A simple, high-fidelity testbed for building climate control

Kevin J. Kircher[a,*], Walter Schaefer[b], K. Max Zhang[a]

[a] *Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA*
[b] *Energy and Resource Solutions, New York, NY, USA*

Abstract

Advanced building climate control systems have the potential to significantly reduce greenhouse gas emissions and energy costs, but more research is needed to bring these systems to market. A key component of building control research is testing algorithms through simulation. Many high-fidelity simulation testbeds exist, but they tend to be complex and opaque to the user. Simpler, more transparent testbeds also exist, but they tend to neglect important nonlinearities and disturbances encountered in practice. In this paper, we develop a simulation testbed that is both simple and high-fidelity. We validate the testbed empirically, then demonstrate its use through the examples of system identification, online state and parameter estimation, and model predictive control. The testbed is intended to enable rapid, reliable analysis of building control algorithms, thereby accelerating progress toward reducing greenhouse gas emissions at scale.

Keywords: buildings, heating, cooling, simulation, estimation, control

1. Introduction

Heating and cooling cause about one quarter of anthropogenic greenhouse gas emissions. [1] In the U.S., these activities cost about $300 billion per year. [2] Experiments suggest that advanced controls could reduce the emissions and costs associated with heating and cooling by 10-25%. [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

The benefits of advanced heating and cooling controls are well-established, but not much research has been done into the costs of developing and deploying them. The only cost-benefit analysis in the literature concludes that, at present, the required initial investment appears to outweigh the operating cost savings. [11] To realize the potential emission reductions of advanced heating and cooling controls at scale, their cost-benefit ratio will likely need to be improved. This could be accomplished, for example, by streamlining the process of building modeling or improving the performance of estimation and control algorithms.

Many methods have been proposed for these purposes. Since 2010, for example, researchers have tested artificial neural networks [9, 12], subspace system identification [13], prediction error methods for system identification [13, 14], online state or parameter estimation with linear, extended, and unscented Kalman filters [11, 15, 16, 17, 18, 19], and model predictive control (MPC) in its certainty-equivalent [3, 4, 5, 6, 7, 8, 10, 11, 20], robust [21, 22], and stochastic [23, 24, 25, 26] forms.

It is not clear how different combinations of these methods balance performance against ease of implementation. Answering this question in general is difficult; buildings have a wide variety of geometries, construction types, mechanical systems, weather conditions and control objectives. Therefore, building control algorithms are typically evaluated and compared through case studies.

Experimental case studies in occupied buildings and varying weather conditions, such as [3, 5, 6, 7, 8, 11], allow the strongest conclusions to be drawn. They generally include the nonlinearities, disturbances, and other complications encountered in practice. Experiments have drawbacks, however. They tend to be costly and time-consuming. Experimentally comparing algorithms requires either side-by-side testbeds, or

*Corresponding author: kjk82@cornell.edu.

replicating weather conditions and occupant behavior. Some questions, such as sensitivity to construction parameters like thermal mass, or robustness to rare disturbances like extreme weather events, are difficult to investigate experimentally. For these reasons, most case studies are done in simulation testbeds.

We believe that two desirable attributes of a simulation testbed are *simplicity* and *fidelity*. Simple testbeds reduce modeling effort, make it easy to vary parameters, and enable fast simulation and clear interpretation of results. Tellingly, the majority of the case studies mentioned above considered either one [4, 10, 14, 17, 18, 21, 22, 23, 24, 26] or two [6, 13, 15, 16] thermal zones.

Testbed fidelity is also important. Real buildings have nonlinear dynamics caused, for example, by long-wave radiation and by convection with temperature-dependent film coefficients. They also have significant disturbances from weather and occupant behavior. Solar forcing, in particular, is a strong effect that is nontrivial to accurately model or predict. Simulation testbeds that neglect nonlinearities or oversimplify disturbances could bias case studies in unknown ways.

Existing simulation testbeds tend to be either simple or high-fidelity, but not both. Simple testbeds, such as the low-order resistor-capacitor networks often used for MPC, usually neglect nonlinearities or simplify solar forcing. High-fidelity testbeds, such as EnergyPlus [27] and TRNSYS [28], usually require significant effort to learn the software, model a building, vary its parameters, apply randomly generated disturbances, simulate a control algorithm, or interpret results.

Developing a simple, high-fidelity simulation testbed is the main purpose and contribution of this paper. To maximize simplicity, we restrict our attention to a family of single-zone buildings indexed by a small number of parameters. Varying these parameters generates buildings of varying size, thermal mass, insulation, draftiness, and susceptibility to solar forcing. To maximize fidelity, we model and simulate this family of buildings using the assumptions and methods that underpin state-of-the-art tools such as EnergyPlus and TRNSYS. In particular, we include nonlinear thermal radiation exchange between surfaces, nonlinear convection with temperature-dependent film coefficients, wall temperatures governed by partial differential equations, and solar radiation treated via spherical geometry and optical physics.

We call the resulting testbed and supporting functions the `bldg` toolbox (or simply `bldg`). It is implemented in Matlab, a common environment for control design and analysis, with no links to external software. At its core, `bldg` represents a building as a discrete-time system with nonlinear, time-varying dynamics. The building's thermal behavior is determined by the difference equations

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, 2, \ldots.$$

Here k indexes discrete time, x_k is the building state (a high-dimensional vector of temperatures of the indoor air and components of the building envelope), u_k is the control input (either a heat flow, or a supply air temperature and mass flow), and w_k is the disturbance (the outdoor air temperature, solar irradiance, and internal heat flows from bodies, lights, and equipment). The main purpose of `bldg` is to provide the dynamics functions f_k. This allows users to decide u_k by any method (*e.g.*, by MPC using a low-order, linear, time-invariant model that approximates the f_k), then simulate its effect in a high-order, nonlinear, time-varying testbed.

This paper is organized as follows. We review related work in §2. In §3, we discuss the mathematical model underlying the testbed. We discuss the numerical solution scheme in §4 and the required input data in §5. We empirically validate some aspects of the testbed in §6. Algorithm testing examples, including system identification, online state and parameter estimation, and MPC, are presented in §7. We conclude in §8.

2. Related work

The `bldg` toolbox is not a general-purpose building simulator. Although it has a similar mathematical structure to tools such as EnergyPlus and TRNSYS, it does not compete with them. These tools can simulate wide varieties of building geometries, construction types, and mechanical systems. They were designed for diverse purposes: comparing the energy efficiency of architectural designs, predicting the energy

savings of building retrofits, sizing heating and cooling equipment, evaluating solar photovoltaic and solar thermal systems, and many more. This design philosophy, while undoubtedly powerful, necessitates a level of complexity that renders these tools opaque to most users. By contrast, `bldg` was designed for the narrow purpose of simple, high-fidelity building control testing. Its scope is one family of single-zone buildings indexed by a small number of parameters. Its intended audience is building control researchers. Its focus is on accessibility to this community.

Two other Matlab toolboxes have recently been developed for the building control community. These are the Building Resistance-Capacitance Modeling (BRCM) [29] and OpenBuild [30] toolboxes. Both are built on the Building Controls Virtual testbed [31] and MLE+ [32], which enable communication between Matlab and EnergyPlus. Unlike `bldg`, which was designed for control *testing*, BRCM and OpenBuild were designed primarily for control *synthesis*. BRCM and OpenBuild map EnergyPlus building models into state space models and facilitate their use in MPC of real buildings. BRCM has been used in MPC of an occupied office building for seven months [11] and for numerous simulation studies [33, 34, 35]. OpenBuild has been used to simulate demand response [36] and ancillary service provision [37] from commercial buildings, and to investigate the influence of controller model order on MPC performance [38].

