Applied Energy 318 (2022) 119104

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Field demonstration and implementation analysis of model predictive control in an office HVAC system

David Blum *, Zhe Wang **, Chris Weyandt, Donghun Kim, Michael Wetter, Tianzhen Hong, Mary Ann Piette

Lawrence Berkeley National Laboratory, Berkeley, CA, USA

ARTICLE INFO

Keywords:
Model predictive control
HVAC
Modelica
Optimization
MPCPy

ABSTRACT

Model Predictive Control (MPC) is a promising technique to address growing needs for heating, ventilation, and air-conditioning (HVAC) systems to operate more efficiently and with greater flexibility. However, due to a number of factors, including the required implementation expertise, lack of high quality data, and a risk-adverse industry, MPC has yet to gain widespread adoption. While many previous studies have shown the advantages of MPC, few analyzed the implementation effort and associated practical challenges. In addition, previous work has developed an open-source, Modelica-based tool-chain that automatically generates optimal control, parameter estimation, and state estimation problems aimed at facilitating MPC implementation. Therefore, this study demonstrates usage of this tool-chain to implement MPC in a real office building, discusses practical challenges of implementing MPC, and estimates the implementation effort associated with various tasks in order to inform the development of future workflows and serve as an initial benchmark for their impact on reducing implementation effort. This study finds that the implemented MPC saves approximately 40% of HVAC energy over the existing control during a two-month trial period and that tasks related to data collection and controller deployment activities can each require as much effort as model generation.

1. Introduction

1.1. Background

Predictive controls for heating, ventilation, and air-conditioning (HVAC) systems can help meet the growing needs for energy efficient and flexible load management in buildings. Such controls can optimize set points for greater system-level efficiency and adjust HVAC operation in response to price, demand response, or carbon intensity signals from the electric grid, or in response to emergency conditions like wild fire smoke or power outages. Model Predictive Control (MPC) is such a control technique that makes use of building models, disturbance forecasts, such as weather and occupancy predictions, objective and constraint definitions, and optimization algorithms to optimize control while considering both current and future states of the building [1]. Though widely studied in research communities, MPC has yet to gain mainstream adoption by the building industry. Contributing factors include:

1. The required level of expertise in telemetry, modeling, optimization, and building operation to implement.
2. Low data availability in buildings.
3. Complicating practical nuances in the design and operation of buildings.
4. Risk aversion in the delivery and operation of buildings.
5. Lack of guidance for how to implement MPC in practice.

These advantages and remaining challenges motivate the need for continued work towards achieving widespread adoption of predictive controls like MPC in buildings.

1.2. Paper objectives and contributions

The objectives of this paper are as follows:

1. Provide a review of MPC implementations in real buildings.
2. Demonstrate the use of an open-source and free software tool-chain called MPCPy [2] in a real office building.
3. Present practical challenges that were overcome to implement and operate the MPC.

* Corresponding author.
** Correspondence to: Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region
E-mail addresses: dhblum@lbl.gov (D. Blum), cezhewang@ust.hk (Z. Wang).

https://doi.org/10.1016/j.apenergy.2022.119104
Received 14 January 2022; Received in revised form 4 April 2022; Accepted 7 April 2022
Available online 6 May 2022

Nomenclature

$\boldsymbol{\omega}$	MPC model disturbance vector
θ_e^*	MPC model parameters used for control optimization
θ^l	MPC model parameter lower limits
θ^u	MPC model parameter upper limits
θ_e	Parameters to be estimated
θ_f	MPC model parameters that are fixed
$\tilde{\omega}$	MPC model forecasted disturbance vector
$\dot{m}_{oa}^l$	Lower limit on outside air flow rate
$\dot{m}_{sa}^l$	Lower limit on supply air flow rate
$\dot{m}_{0,sf}$	Supply fan nominal air flow rate
$\dot{m}_{sf}$	Supply fan air flow rate
$\dot{Q}_{0,dx}$	DX nominal sensible cooling rate
$\dot{Q}_{cool}$	DX sensible cooling rate
$\hat{y}$	MPC model output data
$\dot{\mathbf{x}}$	MPC model state vector time derivative
$\mathbf{u}$	MPC model control input vector
$\mathbf{x}$	MPC model state vector
$\mathbf{x}_0^*$	MPC model state vector initial condition used by control optimization
$\mathbf{x}_0$	MPC model state vector initial condition
a_{dx}	DX power estimated coefficient
a_{rf}	Return fan power estimated coefficient
a_{sf}	Supply fan power estimated coefficient
b_{dx}	DX power estimated coefficient
c_p	Specific heat capacity of air
Ci	Internal capacitance
COP_{hp}	Estimated ASHP COP
Cw	Wall capacitance
d	Day indicator
D_d	Daily discomfort for day
E_d	Daily energy consumption for day
f	MPC model
$Fint$	Internal gain fraction
$Fsol_i$	Solar fraction to internal
$Fsol_w$	Solar fraction to wall
J	Objective variable
k	Control regularization weights
P	MPC model sum of supply and return fan power, DX power, UFT reheat power
$P_{0,rf}$	Return fan nominal power
$P_{0,sf}$	Supply fan nominal power
P_{dx}	DX power
P_{hvac1}	HVAC panel 1 power measurement
P_{hvac2}	HVAC panel 2 power measurement
P_{rf}	Return fan power
P_{sf}	Supply fan power
$\dot{Q}_{hw}$	Reheat water heat flow rate measured at ASHP
R^n	Set of real numbers with dimension n
$Rw1$	Wall thermal resistance 1
$Rw2$	Wall thermal resistance 2
s	Slack variables
T_z^c	Zone cooling set point
T_z^h	Zone heating set point
T_z^l	Lower constraint on zone air temperature
T_{sa}^l	Lower constraint on supply air temperature
T_z^u	Upper constraint on zone air temperature
T_{sa}^u	Upper constraint on supply air temperature
t_0	Start time
t_f	Final time
T_z	Zone air temperature
T_{ma}	Mixed air temperature
T_{sa}	Supply air temperature
u_{dx}	DX control signal
u_{sf}	Supply fan control signal
w	Slack variable weights
y	Measurement data
Z	Total number of zones
z	Zone indicator

Acronyms

ACB	Active Chilled Beams
AHU	Air Handling Unit
ALC	Automated Logic Corporation
API	Application Programming Interface
ARX	Autoregressive Model with Exogenous Input
ASHP	Air Source Heat Pump
BMS	Building Management System
COP	Coefficient of Performance
DX	Direct Expansion
FCU	Fan Coil Unit
GHI	Global Horizontal Irradiation
HP	Heat Pump
HVAC	Heating, Ventilation, and Air Conditioning
IQR	Interquartile Range
IT	Information Technology
LBNL	Lawrence Berkeley National Laboratory
MPC	Model Predictive Control
NERSC	National Energy Research Scientific Computing Center
PI	Proportional–Integral
PMV	Predicted Mean Vote
RC	Resistance–Capacitance
RMSE	Root Mean Square Error
RTU	Rooftop Unit
TABS	Thermally Activated Building Structures
UFAD	Underfloor Air Distribution
UFT	Underfloor Terminal Unit
VAV	Variable Air Volume
VFD	Variable Frequency Drive

4. Estimate and discuss a breakdown of implementation effort.

With the stated objectives, the contributions of this paper are as follows, and specifically aim to address challenges (1), (4), and (5) presented in Section 1.1. The review of MPC implementations in real buildings provides a summary of implementation methods validated in real-world situations. MPCPy was developed in [2] to facilitate the coupling of the various components of MPC and to automate the generation of optimal control, parameter estimation, and state estimation problems using user-defined models written in Modelica [3] and high-level parameterization in Python. Such a tool-chain aims to reduce the level of expertise needed to implement MPC, particularly as the

modeling ecosystem around the use of Modelica grows in the buildings industry, for instance through the development of Spawn [4]. Other Modelica-based tool-chains have been developed for the identification of gray-box envelope models [5] and generation of the optimal control problem using a white-box modeling approach [6], however, neither are open-source and neither facilitate the implementation of parameter estimation, state estimation, and optimal control problems in the same framework. Therefore, this paper demonstrates the feasibility of using MPCPy for MPC implementation in an office building. Finally, few studies describing the implementation of MPC in real buildings discuss practical challenges that were overcome to implement and operate the MPC. In addition, [7] estimated 70% of project costs could be dedicated to model generation during the MPC process and [8] cited model generation as the most significant installation cost driver. However, no studies to date have presented an estimated breakdown of implementation effort. Therefore, the discussion on practical challenges and estimated breakdown of implementation effort provided in this paper can aid in the identification of where more work is needed to reduce installation costs and can serve as a benchmark for comparison with future studies.

This paper achieves these objectives by describing the design, implementation, and field operation of an MPC controller to control the Underfloor Air Distribution (UFAD) HVAC system serving approximately 6038 m^2 (65,000 ft^2) of office space in Berkeley, California, USA. In particular, this paper is structured as follows. Section 2 reviews the literature on MPC field demonstrations for commercial buildings. Section 3 describes the field demonstration site. Section 4 describes the MPC controller design, including integration with the Building Management System (BMS) and other software implementation details. Section 5 presents the results of the field demonstration and describes the performance. Section 6 discusses practical issues, provides an estimation of implementation effort, summarizes lessons learned, and identifies limitations and future work. Finally, Section 7 provides conclusions.

2. Literature review

This review provides a summary of field demonstrations of MPC in real commercial buildings to date. Simulation studies are not included, nor are experiments in test facilities, though notable test facility experiments include [9], which implemented MPC for an HVAC system with ice storage, [10], which implemented MPC for an air-cooled chiller and concrete floor cooling system, and [11], which implemented MPC for active chilled beam and fan coil unit cooling systems. Table 1 summarizes the review, presented chronologically, while discussions afterward focus on approaches to modeling, control formulation, and performance evaluation.

2.1. Modeling approaches

A key step for MPC is to model the system thermo-fluid and energy dynamics, primarily including the dynamics of the building envelope and the variation of performance of the HVAC system. For the sake of simplicity, most surveyed studies modeled the HVAC equipment using quasi-steady state models (ignoring the dimension of time) since their time constant of operation is smaller than that of building envelope [16].

To model the dynamics of the building envelope, the majority of studies applied gray-box or black-box models. Notable exceptions include [19], which developed a white-box model using EnergyPlus, and [23], which developed a white-box model in Modelica. One shortcoming of white-box models that use a building energy simulator to evaluate the cost function is that these simulators typically do not allow computing gradients, and hence gradient-based optimization solvers cannot be used to solve the optimal control problem [25]. For example, in [19], brute force search and a genetic algorithm were applied as the optimization approach. This approach is limited in scalability since it is computationally demanding especially as the problem size grows. However, in [23], a process was implemented to separate a linear thermal envelope model from nonlinear HVAC and weather disturbance models to enable the formulation of a quadratic program that can be efficiently solved. Additionally, a tool-chain for the implementation and efficient solving of nonlinear programming problems resulting from white-box models implemented in Modelica was developed called TACO [6], for which performance results from a real full-scale demonstration are forthcoming as of the time of this writing [26].

A typical example of a gray-box thermal envelope model is the resistance-capacitance (RC) model and an example of a black-box model is the autoregressive model with exogenous input (ARX) model. Such linear envelope models represented in state-space form were the most common approach to envelope modeling. Two studies, [14,21], used transfer function methods. Transfer functions are also considered to be gray-box models since the transfer function has clear physical implications, with the coefficients determined through parameter estimation.

Once a gray-box model structure is chosen, the next step is to estimate model parameters. Existing studies applied two approaches to estimate parameters: directly from operational data or from specially designed and executed system identification experiments. The first approach is adopted more in existing studies due to its simplicity. For example, [8] argued that due to building usage constraints, identification experiments may only be possible on weekends, which may not be sufficient for multi-input multi-output models or models with long time constants such as those for Thermally Activated Building System (TABS). However, [21] mentioned the problem of "data rich but information poor", meaning operational data may not contain enough useful information about the system for robust parameter estimation. Therefore, studies such as [16,21] applied functional step response tests.

2.2. Control objectives

The majority of the reviewed studies optimize the operation of office buildings, except [18] optimized the heating system of a multi-family residential building, [21] optimized the chiller operation of a shopping mall, [15] optimized RTU operation in a gym, and [22] optimized operation in a retail store. Most studies control the HVAC distribution system in connection with the zone performance, though, [21] specifically minimized chiller energy use by adjusting supply chilled water temperature. Fluid mass flows, heat flows, and temperatures were common optimization variables while control signals were relayed to the system as supervisory set points for local PID controllers or to actuators directly, such as in [23] where the MPC optimized the TABS heat flows and a post-processor translated into low-level actuator signals.

Most studies minimized operational costs or energy, though [14] minimized carbon emissions and [15,22] minimized peak demand. All studies consider maintaining thermal comfort while minimizing energy/costs/carbon and most do so by including thermal discomfort as a penalty in the objective function. Two field studies include real-time occupant comfort feedback in MPC. [19] collected occupants' feedback to adjust the comfort temperature range and [14] collected subjective comfort feedback to correct the Predicted Mean Vote (PMV) calculated from the zone conditions and incorporate it into the optimization objectives.

2.3. Performance evaluation

The surveyed studies reported very different energy or costs saving potentials of MPC: ranging from less than 10% [20] to more than 70% [13]. A couple of factors that impact the saving potentials of MPC are building construction and ambient conditions [12] as well as

Table 1
Summary of studies on field demonstrations of MPC: VAV for variable air volume, UFAD for underfloor air distribution, RTU for roof top unit, AHU for air handling unit, TABS for thermally activated building structures, FCU for fan coil unit, HP for heat pump, ACB for active chilled beams.

