Testing building controls with the BLDG toolbox

Invited Paper, 2016 American Control Conference, Boston, MA

Kevin J. Kircher and K. Max Zhang

***Abstract*— We introduce the open-source MATLAB® toolbox `bldg`. The main function, `bsim`, simulates a simple building under forcing from the outdoor air temperature, sun, and internal heat sources. `bsim` retains the mathematical structure of state-of-the-art building simulators, including time-varying solar forcing, PDE-governed wall temperatures, and nonlinearities arising from thermal radiation and temperature-dependent convection coefficients. Supporting functions are provided to instantiate a default building object, import weather data, and generate internal heat source signals. The `bldg` toolbox is intended to eliminate the need to model a building or leave the MATLAB® environment before testing an estimation or control algorithm. We demonstrate the software through the examples of system identification, online state and parameter estimation, and model predictive control. Open research questions that the `bldg` toolbox could help explore are noted.**

I. Introduction

The world's buildings have many different sizes, shapes, locations and weather conditions, material properties, mechanical systems, and patterns of use. This variety makes it difficult to draw general conclusions about the performance of a building estimation or control algorithm. Typical practice, therefore, is to test algorithms through case studies and hope that observed trends generalize.

Many building control algorithms have been proposed in recent years. Since 2010, researchers have tested certainty-equivalent model predictive control (MPC) [1–3], learning-based MPC [4], economic MPC [5, 6], stochastic MPC [7–9], robust MPC [3], as well as subspace system identification [10], MPC-relevant system identification [10], and the linear, extended, and unscented Kalman filters [2, 11–15].

Some research, such as investigating an algorithm's scalability to large multizone buildings, requires a complex building geometry. For many purposes, however, a simple geometry suffices. Indeed, the model in the majority of the case studies mentioned above consisted of either one [3, 4, 7, 8, 14, 15] or two [2, 10] rooms. These simple test cases have the advantages of easy modeling, fast simulation, and transparency.

Model fidelity also matters. For a case study to be meaningful, the system from which control-oriented models are learned, and in which controls are implemented, should behave like a real (*i.e.*, nonlinear and time-varying) building. Empirical case studies such as [1, 3, 4, 9, 10, 13, 14] certainly pass this test. Simulation-based studies, on the other hand, commonly use semiphysical linear systems as truth models [2, 6–8, 11, 15]. Only a few [5, 10, 12] have learned control-oriented models from, or implemented controls in, high-fidelity simulators such as EnergyPlus, TRNSYS, or IDA-ICE. This is likely due to the nontrivial effort required to learn a simulation program, develop a model, and integrate the program with MATLAB®, the standard environment for control design.

We believe that building control researchers could benefit from a simple, control-oriented, high-fidelity MATLAB® building simulator. As a step toward that end, in this paper we introduce the `bldg` toolbox. We review related work in §II. In §III, we give an overview of the main function `bsim`, which mimics the syntax and behavior of the MATLAB® linear system simulator `lsim`. Examples are presented in §IV, including system identification, online state and parameter estimation, and MPC. We conclude in §V.

II. Related work

Two recent projects overlap with our work on control-oriented building simulation in MATLAB®. These are the Building Resistance-Capacitance Modeling (BRCM) [16] and OpenBuild [17] toolboxes. Both are built on the Building Controls Virtual Test-Bed [18] and MLE+ [19], which enable communication between MATLAB® and EnergyPlus.

The main factor that distinguishes `bldg` from BRCM is *fidelity*. BRCM maps an EnergyPlus model to an RC network and generates cost and constraint matrices for MPC. The resulting RC network neglects solar forcing and the nonlinearities arising from longwave radiation and temperature-dependent convection coefficients. These departures from the best practices of white-box simulators make BRCM a poor truth model for algorithm testing. This is to be expected, since BRCM was not designed as a test-bed, but rather to enable MPC in real, complex buildings. The effectiveness of BRCM for MPC is demonstrated in [20].