Unlike BRCM, OpenBuild allows controls decided in Matlab to be simulated in EnergyPlus. Therefore, OpenBuild can be a very high-fidelity, general testbed. In principle, OpenBuild can simulate as complex a building envelope as EnergyPlus can model. OpenBuild also benefits from the extensive vetting of EnergyPlus over the last two decades. Although we have validated core aspects of `bldg` empirically (see §6) and verified others against BESTEST [39] and ASHRAE Standard 140P [40] test cases, these steps are nowhere near as comprehensive as EnergyPlus' validation.

The main feature that distinguishes `bldg` from OpenBuild is that `bldg` is implemented entirely in Matlab with no connection to EnergyPlus. This may make `bldg` somewhat easier for control researchers to use; `bldg` models can be created, modified and simulated from the Matlab command line in a few lines of code. It also has the advantage of transparency. The inner workings of `bldg` can be understood with this paper and a familiarity with Matlab differential equation solvers. By contrast, OpenBuild's connection to EnergyPlus introduces the complexities described in its 1,444-page engineering reference [41] and 2,234-page input-output reference [42]. Finally, `bldg` has some computational advantages. It requires no middleware or cosimulation. Its governing equations are solved by vectorized Matlab methods that exploit structure and sparsity. This facilitates fast simulations over long time spans. For example, simulating a default building for one year with 15-minute time steps takes about 6 seconds on a 1.2 GHz Intel Core M processor.

In summary, BRCM and OpenBuild are powerful tools for synthesizing control-oriented building models. The `bldg` toolbox complements them by enabling algorithm evaluation in a transparent, native Matlab testbed. While OpenBuild is another excellent testbed option, `bldg` has some advantages that may be appealing for research such as sensitivity analysis and Monte Carlo simulation. The tools could be used in series, with `bldg` facilitating rapid prototyping and comparison of algorithms, and BRCM or OpenBuild used to investigate scalability to large multizone buildings.

3. Mathematical model

3.1. Geometry and heat transfer mechanisms

The buildings simulated by `bldg` are a family of single-zone rooms. They have one single-pane window and walls of uniform composition. The building family is indexed by the dimensions of the room and the material properties of the window and walls. Figure 1 illustrates an example building and its environment.

The building is coupled to the sky and ground by longwave radiation, and to the outdoor air by convection and infiltration through gaps in the building envelope. The walls and window receive beam and diffuse sunlight depending on the cloud cover and time of day. The walls and window exchange heat with each other through radiation, and with the indoor air through convection. Heat is generated internally by occupants' bodies, lights, heating and cooling systems, and other equipment.

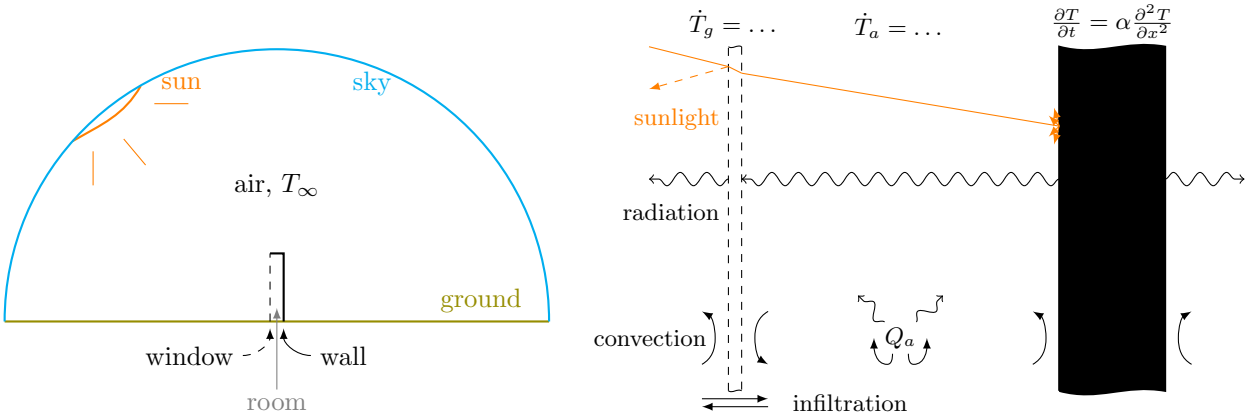

Figure 1: Geometry of the environment (left) and a cross-section of an example building (right). Heat is transferred through convection, radiation, conduction, infiltration, and the transmission, absorption, and reflection of visible light. The internal heat flow Q_a is from bodies, mechanical systems, lights, and other equipment.

3.2. Assumptions

A standard set of assumptions underlies the modeling techniques used by state of the art building simulators. [27] Under these assumptions, all material properties are spatially uniform and independent of temperature. The air in each thermal zone is well-mixed and radiatively nonparticipating. Conduction through surfaces is one-dimensional. Each surface is modeled as isothermal and, with respect to longwave radiation, as opaque, diffuse, and gray. Optical properties are assumed to vary between the shortwave and longwave bands of the electromagnetic spectrum, but within each band they are independent of wavelength. We make all of the above assumptions in the remainder of this paper. In addition, we assume that the outdoor air is dry, still, and isothermal with the sky and ground. Furthermore, we assume that heat transfer is dominated by flows in the direction normal to the window surface, so that it can be treated as one-dimensional.

3.3. Notation

We denote temperature by T (°K), heat flow by Q (W), heat flux by q (W/m^2), convection coefficients by h (W/m^2·°K), and solar irradiance by I (W/m^2). The subscripts g, a, and ∞ denote window glass, indoor air, and the outdoor environment far from the building; quantities absent these subscripts refer to the wall opposite the window. The subscripts $-$ and $+$ indicate left and right boundaries, given the orientation depicted in Figure 1. Initial temperatures have the superscript 0. We denote the beam component of solar irradiance by the superscript b and the diffuse component by d; the subscripts h and $\perp$ indicate incidence on a horizontal surface and a surface normal to the sun. For example, h_{g-} is the convection coefficient at the left glass surface, T_a^0 is the initial air temperature, and I_+^b is the beam irradiance on the right wall surface. For internal heat sources, the subscripts p, c, l, and e indicate people, control systems, lights, and other equipment.

3.4. Governing equations

Under the assumptions in §3.2, the temperature distribution $T(x,t)$ in the wall opposite the window satisfies the heat equation with initial and boundary conditions:

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

Here α (m^2/s) and k (W/m·°K) are the wall's thermal diffusivity and conductivity. The left and right surface fluxes q_- and q_+ (W/m^2) are discussed in §3.5 and §3.6. Heat fluxes and flows to the right are taken to be positive.

As demonstrated in [43], the window temperature dynamics can be accurately modeled by

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

Here C_g (J/°K) is the thermal capacitance of the window and A (m^2) is the window surface area. The fluxes q_{g-} and q_{g+} (W/m^2·°K) at the left and right window surfaces are discussed in §3.5 and §3.6.