Study	Year	Building	System	Control variable	Control objectives	MPC approach	Test period	Results
[12]	2011	Office in Prague, Czech Republic	Radiant ceiling	Supply water temperature of ceiling radiant heating system	Minimize energy	RC model for thermal envelope; quadratic program for optimal control; thermal comfort as objective penalty	8 weeks	15%–28% HVAC energy saving depending on insulation level and outside temperature
[13]	2014	Office in Champaign Illinois, US	VAV	Flow rate and temperature of the mixed-air, discharge air temperatures of heating and cooling coil, and supply temperatures for each zone	Minimize cost	Semi-empirical models for thermal envelope and HVAC system; nonlinear program for optimal control solved by IPOPT	1 week in transition season, 1 week in heating season	20% cost saving for transition season, 70% for heating season
[14]	2014	Offices in Newcastle and Melbourne, Australia	UFAD	Zone temperature set-points where available or supply air temperature if not	Minimize costs and CO_2 emissions	Discrete transfer function for thermal envelope; quadratic program for optimal control	2 months	19% and 32% for each building in the heating season
[15]	2015	Gym in Philadelphia, US	Four RTUs	On/off status of RTU	Minimize energy and peak demand	Auto-regressive with exogenous input (ARX) linearized to state-space; Linear program for optimal control solved by MATLAB	4 days	8% building energy saving and 40% peak demand reduction
[8]	2015	Office in Allschwil, Switzerland	TABS	Heating and cooling heat flux of TABS and AHU, air flow rate of AHU	Minimize cost	RC model for thermal envelope; bilinear problem for optimal control solved by CPLEX; thermal comfort as objective penalty	17 months	17% HVAC energy saving
[16]	2015	Office in Philadelphia, US	VAV	Terminal box and AHU supply air flow rates and temperatures	Minimize energy	Linear state-space model for thermal envelope; nonlinear program for optimal control formed using AMPL, solved by IPOPT, interface with MATLAB; thermal comfort as objective penalty	3 months, including 20 days of MPC operation	>20% average daily HVAC energy saving
[17]	2016	Office in Brussels, Belgium	AHU + FCU + radiator	Heating flux of the gas boiler and heat pumps	Minimize cost	Single zone RC model for thermal envelope implemented in Modelica; nonlinear program for optimal control formed using Jmodelica.org and solved by IPOPT; thermal comfort as objective penalty	During winter	34%–40% daily HVAC cost saving
[18]	2017	Multi-family house in Sigulda, Latvia	HP	Scheduling of heat pump and electrical heater	Minimize heating costs	A reduced order model for building thermal dynamics. No optimization methods are applied	1 month	13% cost saving
[19]	2017	University in Halifax, Canada	District heating	Morning starting time and zone temperature set point	Minimize energy	Used EnergyPlus to develop the building thermal model and applied brute force search and genetic algorithm for optimization	4 months	29% energy saving
[20]	2018	Office in Long Beach, courthouse in Dayton, hospital in New York, high school in D.C., US	VAV	Set-points for AHU supply air temperature and duct static pressure	Minimize energy	Did not develop the MPC controller by themselves, instead, commercially available offering of MPC was tested	7–15 months	0%–9% HVAC energy saving for different buildings
[21]	2018	Shopping mall in Chengdu, China	AHU + FCU	Supply chilled water temperature of the chiller	Minimize chiller energy	Step response method and power spectral density (PSD) method are used to identify the transfer function of manipulated variable and random disturbances. Optimization methods not introduced	1 day	16% chiller energy saving

(*continued on next page*)

comparative baseline [20]. Major barriers also exist that prevent MPC from achieving full savings potential [20], including improper tuning of the HVAC system and special operating requirements, such as strict space pressure control requirements in hospitals or minimum ventilation requirements to mitigate potential transmission of COVID-19 in office buildings.

Table 1 (*continued*).

Study	Year	Building	System	Control variable	Control objectives	MPC approach	Test period	Results
[22]	2018	Retail store in Florida, US	Four RTUs	On/off status of RTU	Minimize energy and peak demand	RC model for thermal envelope with lumped disturbance model; integer linear program for optimal control; thermal comfort as objective penalty	4 months	12% HVAC energy savings and 18% HVAC peak demand reduction
[23]	2020	Office in Brussels, Belgium	GSHP + TABS	Heat flows of TABS, then converted into supply water temperature and valve position of TABS	Minimize energy	Building model in Modelica and linearized; quadratic program for optimal control, solved by GUROBI, interface with MATLAB; thermal comfort as objective penalty	5 months	54% heat pump energy saving and 37% thermal comfort improvement during the transient seasons
[24]	2021	Office in Hamburg, Germany	HP + TABS	Supply temperature of heating circuits	Minimize heating energy	RC model for thermal envelope in Modelica; nonlinear program for optimal control solved using the fmincon in MATLAB and Dymola; thermal comfort as objective penalty	3 months	30% heating energy saving

2.4. Discussion

Our literature search found 14 studies implementing MPC in real buildings during the past decade (2011–2021), mostly located in the U.S. and Europe. All existing studies demonstrate positive savings in terms of energy consumption, peak demand, operational cost, or carbon emission while maintaining indoor comfort requirements. Reviewed studies covered different HVAC distribution systems including single-duct multi-zone VAVs, packaged RTUs, TABS, radiant panels, and radiators. One study controlled a UFAD system.

A key focus for all reviewed studies was demonstrating the feasibility and advantages of MPC technology in real buildings. While important, financial performance is another key driver, or barrier, for the adoption of MPC technology. Of the studies reviewed, only [8] estimated implementation costs to some degree. The study estimated purchasing and installing of new sensor hardware and data services, however, did not estimate model development and engineering effort for MPC implementation. It also stated that these costs remain the largest uncertainty in total costs, while also likely being the most decisive barrier to cost-effectiveness today. The study suggests development in tools for MPC workflow implementation and modeling, as well as workforce development, will help enable cost-effectiveness of MPC. Since that study, there has been development of tools to help automate the MPC implementation process, including the automated model generation toolkit [5] used in [17], the method for accounting for unmeasured disturbances enabling the plug-and-play approach used in [22], and the tool-chain for use of white-box Modelica models in nonlinear programs [6]. However, these tools do not facilitate the implementation of all of the components of MPC for buildings, including parameter estimation for gray or black-box models, state estimation, and optimal control, nor are they open-source to enable community contributions for the various approaches described in this literature review.

Therefore, the primary contributions of this paper with respect to the reviewed literature are two-fold. First, beginning to fill the gap identified by [8] in estimating model development and engineering costs by providing a detailed account of MPC implementation including practical issues needed to be overcome and an estimation of implementation effort broken down by various tasks. Second, demonstrating the feasibility of using the open-source workflow developed in [2], which integrates parameter estimation, state estimation, and optimal control, to implement MPC in a real building. The workflow also utilizes Modelica as a modeling language and automates usage of these models for generation of optimization problems related to parameter estimation, state estimation, and optimal control. The approach of gray-box modeling is taken, as the most popular approach described in Section 2.1. One key difference to previous work is the re-estimation of envelope model parameters on a regular basis, which is helpful for accounting for seasonal operating conditions. In addition, the approach of minimizing energy through control of the HVAC distribution system with zone thermal comfort constraints treated as objective penalties is taken, as the most popular approach described in Section 2.2.

3. Site description

3.1. Building

The field test building is Building 59 on the main campus of Lawrence Berkeley National Laboratory in Berkeley, California, which was constructed in 2015 and is home to the NERSC high-performance computing center. Berkeley has a mild climate (ASHRAE Climate Zone 3C). The lower and second levels serve as the mechanical room and computing center, and the third and fourth levels are office spaces. These top two office space levels account for approximately 6038 m^2 (65,000 ft^2) floor area and are deemed the ground level office floor and the second level office floor. The ground office floor is primarily closed office space, while the second office floor is primarily open office space. In this field study, MPC only controls the HVAC system serving these two office floors. The building structure is steel-framed with an exterior metal curtain wall system with integrated windows and foamed insulation core.

3.2. HVAC and existing control

Heating and cooling is provided to the offices by an underfloor air distribution system (UFAD). The system uses four roof-top units (RTUs) located on the roof with water-cooled DX coils to supply cool air to a common underfloor plenum on each level, where air diffuses directly to the core and perimeter zones through floor diffusers and can be pulled and heated to perimeter zones with fan-powered underfloor terminal units (UFT). This design is shown in cross-section for one RTU in Fig. 1. Each RTU serves the ground level and second level offices between particular column lines of the building, though the areas of service are not separated by internal wall partitions. The design air flow of each RTU is 9.44 m^3/s, with design minimum outdoor air flow rate of 2.36 m^3/s. For each RTU, the supply fan motors are 14.9 kW and the return fan motors are 5.6 kW, each equipped with variable frequency drives (VFD). The cooling capacity of each RTU is 104 kW with two 9.6 kW motor R410A scroll compressors with variable speed control

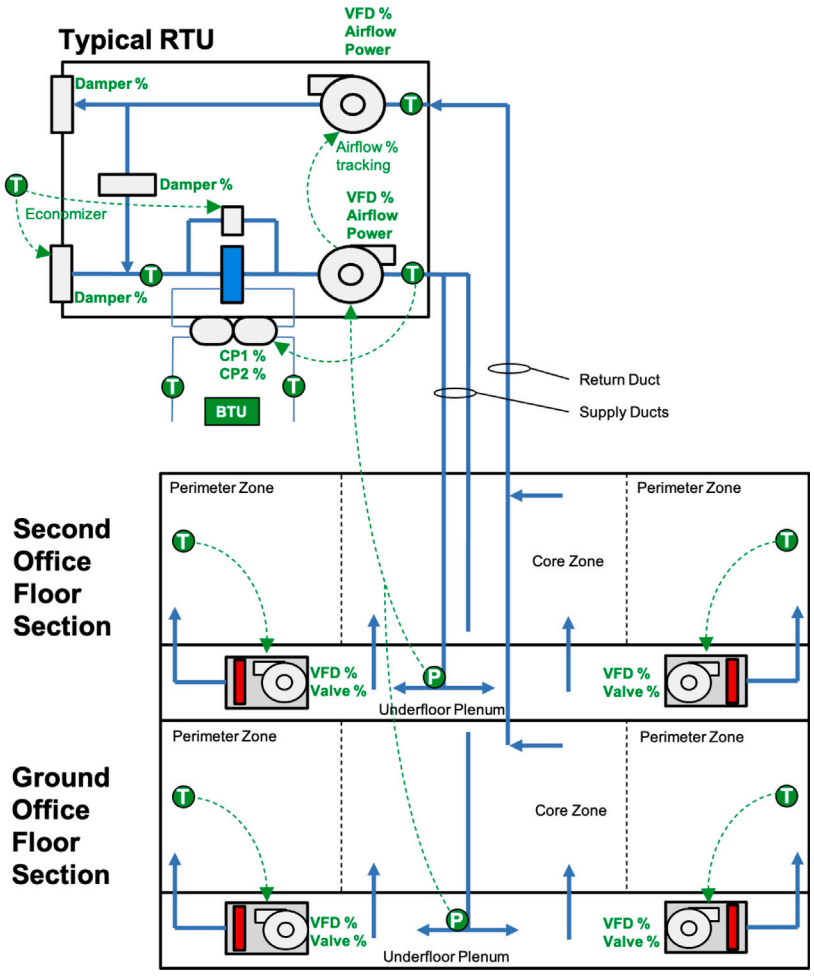


Fig. 1. Design and control schematic for one RTU of the UFAD HVAC system. Important sensor and control points are labeled, including temperatures (T) and static pressures (P).

of each compressor from 10% to 100%. There are 50 fan-powered underfloor terminal units (UFTs) with water heating coil to provide heating and reheat. The condenser water from the RTUs is cooled by cooling towers located next to the building on the mechanical level. These cooling towers are shared with the high performance computing cooling equipment, which dominate the load on the towers. Hot water for the UFT heating coils is produced by a 117 kW air-source heat pump (ASHP) located on the mechanical level of the building.

The UFAD system is controlled by an Automated Logic Corporation (ALC) WebCTRL BMS. The existing control of the HVAC system is the result of a challenging commissioning process. Perimeter zone thermostats control the UFT fan speed for cooling and hot water valves for heating. However, to keep the noise of the UFT fans low for occupants and as a result of system commissioning, strict limits are placed on many UFT fan speeds typically ranging from a minimum of 20% and maximum of 50%. Note that no thermostat-based control of air flow exists for the core zones of the building, as supply air diffuses directly from the underfloor plenums into core spaces. All four RTU supply fan speeds are controlled by a single PI controller taking a constant static pressure set point of 15 Pa and the maximum underfloor plenum static pressure measurement of all units. The return fan speeds of each RTU are controlled to track a flow rate equal to 95% of the supply air flow rate minus an additional 0.1 m^3/s. The supply air temperature for each RTU is controlled to a constant, mild set point of 20 °C by the DX coil and economizer when outside air dry bulb temperature conditions are appropriate. When not economizing, the outside air flow rate for each RTU is controlled to the design minimum outside air flow rate set point using outside air flow measurement stations in each RTU. In light of a concern for ensuring critical IT rooms located in the office floors stay cool, the only active schedule is one that sets back the thermostat heating and cooling set points in the second floor office on Saturdays. However, in general, the RTUs do not turn off.

3.3. Electrical

The electrical system serving the office floors contains two main plug load panels, two main lighting panels, and two main HVAC panels. One plug panel and one lighting panel serve both floors of the north wing, while the other plug panel and other lighting panel serve both floors of the south wing. The two HVAC panels serve the RTU units, two on each where RTUs 1 and 2 are on one panel and RTUs 3 and 4 are on the other panel. These HVAC panels also serve other HVAC equipment, including building exhaust fans and UFT fans, as well as building elevators. The ASHP used to generate hot water for UFT reheat is served by a separate panel that also serves a significant amount of high-performance computing center systems. Electrical metering data for these panels is collected and stored in a database managed by NERSC. Additionally, supply and return fan power consumption for each RTU is collected by the BMS system.

Measurements of the plug and lighting panels allows for some indication of internal load for system identification of zone envelope models, especially if estimated parameters, such as load fractions, can account for disaggregating load from the plug level to the zone level as necessary, and if the spatial resolution of zone models remains relatively low (closer to the wing level rather than individual zone level). This is the approach taken in this study as described later in

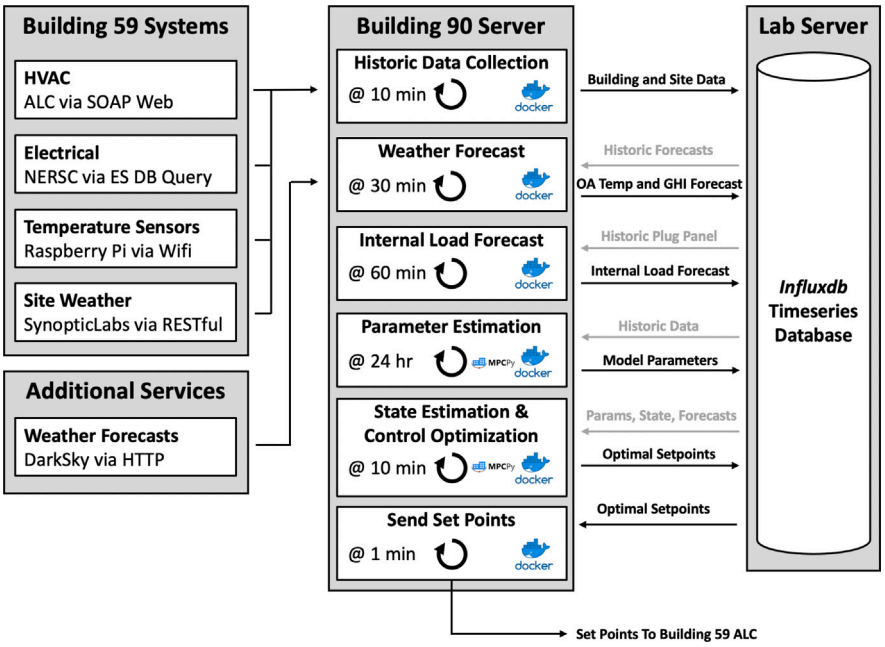


Fig. 2. Architecture of the MPC controller.