Like BRCM, OpenBuild integrates MATLAB® and EnergyPlus. However, OpenBuild uses EnergyPlus as its physics engine, and can therefore be regarded as a valid truth model. The key difference between `bldg` and OpenBuild is *simplicity*. While OpenBuild's link to EnergyPlus allows researchers to study a wide range of building geometries and HVAC systems, it also introduces all of the physical and numerical complexity described in the 1,444-page EnergyPlus engineering reference [21] and the 2,234-page EnergyPlus input-output reference [22]. By contrast, we propose a simple building with few parameters and explain all of our assumptions and numerical methods in the 14-page

The authors are with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14850. {kjk82,kz33}@cornell.edu

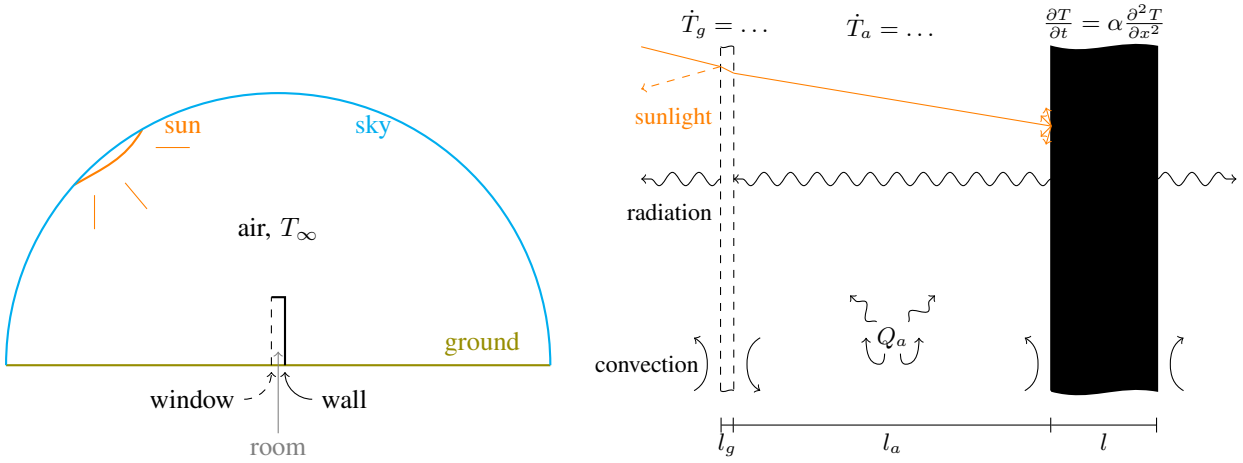

Fig. 1. geometry of the environment (left) and the building (right). The sky, ground, and outdoor air are isothermal at temperature $T_\infty(t)$. The building is a room whose length and height are much greater than the glass thickness l_g, room width l_a, and wall thickness l. The left side of the building is glass with lumped temperature $T_g(t)$. The right side is a uniform wall with temperature distribution $T(x,t)$. The indoor air is well-mixed with temperature $T_a(t)$. Heat is transferred through convection, radiation, and the transmission, absorption, and reflection of visible light. The internal source $Q_a(t)$ includes heat flows from people, control systems, lights, and other equipment (appliances, electronics, *etc.*).

companion paper [23]. This deliberate simplicity is intended to make `bldg` as easy as possible to understand and use.

III. Simulator overview

The `bldg` toolbox simulates the building illustrated in Figure 1. It is a box with two dimensions sufficiently greater than the third that all heat transfer can be considered one-dimensional. For different choices of the geometric and physical parameters, this building can be thermally light or heavy, with weak or strong solar forcing and a drafty or tight envelope. In this section, we briefly review the governing equations and numerical solution scheme, input signals, and simulator syntax. Further details, including empirical validation and sensitivity analysis, can be found in [23].

A. Governing equations

The wall temperature distribution $T(x,t)$ satisfies the heat equation,

$$\begin{aligned} &\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \quad T(x,0) = T^0(x) \\ &-k\frac{\partial T}{\partial x}\bigg|_- = q_-, \quad -k\frac{\partial T}{\partial x}\bigg|_+ = q_+. \end{aligned} \tag{1}$$

Here α (m^2/s) and k (W/m·K) are the wall's thermal diffusivity and conductivity, $T^0(x)$ is the initial temperature distribution, and the subscripts $-$ and $+$ indicate the left and right wall surfaces, respectively. The fluxes q_- and q_+ (W/m^2), which are nonlinear and time-varying, include convection with temperature-dependent coefficients, longwave radiation exchange with the window and surrounding environment, and shortwave radiation from the sun and artificial lighting.

As demonstrated in [24], the window temperature is well-modeled by

$$C_g \dot{T}_g = A(q_{g-} - q_{g+}), \quad T_g(0) = T_g^0, \tag{2}$$

where C_g (J/K) is the thermal capacitance of the window and A (m^2) is its surface area. As with the wall, the window surface fluxes q_{g-} and q_{g+} include nonlinear convection and radiation. The model also accounts for the reflection, absorption and transmission of visible light by the window.

