer processes of a coil model) that many modelers do not possess. Therefore, we adopt a customizable, simple smoothing technique [11] that requires limited knowledge of the component model equations by modelers at all levels. This technique is similar to the one documented in the Modelica buildings library [12]. Similar smoothing techniques have also been seen in other fields, such as phase-field models in the solidification of alloys [13] where discontinuities arise across solid/liquid phases, optimal control of systems in the presence of non-smooth dry or Coulomb friction [14][15], and hydrological models in environmental processes [16] where discontinuities arise from threshold-type behavior.

The purpose of this study is to demonstrate the impact of the smoothing technique on the overall performance of the dynamic building system simulation. In the following sections, the smoothing technique investigated in this study is introduced and its implementation is shown in a practical example. The improvement in robustness and efficiency of the simulation when using the smoothing technique is discussed.

## II. METHODOLOGY

The primary mathematical problem at the heart of HVAC system simulations is the approximation to the solution of a collection of nonlinear equations [17]. Thus, a method that robustly and efficiently approximates the solution to Equation (1) is sought,

$$F(x) = 0, F: \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad (1)$$

where:  $\mathbf{x}$  is the state vector containing  $n$  variables/unknowns;  $\mathbf{F}$  is a set of  $n$  functions that depend exclusively on  $\mathbf{x}$  and is assumed to be continuously differentiable everywhere in  $n$ -dimensional space  $\mathbb{R}^n$ .

A challenge in solving Equation (1), in the case of building dynamic system simulations, is the assumption of a continuously differentiable equation system, which may not always be valid. These discontinuities can prevent convergence and/or lead to excessive iteration that causes long simulation times. A function smoothing technique is one alternative to remove the discontinuity in the system of equations and avoid these pitfalls.

To illustrate the smoothing technique, we simplify the discontinuity in an equation system to a univariate nondifferentiable piecewise function as follows:

$$f(x) = \begin{cases} f_1(x); & x < x_b \\ f_2(x); & x \geq x_b \end{cases} \quad (2)$$

where:  $x_b$  is a point of discontinuity when  $f_1(x_b) \neq f_2(x_b)$ . The system of equations presented in Equation (1) may contain functions of this type.

One can employ a function that varies smoothly, for example, from 0 to 1, to generate a smooth transition of  $f(x)$  between the regions defined by  $f_1(x)$  and  $f_2(x)$  around  $x_b$ , such that  $f(x)$  is differentiable everywhere in  $x$ . Here, a sigmoid function [11], as shown in Equation (3), is employed. In Equation (3),  $z$  determines the midpoint of  $\alpha$ , i.e.,  $\alpha = 0.5$  when  $x = z$ , and  $w$  determines the width of the transition region (i.e., the region between 0 and 1).

$$\alpha = 1 / \left( 1 + e^{\frac{-(x-z)}{w}} \right) \quad (3)$$

The smoothed version of Equation (2) can then be expressed as a convex combination [11] of  $f_1(x)$  and  $f_2(x)$ , such that:

$$f(x) = (1 - \alpha)f_1(x) + \alpha f_2(x). \quad (4)$$

In practice,  $z$  is typically taken to be  $z = x_b$ , so that the transition region between the two discontinuous sections expands symmetrically about  $x_b$ . If  $f_1(x)$  is preferred over  $f_2(x)$ ,  $z > x_b$  can be used. On the other hand, if  $f_2(x)$  is preferred over  $f_1(x)$ ,  $z < x_b$  can be used.