Section 4.3, where aggregated zone models are created at the RTU level. Measurements of the HVAC panel electricity similarly allows for some indication of power consumption of individual RTUs and equipment, particularly if combined with other system data to make sense of what specific equipment may be operating at what times. This is the approach taken in this study particularly for DX compressor power modeling as described later in Section 4.3. However, higher resolution data, especially dedicated measurements of HVAC equipment power, would significantly improve the modeling capabilities, and ease the process of generating and identifying those models.

3.4. Added sensors

The BMS only contains thermostats in the perimeter zones in order to control the UFTs. However, there are no air temperature measurements taken by the BMS in the core zones of the floor plan. Therefore, we deployed 16 air temperature sensors built with Raspberry Pi Zero W and DS18B20 Digital Temperature Sensors. The Raspberry Pis run Raspbian Lite (no desktop), are connected to a power source, and use the lab's wifi network to push air temperature measurements every 10 min to the project's Influxdb database, described in Section 4.

4. MPC controller implementation methods

4.1. Architecture

The MPC controller is implemented in a service-oriented architecture, as presented in Fig. 2. There are six main services that run in parallel, which are described in more detail in the following Sections 4.2 to 4.9. These services run on a dedicated server set up in a different building in the lab campus than the field test building, called Building 90, and exchange data with various Building 59 and site data collection systems, web-based services, and the project database. This database is located on another separate server managed by the LBNL Science-IT program and is an instance of an Influxdb timeseries database [27]. The project database was setup in order to centralize storage for efficient access to relevant data as needed by each service, as well as to provide a means for future analysis of all data. Docker [28] is used for rapid, repeatable deployment of all MPC services. Each MPC service runs within its own Docker container, though each container is based on the same Docker image, which is built to enable all of the functionality of the MPC services. Overall, this service-based architecture deployed with Docker enables an efficient means to maintain each service without interruption of other services.

The control optimization, state estimation, and parameter estimation services are implemented using the open-source software MPCPy [2] commit `ecb9833`. This workflow utilizes a user-defined model written in Modelica along with user-defined data specified in a Python environment to automatically formulate and solve the appropriate optimization problems for each of the control optimization, state estimation, and parameter estimation services. MPCPy uses the JModelica.org toolchain [29], and associated Optimica language extension to Modelica, to write and solve the resulting optimization problems from a given Modelica model. A key limitation to the workflow using Optimica and JModelica.org is the inability to specify and solve integer optimization problems. Such an extension in future workflows would enable Modelica-based optimization workflows for more applications.

4.2. Data collection

Data collection is responsible for gathering data from the various building data systems, storing it in the project database for later use by the MPC algorithms, and managing the project database. Separate Python modules were developed to collect data from each building system, since each building system requires use of a separate API or interface. The service, however, is run by a single Python module that centralizes the use of each of the system-specific modules to collect all system data, tag appropriately for storage in the project database, and transfer to storage in the project database as historic timeseries. For the HVAC system, the ALC SOAP web interface is used to retrieve data from specified points associated with the operation of the RTUs, hot water heat pump, and zone UFTs. For the electrical system, a NERSC web endpoint is used to query an elasticsearch database on which the data is held for power measurements of the panels described in Section 3.3. For the site weather data, an HTTP RESTful web API provided called Synopticlabs is used to obtain data from a weather station located at LBNL Building 46. The installed Raspberry Pi temperature sensors push the measured indoor air temperature to the project database themselves.

In summary, there are four separate data collection systems for retrieving historic operational data, one separate data collection system for retrieving weather forecasts (described later in Section 4.5), and one separate data storage system hosting a database for the controller. Setup and maintenance of each system is a key contributing factor of controller complexity.

4.3. Modeling

The models described here are used by the MPC controller in real time to predict the performance of the building and optimize control. Important values to predict are building space temperatures, RTU air flow rates and temperatures, fan power consumption, DX compressor power consumption, and UFT heating demand. The approach to modeling taken here is a gray-box approach, where simplified, yet physically-based, models are used and implemented in Modelica for use with MPCPy.

A zone-simplification scheme is applied to the building in order to reduce the complexity of models. A key challenge in designing these models is that it is difficult to determine how much supply air flows into each zone since there are no air flow measurements from the UFTs serving perimeter zones and there is no measurement or way of controlling air flow throughout the plenum and through the floor diffusers, which serve both core and perimeter zones. Another challenge is that the plug and lighting electrical loads are measured at the wing level (north and south), and not at individual zone or individual RTU level. Therefore, the scheme implemented combines the individual thermal zones into four zone groups such that each group is associated with one RTU and made up of all of the zones assumed to be served by that RTU. Note that these zone groups cover both office floors served by each RTU. The temperature of each zone group is approximated as a weighted average of all individual zones making up the zone group. Here, an average core temperature (measured by the Raspberry Pi sensors) is given a weight of 50% and an average perimeter temperature (measured by the ALC thermostats) is given a weight of 50%. The core temperatures are averaged equally while the perimeter temperatures are averaged with additional weights determined from their design UFT air flow rate. A single UFT hot water valve position for each zone group is approximated in a similar manner, averaged among all UFTs serving perimeter zones where weights are defined by the design UFT heating capacity. There are no hot water valves serving the core zones.

The total model for a single RTU-zone configuration is shown in Fig. 3. It is composed of one component model representing the thermal envelope and one representing the RTU with supply and return air circulating between the two. The inputs are outside air temperature, solar global horizontal irradiation (GHI), internal loads, hot water valve position control signal, fraction of outside air control signal, supply fan control signal, and compressor control signal. Notable outputs are zone air temperature, supply air temperature, supply air flow rate, outside air flow rate, supply and return fan power consumption, compressor power consumption, and reheat power consumption. Additional blocks are added for regularization of control signals of outside air fraction, supply fan, and compressor. Fan, compressor, and reheat power consumption are summed along with the outputs of the regularization signals to form the objective function of the optimization problem, J, described in Section 4.7.

The thermal envelope zone model is shown in Fig. 4. The basic model is an R2C2 envelope model, with additional components specific to the field test building, such as air-based HVAC supply and return and reheat heat exchanger. The solar irradiation is split so that a fraction is assumed to add energy to the exterior wall and the rest is assumed to add energy to the inside thermal mass. The following parameters are estimated using measured data by the parameter estimation algorithm described in Section 4.4: $Rw1$, $Rw2$, Ci, Cw, $Fsol_i$, $Fsol_w$, $Fint$ (see the schematic model view in Fig. 4 for how these parameters are used). In early parameter estimation trials, both R2C2 and R3C3 models

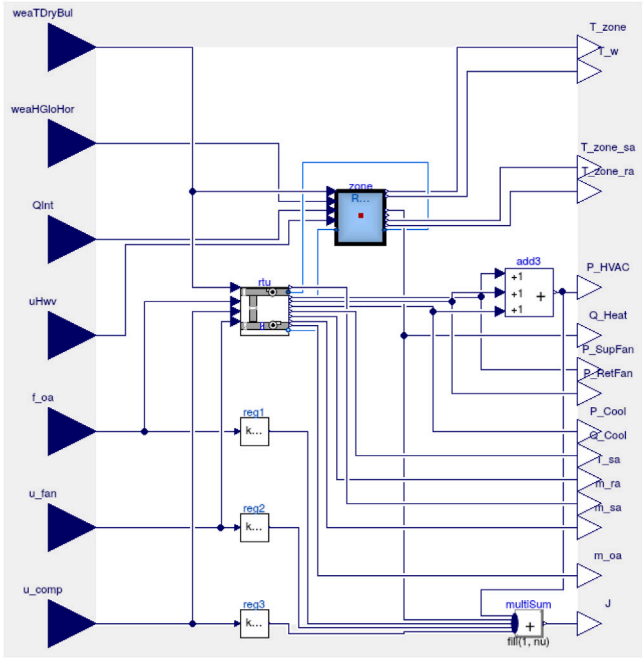


Fig. 3. Modelica model for an RTU-zone system.

were tested, with the R2C2 model performing sufficiently without the additional dimensions of the estimated parameter space added by the R3C3 model.

The RTU model is shown in Fig. 5. It contains a mixing box, supply fan, return fan, and DX model. The mixing box mixes the amount of outside air, and exhausts the corresponding amount of relief air, specified by the outside air fraction control signal and supply air flow rate. The supply fan moves air at a specified flow rate, $\dot{m}_{sf}$, according to $\dot{m}_{sf} = u_{sf}\,\dot{m}_{0,sf}$, where u_{sf} is the control signal and $\dot{m}_{0,sf}$ is the design air flow rate retrieved from design drawings. As a model simplification, the return fan is assumed to move an equal amount of air. The power of each fan is calculated based on supply air flow rate by a cubic expression defined by

$$\frac{P_{sf}}{P_{0,sf}} = a_{sf}\,u_{sf}^3 \tag{1}$$

and

$$\frac{P_{rf}}{P_{0,rf}} = a_{rf}\,u_{sf}^3, \tag{2}$$

where P_{sf} and P_{rf} are the supply and return fan power, $P_{0,sf}$ and $P_{0,rf}$ are the nominal supply and return fan powers specified for $u_{sf} = 1$, and a_{sf} and a_{rf} are estimated using measured data by the parameter estimation algorithm described in Section 4.4. The DX model implements a simple heat exchanger on the air stream where a specified amount of sensible energy is removed from the air stream, $\dot{Q}_{cool}$, according to the product of the compressor control signal and nominal capacity of the coil retrieved from design drawings, $\dot{Q}_{0,dx}$. The resulting power of the compressor is calculated using

$$P_{dx} = a_{dx}\,\dot{Q}_{cool}^2 + b_{dx}\,\dot{Q}_{cool}, \tag{3}$$

$$\dot{Q}_{cool} = u_{dx}\,\dot{Q}_{0,dx}, \tag{4}$$

where P_{dx} is the power consumption of the compressor and a_{dx}, b_{dx} are parameters fit according to measured data as described in Section 4.4.

Neglecting latent load on the DX coil warrants further discussion from two perspectives. First, the implications on accounting for humidity on the power consumption of the DX compressor. Since the DX power model is a data-driven relationship given by Eq. (3), we expect

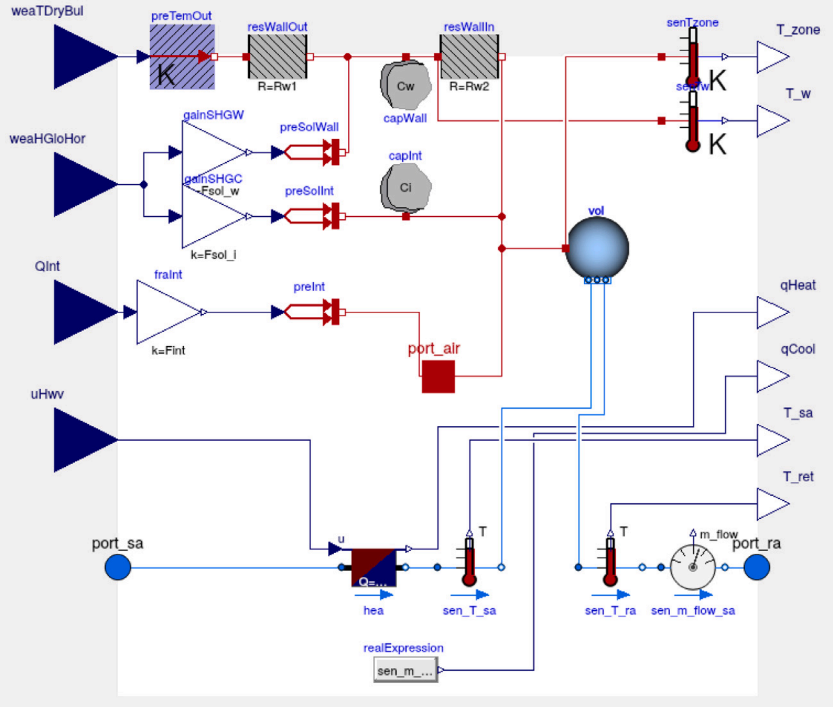


Fig. 4. Modelica model for a single zone.

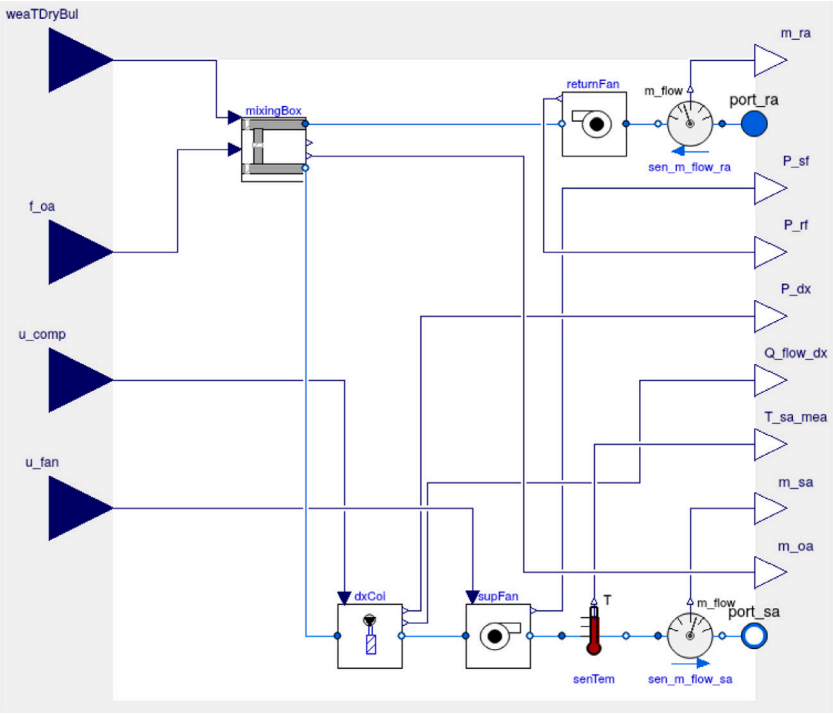


Fig. 5. Modelica model for a single RTU.

latent load to be considered implicitly in the estimated parameters, effectively contributing to a lower derived COP for a given sensible cooling load. Granted, this does not explicitly account for how latent load may vary independently of sensible load and further study would be needed to see how much performance improvement could be expected by taking this into account explicitly. The second perspective is accounting for humidity in the supply air to more accurately predict room conditions. Berkeley is a mild climate and our application is an office building, so humidity build up in rooms is rarely a concern. However, in more humid climates, for certain applications (e.g. libraries, museums, labs), or certain system types (e.g. radiant cooling) where space humidity control is critical, a more accurate model to account for humidity would likely be needed.