The indoor air is assumed to be well-mixed, with temperature T_a governed by

$$C_a \dot{T}_a = A(q_{a-} - q_{a+}) + Q_a, \quad T_a(0) = T_a^0, \tag{3}$$

where C_a (J/K) is the thermal capacitance of the air and any material that is isothermal with it (*e.g.*, ducts, dampers, or furniture). The fluxes q_{a-} and q_{a+} couple the air to the window and wall. The internal heat flow Q_a (W) includes infiltration of outdoor air, as well as heat flows from control systems, people, lights, and other equipment.

By carefully discretizing spatial derivatives, the PDE (1) can be written with $\mathcal{O}((\Delta x)^2)$ truncation error as a system of N coupled ODEs, where N is the number of finite difference nodes in the wall and $\Delta x = l/(N-1)$. These ODEs are coupled to Equations (2) and (3) through q_-, q_{g+}, and q_{a+}, giving a nonlinear, time-varying dynamical system that can be simulated by a MATLAB® stiff ODE solver. The system state is $(T_1, \ldots, T_N, T_g, T_a)$, where T_i is the temperature of the i^{th} wall node.

B. Input signals

We divide the input signals into disturbances and controls, though this distinction is somewhat arbitrary.[1] The disturbance is $(T_\infty, I_h, I_\perp^b, Q_p, Q_l, Q_e)$, where I_h (W/m^2) is the total solar irradiance on a horizontal surface, $I_\perp^b$ (W/m^2) is the beam irradiance on a surface normal to the sun, and Q_p, Q_l, and Q_e (W) are the internal gains from people, lights, and other equipment.

[1] For example, with flexible plug loads Q_e could be controllable.

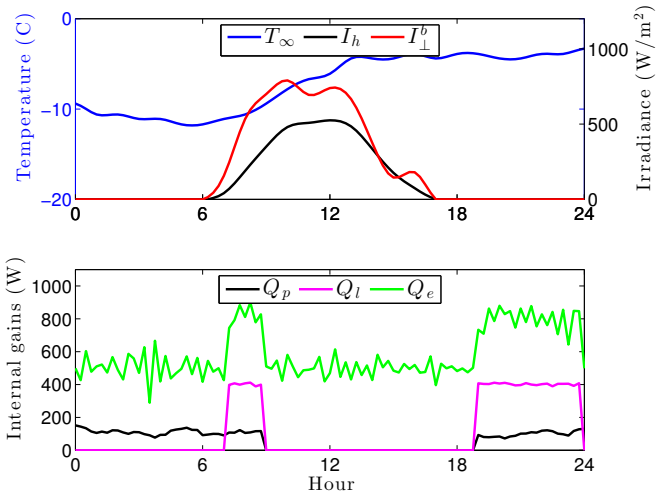

Fig. 2. Input signals from weather (top) and internal heat sources (bottom). The outdoor air temperature (T_∞), total solar irradiance on a horizontal surface (I_h), and beam solar irradiance on a surface normal to the sun ($I_\perp^b$) show that the simulation day is cold and partly cloudy. The internal heat sources model a home on a weekday, with decreased heat flows from people (Q_p), lights (Q_l), and equipment (Q_e) overnight and during work hours.

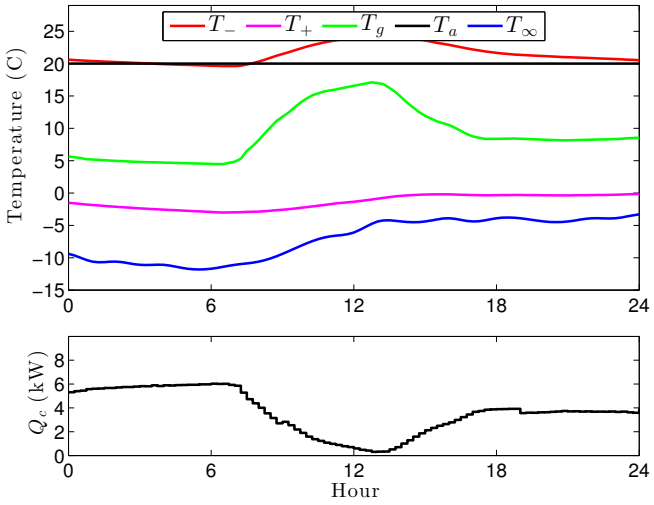

Fig. 3. Temperatures of the air (T_a), glass (T_g), left and right wall surfaces (T_- and T_+), and outdoor environment (T_∞). The bottom plot shows the heat flow required to perfectly regulate T_a at setpoint $T_s = 20$ C under the input signals in Figure 2.