Under the well-mixed assumption, the indoor air temperature satisfies

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

where C_a (J/°K) is the thermal capacitance of the air and any material that's isothermal with it. The fluxes q_{a-} and q_{a+} at the left and right air boundaries are discussed in §3.6. The internal heat flow is

$$\begin{aligned} Q_a = {} & \dot{m}_\infty c_a (T_\infty - T_a) \\ & + \zeta_c Q_c + \zeta_p Q_p + \zeta_l Q_l + \zeta_e Q_e, \end{aligned} \tag{4}$$

where $\dot{m}_\infty$ (kg/s) is the mass flow rate of outdoor air entering the space by infiltration, c_a (J/kg·°K) is the specific heat of air at constant pressure, and $\zeta_c Q_c$, $\zeta_p Q_p$, $\zeta_l Q_l$, and $\zeta_e Q_e$ (W) are the convective heat flows from control systems, people, lighting, and other equipment, respectively. Each ζ_* is between zero and one; the remaining fraction $1 - \zeta_*$ of Q_* radiates to the wall and window. The internal heat sources are discussed in §5.1

3.5. Outdoor heat fluxes

The flux at the outdoor surface of the wall is

$$\begin{aligned} q_+ = {} & h_+ (T_+ - T_\infty) + \sigma\varepsilon (T_+^4 - T_\infty^4) \\ & - \alpha_s (I^d + I_+^b), \end{aligned} \tag{5}$$

where $\sigma = 5.67 \times 10^{-8}$ W/m^2°K^4 is the Stefan-Boltzmann constant, ε is the wall's longwave emissivity, α_s is its shortwave absorptivity, I^d (W/m^2) is the (isotropic) diffuse solar irradiance, and I_+^b (W/m^2) is the beam component of the solar irradiance incident on the wall. The longwave term in equation (5) follows from the fact that the radiative transfer factor between a small, convex, gray object (in this case, the wall's outer surface) and the much larger, isothermal environment that contains it is simply the small object's emissivity; see §10.4 of [44] for details.

The flux at the outdoor surface of the window is

$$\begin{aligned} q_{g-} = {} & h_{g-} (T_\infty - T_g) + \sigma\varepsilon_g (T_\infty^4 - T_g^4) \\ & + \alpha_g(\theta_g) I_{g-}^b + \bar{\alpha}_g I^d, \end{aligned} \tag{6}$$

where ε_g is the longwave emissivity of the glass and I_{g-}^b (W/m^2) is the beam component of the incident sunlight. The diffuse sunlight I^d is assumed to be isotropic, so it equally irradiates the wall and window. Following §1.5 of [45], the absorptivity of beam sunlight $\alpha_g(\theta_g)$ and the absorptivity of diffuse sunlight $\bar{\alpha}_g$ are derived from the beam angle of incidence θ_g and the index of refraction n_g, attenuation coefficient μ_g (m^{-1}), and thickness l_g of the window glass. The derivation uses Snell's law, the Fresnel equation and Beer's law.

The convection coefficients h_+ and h_{g-} (W/m^2°K) depend nontrivially on the difference between the indoor wall surface temperature (or window, respectively) and the outdoor air temperature T_∞. The EnergyPlus engineering reference [41] catalogs 37 different empirical formulas for convection coefficients at

surfaces of various orientations. For simplicity, we use the ASHRAE model for natural convection along a vertical surface,

$$h = 1.31\,|\Delta T|^{1/3}\,.$$

Here ΔT is the temperature difference between the surface and the surrounding air. The dependence of h on ΔT makes this model of convection nonlinear in the temperatures of the indoor air and the wall and window surfaces.

3.6. Indoor heat fluxes

The flux at the indoor surface of the wall is

$$q_- = q_{a+} + q_-^{\text{short}} + q_-^{\text{long}}, \tag{7}$$

where

$$q_{a+} = h_-(T_a - T_-)$$

is the convective flux from the indoor air to the wall. Similarly, the indoor window surface flux is

$$q_{g+} = q_{a-} + q_{g+}^{\text{short}} + q_{g+}^{\text{long}}, \tag{8}$$

where

$$q_{a-} = h_{g+}(T_g - T_a)$$

is the convective flux from the window to the indoor air. For the indoor surface film coefficients, we use a model for a vertical surface near a heating/cooling source [46],

$$h = 1.98\,|\Delta T|^{0.32}\,.$$

The shortwave radiation fluxes q_-^{short} and q_{g+}^{short} involve three sources: beam and diffuse sunlight transmitted through the window and artificial light from indoors. They can be computed by ray tracing; under the diffusivity and isotropy assumptions in §3.2, the results can be obtained in closed form by summing geometric series.

The longwave radiation fluxes q_-^{long} and q_{g+}^{long} involve two effects: gray body radiation exchange between the window wall surfaces, and the radiative components of indoor heat flows from bodies, control systems, lights and other equipment. The longwave fluxes from both effects can be computed using the radiosity method. [47] For a generic enclosure with n surfaces, this involves computing the n^2 view factors between the surfaces, then numerically solving a system of n linear equations. For our simple case with two dominant indoor surfaces and view factors close to one, the fluxes can be computed in closed form.

4. Numerical solution

Simulating the building requires solving the governing equations (1), (2), and (3). The complexity of the boundary conditions and disturbances makes analytical solution intractable, so we use a finite difference approach called the numerical method of lines. [48, 49, 50] The method of lines involves approximating spatial derivatives with algebraic expressions, reducing the PDE (1) to a system of ODEs that can be combined with equations (2) and (3), then fed to an ODE solver. This is accomplished through the central function in the `bldg` toolbox, `bsim`.

4.1. Space discretization

We define the space coordinate x such that $x = 0$ at the left-hand side of the wall and $x = l$ at the right. To discretize space, we introduce N nodes on $[0, l]$, uniformly spaced by $\Delta x = l/(N-1)$ and indexed by $i \in \{1, \ldots, N\}$ such that $x_i = (i-1)\Delta x$. We approximate each temperature $T(x_i, t)$ by a function $T_i(t)$, chosen such that $T_i(t) \to T(x_i, t)$ as $N \to \infty$.

The central difference approximation to the spatial derivative in the heat equation, evaluated at x_i, is

$$\frac{\partial}{\partial t}T(x_i,t) = \alpha\left(\frac{T(x_{i-1},t) - 2T(x_i,t) + T(x_{i+1},t)}{(\Delta x)^2}\right) + \mathcal{O}((\Delta x)^2).$$

Dropping the truncation error term and replacing the true temperatures by their approximations gives

$$\dot{T}_i = r(T_{i-1} - 2T_i + T_{i+1}), \tag{9}$$

where we have suppressed the time arguments and defined $r = \alpha/(\Delta x)^2$.

Evaluating equation (9) at each $i \in \{1, \ldots, N\}$ defines N ODEs in the $N+2$ temperatures $T_0, \ldots, T_{N+1}$. Since $x_0 = -\Delta x$ and $x_{N+1} = l + \Delta x$, however, T_0 and T_{N+1} represent temperatures outside of the wall. They can be eliminated using the boundary conditions

$$-k\left.\frac{\partial T}{\partial x}\right|_{0,t} = q_-, \quad -k\left.\frac{\partial T}{\partial x}\right|_{l,t} = q_+.$$

To preserve the $\mathcal{O}((\Delta x)^2)$ truncation error of the method, we replace the spatial derivatives by their central difference approximations,

$$\left.\frac{\partial T}{\partial x}\right|_{0,t} = \frac{T(x_2,t) - T(x_0,t)}{2\Delta x} + \mathcal{O}((\Delta x)^2)$$
$$\left.\frac{\partial T}{\partial x}\right|_{l,t} = \frac{T(x_{N+1},t) - T(x_{N-1},t)}{2\Delta x} + \mathcal{O}((\Delta x)^2).$$

Dropping the truncation errors and replacing true by approximate temperatures, we have

$$T_0 = T_2 + \left(\frac{2\Delta x}{k}\right) q_-$$
$$T_{N+1} = T_{N-1} - \left(\frac{2\Delta x}{k}\right) q_+.$$