Fig. 1 shows the application of smoothing to a simple example in Equation (5) using various choices of  $z$  and  $w$ . As shown in the first plot of Fig. 1, by setting  $z = 0$ , all smoothed versions of the original function intersect at  $x = 0$  (i.e., the midpoint of the transition region). A larger  $w$  results in a wider transition region, and a smaller  $w$  results in a stiffer transition region. As shown in the second plot of Fig. 1,  $z$  determines the midpoint of the transition region. The transition region shifts from left to right by increasing  $z$ .

$$f(x) = \begin{cases} -x - 1; & x < 0 \\ x + 3; & x \geq 0 \end{cases} \quad (5)$$

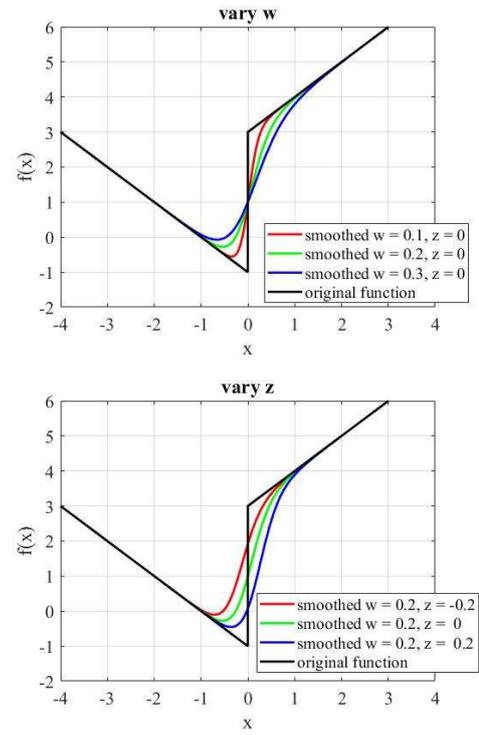


Fig. 1. Smoothing technique example.

### III. NUMERICAL EXPERIMENTS

In this section, the testbed used in the example will be introduced, followed by detailed simulation results and discussion.

#### A. Introduction of the Testbed

The testbed used in this study was developed in the HVACSIM+ environment. Developed by the U.S. National Institute of Standards and Technology (NIST), HVACSIM+ is a dynamic component-based simulation tool and computational environment [18] that includes a collection of subroutines in three categories: pre-processing, simulation, and post-processing. During the pre-processing stage, a simulation work file is created by use of the interactive front-end program of HVACSIM+. The simulation work file is then converted to a model definition file to make the model information readable by the main simulation program, MODSIM.

HVACSIM+ employs a hierarchical structure, illustrated in Fig. 2.

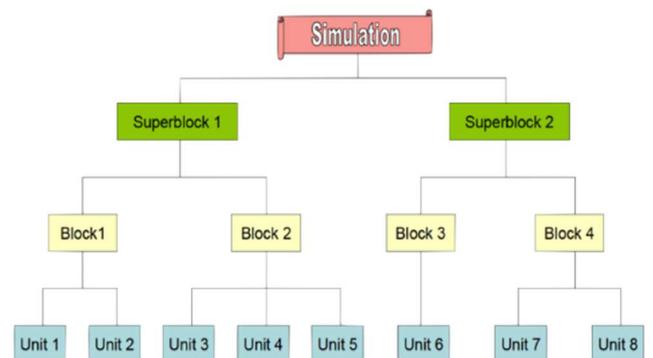


Fig. 2. Hierarchical simulation setup.

In this structure, a UNIT is an individual instance of a generic component model representing a specific component in the system. Each UNIT is assigned with a TYPE that contains equations for a component (e.g., a component that calculates the inlet pressure for a fluid resistance given the outlet pressure and the inlet mass flow rate). Users link UNITS together through their inputs and outputs to represent their relationships in the real physical system. Closely coupled UNITS are grouped by the user into blocks for simultaneous solution. Blocks are then grouped into a superblock for simultaneous solution. State variables in different superblocks are weakly coupled, and each superblock is treated as an independent subsystem of the overall simulation [18]. The simultaneous equation system resulting from input/output connections and discretization of differential equations in each subsystem, as presented in Equation (1), is solved using Powell's Hybrid method [19]. Other solution methods, such as Levenberg-Marquardt [20][21] and Newton-Krylov methods [22][23], can also be used for solving the equation system.