The control can be either Q_c (W), the heat supplied by control systems, or $(\dot{m}_c, T_c)$, where $\dot{m}_c$ (kg/s) is the mass flow rate of supply air and T_c is the supply air temperature. In the latter case, the heat supplied by control systems is state-dependent and nonlinear: $Q_c = \dot{m}_c c_a (T_c - T_a)$, where c_a (J/kg·K) is the specific heat of air at constant pressure. If no control is specified, `bsim` returns the heat flows required to perfectly regulate the air temperature at its initial value. This is useful for quick simulations or to size heating and cooling systems.

C. Syntax

Under the default parameter values, `bsim` simulates a passive solar building that stores the sun's energy in the thermally massive wall and releases it overnight. The user can modify this behavior by adjusting parameters in the `bldg` object. For example, the thermal mass can be decreased by increasing the wall diffusivity α (`bldg.a`), and the solar forcing can be weakened by decreasing the wall shortwave absorptivity α_s (`bldg.as`). The sensitivity analysis in [23] contains more examples.

The following code instantiates a default building object, imports and interpolates a year of hourly weather data from a TMY3 file for New York City, generates plausible internal gains for a home, and defines a 24-hour simulation time span with step $\Delta t = 15$ minutes, starting at midnight ($t = 0$) on February 7 (day number $n_d = 38$). The last line packs the disturbances into the matrix form accepted by `bsim`.

```
b = bldg; dt = 15*60;
gains = generateGains(b,weather.tw);
nd = 38; t0 = 0; tf = 24*3600;
t = getTiming(weather.tw,nd,t0,tf);
W = getDisturbances(weather,gains,t);
```

Figure 2 shows the input signals generated by this code.

Simulating the building requires an initial state, which can be difficult to produce *a priori*. One is easily generated, however, using

```
x0 = precondition(...
     b,t,weather,gains,N,Ts);
```

The `precondition` function starts the building at a naive initial state two weeks before t_0 and integrates it forward with T_a perfectly regulated at setpoint T_s. The building can then be simulated by

```
[X,Qc] = bsim(b,t,W,x0);
```

With $N = 50$ wall nodes, this 24-hour simulation takes about a tenth of a second to run on a 2 GHz Intel Core 2 Duo processor. Figure 3 shows the output.

IV. Examples

A. System identification

In this example, we consider the problem of learning a low-order, linear model of the building dynamics from measurements of the indoor air temperature and weather signals. We specify the first-order ARX structure

$$T_a^{k+1} = \beta_1 T_a^k + \beta_2 Q_c^k + \beta_3 T_\infty^k + \beta_4 I_h^k + w_a^k, \qquad (4)$$

where the w_a^k are independently, identically $\mathcal{N}(0, \sigma_w^2)$ distributed. This model is naive, since it neglects the internal gains and the dynamics of the building envelope, and since the true solar forcing is time-varying and cannot be determined from I_h alone. Nevertheless, the model is sufficiently accurate to give fair controller performance (see §IV-C).

To fit the model, we simulate the (nonlinear, time-varying, high-dimensional) dynamics in `bsim` for the last three weeks of January under a sequence of pseudorandom binary control inputs. We assume perfect knowledge of Q_c, T_∞, and I_h,

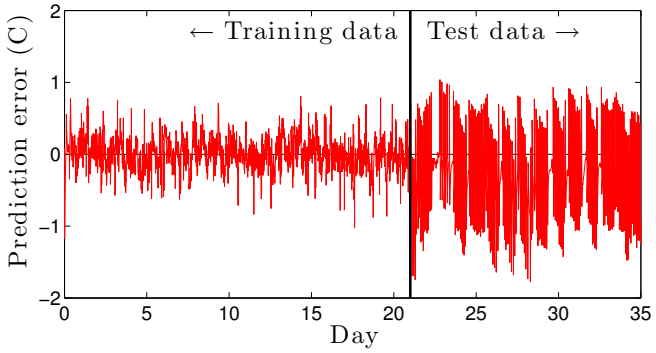

Fig. 4. One-step prediction errors in the training and test data with the static model in §IV-A. The model fit is poor: The errors are autocorrelated and, in the test set, predictions are biased by -0.31 C.