Substituting these expressions into the approximate heat equation at x_1 and x_N gives

$$\begin{aligned} \dot{T}_1 &= 2r\,(-T_1 + T_2) + \left(\frac{2\alpha}{k\Delta x}\right) q_- \\ \dot{T}_N &= 2r\,(T_{N-1} - T_N) - \left(\frac{2\alpha}{k\Delta x}\right) q_+. \end{aligned} \tag{10}$$

4.2. Initial value problem

Equations (10), along with equation (9) evaluated at each $i \in \{2, \ldots, N-1\}$, defines a system of N ODEs in $T_1, \ldots, T_N$. These ODEs involve the internal surface flux q_-, which couples the system to equations (2)

and (3) for T_g and T_a. The full problem is therefore

$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \\ \vdots \\ \dot{T}_{N-1} \\ \dot{T}_N \\ \dot{T}_g \\ \dot{T}_a \end{bmatrix} = \begin{bmatrix} 2r\left(-T_1 + T_2\right) + 2\alpha q_- / k\Delta x \\ r(T_1 - 2T_2 + T_3) \\ \vdots \\ r(T_{N-2} - 2T_{N-1} + T_N) \\ 2r\left(T_{N-1} - T_N\right) - 2\alpha q_+ / k\Delta x \\ A(q_{g-} - q_{g+})/C_g \\ (A(q_{a-} - q_{a+}) + Q_a)/C_a \end{bmatrix}, \tag{11}$$

with the corresponding initial condition. The fluxes q_-, q_+, q_{g-}, q_{g+}, q_{a-}, and q_{a+}, defined in §3.5 and §3.6, are polynomials of degree up to four in the temperatures, so system (11) is nonlinear. The solar fluxes depend on the sun's position in the sky, so the system is also time-varying.

Depending on the physical parameters, control scheme, and choice of N, system (11) may be stiff. An implicit solver should therefore be used. [48] In [51], Crowley compared fifteen numerical methods for a similar problem, and recommended combining the trapezoidal rule and the second-order backward difference formula, as implemented in the Matlab solver `ode23tb`. This scheme has second-order accuracy in both time and space. It is also possible to simulate perfect control, where the indoor temperature is maintained exactly at the setpoint. In this case, the last ODE reduces to the algebraic equation $A(q_{a-} - q_{a+}) + Q_a = 0$, and Problem (11) becomes a system of differential-algebraic equations that can be solved, *e.g.*, by Matlab's `ode15s` solver.

Solving system (11) numerically requires specifying the initial temperatures in the wall, window, and air. Reasonable initial conditions can be produced by preconditioning the building as follows. For a simulation starting at time t_0, the building state can be initialized isothermal with the environment some time earlier, for example one or two weeks, then integrated forward to t_0 under the appropriate weather conditions and internal heat sources.

5. Input data

Simulating the building requires specifying the model parameters, an initial state, the solar time span, and values of the input signals in Table 1 at every time step. The input signals can be divided into internal heat sources (§5.1) and weather (§5.2). We discuss the model parameters in §5.3.

5.1. Internal heat sources

The internal sources are the heat flows Q_c, Q_p, Q_l, and Q_e from control systems, people, lighting, and other equipment. Building simulators typically require user-specified schedules for occupancy, lighting, and equipment, from which Q_p, Q_l, and Q_e are determined.

In general, the control heat flow Q_c may depend on the control objectives, the building state, and predictions of weather and internal heat sources over the control horizon. We assume that Q_c is either provided by the user, or computed internally in order to perfectly regulate T_a at a specified setpoint. In the latter case, $\dot{T}_a = 0$, and equations (3) and (4) give the heating load of the building:

$$\begin{aligned} Q_c = \frac{1}{\zeta_c}\big(&A(q_{a+} - q_{a-}) - (\dot{m}_\infty c_a (T_\infty - T_a) \\ &+ \zeta_p Q_p + \zeta_l Q_l + \zeta_e Q_e)\big). \end{aligned} \tag{12}$$

Perfect regulation gives useful estimates of peak heating and cooling loads.

Table 1: The input signals to the building model include the internal heat flows and weather.

Signal	Symbol	Unit
Control heat flow	Q_c	W
Heat flow from people	Q_p	W
Heat flow from lighting	Q_l	W
Heat flow from equipment	Q_e	W
Outdoor air temperature	T_∞	°K
Total horizontal irradiance	I_h	W/m^2
Beam normal irradiance	$I^b_\perp$	W/m^2

Table 2: The input parameters to the building model include the building geometry and material properties.

Parameter	Units
Wall thickness, l	m
Glass thickness, l_g	m
Room width, l_a	m
Wall surface area, A	m^2
Wall azimuth angle, γ	rad
Latitude, ϕ	rad
Wall thermal diffusivity, α	m^2/s
Wall thermal conductivity, k	W/m·°K
Wall longwave emissivity, ε	-
Wall shortwave absorptivity, α_s	-
Glass thermal capacitance, C_g	J/°K
Glass longwave emissivity, ε_g	-
Glass index of refraction, n_g	-
Glass attenuation coefficient, μ_g	m^{-1}
Room thermal capacitance, C_a	J/°K
Infiltration mass flow rate, $\dot{m}_\infty$	kg/s
Control convective fraction, ζ_c	-
People convective fraction, ζ_p	-
Lighting convective fraction, ζ_l	-
Equipment convective fraction, ζ_e	-
Lighting efficiency, η	-

5.2. Weather

We take as basic weather inputs only quantities that are easily measured: The outdoor air temperature T_∞, the total solar irradiance I_h on a horizontal surface (measured with a pyranometer), and the beam solar irradiance $I^b_\perp$ on a surface normal to the sun (measured with a pyrheliometer). These data are also available in typical meteorological year files for various locations. [52] As discussed in §1.6 of [53], the beam angles of incidence on the wall and window, θ and θ_g, can be computed from the day, time, latitude, longitude, and building orientation. Similarly, the diffuse irradiance I^d and the beam components I^b_+ and I^b_{g-} can be inferred from I_h and $I^b_\perp$.

5.3. Parameters

The building is defined by the 21 parameters in Table 2; all other parameters are derived from them. Six of these parameters define the orientation and dimensions of the building. Nine are material properties: Four for the walls, four for the window glass, and one for the room. The remaining six parameters are associated with the internal heat sources.

6. Empirical validation

In this section, we empirically validate the model and simulator through a series of frequency response experiments. In each experiment, known input signals are sent to (1) a scaled laboratory testbed and (2) `bldg` with parameters reflecting the laboratory testbed's geometry and material properties. Temperature responses within the testbed are measured and compared to the corresponding simulator outputs. The experiments validate the numerical solution scheme and the conduction, convection and longwave radiation models.

The empirical testbed is an enclosure with five plywood surfaces and one clear acrylic window. The side, top, and bottom surfaces are insulated with rigid polystyrene and radiation shielded with low-emissivity aluminum foil. This directs heat transfer primarily through the window and the wall opposite. Joints are tightly sealed; the ambient airflow into the box is negligible.

The `bldg` parameters are determined as follows. The lengths l, l_g and l_a and area A are measured directly. The material properties α, k, ε, ε_g, and C_g are drawn from engineering tables. Nominally, the indoor air's thermal capacitance C_a is the product of its volume, density, and specific heat at constant pressure. As discussed in the section titled *Zone Sensible Heat Capacity Multiplier* of the EnergyPlus engineering reference [41], however, it is common practice to increase C_a over this nominal value to account for unmodeled thermal mass that is nearly isothermal with the air. In keeping with this practice, we increase C_a by an order of magnitude.

