Title

Implementation and test of an automated control hunting fault correction algorithm in a fault detection and diagnostics tool

Permalink

Authors

Lin, Guanjing
Pritoni, Marco
Chen, Yimin
et al.

Publication Date

2023-03-01

DOI

10.1016/j.enbuild.2023.112796

Copyright Information

Energy & Buildings 283 (2023) 112796

Contents lists available at ScienceDirect

Energy & Buildings

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

Implementation and test of an automated control hunting fault correction algorithm in a fault detection and diagnostics tool

Guanjing Lin *, Marco Pritoni, Yimin Chen, Raphael Vitti, Christopher Weyandt, Jessica Granderson

Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:
Received 23 October 2022
Revised 24 December 2022
Accepted 9 January 2023
Available online 16 January 2023

Keywords:
Fault correction
Fault detection and diagnostics
Control hunting
Field testing
Energy management and information system
Smart building

ABSTRACT

Control hunting due to improper proportional–integral–derivative (PID) parameters in the building automation system (BAS) is one of the most common faults identified in commercial buildings. It can cause suboptimal performance and early failure of heating, ventilation, and air conditioning (HVAC) equipment. Commercial fault detection and diagnostics (FDD) software represents one of the fastest growing market segments in smart building technologies in the United States. Implementation of PID retuning procedures as an auto-correction algorithm and integration into FDD software has the potential to mitigate control hunting across a heterogeneous portfolio of buildings with different BAS in a scalable way. This paper presents the development, implementation, and field testing of an automated control hunting fault correction algorithm based on lambda tuning open-loop rules. The algorithm was developed in a commercial FDD software and successfully tested among nine variable air volume boxes in an office building in the United States. The paper shows the feasibility of using FDD tools to automatically correct control hunting faults, discusses scalability considerations, and proposes a path forward for the HVAC industry and academia to further improve this technology.

1. Introduction

Heating, ventilation, and air conditioning (HVAC) systems account for approximately 50 % of the total energy consumed by a building [12]. Detecting and fixing faults in HVAC system operation can reduce energy use and improve occupant comfort. Control hunting is one of the most common faults in HVAC systems [9,23]. It is a fault where a controller output (e.g., fan speed control command, valve control command, damper control command) oscillates with a frequency higher than justified by the system conditions [8]. The oscillations can cause suboptimal performance in regulating the desired signals and even early failure of the equipment due to unnecessary component wear. In many cases, the hunting behavior is caused by improper proportional integral derivative (PID) parameters in control loops of the local building automation system (BAS) controllers. The hunting faults can be eliminated by the PID parameter retuning process, which is to find the best combination of the PID values so the desired controller output can be achieved and control performance can be ensured.

1.1. PID retuning to correct control hunting faults and their implementation

The PID retuning methods can be categorized into two types: manual methods and intelligent methods [5]. Classical manual methods include trial and error methods and several heuristic tuning methods that require one or more on-site visits from the facility staff. The trial-and-error methods adjust the PID parameters and observe the resulting control response until the control performance is satisfied. To facilitate the tuning process, heuristic retuning methods are also performed manually but employ closed-loop or open-loop rules to obtain approximate or qualitative results. Closed-loop tuning rules require adjusting (increasing and/or decreasing) specific PID parameters until the process variable oscillates indefinitely at a steady state with constant amplitude [16]. For example, the Ziegler–Nichols (ZN) closed-loop tuning method [2] includes three steps: (1) remove integral and derivative terms so only the proportional term is used in the controller; (2) create a small disturbance in the loop by changing the setpoint and then increase and/or decrease the proportional parameter until the oscillations of the process variable has constant amplitude; (3) insert the values of the proportional parameter and process variable's period of oscillation into the ZN closed-loop equations and determine the necessary settings for the controller.

* Corresponding author.
E-mail address: gjlin@lbl.gov (G. Lin).

https://doi.org/10.1016/j.enbuild.2023.112796
0378-7788/

Nomenclature

AHU	Air Handling Unit	PB	proportional band
ALC	Automated Logic Controls	PV	Process Variable
BAS	Building Automation System	SP	Setpoint
CO	Controller Output	g_m	Measured process gain
DAT	Discharge Air Temperature	t	Present time
e	Error between the setpoint and the measured process variable	T	Sample time and corresponds to the control loop cycle time in ALC
EMIS	Energy Management and Information Systems	T_d	Derivative time
FDD	Fault Detection and Diagnostics	T_i	Integral time
HVAC	Heating Ventilation and Air Conditioning	$Time_D$	Measured dead time of the process
K_i	Integral gain	VAV	Variable Air Volume
K_{iT}	Integral gain multiplied by sample time	ZN	Ziegler-Nichols
K_p	Proportional gain	T_m	Process time constant
PID	Proportional Integral Derivative	λ_C	An arbitrary constant applied to T_m

On the other end, open-loop tuning rules introduce a disturbance into the system, and the PID parameters are obtained from the process reaction curve. For example, lambda open-loop tuning rules [10] implement a change to the controller output, to iteratively learn the PID parameter from the response of the process variable [33]. The manual tuning methods are time consuming and expensive [19], especially for systems with a slow response. For example, when tuning the air pressure loop during the HVAC system commissioning stage, it may take control engineers one to three days to determine the proper PID parameters [4].

In addition to the classical methods, many intelligent methods (known as auto-tuning or self-tuning) have been developed and embedded into the HVAC controllers or added as a new feature with extra charges of BAS software so the tuning process can be achieved without operator intervention [31]. Essentially, the intelligent methods enable the PID controller to learn how the process responds to a disturbance or change in setpoint, and calculate appropriate PID settings. Several journal publications and technical reports discuss intelligent tuning techniques for PID loops of HVAC equipment. Zhou and Liu developed an on-line self-tuning algorithm of a PI controller for heating and cooling coil valves [36]. Seem developed a novel pattern recognition adaptive controller, which is suitable for first-order plus dead-time systems commonly found in HVAC applications [32]. Qu and Zaheeruddin developed an adaptive PI control strategy for single loop HVAC processes, using the recursive least squares method to update the model parameters while the system remains in a closed loop [28]. Jin et al. [17] employed a model reduction method and explicit PID tuning rules on the basis of a fractional order plus time delay model to enhance the PID auto-tuning process. Dey and Mudi proposed an improved auto-tuning scheme for ZN-tuned PID controllers [11]. Romero et al. developed a PI and PID auto-tuning procedure that can be applied for any linear model structure, including dead time and nonminimum phase systems [29]. Although some HVAC controllers include the intelligent tuning routines, these features are inconsistent across vendors, often proprietary and with opaque behavior. Replacing the existing controllers with new ones is also costly and requires full recommissioning of the associated system. Most controllers in the existing buildings are still regular controllers without intelligent tuning capability.

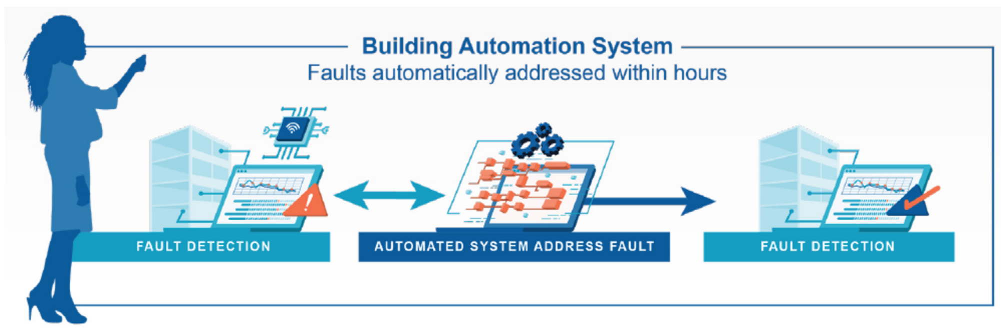
1.2. PID retuning to correct the control hunting fault with fault detection and diagnostics tools

Implementation of PID retuning procedures as automated correction algorithms and integrated into fault detection and diagnostics (FDD) software have the potential to mitigate control hunting across a heterogeneous portfolio of buildings with different BAS in a scalable way. An FDD tool is a type of smart building technology referred to as energy management and information systems (EMIS), which are used to improve building operation efficiency with building data and analytics [15]. As shown in Fig. 1(a), the FDD tool periodically collects time-series data (e.g., temperature, pressure, flow rate, control setpoint) from the BAS, identifies the presence of faults and efficiency improvement opportunities with rule-based and/or data-driven analytics, and presents the results to the user through a graphical interface. The building operators review the results, and then plan and execute a set of actions (program BAS or repair the equipment) to fix the identified issues. Commercial FDD software products represent one of the fastest growing and most competitive market segments in smart building technologies. There are dozens of full-featured FDD software product offerings for buildings now available in the United States, and new products continue to enter the market [20]. FDD applications in commercial buildings have grown significantly in the past 10 years to help building operators reduce energy use and improve operations and maintenance.