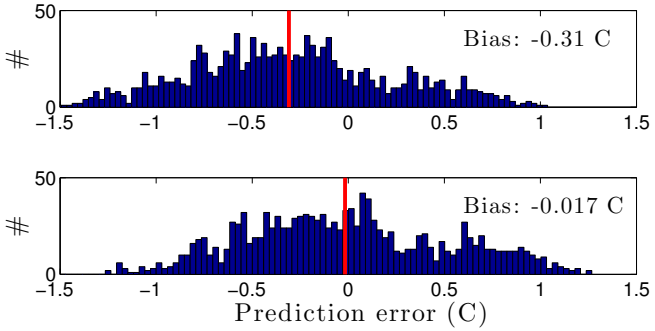

Fig. 5. Histograms of the one-step prediction errors without parameter adaptation (top), and with it (bottom). The online estimation algorithm discussed in §IV-B nearly eliminates the prediction bias. This is accomplished by adjusting the model parameters β_1 and β_3, as shown in Figure 6.

but corrupt T_a with zero mean, white, Gaussian noise with standard deviation $\sigma_v = 1/6$ C. The model parameters β and σ_w are estimated using linear regression.

Figure 4 shows the one-step prediction errors during the three training weeks, and in two subsequent test weeks under thermostatic control with deadband [18 C, 22 C]. The maximum absolute error of about 2 C is large. A higher-order ARX, ARMAX, or RC network model could achieve better predictions at the cost of introducing more states and parameters. This accuracy/complexity trade-off was explored for a family of RC networks in [15]. The authors found that a second-order RC network with five parameters is able to capture the essential building behavior (a result that dates back at least to 1980; see [25, 26]). Less is known about accuracy/complexity trade-offs for buildings with thermally massive construction or strong solar forcing. These questions could be explored in `bsim` by varying physical parameters such as the wall diffusivity α (`bldg.a`) and shortwave absorptivity α_s (`bldg.as`) and studying the model fit.

Another interesting topic is the relationship between gray- and black-box models. The parameters in model (4), for example, can be interpreted in terms of the first-order RC network

$$C_{\text{eff}}\dot{T}_a = \frac{T_\infty - T_a}{R_{\text{eff}}} + Q_c + A_{\text{eff}} I_h,$$

where C_{eff}, R_{eff}, and A_{eff} are the effective capacitance, resistance, and solar absorption area. Forward Euler discretization gives

$$T_a^{k+1} = \left(1 - \frac{\Delta t}{R_{\text{eff}} C_{\text{eff}}}\right) T_a^k + \frac{\Delta t}{C_{\text{eff}}} Q_c^k + \frac{\Delta t}{R_{\text{eff}} C_{\text{eff}}} T_\infty^k + \frac{\Delta t A_{\text{eff}}}{C_{\text{eff}}} I_h^k,$$

2 which is consistent with model (4) only if $\beta_1 + \beta_3 = 1$. In this case, the 'cost of grayness' is the accuracy lost by imposing the constraint $\beta_1 + \beta_3 = 1$ on the least-squares fitting problem, along with the resulting controller performance reduction (if any). While several authors have argued that the RC network structure has benefits – *e.g.*, it can provide initial guesses to estimation algorithms and sanity checks on their output – to our knowledge the cost of grayness has not been well-studied.

B. Online estimation

Building dynamics are naturally time-varying: Solar forcing depends on the season and time of day, infiltration rates change as occupants open and close windows and doors, and control setpoints and delivery mechanisms differ in heating and cooling modes. We therefore expect model parameters to change over several time scales. The framework of online estimation allows parameters to be continuously calibrated to measurements, enabling adaptive control. Online estimation of building model parameters has been studied in [2, 11–14].

In this example, we consider the problem of adapting a subset of the parameters of the ARX model identified in §IV-A, in order to reduce the prediction bias apparent in Figure 4. We follow the typical approach of simultaneously estimating the states and parameters with an unscented Kalman filter. [27] We allow the parameters β_1 and β_3, initialized with the fit from §IV-A, to vary under the random walk model $\beta_i^{k+1} = \beta_i^k + w_{\beta_i}^k$. This gives the augmented system model

$$\begin{aligned} x^{k+1} &= f(x^k, u^k) + w^k \\ y^k &= x_1^k + v^k, \end{aligned} \tag{5}$$

where $x = (T_a, \beta_1, \beta_3)$, $u = (Q_c, T_\infty, I_h)$, $w = (w_a, w_{\beta_1}, w_{\beta_3})$, $f_1(x,u) = x_2 x_1 + \beta_2 u_1 + x_3 u_2 + \beta_4 u_3$, and $f_i(x,u) = x_i$ for $i = 2, 3$. We model the disturbance w as zero-mean, white, and Gaussian. The RC network analogy developed in §IV-A suggests that the parameters β_1 and β_3 should (roughly) sum to one, so we specify a strong negative correlation between w_{β_1} and w_{β_3}.