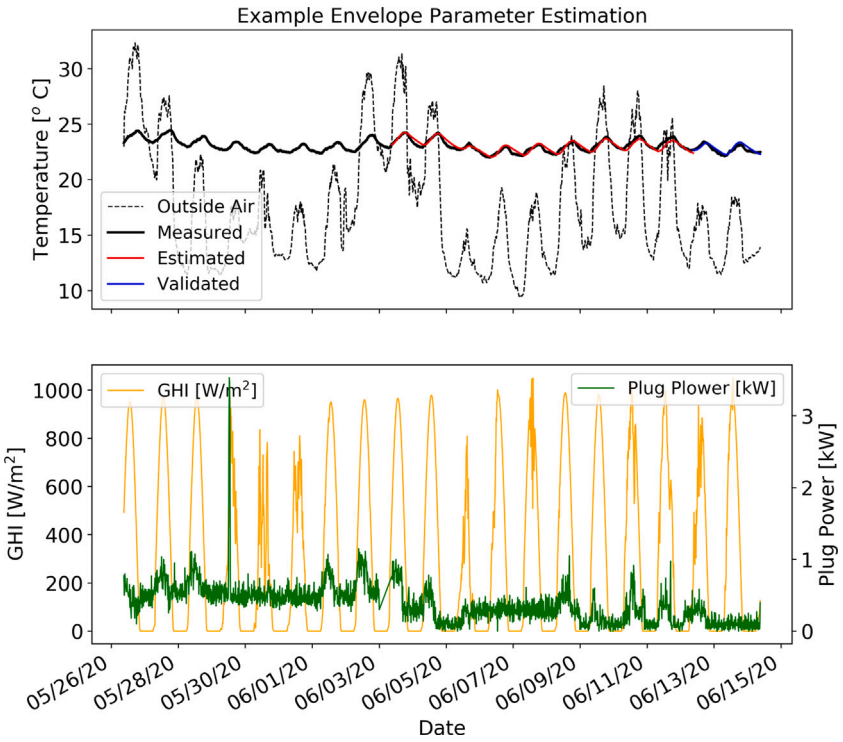


Fig. 6. Example result of parameter estimation process for the envelope model of RTU1 initialized on June 15, 2020 showing measured, estimated, and validated zone temperature along with outside air temperature (top) and other key disturbance inputs of global horizontal irradiation (GHI) and electrical plug power (bottom).

4.4. Parameter estimation

The parameter estimation is responsible for estimating parameters for the MPC models for the envelope and fans described in the previous section. The parameters identified are time-invariant from the perspective of their use in the control optimization. However, since the model is rather simple, it is difficult to say that true, time-invariant parameters exist for the model that will be valid for all time (as would be the case for a very detailed physics-based model). Instead, the assumption is that the solution of the parameter estimation problem is relevant for the data utilized for the estimation of parameters, with some extrapolation outside of those bounds as permitted by the physical implications of the model. Therefore, the approach utilized here is that the parameter estimation problem is formulated as an optimization problem, according to Eqs. (5)–(8), and solved at regular intervals to continually update the parameters based on recent data. This is particularly useful for updating the models as the weather changes over the year or if building operation changes. The parameter estimation problem is formulated as

$$\min_{\theta_e \in R^n} \quad J = \int_{t_0}^{t_f} (y - \hat{y})^2 dt \tag{5}$$

$$\text{s.t.} \quad f(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \boldsymbol{\omega}, \boldsymbol{\theta}_e, \boldsymbol{\theta}_f) = 0 \tag{6}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{7}$$

$$\theta_i^l \leq \theta_i \leq \theta_i^u \quad \forall i \in \{1, \ldots, n\}, \tag{8}$$

where $\boldsymbol{\theta}_e \in R^n$ is the set of n parameters to be estimated, t_0 is the start time of the estimation period, t_f is the final time of the estimation period, y is the measurement data, $\hat{y}$ is the model output, $\mathbf{x}$ is the model state vector, $\mathbf{u}$ is the model control signal vector, $\boldsymbol{\omega}$ is the model disturbance vector, $\boldsymbol{\theta}_f$ are other parameters of the model that are fixed, $\boldsymbol{\theta}^l$ are the estimated parameter lower limits, and $\boldsymbol{\theta}^u$ are the estimated parameter upper limits. Note that this parameter estimation problem is formulated automatically in the Modelica language extension of Optimica by MPCPy given the model f written in Modelica and specification of measured variable, tunable parameters and associated limits, and input data in Python.

One challenge confronted during the implementation of the parameter estimation approach was the tendency for missing data in one of the many model inputs. While missing data for a short duration can be tolerated, lengthy periods of missing data means that period is unable to be modeled. Therefore, a scheme was implemented to allow for continuous parameter estimation while accounting for, and avoiding, using the model during periods of missing data. The scheme started with defining a window of historic data to use for the parameter estimation process. Periods of missing data were automatically identified, based on an algorithm that analyzed the most common data sample rates, and used to split the window into usable sections of at least a specified number of days. The most recent usable section was selected for use by the parameter estimation process. Based on a specified number of validation days, a further division of this section was made into an estimation period and validation period, with the validation period being more recent.

An example result of the parameter estimation process for the envelope model of RTU1 initiated on June 15, 2020 is shown in Fig. 6. The measurement data, y, is the zone temperature as determined from the zone grouping method described in Section 4.3. The full window of historic data was 20 days, the allowable duration of missing data was five hours, the minimum time needed for a usable section of data was five days, and the length of the validation period was two days. Notice that the estimation period starts just after the period of missing electrical plug load data identified on June 3, 2020.

An example result of the parameter estimation process for the fan models of RTU1 initiated on June 15, 2020 is shown in Fig. 7. The measurement data, y, is power consumption as reported by the BMS for each VFD. For the fan models, the full window of historic data was 15 days, the allowable duration of missing data was one hour, the minimum time needed for a usable section of data was three days, and the length of the validation period was one day. In the particular case of Fig. 7, no missing input data was identified and the whole historic window was used for the estimation process.

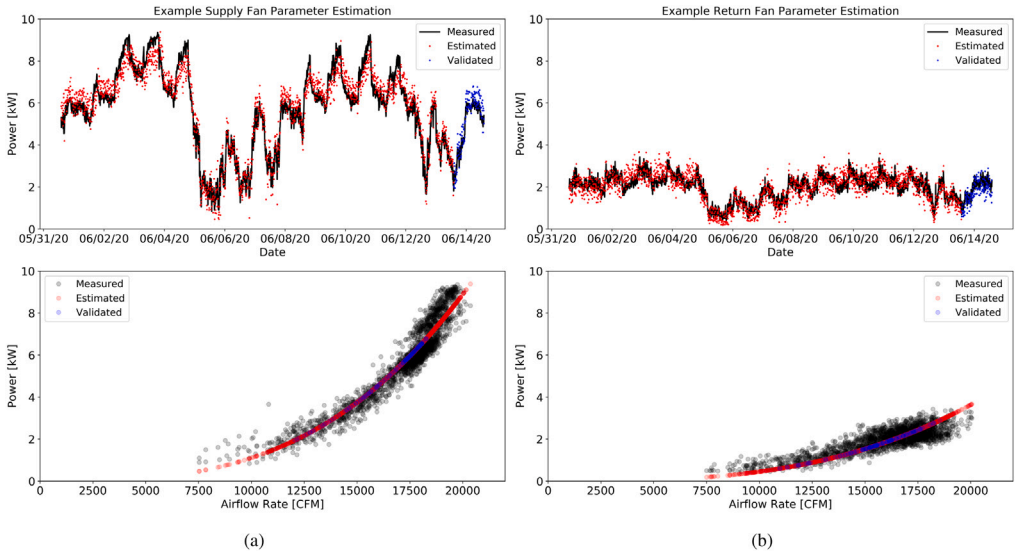


Fig. 7. Example result of parameter estimation process for the supply (left) and return (right) fan models of RTU1 initialized on June 15, 2020 showing measured, estimated, and validated fan power (top) and measured, estimated, and validated fan power as a function of supply air flow rate (bottom).

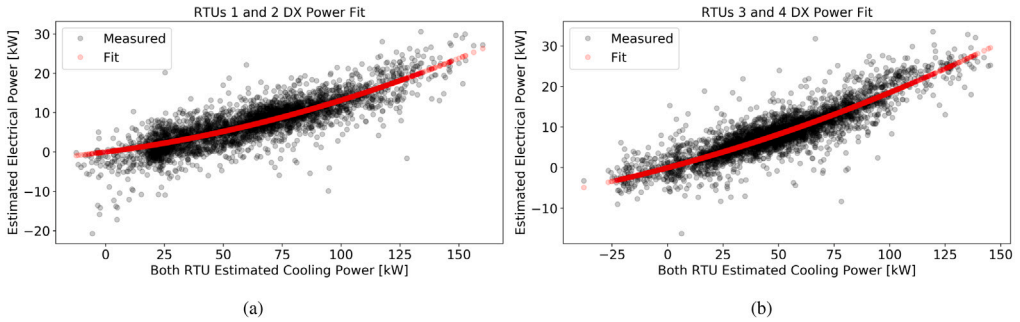


Fig. 8. Parameter estimation results for DX compressor power as a function of cooling power as estimated per HVAC electrical panel serving RTU 1 and 2 (a) and RTU 3 and 4 (b) using data from August 1, 2020 to October 18, 2020.

The estimation of the parameters for the DX power model of Eq. (3) was done differently than the envelope and fan model parameters due to the availability of data and needs for additional data processing. As described earlier, there are no direct power measurements of the compressors for each RTU DX. Instead, there are compressor control signal measurements available from the BMS for each RTU DX compressor, and the HVAC power panel measurements as described in Section 3, where one panel contains RTU 1 and 2 and the other RTU 3 and 4. Both panel electrical measurements, however, include power for the UFTs, exhaust fans throughout the building, and additional ancillary equipment. Therefore, the power for the DX compressors was estimated by filtering the panel measurements for any of the associated RTU compressor signals was greater than 0, subtracting the power measured by the fan VFDs for each RTU, and subtracting a constant offset of power for each panel representing the other equipment served by the panel than the DX compressors. For estimating the parameters of Eq. (3), the cooling power delivered by each RTU was estimated from BMS measurements of mixed air temperature, T_{ma}, supply air temperature, T_{sa} and supply air flow rate according to

$$\dot{Q}_{cool} = \dot{m}_{sf}\, c_p\, (T_{ma} - T_{sa}), \tag{9}$$

where c_p is 1006 J/(kg K), the cooling power for each RTU on a particular panel is summed, and the panel power offset is determined within the parameter estimation process such that the trend of estimated power as a function of cooling delivered passes through the origin. The resulting parameters determined for each panel are used for both RTU DX power models associated with the panel. This process was done once with operational data from August 1, 2020 to October 18, 2020 and the results are shown in Fig. 8. For RTUs 1 and 2, the RMSE is 3.1 kW and R2 is 0.71, while for RTUs 3 and 4 they are 2.9 kW and 0.77 respectively. Note that the estimation of negative cooling power, especially for RTU 3 and 4, indicates calibration needed for the supply and mixed air temperature sensors. Note that the display of negative electrical power results from estimating the power of other equipment on the panels that could not be disaggregated with a constant offset and subtracting this offset from panel measurements such that the fitted DX power model passes through zero electrical power at zero cooling power. In addition, the estimation of parameters was done with the Python package *scipy* [30] function *curve_fit*. In the future, adding sensors for measuring compressor power directly for each RTU individually and improved sensor calibration would help improve the model and ability to estimate parameters.

4.5. Weather forecasting

Weather forecasting is responsible for obtaining forecasts of the weather-related input variables of the MPC model, described in Section 4.3. The Weather Underground [31] and, later, Dark Sky services [32] are used to provide forecasts at a time step of one hour and horizon of 24 h for a number of weather parameters, including among others temperature, humidity, dew point, pressure, wind speed, wind direction, and cloud cover. A Python module has been written to use these services' HTTP RESTful APIs to query, retrieve, format, and write forecast data to the project database.

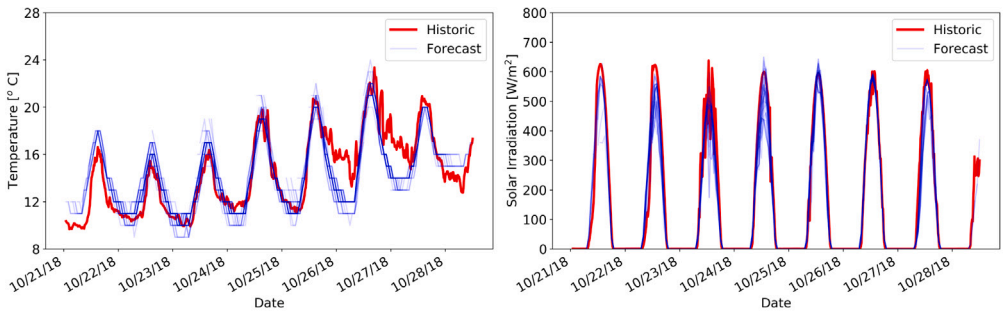


Fig. 9. Measured historic (red) and forecasted (blue) values over a one-week period for outside dry bulb temperature (a) and global horizontal solar irradiance (b). Forecasts retrieved from the Weather Underground service [31].

Unfortunately, these services do not provide solar irradiance explicitly. Instead, they provide cloud cover forecasts. In this study, these cloud cover forecasts are used to determine solar irradiance, specifically global horizontal irradiance, through the use of a model. The implementation of this model uses the machine learning algorithm k-Nearest Neighbors. The machine-learned model is trained using past measurements of solar irradiation from LBNL's weather station in combination with past 1-step cloud cover and temperature forecasts as well as predicted clear-sky irradiance. Then, the model can be used to predict solar irradiation using cloud cover and temperature forecasts as well as predicted clear-sky irradiance as inputs. The model itself predicts the ratio of actual solar irradiation to clear-sky prediction, and then uses the clear-sky prediction to multiply by this ratio to get the final solar irradiation forecast. The algorithm is implemented in Python using the scikit-learn v0.20.2 Python package [33]. For the algorithm, the number of nearest neighbors is five and the amount of previous data is 30 days. Clear-sky predictions are made for the project demonstration site location using the Python package pvlib v0.6.0 [34]. A new model is trained based on the previous data at every call to forecast the weather. The solar irradiation forecasts are added to the rest of the weather forecast points and written to the database. Fig. 9 shows sample forecast and historic measured data using an original implementation with Weather Underground. In the figure, 24-h forecasts are drawn using translucent blue lines starting at each hour the forecast was retrieved, such that darker blue areas represent values at particular times that multiple forecasts predict. Additional data taken during the evaluated MPC performance period in Section 5, and using the Dark Sky service after a required switch, can be found in Appendix.

Future work could include the testing and comparison of various solar irradiation forecast techniques to improve the solar irradiation forecasts, such as the use of an empirical model [35] developed or the use of pvlib's solar irradiation forecasting capability.

4.6. Internal load forecasting

Internal load forecasting is responsible for obtaining forecasts of the internal heat gain variables of the MPC model, described in Section 4.3. Internal loads come from three components: plug load, lighting, and occupant heat gains. [36] found plug load is a good proxy variable for total internal load. Therefore, we forecast plug load rather than the total internal heat gains, which could achieve an acceptable accuracy while reducing model complexity. A simple plug load prediction algorithm was used which predicts plug load as the average value of the same hour of day and the same day of week of the previous 90 days.

Fig. 10 shows sample results from the forecasts with measured data from before the COVID19 pandemic. Data from the MPC performance evaluation period in Section 5, occurring during the COVID19 pandemic with significantly lower occupancy, can be found in the Appendix.