Automated fault correction after the detection of a fault has been demonstrated in research environments [18,6,14,15,7] and more recently tested in commercial products [21,25]. Automatically correcting faults (e.g., incorrectly programmed schedules, suboptimal control setpoints) by manipulating the BAS parameters can resolve a compelling set of operational problems that are detectable in today's FDD tools. Fig. 1(b) shows the enhanced workflow with automated fault correction. After the FDD algorithm identifies a specific fault, the appropriate correction algorithm is initiated to determine the corrective action. With user review and approval or auto-approval, the correction commands are sent to the BAS to change the values of specific control variables. Realizing automated fault correction in commercial FDD tools can fix the faults as soon as they are identified, increase the savings realized through the use of FDD tools, and ensure corrections to common faults are done consistently [26].

Control hunting faults also can be addressed using this automated correction process. After control hunting is identified, the FDD tool triggers a PID retuning procedure, which identifies the proper values of PID parameters and writes back to the PID controller via BACnet or other protocols. Fernandez et al. developed a passive diagnostic algorithm to detect hunting damper faults, and to correct those faults by automatically tuning the PID parameters [13]. The tuning algorithm employs historical damper signal values collected from damper controllers. When the hunting fault is diagnosed, the algorithm automatically decreases the proportional gain by a 10% value at each time step until the damper hunting symptom is eliminated. Since it employs a trial-and-error

(a)



(b)

Fig. 1. Traditional FDD workflow (a) and enhanced workflow with fault auto-correction (b).

procedure, the tuning is quite slow for control loops that start with a high proportional gain. Further, the algorithm was not tested in a laboratory experiment or real building. Previous work by the authors demonstrated a proof-of-concept implementation of PID tuning using an FDD tool [25]. The algorithm was integrated into a commercial FDD product; however, such implementation was not fully automated (i.e., it required facility managers to write new parameters into the BAS), was tested on a single variable air volume (VAV) box, and lacked important details to allow other researchers to reproduce it.

1.3. Gaps and contribution

PID retuning is an effective approach to eliminate control hunting. The classic tuning methods require significant human intervention, resulting in high costs and low scalability. In contrast, the intelligent methods, are often embedded into various BAS vendors' controllers or proprietary software. These commercial features are inconsistent across vendors and not transparent. Further, upgrading controllers that do not have these features typically requires large capital investments. The majority of the controllers or BAS in existing buildings don't have intelligent tuning features and many of them are out of tune. Finding an effective and scalable way to solve control hunting problems in this segment is what we are targeting in our study. FDD tools are one of the fastest growing smart building technologies being implemented in commercial buildings, providing a compelling solution to address building control problems. The new auto-correction capabilities, demonstrated in recent work [21,25], utilize the two-way communications (read and write capability) with the BAS and allows it to correct control hunting faults with PID retuning through FDD software. Using FDD tools to correct control hunting faults combines a method to detect the fault, a procedure to recalculate tuning parameters, and a process to push back the new parameters to the BAS. It provides a cost-effective way to solve hunting problems across a great number of controllers from various BAS vendors in many buildings. While tuning methods have been extensively covered in the literature, with the exception of a recent publication [25] the PID tuning algorithms were never integrated with commercial FDD products. While [25] represents important progress, the implementation was not fully automated, the implementation details were simply described, and the test was limited to a single piece of equipment.

To bridge the research gaps, this paper details the development, implementation, and field testing of a control hunting fault auto-correction algorithm in an FDD tool based on lambda open-loop tuning rules. The algorithm was implemented as separate software modules in a commercial FDD software.[1] It can automatically correct the control hunting faults of zone-level reheat valves – discharge air temperature (DAT) control loops in VAV boxes by calculating improved PID parameters and pushing these improved parameters into the BAS. We selected zone-level reheat valves as the study focus for two reasons: (1) facility staff of the test facility indicated that it was a priority undertaking, as there can be hundreds of zones in a building and hunting reheat valves account for a large number of faults; and (2) the process response time of the reheat valve – DAT control loop is slow enough to accommodate a one minute data collection interval in the BAS and FDD tools, unlike the fan speed – air handling unit (AHU) return air flow rate control loop that has the response time of less than one minute. The algorithm in the FDD tool was tested on nine VAV boxes in an office building in Berkeley, California, United States. This paper makes three main contributions to the existing body of knowledge:

- It successfully develops and implements a fully automated control hunting correction algorithm in a commercial FDD tool. It also provides details about its implementation to allow other researchers and practitioners to reproduce it.

[1] SkySpark® by SkyFoundry

- It successfully field-tests the software in nine VAV boxes in a real building.
- It discusses scalability issues and proposes a path forward for industry and academia to improve this technology.

The rest of the paper is organized as follows: Section 2 describes the classic lambda open-loop tuning approach and the modifications for this study. Section 3 presents the results of implementing the control hunting fault auto-correction algorithm in an FDD tool based on lambda open-loop tuning rules. Section 4 presents the field test results with the enhanced FDD tool in nine VAV boxes of an office building. Section 5 discusses the path forward. Section 6 summarizes the conclusions and future work.

2. Classic lambda open-loop tuning and modifications

In the literature, we find both closed-loop and open-loop rules to identify improved PID parameters that prevent control hunting behavior. It is challenging to implement closed-loop rules because it is difficult to determine the number of adjustments needed before the intended behavior is achieved. Further, the tuning process is highly dependent on different control loops in different buildings. In contrast, the open-loop tuning rules are easy to implement, as they only need to perform designed step-change tests to obtain a process reaction curve. Lambda open loop tuning [10] has been successfully applied to thousands of process control applications [27,22]. Many HVAC controls like zone-level reheat valves do not need to act fast and can benefit from a slow, stable response. Therefore, the lambda open-loop tuning rules were selected as the core of the proposed correction algorithm because they are designed for a slower response, of the same order as the process response time.

Section 2.1 introduces the classic lambda open-loop tuning rules and equations that determine the improved PID parameters for a standard PID equation (Eq. (1)). Section 2.2 presents the modified lambda tuning equations that were tailored to the specific implementation of the PID equation in the equipment at the test site. The developed fault auto-correction algorithm (see Section 3) was built based on the classic lambda open-loop tuning rules and modified tuning equations.

2.1. Lambda open-loop tuning rules and equations

PID controllers (Fig. 2) are widely used in HVAC systems to control variables in the built environment [32]. A PID controller compares a process variable (PV, i.e., the property that is to be controlled, typically measured by a sensor, such as temperature or pressure) to its setpoint (SP) to calculate error $e(t)$ between the SP and the value of PV. The controller output (CO) is determined based on a PID equation. This output changes the behavior of an actuator, to generate actual control actions so the PV tends toward the SP.

Eq. (1) shows a standard PID equation, which can be expanded into a proportional (P) term, an integral (I) term, and a derivative (D) term. As the names imply, the P term is proportional to error $e(t)$, the I term is proportional to the integral of the error over time, and the D term is proportional to the derivative of the error.

$$CO(t) = K_p\left[e(t) + \frac{1}{T_i}\cdot\int e(t)dt + T_d\cdot\frac{de}{dt}(t)\right] \quad (1)$$

where $CO(t)$ is the controller output, K_p is the proportional gain, T_i is the integral time, T_d is the derivative time, $e(t) = SP(t) - PV(t)$ which is the error between the SP and the value of PV, and t is the present time.