Controlled heat flows are supplied to the box through two resistive heating elements, each thermally glued to an aluminum heat sink. Temperatures are measured by fifteen thermistors accurate to ± 0.1 °C. Twelve thermistors are arranged in two lines normal to the window surface, with one line of six at one quarter of the wall height and another at three quarters. Each line measures the inner and outer window surface temperature, the indoor air temperature, the inner and outer wall surface temperatures, and the temperature at the midpoint of the wall. The remaining three thermistors measure the ambient temperature outside the box at different locations. Control and measurement signals are exchanged between the testbed and a laboratory computer running Matlab and a National Instruments sensor toolbox.

Figure 2 shows the inputs and outputs of one experiment. Each experiment involves a six-hour warm-up phase: the heating system is turned on to half capacity and the box temperatures settle into a steady state. After the warm-up phase, the internal heat flow is varied sinusoidally between zero and full capacity for ten oscillations. Ten such experiments are conducted. Oscillation periods vary from 17 minutes (a nine-hour experiment, including the six warm-up hours and ten oscillations) to 27 hours (an 11-day experiment). The total experiment runtime, including ten oscillations at each of the ten frequencies, is about 31 days.

Figure 3 summarizes the results of the ten frequency response experiments. The top row of plots shows the amplitudes of the temperature waveforms, in units of decibels, caused by control heat flow sinusoids of varying frequencies. The bottom row of plots shows the phase shift between the temperature and heat flow waveforms. Each error bar shows one standard deviation from the mean of ten oscillations at each frequency in the experimental data. The model accurately predicts the full temperature waveforms (both amplitudes and phases) caused by oscillating heat flows at most frequencies. This empirical validation supports both the mathematical model and the numerical solution scheme.

7. Examples

In this section, we demonstrate use of the `bldg` toolbox through the examples of system identification (§7.1), online state and parameter estimation (§7.2), and model predictive control (§7.3). Along the way, we highlight several open research questions that `bldg` could help investigate. Further examples are provided in the `bldg` documentation. [54]

The context of the examples in this section is the building described in §3 during winter in New York City. A building model with default parameter values can be created by instantiating an object of class `bldg`:

```
b = bldg;
```

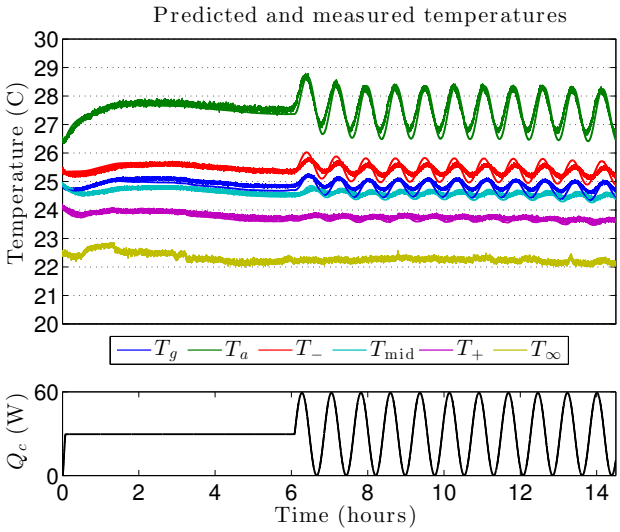

Figure 2: The temperatures (top plot) and control heat flow (bottom) in an experiment with oscillation period of 3.5 hours. Thick, fuzzy lines are measurements. Thin lines are predictions. Nine similar experiments were conducted to build the frequency response plots in Figure 3.

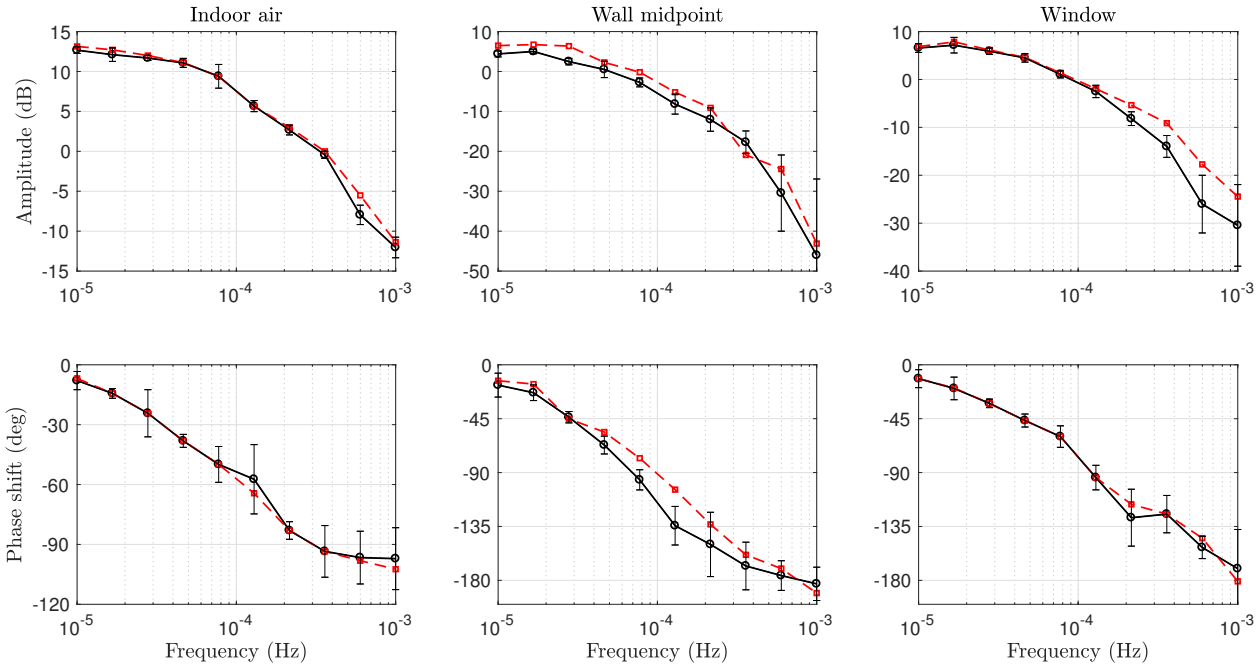

Figure 3: Predicted (dashed red) and measured (solid black) frequency responses. The temperature amplitudes (top row) and phase shifts (bottom row) agree closely with measurements over the 31 days of total experiment time.

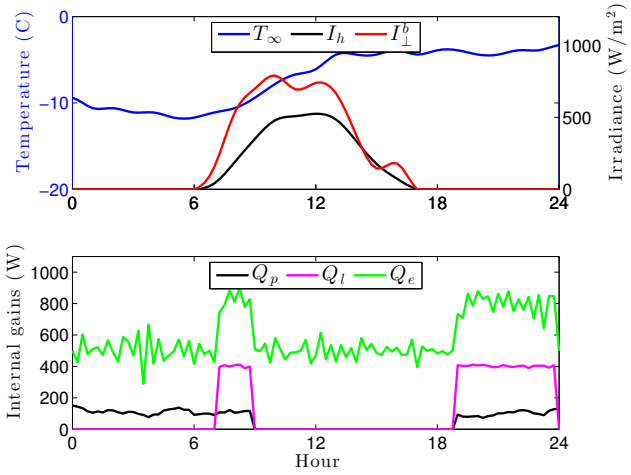

Figure 4: Input signals from weather (top) and internal heat sources (bottom). The outdoor air temperature (T_∞), total solar irradiance on a horizontal surface (I_h), and beam solar irradiance on a surface normal to the sun ($I_\perp^b$) show that the simulation day is cold and partly cloudy. The internal 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.