The testbed used for the example study is based on a small commercial building simulation model developed as part of the ASHRAE 1312 project [24]. This model simulates a single duct, dual fan variable air volume (VAV) system serving four building zones: west-facing, south-facing, east-facing, and internal spaces. Each superblock in the testbed is constituted by one block in which equations are solved simultaneously.

In the testbed, components are grouped into five superblocks as follows: control, actuator, airflow (i.e., equations related to mass flow and pressure), thermal (i.e., equations related to heat and moisture transfer), and sensor. This testbed imposes a weak coupling among equations from different subsystems. At each time step, equations within each superblock (or block) are solved simultaneously. The solution of each superblock is provided to the superblocks that numerically follow the solved superblock (e.g., the actuator superblock is solved after the control superblock). The sensor superblock is computed last in a time step  $t$ , and captures the values of time step  $t$ , such as the temperatures, which are computed earlier in the same time step by the thermal superblock. These values will be passed to the control superblock at the next time step,  $t + 1$ , to determine whether adjustments of damper and valve positions are needed. Then, at time step  $t + 1$ , based on the determination of the control superblock, command signals are passed to the actuator superblock within the same time step to move dampers and

valves. In the same time step  $t + 1$ , mass-pressure equations in the airflow superblock and then energy balance equations in the thermal superblock are solved simultaneously for a new solution based on the new damper and valve positions. Among these superblocks, the three superblocks that contain variables that require simultaneous solution are the: control superblock, airflow superblock, and thermal superblock. The simple coil component model is simulated simultaneously in the thermal superblock, therefore only the solver performance for that superblock will be discussed in this example.

### B. Smoothing of the discontinuous simple coil component model

In the testbed, a set of simple coil functions is used to simulate the dynamic behaviour of both the heating coil and the cooling coil. This set of functions assumes a coil is either completely dry or completely wet, based on whichever one predicts a greater heat transfer rate, even if the coil is partially wet. The solution process of a simple coil model can be found in Fig. 3.a. The advantage of this model is an increase in computational speed. However, oscillations in the outputs will be found at the transition between completely dry and completely wet (i.e., partially wet). Moreover, while coupling the simple coil model into the HVAC system, as two sets of distinct functions are being evaluated during a single time step, drastic changes to the Jacobian/derivative occur between iterations when  $\dot{Q}_{dry} \cong \dot{Q}_{wet}$ , making numerical convergence difficult.

Employing a detailed coil model can remove the convergence difficulties caused by the coil component, and its solution is more accurate. However, more time is consumed in solving such a detailed model. For example, the computational time of the simple coil model is about 0.00017 seconds per function evaluation on a personal laptop, while a detailed model [7] with six control volumes requires 10 fold more computational time. Thus, instead of solving a detailed model, the function smoothing technique presented in Section II is employed to make the simple coil model more stable and more compatible with nonlinear equation solution techniques, while maintaining its efficiency.

Equation (3) can be rewritten to Equation (6) as follows:

$$\alpha = 1 / \left( 1 + e^{\frac{-[(\dot{Q}_{dry} - \dot{Q}_{wet}) - z]}{w}} \right). \quad (6)$$