In the HVAC control area, PI controllers are most commonly used [35] because the relatively slow HVAC processes do not need the derivative component, which is often used to mitigate overshoots when implementing a faster control response.

Lambda open-loop tuning conducts a step change open-loop test as illustrated in Fig. 3 and includes the three steps below [33]. The intent of open-loop tests is to observe the reaction of the process variable (e.g., temperature) to a change in the controller output (e.g., valve control command) without interaction from the PID controller.

1. Stabilize the process by overriding the CO to a value CO_0 within its normal operating range. Allow the PV to settle into a steady state PV_0.
2. Noting the time, override the CO to another value, CO_1, within its normal operating range. Document the difference between the new value and the prior value, which is the step change ($change\ in\ CO = CO_1\text{-}CO_0$).
3. Allow the PV to settle at a new steady state value, PV_1. The step change should be large enough that the magnitude of change to the PV is significantly greater than the signal noise and then changes caused by outside disturbances to the process.

If a controller uses a standard PID equation (Eq. (1)), lambda open-loop tuning equations give a proposed controller gain and integral time (K_p and T_i) based on measurements taken on the PV and CO after the step change open-loop test is completed:

$$K_p = \frac{T_m}{g_m\cdot(\lambda_C\cdot T_m + Time_D)} \quad (2)$$

$$T_i = T_m \quad (3)$$

where T_m is the process time constant, which is the time it would take the process to go from PV_0 to PV_1 at the maximum observed rate of change.

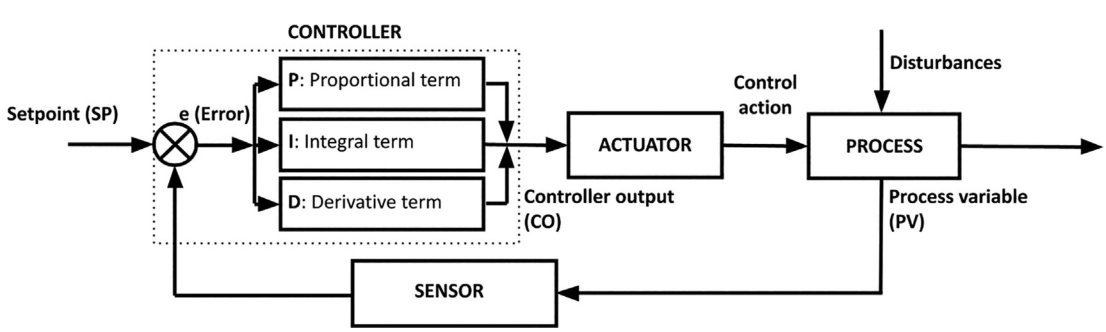


Fig. 2. Classic block diagram of a single PID control loop.

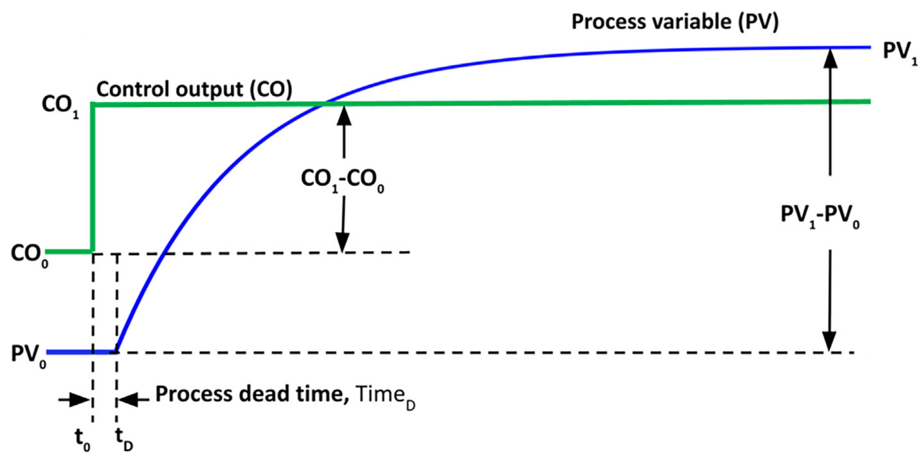


Fig. 3. Step change open-loop test performed for lambda tuning.

g_m is the measured process gain,

$$g_m = \frac{PV_1 - PV_0}{CO_1 - CO_0} \tag{4}$$

λ_C is an arbitrary constant applied to T_m, typically between 1 (faster tuning) and 3 (slower tuning). A value of 2 was used in this study's tests.

$Time_D$ is the measured dead time of the process, measured as the difference between the time t_0 when CO changes from CO_0 to CO_1 and the time t_D when the PV starts to deviate from its stable-state value PV_0.

If a controller does not use a standard PID equation, changes must be made to eqs. (2) and (3) accordingly. Section 2.2 presents the modified tuning equations for the controllers of the testing site using a parallel PID equation in which the parameters of the P, I, D terms are independent from each other.

2.2. Modified lambda tuning equations specific to the test site

The classic lambda tuning equations (eqs. (2) and (3)) were adjusted to the specific equipment on the test site in this study. The targeted reheat valve – DAT control loops are PI control loops in the Automated Logic Controls (ALC)[2] controllers of VAV boxes. The VAV box DAT is the PV. The reheat valve command is the CO. As implemented at the test site, the controllers use the discrete form (Eq. (6)) of a parallel PI equation (Eq. (5)). The CO consists of a proportional term and an integral term.

$$CO(t) = CO_p(t) + CO_i(t) = K_p \cdot e(t) + K_i \cdot \int e(t)dt \tag{5}$$

$$\begin{aligned} CO(t) &= K_p \cdot e(t) + CO_i(t-1) + K_i \cdot T \cdot e(t) \\ &= K_p \cdot e(t) + CO_i(t-1) + K_{iT} \cdot e(t) \end{aligned} \tag{6}$$

where $CO(t)$ is the control output in unit %, $CO_p(t)$ is the proportional term, $CO_i(t)$ is the integral term, K_p is the proportional gain, K_i is the integral gain, and t is the present time. T is the sample time and corresponds to the control loop cycle time in ALC. T is set to one minute in the target controllers at the testing site, which means $CO(t)$ is updated once per minute. The targeted controllers use reverse-acting controls, which match the standard error calculation used with Eq. (1), $e(t) = SP(t) - PV(t)$. $e(t)$ is in the unit of the PV, therefore K_p is in $\frac{\%}{unit_{pv}}$. $K_{iT} = K_i \cdot T$, which is also in $\frac{\%}{unit_{pv}}$.

As shown in Eq. (6), the parameters of the control blocks that can be overwritten are K_p and K_{iT}.

The lambda open-loop tuning equations (Eqs. (2) and (3)) presented in Section 2.1 are written for the continuous version of the standard PID equation (Eq. (1)) and determine the value of K_p and T_i. Therefore, they must be modified to yield the correct parameters K_p and K_{iT} for the actual, discrete PI control loop equation used in the testing site controllers (Eq. (6)). The process to derive the value of K_{iT} based on the values of K_p and T_i is illustrated below through Eq. (7) and Eq. (8).

Separating and equating the integral terms in Eq. (1) and Eq. (5) gives:

$$K_i = \frac{K_p}{T_i} \tag{7}$$

From eqs. (2), (3), (6), and (7):

$$K_{iT} = K_i \cdot T = \frac{K_p \cdot T}{T_i} = \frac{K_p \cdot T}{T_m} = \frac{T}{g_m \cdot (\lambda_C \cdot T_m + Time_D)} \tag{8}$$

The lambda open-loop tuning equations to obtain ALC parameters for the testing equipment are therefore eqs. 2 and 9, with g_m measured process gain described in Eq. (4) in unit of $\frac{unit_{pv}}{\%}$ and all times (T_m, T, and $Time_D$) measured in the same time unit.