4.7. Control optimization

The optimal control problem is defined for the current time, t_0, to the time at the end of the prediction horizon, t_f, by

$$\min_{u_{sf},u_{dx},u_{oa},u_{hw},s_1,s_2,s_3} \quad J = \int_{t_0}^{t_f} (P + w_1 s_1^2 + w_2 s_2^2 + w_3 s_3^2 + k_1 u_{sf}^2 + k_2 u_{dx}^2 + k_3 u_{oa}^2 + k_4 u_{hw}^2) dt \tag{10}$$

$$\text{s.t.} \quad f(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \tilde{\boldsymbol{\omega}}, \boldsymbol{\theta}_e^*, \boldsymbol{\theta}_f) = 0 \tag{11}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0^* \tag{12}$$

$$T_z - s_1 \leq T_z^u \tag{13}$$

$$T_z + s_2 \geq T_z^l \tag{14}$$

$$T_{sa} + s_3 \leq T_{sa}^l \tag{15}$$

$$T_{sa} \leq T_{sa}^u \tag{16}$$

$$\dot{m}_{sa} \geq \dot{m}_{sa}^l \tag{17}$$

$$\dot{m}_{oa} \geq \dot{m}_{oa}^l \tag{18}$$

$$0 \leq u_{sf}, u_{co}, u_{oa}, u_{hw} \leq 1 \tag{19}$$

$$s_1, s_2, s_3 \geq 0. \tag{20}$$

The total power P is equal to the sum of supply and return fan power, DX power, and UFT reheat power. Upper and lower constraints on zone temperature, $T_z^u = 23.2$ °C and $T_z^l = 22.2$ °C, and a lower constraint on supply air temperature, $T_{sa}^l = 17.2$ °C are implemented with slack variables s_1, s_2, s_3 and corresponding weights w_1, w_2, w_3. These constraints are implemented with slack variables instead of hard constraints to ensure feasibility. This is typical for the zone temperature, and used for the lower limit of supply air temperature in the case the outside air temperature is too low to maintain the minimum supply air temperature and minimum outside air flow requirement. The upper limit on supply air temperature, $T_{sa}^u = 21.1$ °C, minimum supply air flow $\dot{m}_{sa}^l = 4.75$ m^3/s, and minimum outside air flow, $\dot{m}_{oa}^l = 2.36$ m^3/s are implemented as hard constraints, along with control signals to be between 0 and 1 and slack variables to be non-negative. The minimum supply air flow constraint is added as a safety measure to ensure sufficient air flow is provided through the DX coil at all times to prevent freezing. Regularization on the RTU control signals is added in the objective function for the opportunity to help prevent oscillatory control solutions with weights k_1, k_2, k_3, k_4. For this field study, through offline tuning, it was determined that all weights be equal to 1 except $k_1 = 1e4$. In addition, $\boldsymbol{\theta}_e^*$ is the set of estimated parameters resulting from the parameter estimation processes, $\tilde{\boldsymbol{\omega}}$ are forecasted disturbances, and $\mathbf{x}_0^*$ is the initial state estimation by the state estimator described in Section 4.8.

Note that this optimal control problem is formulated automatically in the Modelica language extension of Optimica by MPCPy given the

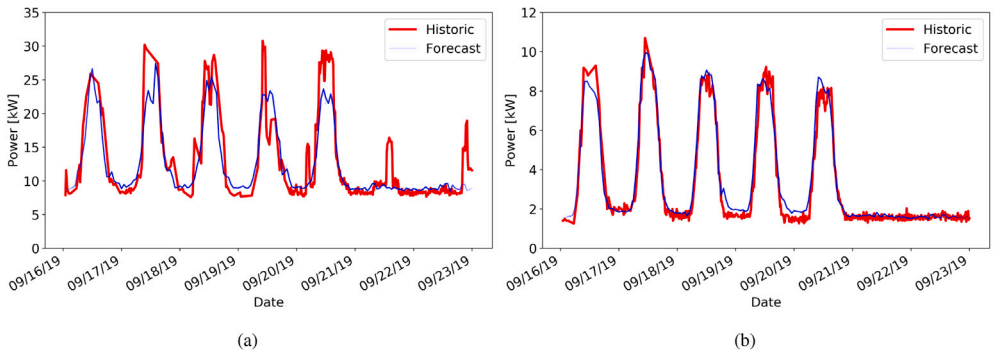


Fig. 10. Measured historic (red) and forecasted (blue) values over a one-week period before the COVID19 pandemic for north (a) and south (b) plug load electrical panels.

model f written in Modelica and specification in Python of the objective (energy minimization), objective variable (the output J in Fig. 3), constraint type, data, and applicable model variable (including whether to use a hard constraint or slack variable with specified weight), and input data.

For this study, the prediction horizon is 24 h. The set of estimated parameters are those from the most recent parameter estimation process results that achieved validation RMSE's for envelope of less than 1 Kelvin, supply fan power of less than 1 kW, and return fan power of less than 0.5 kW. These values were deemed reasonable for this application because the limits on fan power represented approximately 10% of respective design motor powers and 1 Kelvin is a reasonable value based on previous studies [5,37]. Also, since the existing HVAC control does not utilize enable/disable schedules or thermostat resets, except for second floor office thermostat set backs on Saturdays, the limits used for optimal control constraints are also constant in time.

4.8. State estimation

The state estimation process estimates the initial value of states to be used for the control optimization problem at each control step, $\mathbf{x}_0^*$ in Eq. (12). The state estimator used is similar to that which was implemented in [17] and represents a simple moving horizon estimator. The estimator solves an optimization problem from the start of a historic time horizon t_0 to current time t_f, defined by

$$\min_{\mathbf{x}_0} \quad J = \int_{t_0}^{t_f} (y - \hat{y})^2 dt \tag{21}$$

$$\text{s.t.} \quad f(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \boldsymbol{\omega}, \boldsymbol{\theta}_e^*, \boldsymbol{\theta}_f) = 0 \tag{22}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{23}$$

where $\boldsymbol{x}_0^*$ of Eq. (12) is equal to $\boldsymbol{x}(t_f)$ of the state estimation problem. Note that this state estimation problem is formulated automatically in the Modelica language extension of Optimica by MPCPy given the model f written in Modelica and specification of measured variable, tunable states, and input data in Python. In this study, the historic time horizon is 24 h, tuned based on offline trial and error for reasonable performance.

4.9. BMS integration

The implementation scheme for the MPC controller to send set points to the Building 59 ALC system is presented in Fig. 11. The MPC controller on the server in Building 90 runs a service called "Set set points" which, every 1 min, pulls optimized set points from the Influxdb database, checks that the set points are valid, and sends them to the ALC RTU Manager program via the ALC SOAP web interface. These inputs are checked by logic in the ALC RTU Manager and any constraints on their allowable ranges and ramps are applied. The resulting checked inputs are sent to a signal selector, which selects which set point to finally send to the field controllers between the checked MPC inputs and the default set points from existing sequence logic. The signal selector is controlled by three sets of logic. One allows for an operator to manually enable/disable MPC control at the manager level and for each individual RTU. A second is an automatic "watchdog" that verifies signals have been received within the last 20 min from the MPC controller, disabling MPC control if the time has expired. The third allows for enable/disable signals sent from the MPC controller. Email and text message alerts are sent out to appropriate individuals stating when and how the MPC control is enabled or disabled.

The ALC points that can be written to by the MPC controller are collected into a group with a specific privilege set that allows them to be commanded by a user account created for the MPC. All other (existing) points on the RTU Manager program are configured with privileges sets that cannot be commanded by the MPC account. Finally, location-based privileges prevent the MPC account from commanding any points outside of the RTU Manager program.

4.10. Other implementation notes

The implementation of MPC often requires a high degree of expertise in a number of topics, including telemetry, building operation, physical modeling, and optimization. Researchers in building science or controls often do not have vast experience or expertise in software development and deployment. Therefore, this section includes a few notes that are not often included in the report of MPC demonstrations that are meant to help facilitate software implementation and operation in future studies.

4.10.1. Forecast data

The first note is on the storage of forecast data, including not only weather and internal load predictions, but also MPC predictions resulting from the control optimization service. Storing forecast data proved beneficial for two reasons. First, it allows for the execution of the control optimization routine at any time in the past exactly as it was (i.e. with all of the same data), or would have been, executed at that time in the past. This was very helpful in debugging, tuning, and further understanding MPC performance. Second, storing forecast data allows for further analysis comparing forecast data to realized historic data, eventually allowing for a better understanding of the accuracy of forecasts and their impact on MPC performance in real-world, practical implementations.

Storing forecast data, however, is not straight-forward. Unlike historic timeseries data that is made up of time–value pairs, forecast data is made up of time–timeseries pairs. Consider a forecast at a given time, say midnight, is created for a given forecast horizon, say 24 h. Then, one hour later, at 1 AM, another forecast is created for 24 h. Notice that for the given measurement, 2 AM has two forecasts, one associated with the one from midnight, and one associated with the one from 1 AM. This makes it difficult to simply assign a single timestamp–value

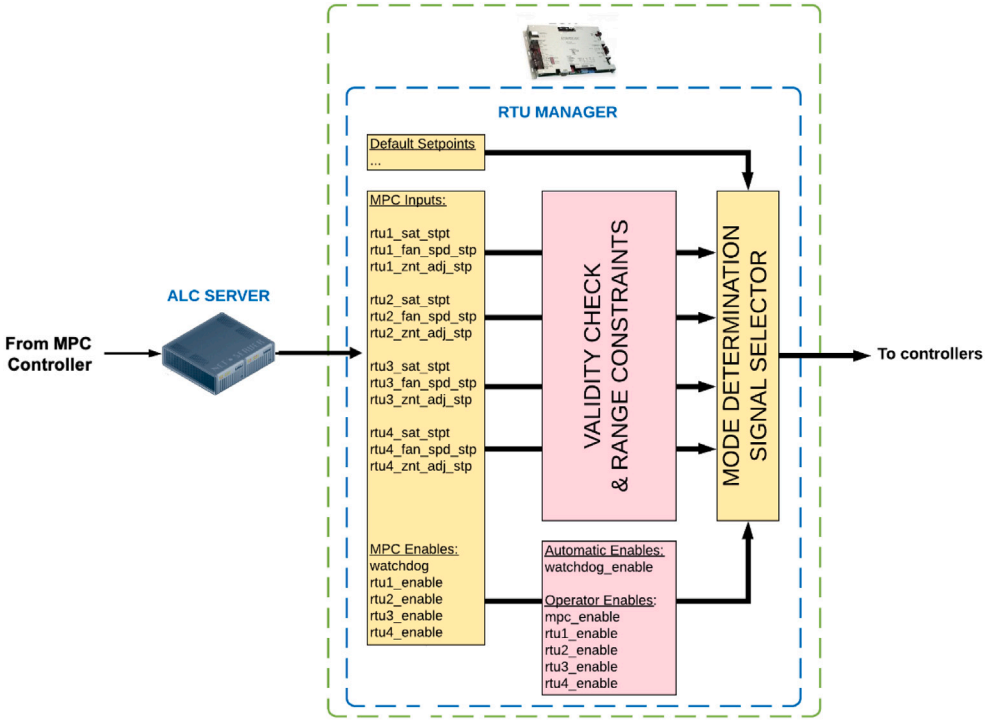


Fig. 11. Integration of MPC controller with BMS system. Refer to Fig. 2 for the MPC controller architecture.

pair to the forecast value of the measurement at 2 AM, or any other time. Therefore, for the InfluxDB database, a scheme was developed to store a forecast with a given time step, Δt, and starting time, t_0, which may be different than the creation time, under a measurement name and a timestamp equal to the time the forecast was created t. Then, for each measurement name and timestamp, additional fields are created with names corresponding to $1, 2, \ldots, n$ and values equal to the forecast value $t_0 + n\Delta t$ in the future. The timestep is automatically detected based on the forecast data obtained and additionally stored under the measurement name and timestamp of the forecast. Therefore, for a given measurement name, forecasts are stored according to the time at which they were created t, as a collection of fields with a known order $1, 2, \ldots, n$, time step Δt, and starting time t_0. When the forecast data is requested for the measurement name and timestamp, the forecast is built by retrieving each field value in order, and pairing them with the corresponding timestamp equal to $t_0 + n\Delta t$.

4.10.2. TMUX and CRON

Execution of the parallel processes on a single remote server was greatly aided by the use of two utility software packages not often known to MPC researchers and developers: *tmux* and *cron*. For Unix-like operating systems, *tmux* enables access to multiple terminal sessions from within a single window, which is often what is available when configuring operation in a remote server. Ultimately, this allows a user to execute multiple commands from command-line prompts in parallel terminal sessions. In the case of this MPC implementation, this allowed for the starting, stopping, and configuration of each service independently and without interruption of the others on the single remote implementation server. Also for Unix-like operating systems, *cron* is a time-based job scheduler, where commands are executed as specified in a *crontab* file. In the case of this MPC implementation, this allowed for the execution of service tasks at regular intervals once the service was deployed. For example, after configuring the control optimization service and deploying using Docker and *tmux*, *cron* was used to execute a run command within the deployed Docker container every 0:10, 0:20, 0:30, 0:40, and 0:50 of every hour of every day.

5. MPC performance and analysis

MPC control was activated approximately every-other week between October 18 and December 15, 2020. Exceptions were made according to planned building maintenance schedules and issues that arose during testing, described in Section 6. In total, there were 31 days of MPC ON control and 27 days of MPC OFF control. The approach of alternating between MPC ON and MPC OFF has the aim of obtaining comparable weather conditions for both methods of control. One week was considered long enough to alleviate the influence of initial conditions after switch-over. Fig. 12 shows the weather conditions and indication of the MPC ON and MPC OFF periods during the test.

As shown in Fig. 2, the historic data collection process executes every 10 min, the weather forecast process every 30 min, the internal load forecast process every 60 min, the parameter estimation processes for the envelope and fans every 24 h, the control optimization process every 10 min, and the process setting the set points in the ALC every 1 min, at higher frequency than the optimization process so that the process can be attempted many times in case of connection losses lasting only very short durations.