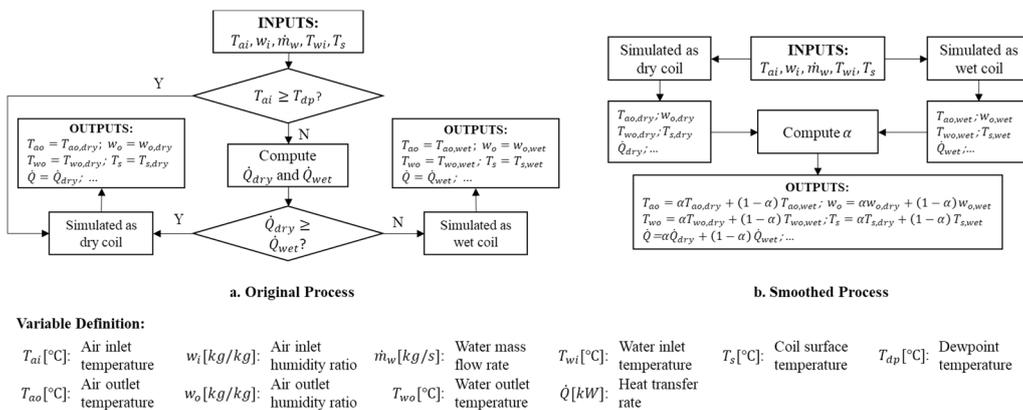


Fig. 3. Solution process of a simple coil model (a. Original; b. Smoothed).

Typically, we set  $z = 0$ , such that the completely dry and wet coil equations have an equivalent contribution to the final solution when  $\dot{Q}_{dry} = \dot{Q}_{wet}$ . The heat transfer rate of the smoothed simple coil model can then be expressed as,

$$\dot{Q} = \alpha \dot{Q}_{dry} + (1 - \alpha) \dot{Q}_{wet}. \quad (7)$$

By applying this technique, the diagram for the solution process of a simple coil model is transformed as shown in Fig. 3.b. Instead of calculating either dry coil functions or wet coil functions, the smoothed coil model requires calculations for both sets of functions, which increases the computational effort compared to the original coil model. However, it is still more efficient than detailed models. Moreover, the smoothed coil model will converge more quickly and accurately than the original coil model, and hence improve the robustness and efficiency of the overall simulation.

Numerical experiments are performed to evaluate the smoothing technique. Multiple choices ( $w$ : 0.05, 0.1, 0.3, 0.5, 0.8) for the transition region's width with  $z = 0$  are tested. The simulation time step is 2.5 seconds. For all cases, each time step of the simulation terminates when either one of the criteria summarized in TABLE I. is satisfied. Among the two converged conditions, it is preferred that the residual test, i.e.,  $\|F\|_2 \leq ftol$ , is satisfied. If only the stalling test, i.e.,  $\|x^{k+1} - x^k\|_2 \leq xtol$ , is satisfied, the solution may be trapped at a local minimum with large residual, resulting in a lower degree of success in convergence. If neither of the two convergence conditions is satisfied, the time step is terminated when the maximum number of function evaluations is reached to prevent unnecessary computational effort. For the 4Z5B testbed, the number of simultaneous variables to be solved in the superblock,  $n$ , is 7, 37, and 48 for the control, airflow, and thermal superblocks, respectively. The 4Z5B testbed employing the original discontinuous simple coil component model is set as the baseline. Only the results for the thermal superblock during the operating period (6:00-18:00) on a summer day (09/02/2007) in Ankeny, IA are shown because the smoothing technique only takes effect when the cooling coil is operating. The performance of each case is evaluated by the termination condition, the average norm of the final residual, and the number of function evaluations.

The number of time steps when the final residual did not reach the residual tolerance (either failed to converge or considered converged with a lower degree of success) are summarized in TABLE II. ; a smaller number indicates better convergence. The results indicate that when  $w$  is greater than or equal to 0.3, the number of time steps with the final residual not reaching the residual tolerance drastically decreased compared to the baseline. This suggests that a smoothed simple coil model with  $w \geq 0.3$  delivers satisfactory convergence properties. A smaller  $w$  would approximate the original discontinuous model more closely but would result in a transition region that is still too "steep" so that the model would not perform as well as with larger  $w$ .

TABLE III. shows the average norm of the final residual. Even though there are still a significant number of time steps where the residuals do not reach the tolerance in the cases with  $w = 0.05$  and  $w = 0.1$ , the residuals in these cases are significantly smaller than the residuals in the baseline case.