$K_p = \frac{T_m}{g_m \cdot (\lambda_C \cdot T_m + Time_D)}$(Eq. (2)), $K_{iT} = \frac{T}{g_m \cdot (\lambda_C \cdot T_m + Time_D)}$ (Eq. (9)), with $g_m = \frac{PV_1 - PV_0}{CO_1 - CO_0}$ (Eq. (4))

3. Implementation of an automated control hunting correction algorithm in an FDD tool

This section presents the implementation details of a control hunting fault auto-correction algorithm in an FDD tool based on the modified lambda open-loop tuning method presented in Section 2. The auto-correction algorithm was developed as four separate software modules in the FDD tool: one fault detection algorithm (3.1) and three separate modules for correction (3.2), (3.3), and (3.4) (Fig. 4):

- Detection of hunting faults (3.1)
- Management of active tests (3.2)
- Calculation of improved PID parameters (3.3)
- Database and interface to access test results (3.4)

After the hunting fault is detected by module 3.1, the auto-correction is initiated by the facility operator. As shown in Fig. 4, modules 3.2, 3.3, and 3.4 are executed to obtain improved PID

[2] Automated Logic. https://www.automatedlogic.com/en/

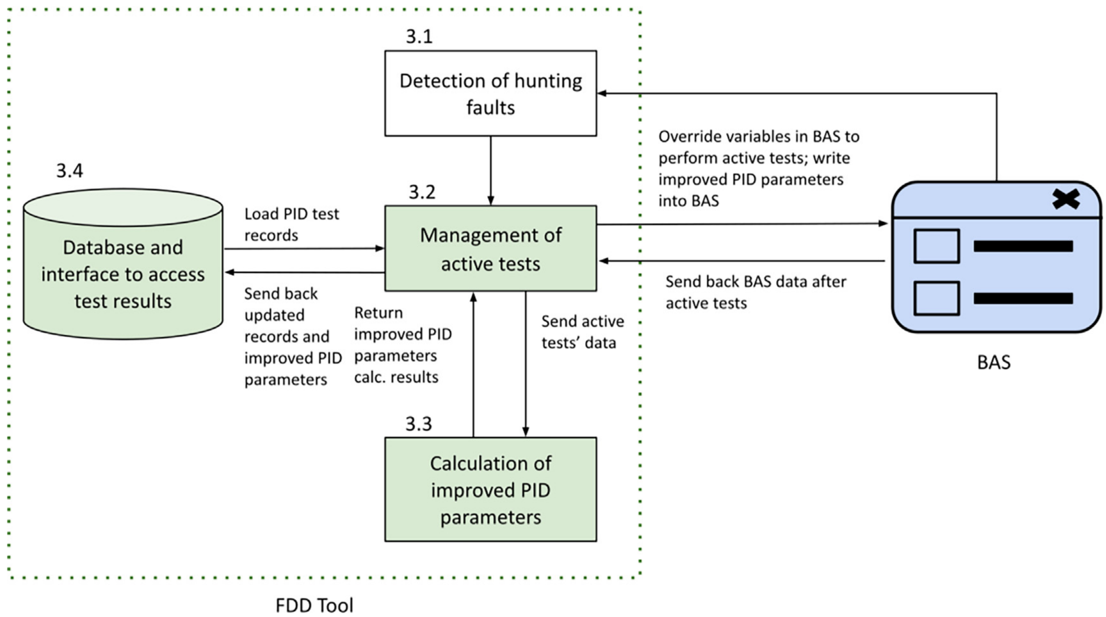


Fig. 4. Software modules created (in green) or updated (in white) in the FDD tool to implement the control hunting auto-correction algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

parameters based on the results of active step-change tests, then the improved PID parameters are written back into the BAS. The source codes of four modules are available at https://github.com/LBNL-ETA/haxall-based-fault-correction.

3.1. Detection of hunting faults

A fault detection algorithm is run continuously in the background to identify what variables are hunting and on which equipment. The hunting fault detection algorithm, which is designed to detect severe cycling behaviors with minimal false positives, is applied to specific COs with the tag "cmd," and units in %. For a given CO, up to one day of trend data is analyzed at once.

- The rate of change is calculated as follows:
  - o At t_i, the rate of change is $[CO(t_i) - CO(t_{i-1})]/(t_i - t_{i-1})$.
  - o $CO(t_{i-1})$ is the previous valid measurement at t_{i-1} for which $CO(t_{i-1})$ is a number (not N/A).
- A flag is raised if the calculated rate change exceeds a user defined threshold (customized threshold depending on equipment and control loops, e.g., 5%/min is used for the reheat valve – DAT control loops in this study[3]).
- Two flags within a user defined time threshold (customized threshold depending on equipment and control loops, e.g. 30 min in this study[4]) of each other are merged into a flagged period.
- If a flagged period exceeds or is equal to the time threshold defined above (30 min), a hunting fault is reported.

[3] The threshold of 5%/min, used to detect hunting faults for reheat valve – DAT control loops in this study, was determined experimentally, with the goals of minimizing false positives and obtaining a manageable number of faults. Although this threshold applies to small and large oscillations alike, the starting point was determined based on allowing one full cycle of the reheat valve command (0% – 100% – 0%) in 1 h, which is 200% / 60min = 3.3%/min, and was then relaxed to reduce the number of detected faults. As faults are addressed, the user can lower the threshold to capture less egregious faults.

[4] The 30-minute threshold is a parameter that is tuned so as not to flag high rates of change that can occur over a brief period of time, for example when the discharge air temperature setpoint changes.

3.2. Management of active tests

When the FDD user initiates the auto-correction of a detected hunting fault for a given control loop, software module 3.2 starts performing active step-change tests of the control loop, as needed to apply the lambda open-loop tuning rules (Section 2.1). Step changes are performed by keeping the CO stable for a few minutes, then changing its value suddenly and maintaining it at the new value until the PV stabilizes. During the test, the control loop is therefore bypassed and the system response is observed in an open-loop control mode. To ensure the reliability of test results, multiple successful tests may be required. The new PID parameters are determined from results of multiple tests and written back to the BAS. Module 3.2 monitors the status of the tests and writes new PID parameters to the BAS upon reaching the target number of successful tests for a given control loop.

A persistent task runs in this module every hour to review all the PID loop records that have auto-testing enabled. If a PID loop requires a test (number of successful tests < target number of tests) and a test was not recently completed (in the last 24 h), it queues up a new test by creating a PID test record in the database (module 3.4) and creates a new temporary dedicated task.

This temporary dedicated task runs every-five minutes to perform the following steps sequentially:

- Load the corresponding PID test record, which contains testing status information and testing parameters (see module 3.4 "Database and interface to access test results").
- Synchronize time-series data to have the most recent data available for the PV and the CO.
- Check if testing conditions are met (e.g., reheat valve testing requires supply airflow through the terminal unit).
- If testing conditions are not met, the current test is put on hold or fails.
- If testing conditions are met, initialize an active step-change test described in Section 2.1 by reviewing recent CO data and determine a target CO_0 and a future target CO_1.
  - o Override CO to CO_0 using a temporary override (start of test). Review recent data to confirm that the override is successful. If the override fails, the test stops and is considered failed.

- o Review recent PV data. If the PV is stable at PV_0 or a maximum time duration has elapsed (defined based on control loop type; e.g., 20 min in the tests presented in this paper), initiate a step change by overriding the CO to CO_1 using a temporary override.
- o Review recent PV and CO data to confirm that the override of the CV to CO_1 is successful. If it succeeds, send the data collected so far to module 3.3 "Calculation of improved PID parameters."
- o If module 3.3 returns improved PID parameters, stop the test, mark it as successful, and record the improved PID parameters. Release the override (end of test).
- o If module 3.3 fails and does not return improved PID parameters, continue the test until the maximum testing duration has been reached (e.g., one hour in the tests presented here), at which point the test fails.
- o Store information about the test and test results, if available, in the PID test record in the database (module 3.4.)

3.3. Calculation of improved PID parameters

This module is invoked during testing if enough data have been collected during the execution of module 3.2. In this module, the improved PID parameters are calculated from the collected data (CO and PV) with the specific lambda tuning equations for the targeted equipment (Eqs. 2, 4, and 9).