Figs. 13 and 14 show example performance of RTU1 with MPC ON and MPC OFF during representative warm days (MPC ON is October 23, 2020 and MPC OFF is October 28, 2020) and cold days (MPC ON is November 15, 2020 and MPC OFF is November 10, 2020). During warm days, the MPC OFF control keeps the supply air temperature set point constant for the duration of the day. In addition, the constant static pressure set point and low controllability of air flow rate at the zone level cause the supply air flow rate to be relatively constant throughout the day at about 80% of the design flow rate. Notice the DX unit is utilized in the afternoon when the economizer is disabled. During this time, the outside air flow rate is maintained at the minimum for ventilation. Meanwhile, for a similar type of day with MPC ON, the MPC lowers the supply air temperature set point during the early morning and increases it slowly as the outside air temperature increases. With cooler supply air, the MPC also lowers the fan speed, supplying only half of the design air flow rate. When the cooling load increases in the afternoon, the MPC first reduces the supply air temperature

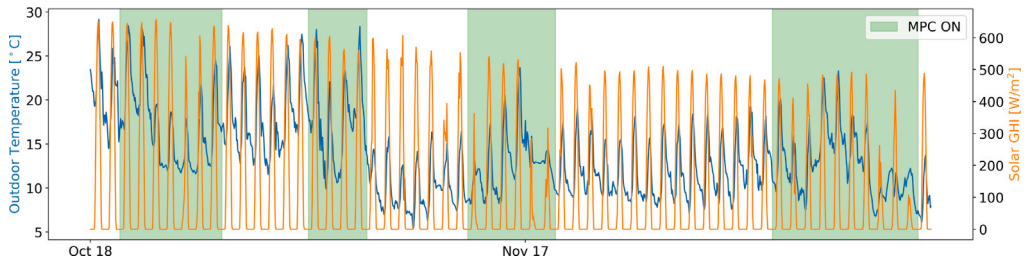


Fig. 12. Outside air temperature (left axis) and global horizontal solar irradiation (right axis) during the test period. Times when MPC ON control was active are shown in green.

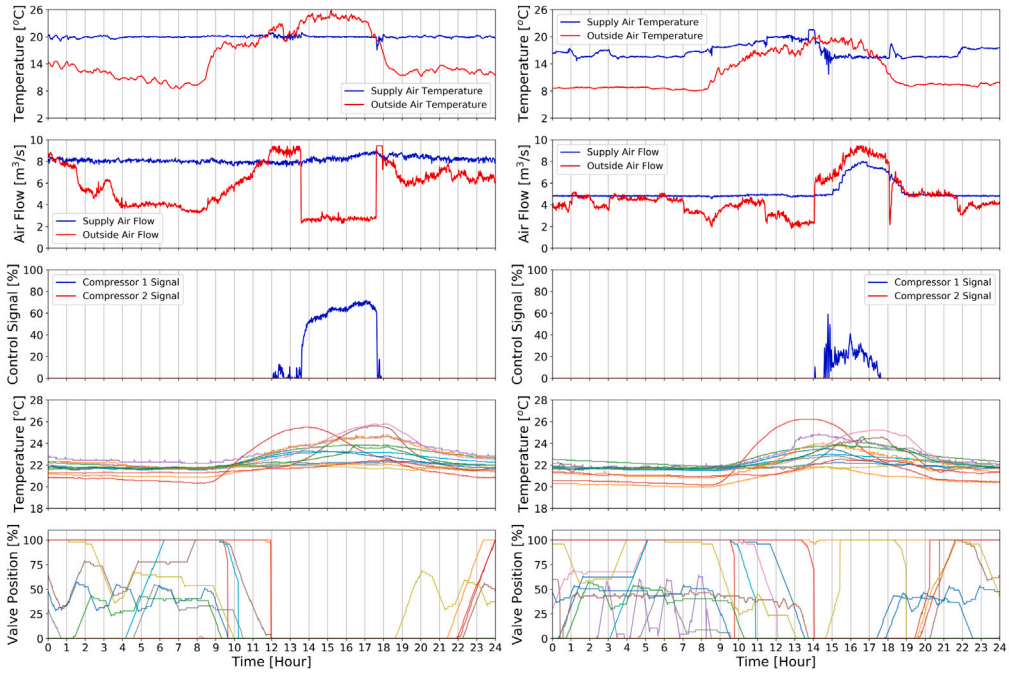


Fig. 13. Warm day supply (blue) and outside (red) air temperatures (top), supply (blue) and outside (red) air flow rates (second), compressor 1 (blue) and compressor 2 (red) control signals (third), individual zone air temperatures (fourth), and individual zone reheat valve positions (bottom) for RTU1 under MPC OFF (a) and MPC ON (b) control. MPC OFF is October 28, 2020 and MPC ON is October 23, 2020.

set point and then increases the supply air flow rate as needed. The economizer is never disabled on this day as the outside air temperature stays relatively cool.

During cooler days, the MPC OFF control keeps the supply air temperature set point constant all day, although it is not maintained in the early morning due to the need to supply minimum outside air with very low outside air temperatures. Similar to the warm day, the supply air flow rate is relatively high and unchanged through the whole day. Meanwhile, for a similar type of day with MPC ON, the MPC shows similar behavior with the supply air temperature set point as the warmer day, but increases the supply air flow rate in the early morning in order to maintain the supply air temperature set point and minimum outside air requirement with increased recirculation air. Since the day has very low cooling loads, the MPC does not increase the supply air flow rate and increases the supply air temperature set point as needed to avoid DX usage.

Fig. 15 shows the distribution of temperatures for all zones served by each RTU for each hour of the day over all days with MPC ON and MPC OFF. The boxes indicate the interquartile range (IQR), the whiskers represent data that falls within $1.5IQR$, and the dots represent data that falls outside of this range (outliers). Overall, the distributions shown for MPC ON are similar to those for MPC OFF, although MPC control does show upper whiskers that are approximately 1 °C warmer and 0.5 °C colder in the afternoon for RTU1. Individual zone measurements in Figs. 13 and 14 show that zones that are already particularly sensitive to the sun under MPC OFF control can get warmer under MPC ON control. This is likely due to lower total air flow into the plenum under MPC control and lower air flow through the perimeter underfloor terminal units. In the future, though more complex to generate, utilizing an envelope model with increased zone resolution would help account for individual zone performance.

Fig. 16 shows daily HVAC energy and discomfort data as a function of daily mean outside air temperature under MPC ON and MPC OFF control. Since second floor office thermostats are setback on Saturdays, likely leading to less energy consumption than usual, and restored on Sundays, likely leading to more energy consumption than usual, these days are specifically highlighted. Dashed lines show five-parameter piece-wise linear regressions for each of MPC ON and OFF generated using the inverse modeling open-source software described in [38]. The daily energy consumption, E_d, for a day d is calculated using power measurements from the two HVAC electrical panels, P_{hvac1} and P_{hvac2}, along with the measured reheat water heat flow rate at the ASHP, Q_{hw}, and estimated heat pump COP, COP_{hp}, of 2.0, according to

$$E_d = \int_{t_0}^{t_f} P_{hvac1} + P_{hvac2} + \frac{Q_{hw}}{COP_{hp}} dt. \tag{24}$$

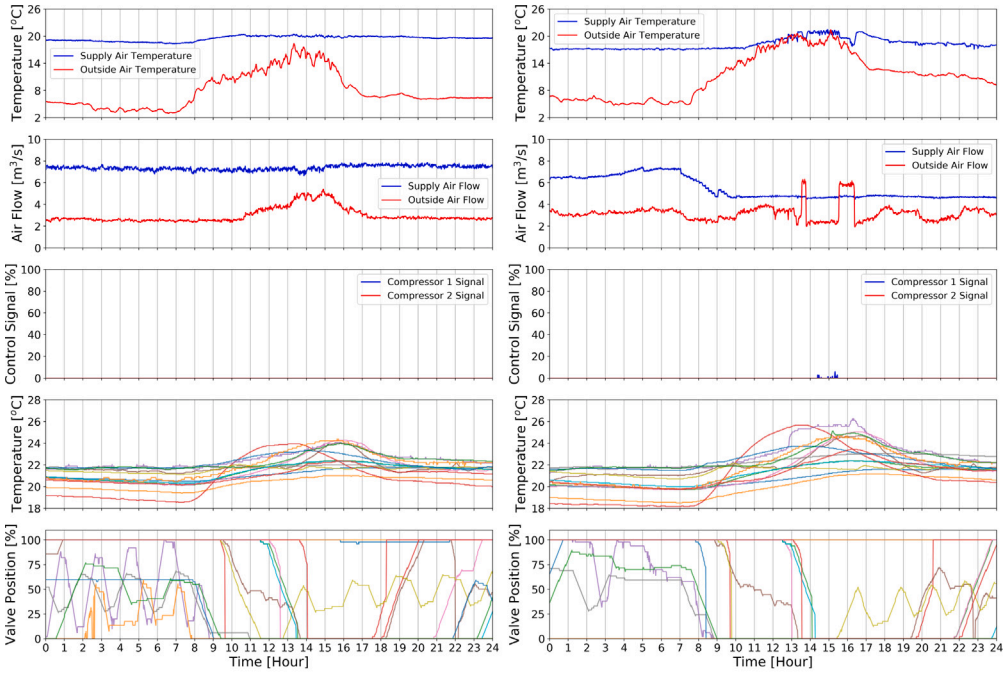


Fig. 14. Cool day supply (blue) and outside (red) air temperatures (top), supply (blue) and outside (red) air flow rates (second), compressor 1 (blue) and compressor 2 (red) control signals (third), individual zone air temperatures (fourth), and individual zone reheat valve positions (bottom) for RTU1 under MPC OFF (a) and MPC ON (b) control. MPC OFF is November 10, 2020 and MPC ON is November 15, 2020.

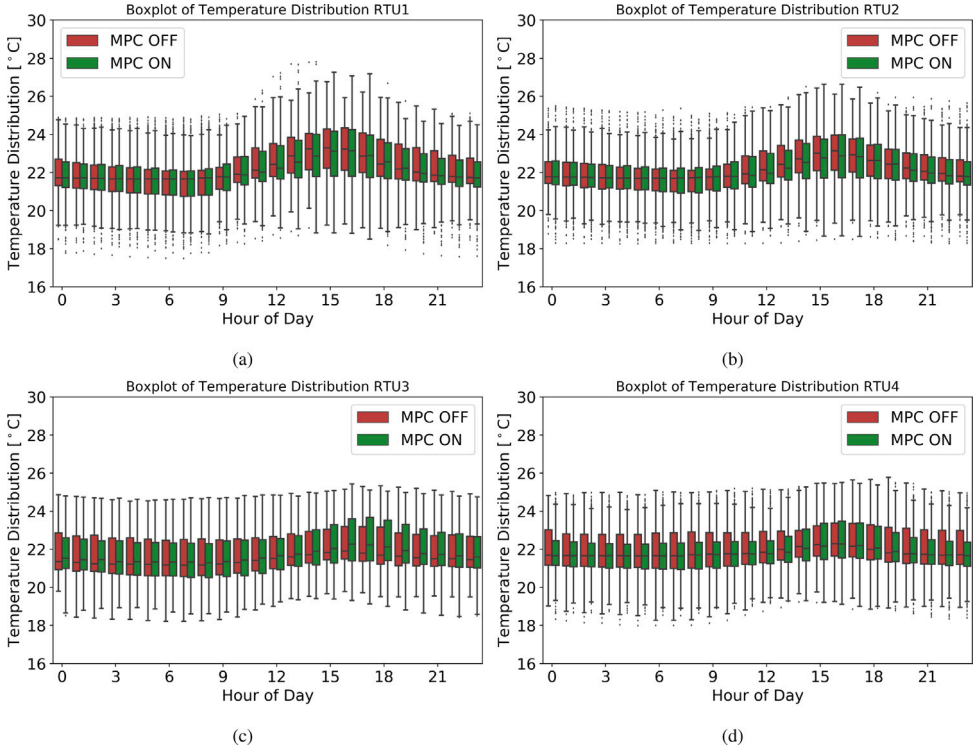
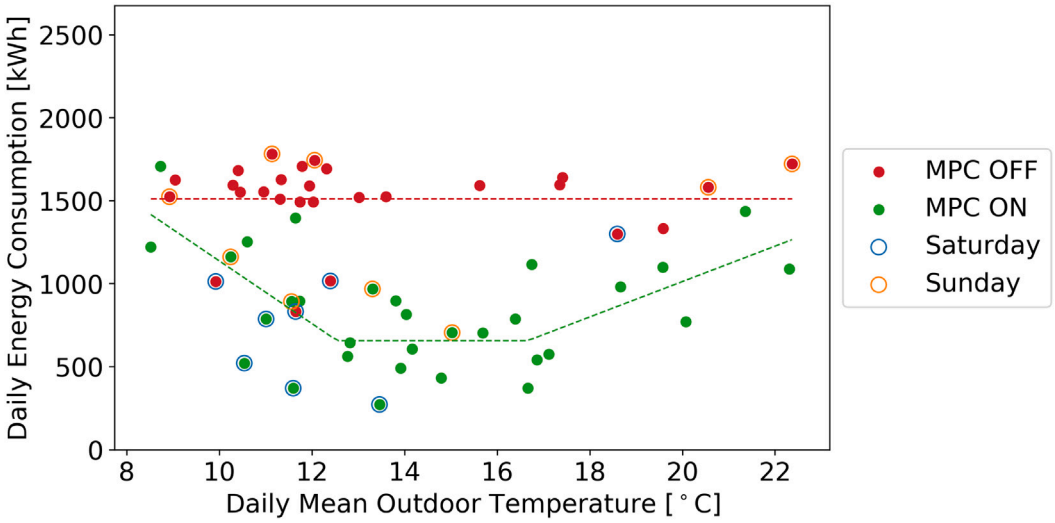
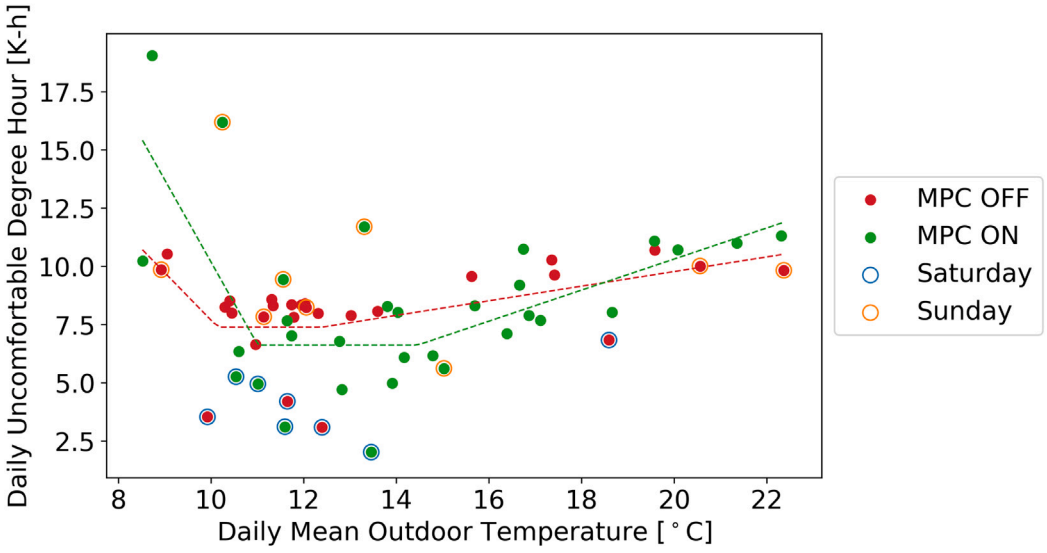


Fig. 15. Hourly zone temperature distributions for zones under RTU1 (a), RTU2 (b), RTU3 (c), and RTU4 (d) for MPC OFF (red) and MPC ON (green) control.



(a)



(b)

Fig. 16. Daily HVAC energy consumption (a) and discomfort (b) under MPC OFF (red) and MPC ON (green) control with five-parameter piece-wise linear regressions (dashed).