Under the default parameter values, the building is thermally massive and highly susceptible to solar forcing. The user can modify the building by adjusting parameters in the `bldg` object `b`. For example, the thermal mass can be decreased by increasing the wall diffusivity α (`b.a`) or thickness l (`b.l`). The solar forcing can be weakened by decreasing the wall shortwave absorptivity α_s (`b.as`) or by changing the window orientation through the azimuth angle γ (`b.gam`).

The following code 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 `bldg` toolbox uses SI units throughout, so times are given in seconds.) The last line packs the disturbances into the matrix form accepted by the building simulation function `bsim`.

```
dt = 15*60;
[b,weather] = importWeather(b,'NYC_TMY3.csv',dt);
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 4 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);
```

As discussed in §4.2, the `precondition` function starts the building isothermal with the outdoor air two weeks before t_0. It then integrates the building forward with the indoor air temperature 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 5 shows the output.

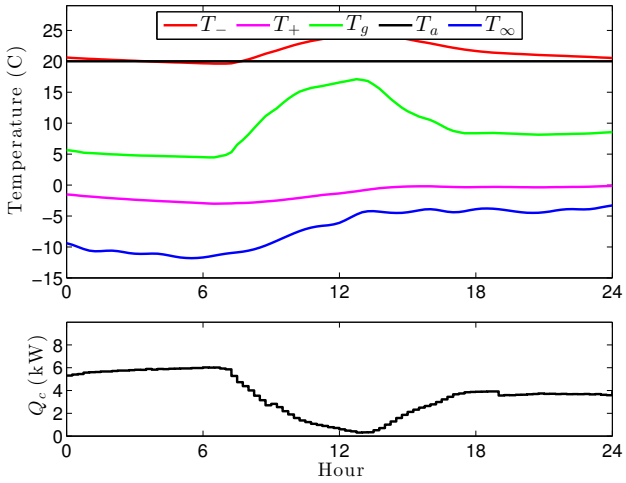

Figure 5: Temperatures of the air (T_a), glass (T_g), left and right wall surfaces (T_- and T_+), and outdoor environment (T_∞). The bottom plot shows the heat flow required to perfectly regulate T_a at setpoint $T_s = 20$ C under the input signals in Figure 4.

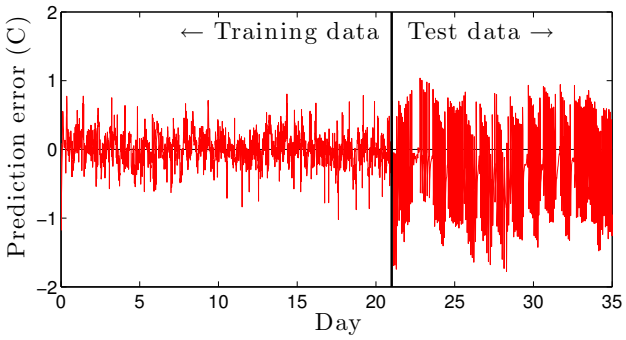

Figure 6: One-step prediction errors in the training and test data with the static model in §7.1. The model fit is poor: The errors are autocorrelated and, in the test set, predictions are biased by −0.31 °C.

7.1. 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

$$T_a^{k+1} = \beta_1 T_a^k + \beta_2 Q_c^k + \beta_3 T_\infty^k + \beta_4 I_h^k + w_a^k, \tag{13}$$

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 §7.3.

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, 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 6 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

1.5 °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 [14]. The authors found that a second-order RC network with five parameters is able to capture the essential building behavior (confirming a result from the 1980s; see [55, 56]). 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 (`b.a`) and shortwave absorptivity α_s (`b.as`) and studying the model fit.

Another interesting topic is the relationship between gray- and black-box models. The parameters in model (13), 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 thermal capacitance, thermal resistance, and solar absorption area. Forward Euler discretization gives

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

This is consistent with the ARX model (13) only if $\beta_1 + \beta_3 = 1$. In this case, the 'cost of grayness' could be defined as the prediction accuracy lost by imposing the constraint $\beta_1 + \beta_3 = 1$ on the least-squares fitting problem, along with any resulting controller performance reduction. 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.

7.2. 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, temperature control equipment differs in heating and cooling seasons, and material properties change as materials age. 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 [15, 16, 17, 18, 19].

In this example, we consider the problem of adapting a subset of the parameters of the ARX model identified in §7.1. The goal is to reduce the prediction bias apparent in Figure 6. We simultaneously estimate the indoor air temperature and model parameters using an unscented Kalman filter. [57] We allow the parameters β_1 and β_3, initialized with the fit from §7.1, to vary under the random walk model $\beta_i^{k+1} = \beta_i^k + w_{\beta_i}^k$. This gives the augmented system model

$$\begin{aligned} x^{k+1} &= f(x^k, u^k) + w^k \\ y^k &= x_1^k + v^k, \end{aligned} \tag{14}$$

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 §7.1 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 7 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 8. 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,

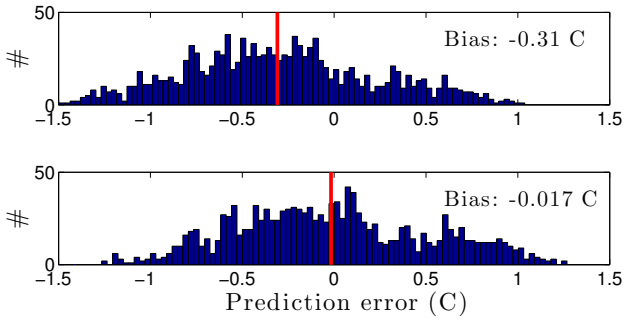

Figure 7: Histograms of the one-step prediction errors without parameter adaptation (top), and with it (bottom). The online estimation algorithm discussed in §7.2 nearly eliminates the prediction bias. This is accomplished by adjusting the model parameters β_1 and β_3, as shown in Figure 8.

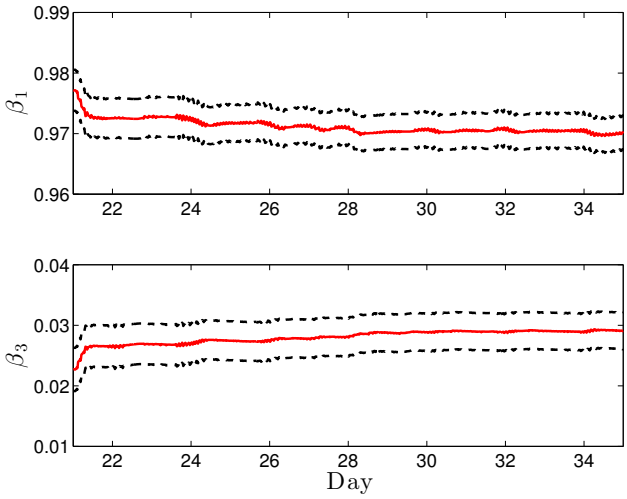

Figure 8: Unscented Kalman filter parameter estimates and 95% confidence intervals during the two test weeks. The filter remains stable after initially adjusting β_1 and β_3 to reduce prediction bias.

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, $\dot{m}_\infty$ (`b.mdot`), or the fraction of the control heat that convects to the indoor air, ζ_c (`b.zc`).