Figure 5 shows histograms of the one-step prediction errors in the test data with and without parameter adaptation. Adaptation nearly eliminates the prediction bias. The unscented Kalman filter accomplishes this by decreasing β_1 and increasing β_3 over the course of about twelve hours, as shown in Figure 6. The parameters stabilize after the initial adjustment. This is expected, since the underlying physical model remains nearly constant over the estimation period. It is less clear how the filter would respond to a large,

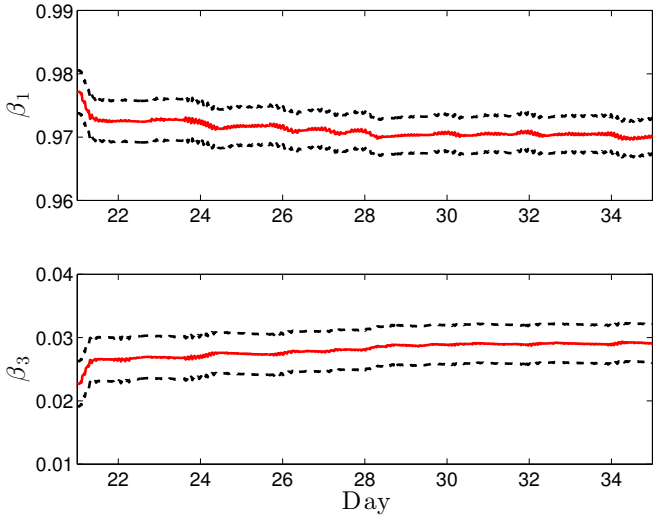

Fig. 6. Unscented Kalman filter parameter estimates and 95% confidence intervals during the two test weeks. The filter remains stable after initially adjusting β_1 and β_3 to reduce prediction bias.

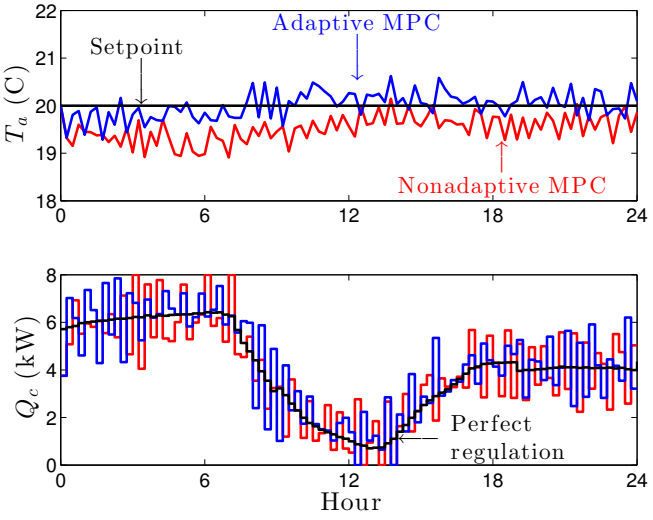

Fig. 7. Indoor air temperature T_a (top) and control heat flow Q_c (bottom) under MPC, with and without parameter adaptation. Both controllers attempt to regulate T_a just above the minimum permissible temperature of 20 C, giving heat flows that resemble the 'perfect regulation' case computed by `bsim`. Both MPC variants frequently under-heat due to model error. Parameter adaptation mitigates, but does not eliminate, constraint violation.

sudden change in the underlying system, *e.g.*, a window being opened or a mechanical component failing. These questions, which lie in the domain of fault detection, could be explored in `bsim` by perturbing the mass flow rate of infiltration air, $\dot{m}_\infty$ (`bldg.mdot`), or the fraction of the control heat that convects to the indoor air, ζ_c (`bldg.zc`).

C. Model predictive control

This example involves efficiently heating a building. We consider the stochastic optimal control problem

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\left[\Delta t \sum_{k=0}^{M} Q_c^k\right] \\ \text{subject to} & T_a^{k+1} = \text{(nonlinear } \texttt{bsim} \text{ dynamics)} \\ & y^k = T_a^k + v^k \\ & T_a^{k+1} \geq T^{\min} \text{ almost surely} \\ & Q_c^k = \mu^k(y^0, \ldots, y^k) \in [0, Q_c^{\max}], \end{array} \tag{6}$$

where the constraints hold for $k = 0, \ldots, M$. The optimization variable is the control policy $(\mu^0, \ldots, \mu^M)$, where $\mu^k : \mathbf{R}^{k+1} \to \mathbf{R}$ maps observations into controls. The expectation is taken with respect to the joint distribution of the disturbance and noise sequences and the initial state.