TABLE I. TERMINATION CRITERIA

Condition	Criteria	Specified Value
Converged	$\ F\ _2 \leq ftol$	$ftol = 5 \times 10^{-5}$
Converged (less success)	$\ x^{k+1} - x^k\ _2 \leq xtol$	$xtol = 1 \times 10^{-4}$
Not Converged	$NFE \geq NFE_{max}$	$NFE_{max} = 200(n + 1)$
<b>Definition:</b> (1) $x^{k+1}$ : solution of the $(k + 1)$ -th iteration; (2) $x^k$ : solution of the $k$ -th iteration; (3) $\ F\ _2$ : Euclidean norm of the residual; (4) NFE: Number of Function Evaluations; (5) $n$ : the number of simultaneous variables to be solved in a superblock.		

TABLE II. TERMINATION CONDITIONS  
(THERMAL SUPERBLOCK, 6:00-18:00, SUMMER DAY 09/02/2007 AT ANKENY, IA)

	Baseline	$w$				
		0.05	0.1	0.3	0.5	0.8
Number of Time Steps with $\ F\ _2 > ftol$	2463	1506	679	13	9	26

TABLE III. AVERAGE NORM OF THE FINAL RESIDUAL  
(THERMAL SUPERBLOCK, 6:00-18:00, SUMMER DAY 09/02/2007 AT ANKENY, IA)

	Baseline	$w$				
		0.05	0.1	0.3	0.5	0.8
$\ F\ _2$	$1.3 \times 10^{-2}$	$2.3 \times 10^{-5}$	$1.6 \times 10^{-5}$	$1.5 \times 10^{-5}$	$1.9 \times 10^{-5}$	$1.8 \times 10^{-5}$

Under the partially wet condition, the use of a discontinuous coil model causes the computation to iterate back and forth between the completely dry and completely wet conditions, significantly increasing the number of function evaluations. By adding a smooth transition between the two conditions, the computation quickly determines function values between the two conditions, thus reducing the number of function evaluations. According to TABLE IV. , simulations employing the smoothed coil model require 20 % fewer function evaluations than the simulation employing the original coil model.

As a consequence of introducing a smooth transition region to the coil model, it is expected that the output discontinuity/oscillation issues can be resolved. The solution of the cooling coil outlet air temperature and humidity ratio, which are the two most important outputs of the cooling coil, are presented in Fig. 4. These figures only capture the solution from 8:00 to 11:00, during which the cooling coil is dominated by the partially wet condition. The baseline model has large oscillations, which is consistent with the statement in [5]. The solutions are much more stable for the smoothed coil simulations. There is no clear relationship between the smoothing level and the oscillation magnitude.

TABLE IV. NUMBER OF FUNCTION EVALUATIONS "NFE"  
(THERMAL SUPERBLOCK, 6:00-18:00, SUMMER DAY 09/02/2007 AT ANKENY, IA)

	Baseline	$w$				
		0.05	0.1	0.3	0.5	0.8
Avg. NFE	64.9	50.2	50.2	50.1	50.0	50.0

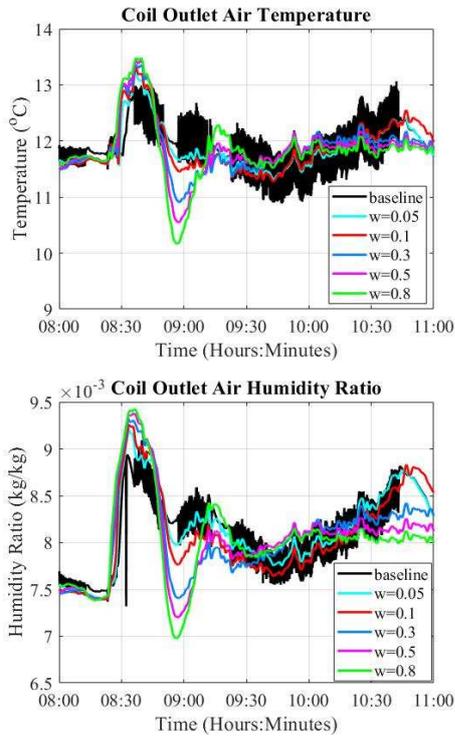


Fig. 4. Solution of Cooling Coil Outlet Air Temperature and Humidity Ratio – Time Domain (Thermal Superblock, 8:00-11:00, Summer Day 09/02/2007 Ankeny, IA).