Eqs. 2 , 4, and 9 include three key parameters: the measured process gain g_m, the process time constant T_m, and dead time $Time_D$ whose values need to be determined from the data of CO and PV during the step-change test. Additionally, the control loop cycle time T is needed, which is a configuration parameter in the BAS. In this specific implementation of the algorithm at the testing site, the four parameters are obtained as follows and illustrated in Fig. 5 with artificial data:

- The cycle time T of the control loop in the test site is one minute, as stated in Section 2.2.
- The measured process gain g_m can be calculated with CO_0, CO_1, PV_0, and PV_1. The value of CO_0 and CO_1 are predetermined in module 3.2 when performing the active test. The derivative of PV $\left(\frac{\delta PV}{\delta t}\right)$ is used to determine the boundary of the PV's reaction (PV_0 and PV_1). PV_0 is the measured value of the PV when the $\frac{\delta PV}{\delta t}$ is last below 10% of the maximum measured rate of change $\left[0.1 \times \left(\frac{\delta PV}{\delta t}\right)_{max}\right]$ before the step change launches. The corresponding time is t_{m0}. PV_1 is the measured value of PV when the $\frac{\delta PV}{\delta t}$ first falls below $0.1 \times \left(\frac{\delta PV}{\delta t}\right)_{max}$ after the step change launches. The corresponding time is t_{m1}.
- The process time constant is calculated as $T_m = t_{m1} - t_{m0}$.
- The dead time $Time_D = t_D - t_0$ is the difference between the time t_0 when CO changes from CO_0 to CO_1 and the time t_D when the PV starts to deviate from its original value PV_0, as described in Section 2.1. In the calculation of this study, $t_D = t_{m0}$ and t_0 is the time when CO was last measured at CO_0. $Time_D$ has a minimum value of 0 s.

Conducting this exercise from the FDD tool means we are using data from the BAS, at BAS trending intervals (i.e., one minute), and the sampling interval creates some inaccuracies. Therefore, the PV data are rolled up (e.g., to two minutes) using an averaging function to ensure measurement accuracy, a constant x-axis interval, and to smooth out the signal. Because of the sampling interval and implemented roll-up, it is not uncommon to measure a dead time of zero.

3.4. Database and interface to access test results

This module stores and displays the results from module 3.2. Information about the PID control loops and about each active test is recorded in a database which is structured as a combination relational and time-series databases in the FDD tool, including start and end timestamps, whether the active test is successful or has failed, and, for successful tests, relevant measurements such as process gain.

PID loop records are created semi-manually through an interface that allows the user to select one PV and one CO from data streams that are already monitored by the FDD tool. Each **record of the PID loops** in the database contains:

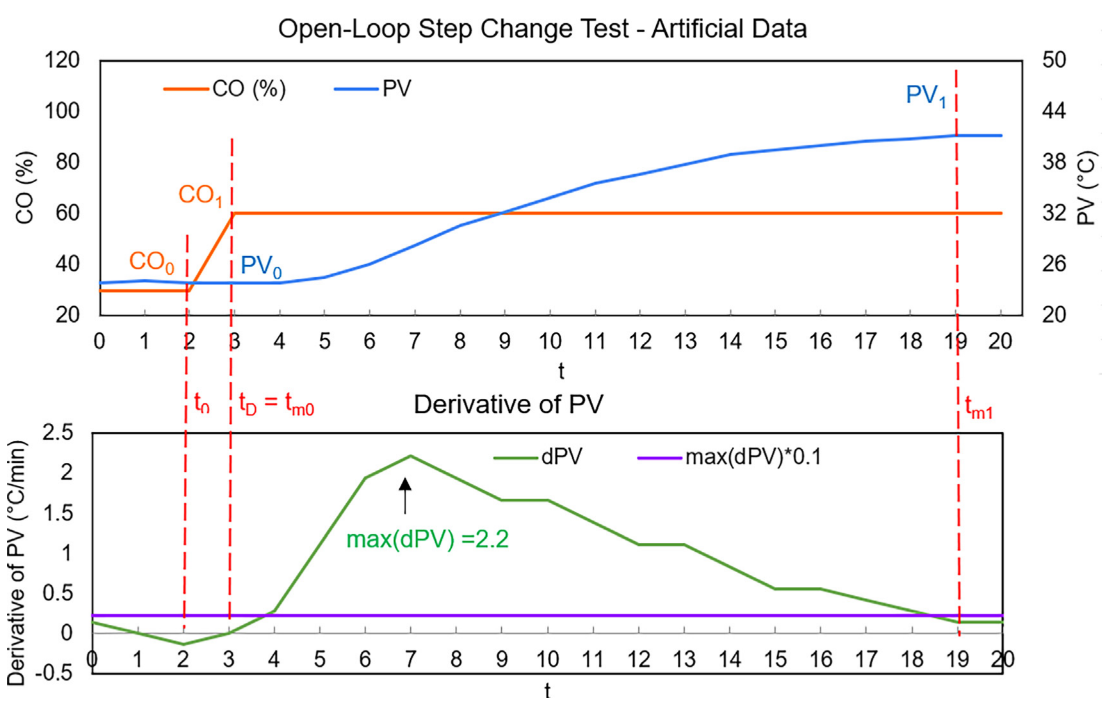


Fig. 5. Illustration of key parameters with artificial data.

- Reference to a specific PV (e.g., VAV-001 Discharge Air Temperature)
- Reference to a specific CO (e.g., VAV-001 Reheat Valve Command)
- Auto-testing boolean (toggle on/off)
- Auto-testing increments target (how many tests will be performed, e.g., two)
- Expiration period for previous tests (tests do not count if they are older, e.g., three months)
- Automated PI parameter update boolean (should module 3.2 push new PI parameters after the target number of successful tests is achieved)

The database also contains records for each test. These are created automatically by module 3.2 and contain test status information, testing parameters, and test results. Each record of the step change tests in the database contains:

- Reference to a PID loop record
- Maximum test duration on hold (no overrides, waiting for allowed conditions), e.g., one week
- Maximum test duration in active mode (override active), e.g., one hour
- Timestamp of record creation (initialization of test)
- Timestamp at which active open-loop testing started
- Timestamp at which a stable state was achieved prior to the step change
- Timestamp at which the test succeeded or failed
- Success or failure status
- Diagnostic message
- Measurements of the reaction of the process to the step change (g_m, T_m, and $Time_D$) if the test was successful.

Records also contain temporary information needed during testing, such as the target CO values before and after the step change. An interface is also created that allows users to view these results, as well as the combined results of multiple successful tests on the same PID loop.

4. Field testing

4.1. Test procedure

The FDD tool-based control hunting correction algorithm (Section 3) was tested on the reheat valve – DAT control loops of nine VAV boxes that require heating in the testing period in an office building in Berkeley, California, United States. The test procedure is as follows:

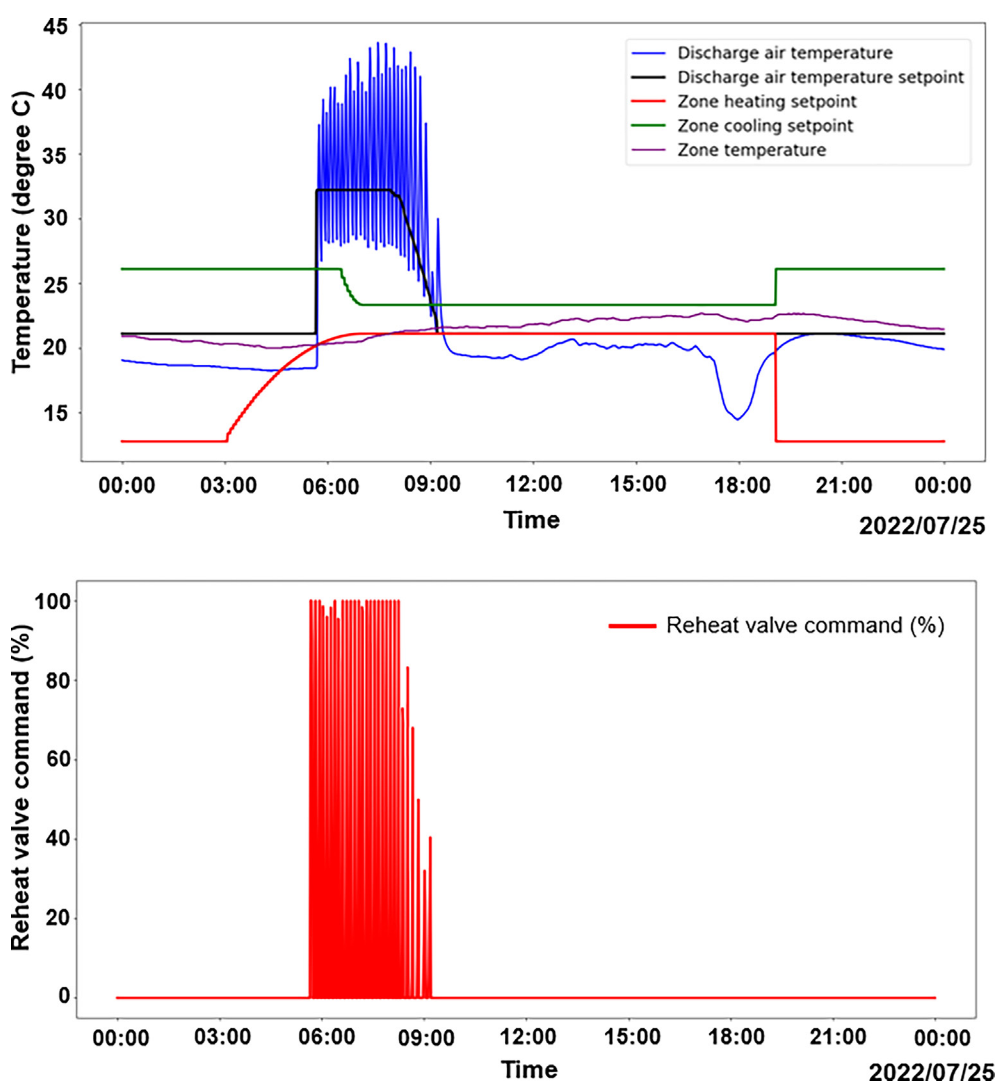


Fig. 6. Reheat valve command (CO) and discharge air temperature (PV) and setpoint of VAV-105, zone setpoints and temperature before fault correction (July 25).

- Impose the control hunting fault in “clean” (fault-free) equipment.
- Observe and document the FDD tool’s detection and diagnosis results.
- Execute the FDD-embedded correction routine described in Section 3. The routine determines the improved value of PID parameters K_p and K_{iT} and overwrites their values in the corresponding ALC controller.
- Observe and document the effect of the automated fault correction.

4.2. Test results of one VAV box

Multiple tests were conducted in Summer 2022 to observe the effect of control hunting auto-correction. The test results of one VAV box (VAV-105) are presented explicitly in this section, and the results of all VAV boxes are summarized in Section 4.3.

4.2.1. System behaviors before auto-correction

During the test period the facility management team purposefully introduced hunting faults in a series of VAV boxes, by chang-

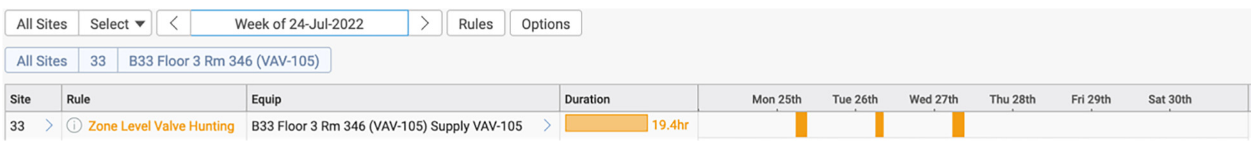


Fig. 7. Visualization of the faulty behavior using the FDD tool interface.

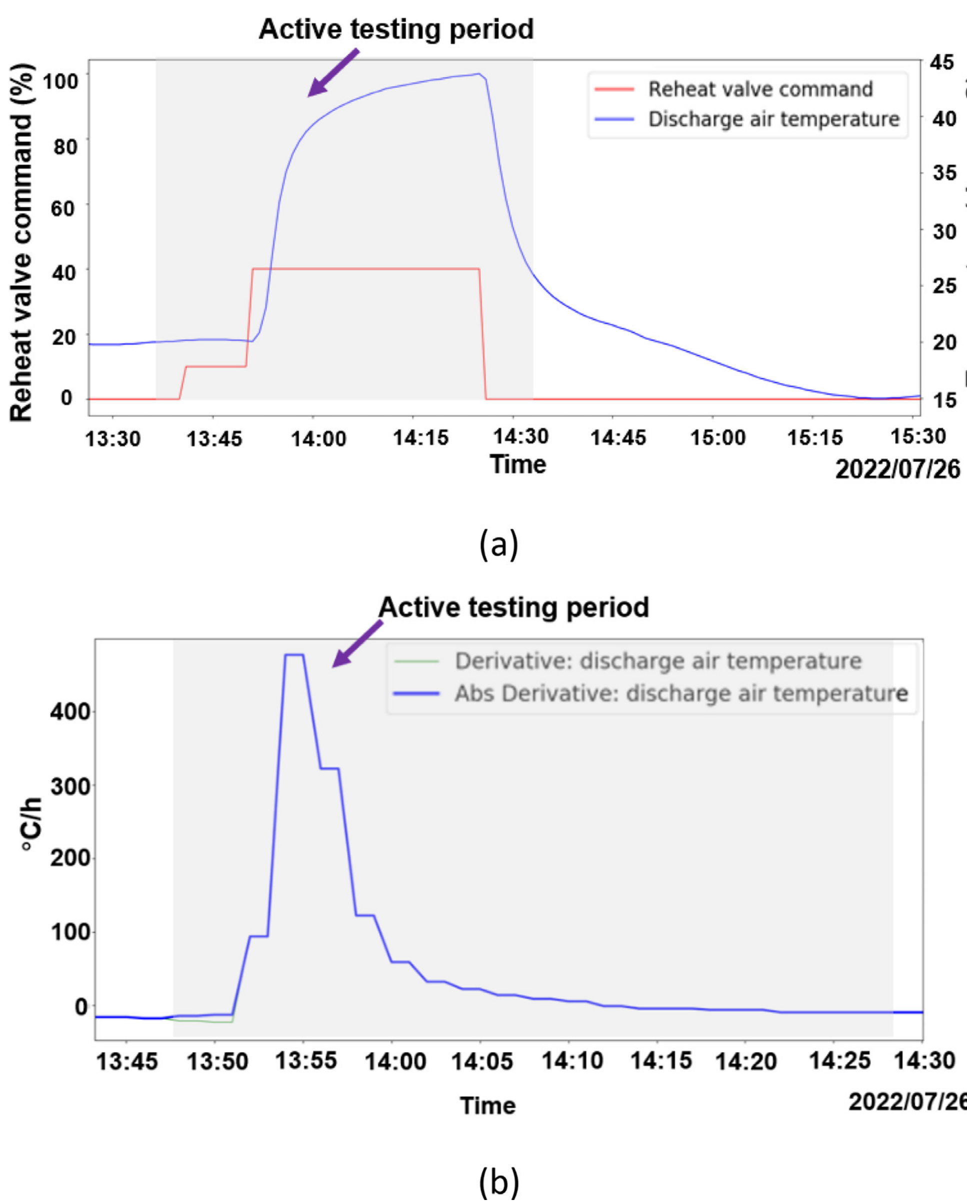


Fig. 8. Controller output (reheat valve command), process variable (discharge air temperature), and derivative of the PV for the active step test conducted on the VAV-105 on July 26.