Problem (6) is analytically intractable due to the nonlinear dynamics, imperfect state information, and optimization over infinite-dimensional objects (the functions μ^k). While it is not difficult to generate good approximate solutions – a well-tuned thermostat works – we apply two variants of MPC to Problem 6 in order to illustrate the research value of the `bldg` toolbox. The first MPC variant uses the ARX model from §IV-A, with no parameter adaptation. At each time step, we estimate the temperature using the linear Kalman filter, then solve a truncated, certainty-equivalent version of problem (6) with horizon $H = 6$ hours. Each MPC subproblem is a deterministic linear program that generates a planned control trajectory, of which the first control is implemented. We then allow the system to evolve according to the nonlinear, time-varying dynamics in `bsim` and repeat the process.

The second MPC variant is identical, except that state and parameter estimates are simultaneously updated at each time step using the unscented Kalman filter from §IV-B. Simulating the building under MPC for one day with fifteen-minute time steps takes about 42 seconds, with linear programs solved by Gurobi on a 2 GHz Intel Core 2 Duo processor. About 93% of that time is spent in optimization, 6% in `bsim`, and 1% in the unscented Kalman filter.

Figure 7 shows the indoor air temperatures and control heat flows under both MPC variants. The simulation takes place on February 7, under the exogenous input signals shown in Figure 2. Both MPC variants attempt to regulate T_a at the minimum feasible temperature of $T^{\min} = 20$ C. This generates control trajectories that resemble noisy versions of the 'perfect regulation' case discussed in §III-B. Due to model error, both MPC variants frequently allow T_a to drop below $T^{\min}$. Adaptive MPC performs somewhat better due to its reduced prediction bias, achieving a time-averaged constraint violation of 0.11 C, compared to 0.48 C for the nonadaptive case. However, the adaptive MPC violations are still an order of magnitude larger than the 70 Celsius-hours per year (0.008 C average) specified by European building codes.

The constraint violations are an artifact of model mismatch between the controller and the underlying `bsim` dynamics. They could be reduced by using a more accurate, higher-order model, by modifying the MPC objective function (*e.g.*, by replacing $T^{\min}$ with $T^{\min} + \delta$ for some $\delta > 0$), or by moving the MPC optimization to a stochastic or robust framework. While several studies have applied stochastic and/or robust MPC to buildings [3, 7–9], it is not clear

how these approaches, which add significant conceptual and computational complexity, compare to increasing the model order or modifying the objective function.

V. Conclusion

In this paper, we introduced the MATLAB® building simulation toolbox `bldg`. We reviewed the building geometry and governing equations, then demonstrated the use of the main function `bsim` as a nonlinear, time-varying truth model for system identification, online estimation, and MPC. In MPC simulations, the computation time added by `bsim` was a modest 6% of the time spent solving linear programs.

Many research questions, such as those involving scalability to large multizone buildings, are beyond the scope of `bldg`. We also emphasize that simulation is no substitute for empirical validation, regardless of the simulator quality. Nevertheless, our experience suggests that `bldg`'s speed, fidelity, and easily varied parameters could make it a useful tool for exploring a variety of open research questions, *e.g.*:

1) How does the trade-off between a learned model's accuracy and complexity depend on the building's thermal mass and the strength of solar forcing?
2) What are the costs, in terms of model accuracy or controller performance, of imposing gray-box structure on learned models?
3) Of the various model structures and filtering algorithms in the literature, which are best for fault detection?
4) How do the costs and benefits of using robust or stochastic MPC compare to those of using a more accurate model or of modifying the certainty-equivalent MPC optimization problem?

The `bldg` toolbox is free and open source. Readers who are interested in using or modifying it can find `bldg` online at [28].

Acknowledgments

The authors are grateful to the Hydro Research Foundation and the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy for support.