7.3. Model predictive control

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

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\,\Delta t \sum_{k=0}^{M} Q_c^k \\ \text{subject to} & T_a^{k+1} = (\text{nonlinear } \texttt{bsim} \text{ dynamics}) \\ & y^k = T_a^k + v^k \\ & T_a^{k+1} \geq T^{\min} \\ & Q_c^k = \mu^k(y^0, \ldots, y^k) \in [0, Q_c^{\max}], \end{array} \tag{15}$$

where the constraints should hold for each $k = 0, \ldots, M$ almost surely. 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

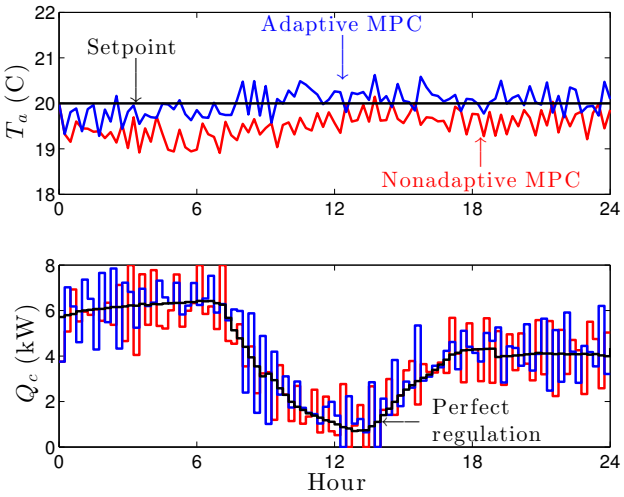

Figure 9: Indoor air temperature T_a (top) and control heat flow Q_c (bottom) under MPC, with and without parameter adaptation. Both controllers attempt to regulate T_a just above the minimum permissible temperature of 20 °C, giving heat flows that resemble the 'perfect regulation' case computed by `bsim`. Both MPC variants frequently under-heat due to model error. Parameter adaptation mitigates, but does not eliminate, constraint violation.

taken with respect to the joint distribution of the disturbance and noise sequences and the initial state.

Problem (15) 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 (15) in order to illustrate the research value of the `bldg` toolbox. The first MPC variant uses the ARX model from §7.1 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 (15) 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 §7.2. 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 9 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 4. 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 §5. 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.

The constraint violations are an artefact of model mismatch between the controller and the underlying `bsim` dynamics. They could be reduced by using a more accurate, higher-order model, by specifying a safety margin (*e.g.*, by replacing the constraint $T_a^k \geq T^{\min}$ with $T_a^k \geq T^{\min} + \delta$ for some positive δ), or by moving the MPC optimization to a robust or stochastic framework. While several studies have applied robust or stochastic MPC to buildings [21, 22, 23, 24, 25, 26], it is not clear how these approaches, which add significant conceptual and computational complexity, compare to using a more accurate model or specifying safety margins.

8. Conclusion

In this paper, we developed a simple, high-fidelity Matlab testbed for building climate control algorithms. After reviewing related work, we developed a nonlinear building model and a numerical solution scheme. The model has a small number of governing equations, parameters, and input signals. The numerical solution scheme uses built-in Matlab differential equation solvers that are familiar to many control engineers. We hope that these properties make the `bldg` toolbox accessible and transparent to building control researchers.

The `bldg` toolbox is also sufficiently high-fidelity to be useful for prototyping, analyzing, and comparing building estimation and control algorithms. We supported this claim through empirical validation and the examples of system identification, online state and parameter estimation, and model predictive control. We highlighted several open research questions that the `bldg` toolbox could help investigate.

The scope of the `bldg` toolbox is intentionally narrow. It can simulate one family of single-zone buildings indexed by a small number of parameters. This simplicity has conceptual and computational advantages, as evidenced by the large number of high-quality studies done in one- or two-zone testbeds. [4, 6, 10, 13, 14, 15, 16, 17, 18, 21, 22, 23, 24, 26] However, `bldg`'s narrow scope prevents its use for some important research, such as investigating scalability to large multizone buildings. For such research, we recommend the BRCM [29] and OpenBuild [30] toolboxes.

The `bldg` toolbox is free, open source, and available online at [54]. It is intended to eliminate the need to model a building or leave the Matlab environment before simulating an estimation or control algorithm in a high-fidelity testbed. We hope that this facilitates rapid, reliable research into advanced building controls, accelerating progress toward realizing their environmental and economic benefits at scale.

References

[1] O. Lucon, D. Urge-Vorsatz, A. Ahmed, H. Akbari, P. Bertoldi, L. Cabeza, N. Eyre, A. Gadgil, Chapter 9 - Buildings, in: Climate Change 2014: Mitigation of Climate Change. IPCC Working Group III Contribution to AR5, Cambridge University Press.

[2] Buildings energy databook, chapter 1: Buildings sector, Tech. rep., Office of Energy Efficiency and Renewable Energy, US Department of Energy (2011).

[3] Z. Liao, A. L. Dexter, An inferential model-based predictive control scheme for optimizing the operation of boilers in building space-heating systems, Control Systems Technology, IEEE Transactions on 18 (5) (2010) 1092–1102.

[4] A. Aswani, N. Master, J. Taneja, D. Culler, C. Tomlin, Reducing Transient and Steady State Electricity Consumption in HVAC Using Learning-Based Model-Predictive Control, Proceedings of the IEEE 100.1 (2011) 240–253.

[5] J. Siroky, F. Oldewurtel, J. Cigler, S. Privara, Experimental Analysis of Model Predictive Control for an Energy Efficient Building Heating system, Applied Energy 88 (2011) 3079–3087.

[6] M. Schuss, R. Zach, K. Orehounig, A. Mahdavi, Empirical evaluation of a predective simulation-based control method, in: Proceedings of the 12th International IBPSA Conference, 2011, pp. 918–925.

[7] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, P. Haves, Model predictive control for the operation of building cooling systems, Control Systems Technology, IEEE Transactions on 20 (3) (2012) 796–803.

[8] S. Bengea, A. Kelman, F. Borrelli, R. Taylor, S. Narayanan, Model Predictive Control for Mid-size Commercial Building HVAC: Implementation, Results and Energy Savings, in: International Conference on Building Energy and Environment, 2012.

[9] P. M. Ferreira, A. E. Ruano, S. Silva, E. Z. E. Conceicao, Neural networks based predictive control for thermal comfort and energy savings in public buildings, Energy and Buildings 55 (2012) 238–251.

[10] M. Castilla, J. D. Alvarez, J. E. Normey-Rico, F. Rodríguez, Thermal comfort control using a non-linear MPC strategy: A real case of study in a bioclimatic building, Journal of Process Control 24 (6) (2014) 703–713.

[11] D. Sturzenegger, D. Gyalistras, M. Morari, R. Smith, Model predictive climate control of a Swiss office building: Implementation, results, and cost-benefit analysis, Control Systems Technology, IEEE Transactions on.

[12] H. Huang, L. Chen, E. Hu, A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study, Building and Environment 89 (2015) 203–216.

[13] S. Privara, J. Cigler, Z. Vana, F. Oldewurtel, C. Sagerschnig, Building modeling as a crucial part for building predictive control, Energy and Buildings 56 (2013) 8–22.

[14] Y. Lin, T. Middelkoop, P. Barooah, Issues in identification of control-oriented thermal models of zones in multi-zone buildings, in: Conference on Decision and Control, IEEE, 2012.

[15] P. Radecki, B. Hencey, Online building thermal parameter estimation via unscented Kalman filtering, in: American Control Conference, 2012, pp. 3056–3062.