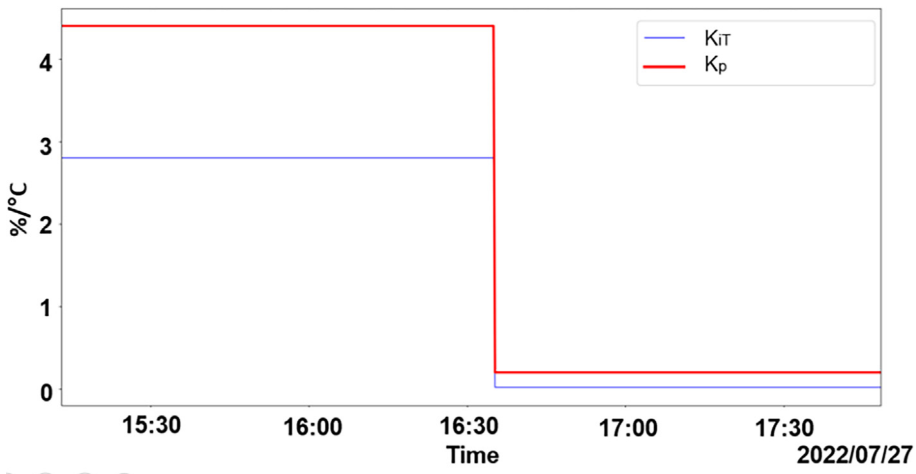


Fig. 9. Illustration of the successful written action of the PI parameters (K_p and K_{iT}) on the VAV-105 on July 27.

ing PID parameters, as prescribed by the procedure in Section 4.1. Fig. 6 illustrates the hunting behavior of the reheat valve command (i.e., the CO) and DAT (i.e., the PV) collected at a 1-min sampling interval of VAV-105 on July 25, 2022. At about 5:30 am the DAT setpoint increased from 21.1°C (70°F) to 32.2°C (90°F) (top of Fig. 6 in black) and the PID loop started modulating the reheat valve (bottom of Fig. 6 in red) to maintain the setpoint. Fast oscillations were observed (i.e., hunting) of both the reheat valve command and the DAT. The PI parameters when the fault was detected were:

$K_p = 4.4\%/^\circ C(8\%/^\circ F)$ and.$K_{iT} = 2.8\%/^\circ C(5\%/^\circ F)$

4.2.2. Fault detection results

The oscillations described above were detected by the FDD tool, as represented in Fig. 7. The faulty behavior was detected on July 25, corroborating the visual information in Fig. 6. The faulty behavior was also detected on July 26 and 27, when the auto-correction was in progress. After the completion of auto-correction on July 28, no more hunting faults were flagged.

4.2.3. Execution of the active tuning test

Following the detection of the fault on July 25, 2022, facility staff initiated the auto-correction procedure illustrated in Section 3. One successful active test was completed on July 26, and a second successful test was completed on July 27. Fig. 8 shows data from the example test executed on July 26, 2022. As shown in the shaded area in Fig. 8(a), the CO (reheat valve command) of the VAV was increased from 10% to 40%; as a result, the discharge air temperature increased from 20 °C (68°F) to 41.1 °C (106°F). In Fig. 8(b), the measured rate of change of PV reached the maximum 477 °C/hour (h) (891°F /h) at 13:54 pm.

Key parameters of the reaction of the process to the step change are:

- $Time_D = 0min$. Dead time is measured at 0 because 13:50 pm was the last timestamp with CO = 10%, and the next timestamp (13:52 pm after averaging roll-up) already shows a strong response of PV from the change of CO, as seen in its derivative of 112°C /h (201 °F /h) at the time, which corresponds to 23% of the maximum derivative, observed at 13:54 pm.
- $T_m = 14min$. Process time constant, measured between 13:50 pm and 14:04 pm.
- $g_m = \frac{PV_1 - PV_0}{CV_1 - CV_0} = \frac{41.1^\circ C - 20^\circ C}{40\% - 10\%} = 0.7^\circ C/\%(1.3^\circ F/\%)$

The arbitrary lambda tuning constant λ_C is set to 2. Therefore, the following ALC-specific PI parameters are obtained from Eq. (2) and Eq. (11), presented in Section 2.2:

$$K_p = \frac{Tm}{g_m \cdot (\lambda_C \cdot Tm + D)} = \frac{14\,min}{0.7^\circ C/\% \cdot (2 \times 14\,min + 0\,min)} = 0.2\%/^\circ C\,(0.4\,\%/^\circ F)$$

$$K_{iT} = \frac{T}{g_m \cdot (\lambda_C \cdot Tm + D)} = \frac{1\,min}{0.7^\circ C/\% \cdot (2 \times 14\,min + 0\,min)} = 0.02\%/^\circ C(0.03\,\%/^\circ F)$$

4.2.4. Writing of updated parameters to the BAS

Following the completion of two successful tests, on July 26 and 27, module 3.2 averaged the measurements from both tests, then calculated the ALC-specific PI parameters below from these averages.

$K_p = 0.2\%/^\circ C(0.4\%/^\circ F)$ and $K_{iT} = 0.02\%/^\circ C(0.03\%/^\circ F)$

Finally, module 3.2 wrote new parameters via BACnet to the BAS on July 27, as shown in Fig. 9.

4.2.5. Effects of the auto-correction routine

After the tests were run and the improved PID parameters were calculated, the new parameters shown in Fig. 9 were automatically sent to the BAS (at 16:35 pm on July 27, 2022). Fig. 10 shows the behavior of the reheat valve command and the DAT with the new PI parameters on July 28. The oscillations of the reheat valve controller command and supply air temperature disappeared after the update of the parameters, and the hunting fault also disappeared from the fault detection interface (Fig. 7).

4.3. Test results of all nine VAV boxes

The same procedure was successfully executed on other VAV boxes during the same period. Table 1 shows the PID parameters before and after the automated fault correction in all VAV boxes where the hunting faults were successfully imposed. The imposed hunting fault clearly disappeared after an update of the PID parameters in the other eight VAV boxes (Fig. 11).

5. Discussion

This paper shows the development and integration of a fully automated control hunting auto-correction algorithm in a com-

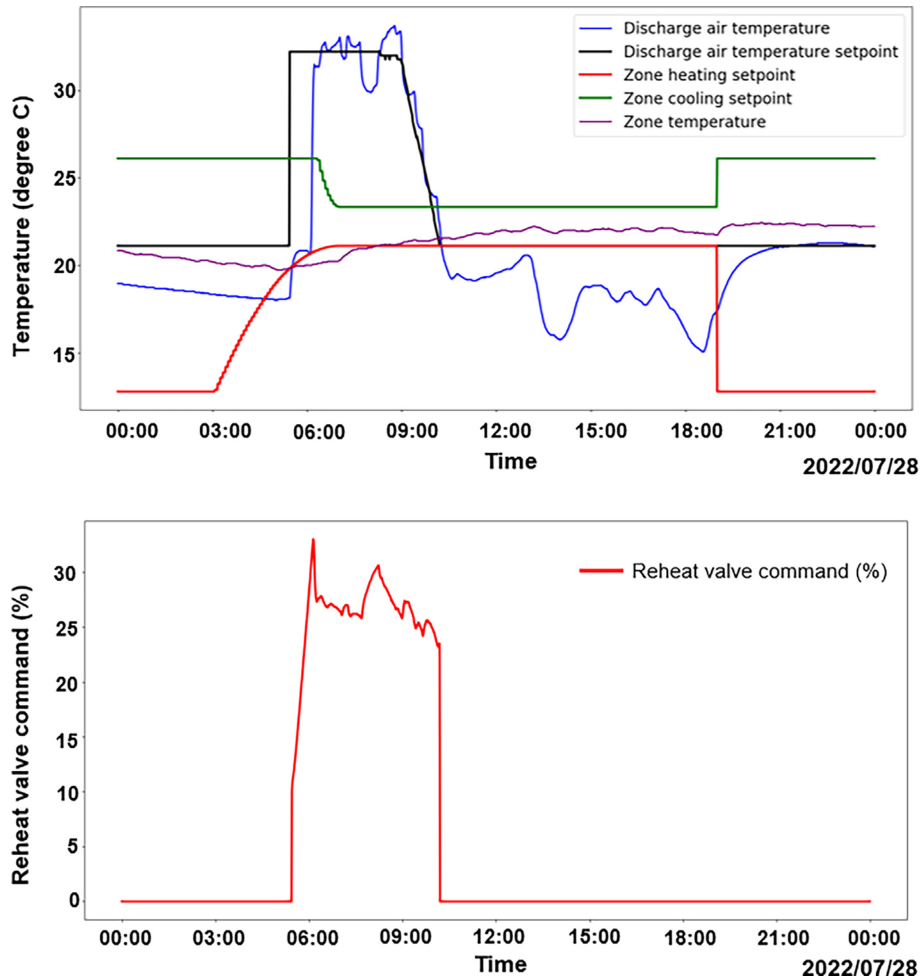


Fig. 10. Reheat valve command (CO) and discharge air temperature (PV) and setpoint of VAV-105, zone setpoints and temperature after fault correction (July 28).

mercial FDD tool and its test in nine VAV boxes in an occupied building. While the integration was successful and the results were promising, the study revealed several challenges in scaling up this approach to buildings with different FDD platforms, BAS software, and types of equipment. This section explores those challenges and proposes a path forward for academia and industry to support the large-scale deployment of this technology.