The values of the smooth function  $\alpha$  during 8:00 to 11:00, which indicates the proportion of dry coil in the solution during this period, are presented in Fig. 5. The smoothed coil model with smaller transition width (e.g.,  $w = 0.05$  and  $w = 0.1$ ) switches faster between completely dry (i.e.,  $\alpha = 1$ ) and completely wet (i.e.,  $\alpha = 0$ ) conditions. This explains the large discrepancies among the solutions observed in Fig. 4 around 9:00.

The cooling coil is a key component in an air handling unit. The accuracy of the cooling coil model has a strong impact on the accuracy of the simulation model. The accuracy of the smoothed simple coil model scheme presented above can be evaluated by comparing the solution of the smoothed simple coil model with the solution of the original simple coil model and the solution of a detailed model that provides more accurate predictions. The Purdue cooling coil model (with dynamic forward modelling) [7] was developed using the control volume method, serves as the detailed model in this comparison. This model agrees well with the experimental data for all coil conditions under varying coil inputs. The coil parameters required by the original and the smoothed simple coil models, and the Purdue cooling coil model are summarized in Table 3-8 and Table 3-7 from [23], respectively. The smoothed coil model used for this comparison employs  $w = 0.3$  and  $z = 0$ .

The original simple coil model, the smoothed simple coil model, and the Purdue cooling coil model are compared under eighteen steady-state conditions using the coil inputs summarized in TABLE V. . By varying the air inlet relative humidity among different steady state inputs, all completely dry, completely wet, and partially wet conditions are investigated.

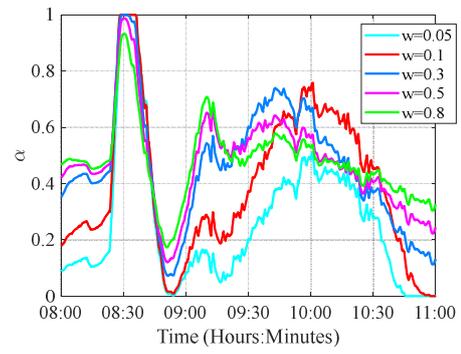


Fig. 5. Values of the Smooth Function  $\alpha$  (Thermal Superblock, 8:00-11:00, Summer Day 09/02/2007 at Ankeny, IA).

TABLE V. STEADY COIL INPUTS

Air inlet temperature $T_{ai}$	Air inlet relative humidity $RH_i$	Air inlet mass flow rate $\dot{m}_{ai}$	Water inlet temperature $T_{wi}$	Water inlet mass flow rate $\dot{m}_{wi}$
26.67 °C (80 °F)	18 choices (every 5 % from 10 % to 95 %)	0.41 kg/s (1000 CFM)	4.44 °C (40 °F)	0.25 kg/s (4 GPM)

Three outputs (air outlet temperature -  $T_{ao}$ , air outlet humidity ratio -  $w_o$ , and heat transfer rate  $\dot{Q}_a$ ) of the three coil models are presented in Fig. 6. The obvious difference between the outputs of the smoothed and the original simple coil models can be seen at 40 % and 45 % relative humidity. The differences are due to the application of the smoothing technique for the partially wet coil condition. Compared to the original simple coil model, the smoothed simple coil model results in more accurate  $T_{ao}$  at 40 % relative humidity, less accurate  $T_{ao}$  at 45 % relative humidity, more accurate  $w_o$  at both 40 % and 45 % relative humidity, and similar accuracy for other humidity ratios and other outputs. The two simple coil models also tend to underestimate the total heat transfer rate, especially when the coil moves toward the completely wet condition (i.e., high relative humidity). This error between the simple model and the detailed model is inevitable due to the lack of detail in the simple coil model.