The zone-wise averaged, daily discomfort for a day d, denoted as D_d, is calculated using the deviation between the temperature measurement, T_z, of each zone z for total number of zones Z, and that zone's heating and cooling set points, T_z^h and T_z^c, according to

$$D_d = \frac{\sum_z^Z \int_{t_0}^{t_f} max(T_z^h - T_z, 0) + max(T_z - T_z^c, 0) dt}{Z}. \quad (25)$$

The set points are taken from the ALC system for perimeter zones served by UFTs and the set points are assumed 21 °C and 24 °C for core zone temperatures measured by the Raspberry Pi sensors.

Fig. 16 shows that MPC ON control leads to energy consumption that is more dynamic with respect to weather conditions compared to MPC OFF control, significantly reducing energy consumption during mild days. The data point at the top left of the discomfort figure under MPC ON control is obtained during the last day of testing during the coldest conditions seen during the testing period. Data obtained after the testing period with MPC OFF control showed similar discomfort under similar cold conditions. Energy savings over the testing period are estimated by integrating the two piece-wise linear models over the range of mean outdoor temperature and taking the difference, resulting in a 40% reduction by MPC ON compared to MPC OFF. The most significant source of this energy savings is the reduction in fan power during low load conditions.

6. Discussion

As implementation effort is a key barrier to wide-spread adoption of MPC, we discuss here a number of practical issues that were overcome during the implementation and operation of the MPC. In addition, we discuss practical solutions that worked particularly well. Finally, we estimate the implementation and operation effort for the MPC.

6.1. Practical issues

An initial type of issue is the number of sources from which data is needed to implement the MPC and varying quality of that data. For the MPC implemented in this study, different sources were needed for each of HVAC system data, electrical system data, weather measurement data, weather forecast data, and additional temperature sensor data. Each source required dedicated access management (e.g. user account, API key, etc.), data interpretation effort, and custom software for accessing and formatting the data for use by the MPC. Missing data

also needed to be identified and accounted for, as was described for the parameter estimation service earlier. In the future, streamlining the processes for accessing, understanding, and cleaning different types of data will reduce the implementation effort for MPC and can be helpful guidance for MPC projects.

Another issue is accounting for deficiencies in any of the data accuracy. One example from this study relates to outside air temperature measurements. In 2018 and 2019, outside air temperature data measured by RTU-specific sensors contained up to approximately a 4 °C error compared to each other. Compared to the outside air temperature measured by the campus weather station, the RTU sensors showed up to a 4 °C error low during night time and approximately 5 °C error high during day time, likely caused by the influence of solar and sky irradiation on nearby RTU surfaces. Early development of the MPC used the campus weather station measurements and eventually in 2020, additional outside air temperature sensors were installed by facilities personnel for the BMS that agreed more closely with the campus weather station and used by the RTUs for economizer control. However, usage of such earlier data during MPC field trials would have required more attention in how to incorporate the data differences within the MPC. Another example is the impact measurement accuracy has on the calculation of derived quantities, such as the DX cooling rate in Eq. (9). At 80% design air flow and a supply air temperature of 20 °C, a 1 °C error in mixed air temperature measurement leads to an error in cooling rate of 9% of the capacity of the DX unit, while a 3 °C error leads to 26%. This highlights the sensitivity to mixed air temperature measurement error that depends not only on the accuracy of the sensor itself, but also its location within or near the mixing box as the return and outside air streams are mixed.

Once data sources are established, another type of issue is the unexpected loss in communication with those sources or other services during operation. An initial example from this study is when ALC maintenance required the unexpected restart of the ALC services needed for the RTU Manager Program to accept set points from the MPC. In this case, the watchdog correctly detected the loss of communication with the MPC, reverted to MPC OFF control after 10 min, and then reverted back to MPC ON control once communication was restored, approximately 30 min later. Another example is when the server hosting the database for the MPC had an unexpected failure that caused it to shutdown. This prompted the MPC services to report errors, the service sending set points to the ALC to not send set points, and ultimately the ALC watchdog to switch to MPC OFF control after 10 min of not hearing from the MPC. Once the database was restarted on a new server the next day, the MPC controller was restarted and MPC ON control was enabled again. Final examples that delayed the start of MPC ON control are when the online service providing the weather station measurements stopped providing updated measurements for a few hours and when the online service providing electrical measurements suddenly required a user account modification to access data, which required a few days to resolve. Detecting and handling these different sources of data failures was made more challenging by the fact that each service was managed by a different entity or group. These incidents highlight the many types and causes of communication loss that may occur and the importance of the watchdog functionality. In the future, streamlining data management and access services would reduce the effort needed to anticipate, identify, and resolve these issues.

A final type of issue is the unexpected disabling of equipment and its potential impact on MPC operation. A first example was the tendency for the RTU3 compressor alarm to trip and disable the availability of its compressors for short time periods, up to a few hours. This would occur during both MPC OFF and ON periods and the final reason could not be found during the study period. The first time this happened during MPC ON, the upper limit of the compressor control signal constraint for RTU3 was set to 0 manually, to make the optimizer aware that the compressor was not available. After the compressor came back online, the constraint change was reverted. After it became clear that the compressor alarm would only activate for relatively short periods and that the cooling load on RTU3 was typically very small, the constraint was not changed again. A second example is when the heating system needed to be suddenly shut down for a few hours for maintenance. In this case, this incident also did not have any adverse affect on MPC operation. However, generally speaking, an improvement to the robustness of any MPC would be to have it be able to detect when alarms are active or equipment is disabled in order to dynamically adjust constraints accordingly, without a human-in-the-loop.

6.2. Practical solutions

It is also worthwhile to highlight some practical solutions that worked particularly well, which could be incorporated into future implementations. From a technology point of view, the gray-box modeling approach worked well with one to three weeks of previous operational data. Regular updating of model parameters allowed for continuous adjustment to any seasonal changes in operating conditions for the envelope and other changes in operation of the HVAC system. For example, in one instance during early offline trials, the coefficients for the power-flow model of the RTU fans were modified when RTU filters were replaced. In addition, treating zone and supply air temperature constraints as objective penalties rather than strict inequalities facilitated feasible optimal control solutions. MPCPy's automatic optimal control problem generation made it easy to add, subtract, and modify (e.g. change between hard or soft treatment and tune penalty weights) constraints during final controller verification phases. Missing measurements added challenges to the parameter estimation of gray-box models and internal load predictions. For parameter estimation, data preprocessing scripts were very useful to robustly automate the process by filling in short (minutes to hours) gaps, avoiding large gaps (hours to days), and selecting the most recent period of data that met our minimum data requirements. For load prediction, we found simple heuristic algorithms (like using the average load of the same hour of week during the past 12 weeks) were more robust to missing values compared with machine learning algorithms such as LSTM.

From an implementation point of view, the architecture of the MPC controller using parallel services coupled through a central database worked well in allowing for independent development, maintenance, and monitoring of each service. The storage of all weather and internal load forecasts through the duration of the implementation facilitated offline tuning and debugging of the MPC controller with consistent boundary condition data seen during MPC operation or previous development phases. Email alerts embedded within the MPC controller software code proved useful in alerting developers of potential problems during operation, which could be fixed before severe failure of the MPC occurred, or could prompt an expected changeover to existing operation. The BMS watchdog that monitored the communication with the MPC controller worked very well to ensure timely changeover to existing operation when MPC service failures did occur. Running the MPC controller in real time for long periods and logging intended control set points to the BMS without the BMS actually utilizing those set points allowed for final verification, and tuning where necessary, of stable operation of all services, including alerts, before enabling closed-loop control.

6.3. Implementation effort

In an effort to begin quantifying the impact of the practical issues described before, and therefore be able to measurably address them, Table 2 estimates the effort in person-days needed for various tasks in the implementation and operation of MPC: Preparation, Model Development and Integration, Controller Software Development, Controller Deployment. The estimation was done from the analysis of progress and reported effort for each quarter of the project on various tasks associated with this study. Note that the effort required for the initial

Table 2
Estimation of effort required for implementation and operation of MPC.

Task	Subtasks	Person-days	Fraction of total
Preparation	System design analysis, Data collection & analysis, New sensor installation, Occupant survey	70	0.29
Model development & integration	Thermal envelope, HVAC, Internal load forecast, Weather forecast	79	0.33
Controller software development	Architecture design, Service implementation, Log & error handling processes	30	0.13
Controller deployment	Facilities coordination, BMS logic editing, Functional testing, Commissioning & maintenance, Performance evaluation	60	0.25

implementation of MPCPy itself is not included in this estimation, since MPCPy was developed to be a more generalized, open-source, documented software tool, adding requirements beyond the scope of a single MPC implementation. Nonetheless, the effect of its availability can be seen by the fact that Controller Software Development for this MPC implementation required the least estimated effort compared to the other tasks. This effort estimation represents a first-time implementation of field-operated MPC by the majority of the research team and facility personnel, and represents the effort of a two-person MPC implementation team and two-person management team, whose effort is included only for their participation in key coordination meetings. As teams implement successive MPC controllers, it should be expected that the required implementation effort is reduced as experience is gained, implementation techniques are learned, and software code can be re-used.

Aside from Controller Software Development, the other three tasks each contributed significantly. While model development required the most effort at approximately a third, it did not by itself dominate the effort required to implement the MPC. Preparation tasks involving collecting and understanding data, installing necessary new sensors, and understanding occupant satisfaction with the existing system required almost as much effort as model development. Moreover, once models are developed, they can readily be adapted to other similar systems. Tasks related to controller deployment, especially those related to coordinating with facility personnel to communicate the intent of the MPC, plan for secure access to data and set point modification, and minimize operational risks, also required a significant amount of effort, approximately a quarter of the total.

This analysis for this case study indicates that, while model development still requires significant effort, practical tasks of implementation require comparable effort. In particular, those related to data collection and understanding and those related to coordinating with facilities personnel and interfacing with the BMS. Improved workflows for collecting, understanding, and managing data are needed to streamline the implementation preparation process. In addition, workforce development training and MPC implementation guides are needed to help communicate with all stakeholders at a broad scale the benefits and operational challenges associated with implementing MPC in order to help alleviate these challenges.

6.4. Key lessons learned

To summarize the discussion and takeaways from the implementation of the field study, these are four primary lessons learned:

1. Many separate sources of data increases the complexity and reduces the robustness of the MPC, since each source needs to be understood, setup, and maintained.
2. Data availability and quality informs model generation and model generation informs optimal control formulation, leading to an iterative design process until desired performance is achieved.
3. Data collection and controller deployment activities, including facilities coordination and implementation preparation, can each require as much effort as model development and integration.
4. Alarms, methods to handle MPC process failures or system equipment failures, and methods to automatically (e.g. watchdog) and manually switch to acceptable fallback control all help guard against total control failure.

6.5. Limitations and future work

This study tested the closed-loop control performance of MPC for two months during a transition season. Longer periods of field tests covering both the heating, cooling, and transition seasons could help to evaluate the performance of MPC for an entire year. This study also built the MPC upon existing building infrastructures, only installing additional sensors to collect air temperature in the core zones. Future work could potentially improve the MPC if other data is collected and incorporated into the MPC operation, such as the power consumption of each RTU compressor to improve modeling, CO_2 measurements to control indoor air quality, occupancy sensors to adjust schedules, and dynamic electricity prices or carbon emission signals to direct flexible load management. Finally, infrastructure implemented for this study can be re-used to implement MPC in other buildings on LBNL's campus, helping to further analyze and improve the scalability of the technology.

7. Conclusion

Previous studies have demonstrated the potential of MPC to help meet the growing needs for energy efficient and flexible load management in buildings. However, MPC has yet to gain mainstream adoption by the building industry because of challenges relating to high implementation costs, low data availability, and risk aversion. In this study, we developed an MPC controller using the free and open-source toolchain MPCPy to help reduce the implementation costs associated with controller software development, operated the controller for the UFAD HVAC system serving two office floors in a real building on LBNL's campus, compared its performance with the existing control, discussed the practical challenges associated with implementing the MPC, and estimated the effort required for implementation.

With regard to performance, compared with MPC OFF operation, MPC ON saved an estimated 40% energy over the two-month testing period, without significant penalty to thermal comfort. While promising, the savings should be understood in the context of the study, where existing control logic utilizes few schedules and set point resets and that the majority of these savings were observed on days when the climate was mild. The concerns for keeping enough cool air in the plenums to maintain cooling to 24-h IT rooms located on the office level combined with lack of explicit feedback of air flow and temperatures in the building core limited the operators' strategies for implementing dynamic schedule, static pressure, and supply temperature resets in the form of static sequences. In this case, the MPC was able to optimize resets (static pressure via fan speed control) to minimize energy while achieving service objectives. This included optimal pre-cooling during night with lower supply air temperatures and air flow in anticipation of the following day's cooling load. Follow-on work is planned to further demonstrate the advantages of MPC in layering additional performance objectives into the optimization, including grid-responsiveness, which would be even more challenging with static control sequences.

With regard to implementation effort, in addition to model generation, significant effort was spent on understanding, setting up, and

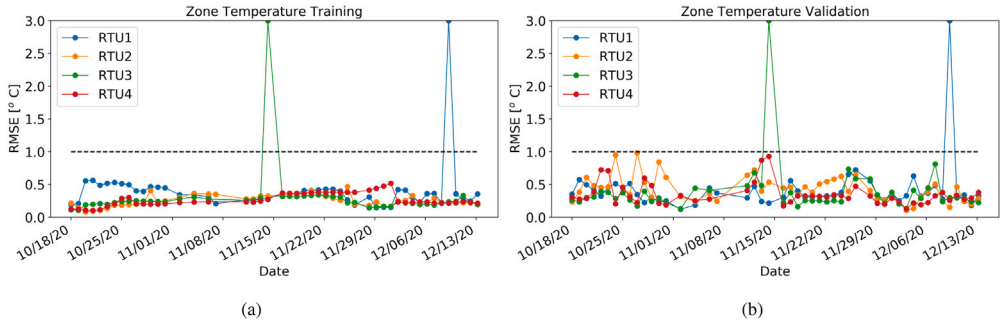


Fig. A.17. Envelope model training (left) and validation (right) air temperature RMSE over the MPC operating period for each RTU.