[16] P. Radecki, B. Hencey, Online thermal estimation, control, and self-excitation of buildings, in: Conference on Decision and Control, IEEE, 2013, pp. 4802–4807.

[17] S. Fux, A. Ashouri, M. Benz, L. Guzzella, EKF based self-adaptive thermal model for a passive house, Energy and Buildings 68 (2014) 811–817.
[18] M. Maasoumy, B. Moridian, M. Razmara, M. Shahbakhti, A. Sangiovanni-Vincentelli, Online simultaneous state estimation and paramater adaptation for building predictive control, in: ASME Dynamic Systems and Control Conference, 2013.
[19] A. Martincevic, A. Starcic, M. Vasak, Parameter estimation for low-order models of complex buildings, in: Innovative Smart Grid Technologies Conference Europe, IEEE, 2014, pp. 1–6.
[20] K. J. Kircher, K. M. Zhang, Model predictive control of thermal storage for demand response, in: American Control Conference (ACC), 2015, pp. 956 – 961.
[21] M. Maasoumy, M. Razmara, M. Shahbakhti, A. Sangiovanni-Vincentelli, Handling model uncertainty in model predictive control for energy efficient buildings, Energy and Buildings 77 (2014) 377–392.
[22] M. Tanaskovic, D. Sturzenegger, R. Smith, M. Morari, Robust adaptive model predictive building climate control, in: IFAC-PapersOnLine, Vol. 50, 2017, pp. 1871–1876.
[23] F. Oldewurtel, A. Parisio, C. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, M. Morari, Use of model predictive control and weather forecasts for energy efficient building climate control, Energy and Buildings 45 (2012) 15–27.
[24] F. Oldewurtel, C. Jones, A. Parisio, M. Morari, Stochastic Model Predictive Control for Building Climate Control, Control Systems Technology, IEEE Transactions on 22 (3) (2014) 1198–1205.
[25] Y. Ma, J. Matusko, F. Borrelli, Stochastic Model Predictive Control for Building HVAC Systems: Complexity and Conservatism, Control Systems Technology, IEEE Transactions on 23.1 (2015) 101–116.
[26] K. J. Kircher, K. M. Zhang, Sample-average model predictive control of uncertain linear systems, in: Conference on Decision and Control, 2016, pp. 6234–6239.
[27] D. Crawley, L. Lawrie, F. C. Winkelman, W. Buhl, Y. J. Huang, C. O. Pederson, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, J. Glazer, Energyplus: creating a new-generation building energy simulation program, Energy and Buildings 33 (2001) 319–331.
[28] S. Klein, W. Beckman, J. Mitchell, J. Duffie, N. Duffie, T. Freeman, J. Mitchell, M. Kummert, TRNSYS 16 - A TRaNsient system simulation program, user manual, Solar Energy Laboratory, University of Wisconsin-Madison, Madison, Wisconsin (2004).
[29] D. Sturzenegger, D. Gyalistras, V. Semeraro, M. Morari, R. Smith, BRCM Matlab Toolbox: Model Generation for Model Predictive Building Control, in: American Control Conference, 2014, pp. 1063–1069.
[30] T. Gorecki, F. Qureshi, C. Jones, OpenBuild: An integrated simulation environment for building control, in: IEEE Conference on Control Applications (CCA), 2015, pp. 1522–1527.
[31] M. Wetter, Co-Simulation of Building Energy and Control Systems with the Building Controls Virtual Test Bed, Journal of Building Performance Simulation 4 (3) (2011) 185–203.
[32] W. Bernal, M. Behl, T. Nghiem, R. Mangharam, 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, pp. 123–130.
[33] G. Darivianakis, A. Georghiou, R. S. Smith, J. Lygeros, The power of diversity: Data-driven robust predictive control for energy-efficient buildings and districts, IEEE Transactions on Control Systems Technology.
[34] S. K. Gupta, K. Kar, S. Mishra, J. T. Wen, Singular perturbation method for smart building temperature control using occupant feedback, Asian Journal of Control.
[35] C. Finck, R. Li, R. Kramer, W. Zeiler, Quantifying demand flexibility of power-to-heat and thermal energy storage in the control of building heating systems, Applied Energy.
[36] F. Qureshi, T. Gorecki, C. Jones, Model predictive control for market-based demand response participation, in: World Congress of the International Federation of Automatic Control, 2014.
[37] I. Lymperopoulos, F. A. Qureshi, T. Nghiem, A. A. Khatir, C. N. Jones, Providing ancillary service with commercial buildings: the Swiss perspective, in: 9th IFAC International Symposium on Advanced Control of Chemical Processes (ADCHEM), 2015.
[38] D. Picard, J. Drgona, M. Kvasnica, L. Helsen, Impact of the controller model complexity on model predictive control performance for buildings, Energy and Buildings 152 (2017) 739–751.
[39] R. Judkoff, J. Neymark, International energy agency building energy simulation test (BESTEST) and diagnostic methoduilding energy simulation test (bestest) and diagnostic method, Tech. Rep. 472-6231, National Renewable Energy Laboratory, Golden, CO (1995).
[40] B. M. Cones, Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs, ASHRAE (2004).
[41] US Department of Energy, EnergyPlus Engineering Reference. (2014).
[42] US Department of Energy, EnergyPlus Input Output Reference (2014).
[43] K. J. Kircher, K. M. Zhang, On the lumped capacitance approximation accuracy in RC network building models, Energy and Buildings 104 (2015) 454–462.
[44] J. H. Lienhard IV, J. H. Lienhard V, A Heat Transfer Textbook, 4th Edition, Phlogiston Press, 2012.
[45] C. P. Underwood, F. W. H. Yik, Modeling Methods for Energy in Buildings, Blackwell Publishing, 2004.
[46] A. Khalifa, Natural convective heat transfer coefficient–a review: II. Surfaces in two-and three-dimensional enclosures, Energy Conversion and Management 42 (4) (2001) 505–517.
[47] A. K. Oppenheim, Radiation analysis by the network method, Transactions of the ASME 78 (1956) 725–735.
[48] M. H. Holmes, Introduction to Numerical Methods in Differential Equations, Springer, 2007.
[49] S. Hamdi, W. E. Schiesser, G. W. Griffiths, Method of lines, Scholarpedia 2 (7) (2007) 2859.

[50] A. A. Sharaf, H. O. Bakodah, A good spatial discretisation in the method of lines, Applied Mathematics and Computation 171 (2005) 1253–1263.
[51] M. Crowley, Evaluation of implicit numerical methods for building energy simulation, Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 212.5 (1998) 331–342.
[52] S. Wilcox, W. Marion., User's Manual for TMY3 Data Sets, Tech. Rep. NREL/TP-581-43156, National Renewable Energy Laboratory (April 2008).
[53] J. Duffie, W. Beckman, Solar Engineering of Thermal Processes, 3rd Edition, Wiley, 2006.
[54] K. J. Kircher, BLDG, a MATLAB® building simulator, https://github.com/kevinjkircher/bldg (April 2016).
[55] L. Laret, Use of general models with a small number of parameters, in: 7th International Congress CLIMA 2000, Budapest, Hungary, 1980, pp. 263–275.
[56] J. Crabb, N. Murdoch, J. M. Penman, A simplified thermal response model, Building Services Engineering Research & Technology 8.1 (1987) 13–19.
[57] E. Wan, R. Van Der Merwe, The unscented Kalman filter for nonlinear estimation, in: The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, 2000, pp. 153–158.