Next, the models are compared under the same dynamic conditions. The dynamic inputs to the coil that result in a partially wet condition can be found in Fig. 3-10 of [23]. The same three outputs as in the steady-state comparisons are presented in Fig. 7. The smoothed simple coil model results are more accurate (fewer errors compared to the results of the detailed model) and have fewer oscillations for all outputs compared to the original coil model. The normalized root mean square errors (NRMSE) of the original and the smoothed simple coil models' outputs are summarized in TABLE VI. . The NRMSE is calculated as,

$$\sqrt{\sum_{t=1}^T (O_s - O_p)^2} / \bar{Q}_p \times 100\% . \quad (8)$$

where:  $O_s$  is an output of either the original or the smoothed simple coil model at time step  $t$ , and  $O_p$  is an output of the Purdue coil model at time step  $t$ ,  $T$  is the final time step, and  $\bar{Q}_p$  is the mean value of a Purdue coil's output.

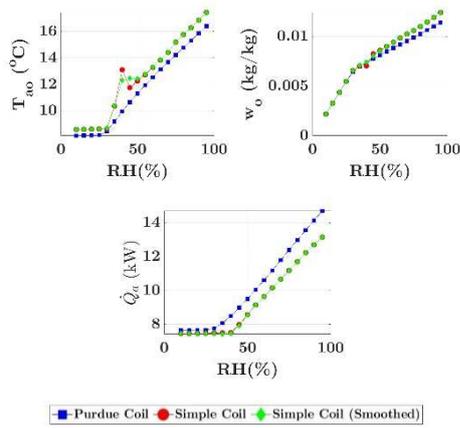


Fig. 6. Comparisons of the steady coil outputs.

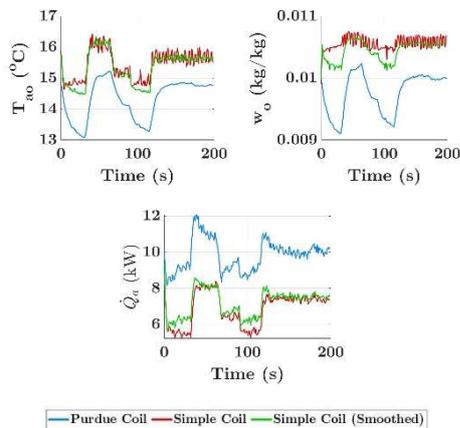


Fig. 7. Comparisons of the dynamic coil outputs.

TABLE VI. NORMALIZED ROOT MEAN SQUARE ERRORS OF THE SIMPLE COIL MODELS WITH RESPECT TO THE PURDUE MODEL

	$T_{co}$	$w_o$	$\dot{Q}_c$
<b>Original</b>	8.2 %	8.8 %	30.8 %
<b>Smoothed</b>	7.4 %	7.3 %	27.3 %

#### IV. CONCLUSION

Removing discontinuities and smoothing spatially and time dependent parameters that are often approximated in large-scale dynamic building system simulation can improve solution efficiency and robustness, as a small number of discontinuous functions in the numerical model can significantly jeopardize the convergence. In this study, an application of a generic smoothing technique in dynamic building system simulation was presented. It was shown in the example that the smoothing technique can improve the efficiency and robustness of the simulation while maintaining similar accuracy of the solution. In practice, tackling every discontinuous function in a numerical simulation model can be very time consuming. Future study may focus on a computational process that locates the sources of discontinuities and applies a smoothing technique automatically.

#### ACKNOWLEDGMENT

The first author thanks the National Institute of Standards and Technology for funding the project under grant number: 60NANB15D251.