References

1. Ma, Y. *et al.* Model predictive control for the operation of building cooling systems. *Control Systems Technology, IEEE Transactions on* **20,** 796–803 (2012).
2. Radecki, P. & Hencey, B. *Online thermal estimation, control, and self-excitation of buildings* in *Conference on Decision and Control* (2013), 4802–4807.
3. Maasoumy, M., Razmara, M., Shahbakhti, M. & Sangiovanni-Vincentelli, A. Handling model uncertainty in model predictive control for energy efficient buildings. *Energy and Buildings* **77,** 377–392 (2014).
4. Aswani, A., Master, N., Taneja, J., Culler, D. & Tomlin, C. Reducing Transient and Steady State Electricity Consumption in HVAC Using Learning-Based Model-Predictive Control. *Proceedings of the IEEE* **100.1,** 240–253 (2011).
5. Ma, J., Qin, J., Salsbury, T. & Xu, P. Demand reduction in building energy systems based on economic model predictive control. *Chemical Engineering Science* **67,** 92–100 (2012).
6. Kircher, K. J. & Zhang, K. M. *Model predictive control of thermal storage for demand response* in *American Control Conference (ACC)* (2015), 956–961.
7. Oldewurtel, F. *et al.* Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings* **45,** 15–27 (2012).
8. Oldewurtel, F., Jones, C., Parisio, A. & Morari, M. Stochastic Model Predictive Control for Building Climate Control. *Control Systems Technology, IEEE Transactions on* **22,** 1198–1205 (2014).
9. Ma, Y., Matusko, J. & Borrelli, F. Stochastic Model Predictive Control for Building HVAC Systems: Complexity and Conservatism. *Control Systems Technology, IEEE Transactions on* **23.1,** 101–116 (2015).
10. Privara, S., Cigler, J., Vana, Z., Oldewurtel, F. & Sagerschnig, C. Building modeling as a crucial part for building predictive control. *Energy and Buildings* **56,** 8–22 (2013).
11. Radecki, P. & Hencey, B. *Online building thermal parameter estimation via unscented Kalman filtering* in *American Control Conference* (2012), 3056–3062.
12. Martincevic, A., Starcic, A. & Vasak, M. *Parameter estimation for low-order models of complex buildings* in *Innovative Smart Grid Technologies Conference Europe* (2014), 1–6.
13. Fux, S., Ashouri, A., Benz, M. & Guzzella, L. EKF based self-adaptive thermal model for a passive house. *Energy and Buildings* **68,** 811–817 (2014).
14. Maasoumy, M., Moridian, B., Razmara, M., Shahbakhti, M. & Sangiovanni-Vincentelli, A. *Online simultaneous state estimation and paramater adaptation for building predictive control* in *ASME Dynamic Systems and Control Conference* (2013).
15. Lin, Y., Middelkoop, T. & Barooah, P. *Issues in identification of control-oriented thermal models of zones in multi-zone buildings* in *Conference on Decision and Control* (2012).
16. Sturzenegger, D., Gyalistras, D., Semeraro, V., Morari, M. & Smith, R. *BRCM Matlab Toolbox: Model Generation for Model Predictive Building Control* in *American Control Conference* (2014), 1063–1069.
17. Gorecki, T., Qureshi, F. & Jones, C. *OpenBuild: An integrated simulation environment for building control* in *IEEE Conference on Control Applications (CCA)* (2015), 1522–1527.
18. Wetter, M. Co-Simulation of Building Energy and Control Systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation* **4,** 185–203 (2011).
19. Bernal, W., Behl, M., Nghiem, T. & Mangharam, R. *MLE+: A Tool for Integrated Design and Deployment of Energy Efficient Building Controls* in *Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings* (2012), 123–130.
20. Sturzenegger, D. *et al. Model Predictive Control of a Swiss Office Building* in *11th RHEVA World Congress Clima* (2013).
21. *EnergyPlus Engineering Reference.* US Department of Energy (2014).
22. *EnergyPlus Input Output Reference* US Department of Energy (2014).
23. Kircher, K. J., Schaefer, W. & Zhang, K. M. A simple first-principles building energy model. *under review* (Sept. 2016).
24. Kircher, K. J. & Zhang, K. M. On the lumped capacitance approximation accuracy in RC network building models. *Energy and Buildings* **104,** 454–462 (2015).
25. Laret, L. *Use of general models with a small number of parameters* in *International Congress CLIMA 2000, Budapest* (1980), 263–275.
26. Crabb, J., Murdoch, N. & Penman, J. M. A simplified thermal response model. *Building Services Engineering Research & Technology* **8.1,** 13–19 (1987).
27. Wan, E. & Van Der Merwe, R. *The unscented Kalman filter for nonlinear estimation* in *The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium* (2000), 153–158.
28. Kircher, K. J. *BLDG, a MATLAB® building simulator* https://github.com/kevinjkircher/bldg. Apr. 2016.