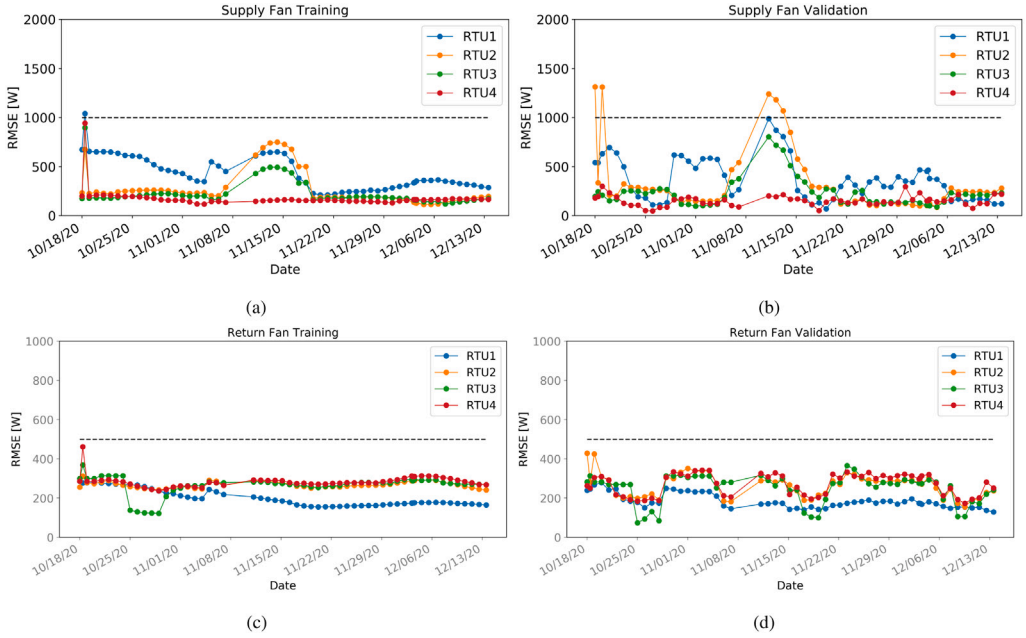


Fig. A.18. Supply (top) and return (bottom) fan model training (left) and validation (right) power use RMSE over the MPC operating period for each RTU. For reference, the supply fan motors have a nameplate power of 14.9 kW and return fan motors of 5.6 kW.

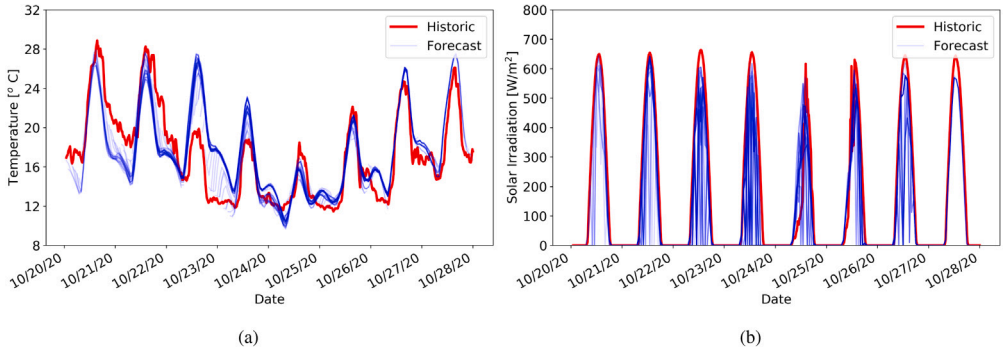


Fig. A.19. Measured historic (red) and forecasted (blue) values over a one-week period for outside dry bulb temperature (a) and global horizontal solar irradiance (b). Forecasts retrieved from the Dark Sky service [32].

maintaining access to data from various sources. Also, the relationships between data collection, model generation, and optimal control formulation lead to an iterative design process since models can only be calibrated with the data available and the chosen modeling approach constrains the optimal control problem formulation. Therefore, workflows that integrate and streamline data collection, access, and understanding processes, along with model generation and optimization problem formulation, would reduce this effort. In addition, implementing automated alert systems were key to allowing for robust operation of the MPC through data communication and equipment failures. Similarly, implementing an automated MPC-disable watchdog was key to enabling robust operation of the building when issues related to MPC operation could not be resolved quickly. Finally, controller deployment activities that involved facility coordination, BMS adjustments, and functional testing required significant effort that should be planned

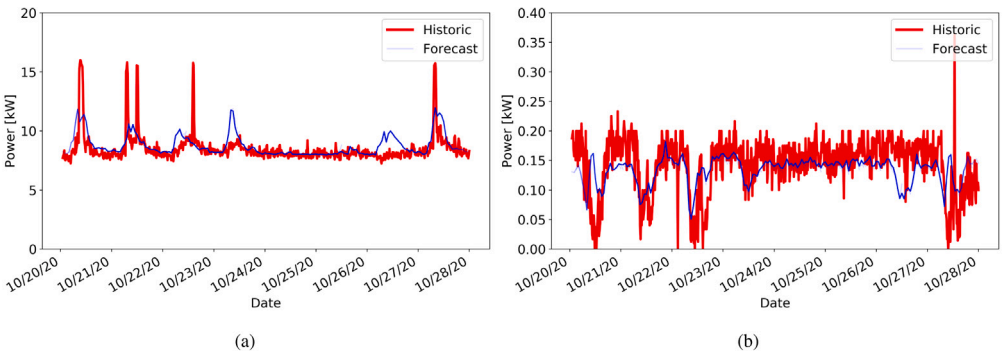


Fig. A.20. Measured historic (red) and forecasted (blue) values over a one-week period during the COVID19 pandemic for north (a) and south (b) plug load electrical panels.

for in future projects. Good communication between the MPC developer and building stakeholders, especially regarding the intent of the control, the security concerns, and operational risks, is key to facilitating MPC implementation. Workforce development and guidelines discussing the benefits, methods, and operational challenges of MPC can help prepare all stakeholders and facilitate the implementation process at scale.

The building data that has been collected before, during, and after the MPC demonstration are available open source in [39].

CRediT authorship contribution statement

David Blum: Conceptualization, Methodology, Software, Investigation, Formal Analysis, Data Curation, Visualization, Writing – original draft, Writing – review & editing, Project Administration. **Zhe Wang:** Conceptualization, Methodology, Software, Investigation, Formal analysis, Data Curation, Visualization, Writing – original draft, Writing – review & editing. **Chris Weyandt:** Methodology, Software, Investigation, Data curation, Writing – review & editing. **Donghun Kim:** Methodology, Writing – review & editing. **Michael Wetter:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Tianzhen Hong:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Mary Ann Piette:** Conceptualization, Writing –review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231 and the U.S.-China Clean Energy Research Center for Building Energy Efficiency (CERC-BEE).

The authors would like to thank Jeff Broughton, Norm Bourassa, Mark Friedrich, John Elliot, Jeff Grounds, Tom Davis, Melissa Abdelbaky, and Raphael Vitti from Lawrence Berkeley National Laboratory, USA very much for their support of this study, providing access to facilities and data, and help in the overall implementation and testing of the MPC controller in Building 59. Authors thank Erika Gupta of Building Technologies Office of United States Department of Energy for her support on the project.

Appendix. Additional data

The parameter estimation services for the thermal envelope and fan models ran throughout the entire study period. Figs. A.17 and A.18 show parameter estimation RMSE for training and validation for the envelope and fan models respectively. Notice that the envelope model validation RMSE stayed at or below 0.5 °C for most days, with some excursions to between 0.5 and 1.0 °C. On two occasions, the parameter estimation failed to find suitable parameters, with RMSE indicated in the Figure to be equal to 3 °C for visual clarity, and the control optimization service was required to use the previous day's results for the associated RTU. Note that the MPC services were shutdown from 11/8/2020 to 11/10/2020 due to a power shutdown in the building hosting the Building 90 server. Additional data is shown in Figs. A.19 and A.20 for the performance of weather and internal load forecasting algorithms during the performance evaluation period.

References

[1] Drgoňa J, Arroyo J, Cupeiro Figueroa I, Blum D, Arendt K, Kim D, Ollé EP, Oravec J, Wetter M, Vrabie DL, Helsen L. All you need to know about model predictive control for buildings. Annu Rev Control 2020;50:190–232. http://dx.doi.org/10.1016/j.arcontrol.2020.09.001, URL: https://www.sciencedirect.com/science/article/pii/S1367578820300584.

[2] Blum D, Wetter M. MPCPy: An open-source software platform for model predictive control in buildings. In: Proceedings of the 15th IBPSA conference. 2017, p. 1381–90.

[3] Mattsson SE, Elmqvist H. Modelica-An international effort to design the next generation modeling language. IFAC Proc Vol 1997;30(4):151–5.

[4] Wetter M, Benne K, Gautier A, Nouidui TS, Ramle A, Roth A, Tummescheit H, Mentzer S, Winther C. Lifting the garage door on Spawn, an open-source BEM-controls engine. In: Proceedings of the 2020 building performance modeling conference and simbuild co-organized by ASHRAE and IBPSA-USA. 2020.

[5] Coninck RD, Magnusson F, Åkesson J, Helsen L. Toolbox for development and validation of grey-box building models for forecasting and control. J Build Perform Simul 2016;9(3):288–303. http://dx.doi.org/10.1080/19401493.2015.1046933, arXiv:https://doi.org/10.1080/19401493.2015.1046933.

[6] Jorissen F, Boydens W, Helsen L. TACO, an automated toolchain for model predictive control of building systems: implementation and verification. J Build Perform Simul 2019;12(2):180–92. http://dx.doi.org/10.1080/19401493.2018.1498537.

[7] Henze GP. Model predictive control for buildings: a quantum leap? J Build Perform Simul 2013;6(3):157–8. http://dx.doi.org/10.1080/19401493.2013.778519.

[8] Sturzenegger D, Gyalistras D, Morari M, Smith RS. Model predictive climate control of a Swiss office building: Implementation, results, and cost–benefit analysis. IEEE Trans Control Syst Technol 2015;24(1):1–12.

[9] Henze GP, Kalz DE, Liu S, Felsmann C. Experimental analysis of model-based predictive optimal control for active and passive building thermal storage inventory. HVAC&R Res 2005;11(2):189–213. http://dx.doi.org/10.1080/10789669.2005.10391134.
[10] Gayeski NT, Armstrong PR, Norford LK. Predictive pre-cooling of thermo-active building systems with low-lift chillers. HVAC&R Res 2012;18(5):858–73. http://dx.doi.org/10.1080/10789669.2012.643752, URL: https://www.tandfonline.com/doi/full/10.1080/10789669.2012.643752.
[11] Yang S, Wan MP, Ng BF, Dubey S, Henze GP, Rai SK, Baskaran K. Experimental study of a model predictive control system for active chilled beam (ACB) air-conditioning system. Energy Build 2019;203:109451.
[12] Široký J, Oldewurtel F, Cigler J, Prívara S. Experimental analysis of model predictive control for an energy efficient building heating system. Appl Energy 2011;88(9):3079–87.
[13] Bengea SC, Kelman AD, Borrelli F, Taylor R, Narayanan S. Implementation of model predictive control for an HVAC system in a mid-size commercial building. HVAC&R Res 2014;20(1):121–35.
[14] West SR, Ward JK, Wall J. Trial results from a model predictive control and optimisation system for commercial building HVAC. Energy Build 2014;72:271–9.
[15] Kim D, Braun J, Cai J, Fugate D. Development and experimental demonstration of a plug-and-play multiple RTU coordination control algorithm for small/medium commercial buildings. Energy Build 2015;107:279–93.
[16] Li P, Vrabie D, Li D, Bengea SC, Mijanovic S, O'Neill ZD. Simulation and experimental demonstration of model predictive control in a building HVAC system. Sci Technol Built Environ 2015;21(6):721–32.
[17] De Coninck R, Helsen L. Practical implementation and evaluation of model predictive control for an office building in Brussels. Energy Build 2016;111:290–8.
[18] Miezis M, Jaunzems D, Stancioff N. Predictive control of a building heating system. Energy Procedia 2017;113:501–8.
[19] Hilliard T, Swan L, Qin Z. Experimental implementation of whole building MPC with zone based thermal comfort adjustments. Build Environ 2017;125:326–38.
[20] Granderson J, Lin G, Singla R, Fernandes S, Touzani S. Field evaluation of performance of HVAC optimization system in commercial buildings. Energy Build 2018;173:577–86.
[21] Zhuang J, Chen Y, Chen X. A new simplified modeling method for model predictive control in a medium-sized commercial building: A case study. Build Environ 2018;127:1–12.
[22] Kim D, Braun JE. Development, implementation and performance of a model predictive controller for packaged air conditioners in small and medium-sized commercial building applications. Energy Build 2018;178:49–60.
[23] Drgoňa J, Picard D, Helsen L. Cloud-based implementation of white-box model predictive control for a GEOTABS office building: A field test demonstration. J Process Control 2020;88:63–77.
[24] Freund S, Schmitz G. Implementation of model predictive control in a large-sized, low-energy office building. Build Environ 2021;197:107830.
[25] Wetter M, Wright J. A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. Build Environ 2004;39(8):989–99. http://dx.doi.org/10.1016/j.buildenv.2004.01.022.
[26] Jorissen F, Picard D, Six K, Helsen L. Detailed white-box non-linear model predictive control for scalable building HVAC control. In: Proceedings of 14th modelica conference 2021, Linköping, Sweden, september 20-24. 2021, p. 315–23. http://dx.doi.org/10.3384/ecp21181315.
[27] InfluxDB. 2021, URL: https://www.influxdata.com/time-series-platform/.
[28] Docker. 2021, URL: https://www.docker.com.
[29] Åkesson J, Årzén KE, Gäfvert M, Bergdahl T, Tummescheit H. Modeling and optimization with Optimica and JModelica.org-languages and tools for solving large-scale dynamic optimization problems. Comput Chem Eng 2010;34:1737–49. http://dx.doi.org/10.1016/j.compchemeng.2009.11.011.
[30] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 10 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. Nature Methods 2020;17:261–72. http://dx.doi.org/10.1038/s41592-019-0686-2.
[31] Weather underground. 2021, URL: https://www.wunderground.com/.
[32] DarkSky. 2021, URL: https://darksky.net.
[33] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.
[34] Holmgren WF, Hansen CW, Mikofski MA. Pvlib python: a python package for modeling solar energy systems. J Open Source Softw 2018;3(29):884. http://dx.doi.org/10.21105/joss.00884.
[35] Zhang Q, Huang J. Development of typical year weather data for Chinese locations. ASHRAE Trans 2002;108.
[36] Wang Z, Hong T, Piette MA. Data fusion in predicting internal heat gains for office buildings through a deep learning approach. Appl Energy 2019;240:386–98. http://dx.doi.org/10.1016/j.apenergy.2019.02.066, URL: https://www.sciencedirect.com/science/article/pii/S0306261919303630.
[37] Blum D, Arendt K, Rivalin L, Piette M, Wetter M, Veje C. Practical factors of envelope model setup and their effects on the performance of model predictive control for building heating, ventilating, and air conditioning systems. Appl Energy 2019;236:410–25. http://dx.doi.org/10.1016/j.apenergy.2018.11.093, URL: https://www.sciencedirect.com/science/article/pii/S0306261918318099.
[38] Li H, Szum C, Lisauskas S, Bekhit A, Nesler C, Snyder SC. Targeting building energy efficiency opportunities - an open-source analytical and benchmarking tool. In: Proceedings of the 2019 ASHRAE winter conference. Atlanta, GA. 2019.
[39] Luo N, Wang Z, Blum D, Bourassa N, Weyandt C, Piette MA, Hong T. A three-year dataset supporting research on building energy management and occupancy analytics. Sci Data 2022;9. http://dx.doi.org/10.1038/s41597-022-01257-x.