#### REFERENCES

- [1] U.S. Energy Information Administration, "March 2020 Monthly Energy Review," 2020.
- [2] J. Goellner et al., "Demand dispatch intelligent demand for a more efficient grid," US National Energy Technology Lab, 2011.
- [3] J. Lebrun, X. Ding, J. Eppe, and M. Wasacz, "Cooling coil models to be used in transient and/or wet regimes-theoretical analysis and experimental validation," Proceedings of SSB, pp. 405-411, 1990.
- [4] R. J. Chillar and R. J. Liesen, "Improvement of the ASHRAE secondary HVAC toolkit simple cooling coil model for simulation," Proceedings of SimBuild, vol. 1, no. 1, 2004.
- [5] P. Haves and L. Norford, "A Standard Simulation Testbed for the Evaluation of Control Algorithms and Strategies. ASHRAE 825-RP Final Report," American Society of Heating, Refrigerating, and Air Conditioning Engineers, Inc. pp. 30329-2305, 1997.
- [6] V. Lemort, "Development of simple cooling coil models for simulation of HVAC systems," ASHRAE Transactions, vol. 114, p. 319, 2008.
- [7] X. Zhou, "Dynamic modeling of chilled water cooling coils," 2005.
- [8] L. Wang, P. Haves, and F. Buhl, "An Improved Simple Chilled Water Cooling Coil Model," IBPSA-USA Journal, vol. 5, no. 1, pp. 647-654, 2012.
- [9] A. Vardhan and P. Dhar, "A new procedure for performance prediction of air conditioning coils," International Journal of Refrigeration, vol. 21, no. 1, pp. 77-83, 1998.
- [10] J. Wang and E. Hihara, "Prediction of air coil performance under partially wet and totally wet cooling conditions using equivalent dry-bulb temperature method," International Journal of Refrigeration, vol. 26, no. 3, pp. 293-301, 2003.
- [11] J. Kitchin, "Smooth transitions between discontinuous functions," 2011. [Online]. Available: <http://matlab.cheme.cmu.edu/2011/10/30/smooth-transitions-between-discontinuous-functions/>.
- [12] M. Wetter, W. Zuo, T. S. Noudui, and X. Pang, "Modelica buildings library," Journal of Building Performance Simulation, vol. 7, no. 4, pp. 253-270, 2014.
- [13] A. A. Wheeler, W. J. Boettinger, and G. B. McFadden, "Phase-field model for isothermal phase transitions in binary alloys," Physical Review A, vol. 45, no. 10, p. 7424, 1992.
- [14] R. Glowinski and A. J. Kearsley, "On the simulation and control of some friction constrained motions," SIAM Journal on Optimization, vol. 5, no. 3, pp. 681-694, 1995.
- [15] A. J. Kearsley, "The use of optimization techniques in the solution of partial differential equations from science and engineering," Ph.D. thesis, Rice University, 1996.
- [16] D. Kavetski and G. Kuczera, "Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration," Water Resources Research, vol. 43, no. 3, 2007.
- [17] J. E. Dennis and R. B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations. Siam, 1996.
- [18] C. Park, D. Clark, and G. Kelly, "HVACSIM+ building systems and equipment simulation program: building-loads calculation," National Bureau of Standards, Washington, DC (USA). Building Equipment Div., 1986.
- [19] M. J. Powell, "A hybrid method for nonlinear equations," Numerical methods for nonlinear algebraic equations, vol. 7, pp. 87-114, 1970.
- [20] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in Numerical analysis: Springer, 1978, pp. 105-116.
- [21] S. Pourarian, A. Kearsley, J. Wen, and A. Pertzborn, "Efficient and robust optimization for building energy simulation," Energy and Buildings, vol. 122, pp. 53-62, 2016.
- [22] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," Journal of Computational Physics, vol. 193, no. 2, pp. 357-397, 2004.
- [23] Z. Chen, "Advanced Solver Development for Large-scale Dynamic Building System Simulation," Ph.D. thesis, Drexel University, 2019.
- [24] J. Wen and S. Li, "Tools for evaluating fault detection and diagnostic methods for air-handling units," ASHRAE RP-1312 Final Report, ASHRAE, Atlanta, 2011.