5.1. Applicability of the approach to other types of HVAC equipment and control loops

To select the candidate equipment for this field test, the research and facility team investigated several options, including cooling towers, fan coils, AHUs, and VAV boxes. Eventually, VAVs were selected, given their ubiquity and relatively simple control logic. We chose to tune the reheat valve – discharge air temperature loop because its hunting behavior destabilizes ancillary systems (i.e., duct pressure control, water pressure control) and it can lead to improper setpoint reset calculations (i.e., AHU supply air temperature setpoint) and cause excessive wear on components. In fact, in this building, the facility staff had to replace nearly a third of all actuators due to early failure from excessive oscillations. Given the successful tests presented above, a natural question is whether this approach is generalizable to other systems, configurations, and control loops.

The process response time (the time that the PV would take to reach the steady-state value after the change of the CO) of the targeted control loop was about 5 to 10 min. To enable the calculation of process characteristics parameters from the trend data, the data collection interval in the FDD tool and BAS were modified to one minute in this study. Data collection intervals in FDD tools depend on the "trend interval" setup in the BAS. Typical BAS trending intervals of 5 to 15 min do not offer enough granularity to capture the reaction of the process to the step changes, nor to perform the measurements discussed in Section 2 and illustrated in Section 3.3. In addition, other control loops may have faster response times. For example, the facility team reported that both the pump speed – chilled water differential pressure control loop and the fan speed – AHU return air flow rate control loop have a response time of less than one minute. These cases require fast data sampling (<1 min) for both detection and correction of control hunting faults. Modern BAS can sample data at a high frequency, but polling (i.e., retrieving) them at rates faster than one sample per minute may cause network traffic problems and/or data storage issues.[5] To overcome this issue, instead changing the configuration on the BAS to let some data collected at one minute as we did in this study, we suggest to

[5] Modern BAS can trend data every second or less, but to avoid network traffic issues, these data are typically temporarily stored in the local controllers' memory. After filling the local memory, the data needs to be stored more permanently on a database, which is typically hosted on a computer in the same network. Since memory is limited for each point trended, with higher sampling rate the memory fills more rapidly and synching with storage has to happen more frequently. The FDD tool can bypass the BAS database and gather data directly from trend objects, but it still needs to sync with the data stored in the controller's memory. This operation takes some network bandwidth and may cause loss of data if the FDD tool is not actively synching before the controller runs out of memory.

Table 1
PID parameters before and after automated correction in nine VAV boxes.

VAV #	PID parameters when hunting faults were detected		Improved PID parameters which eliminated hunting faults	
	K_p in %/° C (%/° F)	K_{iT} in %/° C (%/° F)	K_p in %/° C (%/° F)	K_{iT} in %/° C (%/° F)
003	5.6 (10)	0.6 (1)	0.2 (0.4)	0.02 (0.03)
005	8.3 (15)	1.1 (2)	0.3 (0.6)	0.03 (0.06)
008	2.8 (5)	0.1 (0.06)	0.2 (0.4)	0.02 (0.03)
010	5.6 (10)	0.1 (0.2)	1.2 (2.1)	0.1 (0.2)
012	2.8 (5)	0.6 (1)	0.3 (0.6)	0.03 (0.05)
013	5.6 (10)	1.1 (2)	0.6 (1.0)	0.06 (0.1)
015	8.3 (15)	1.7 (3)	0.4 (0.6)	0.03 (0.05)
048	0.6 (1)	2.2 (4)	0.1 (0.3)	0.01 (0.02)
105	4.4 (8)	2.8 (5)	0.2 (0.4)	0.02 (0.03)

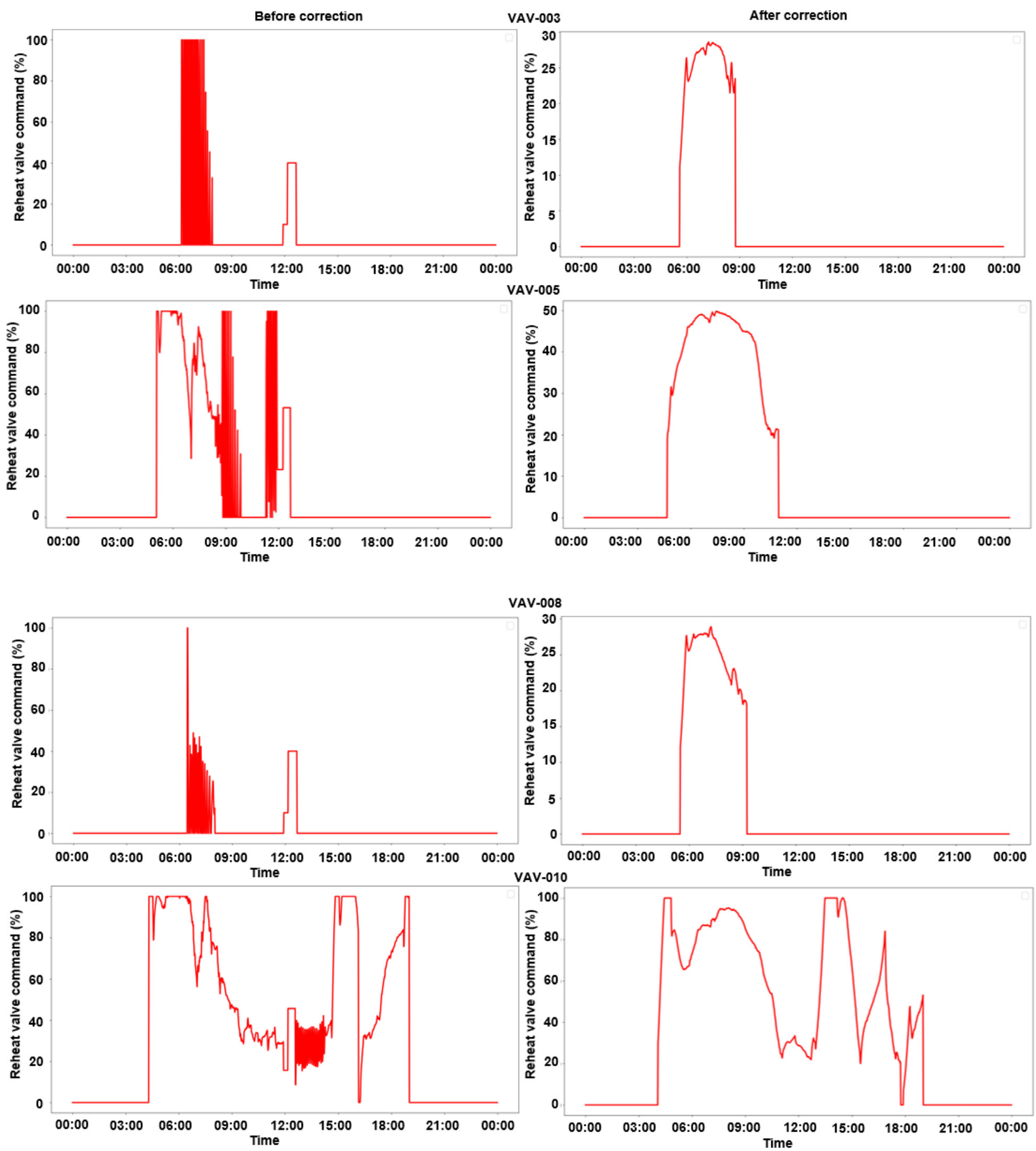


Fig. 11. Reheat valve command (CO) of eight VAV boxes (excluded VAV-105) before and after fault correction.

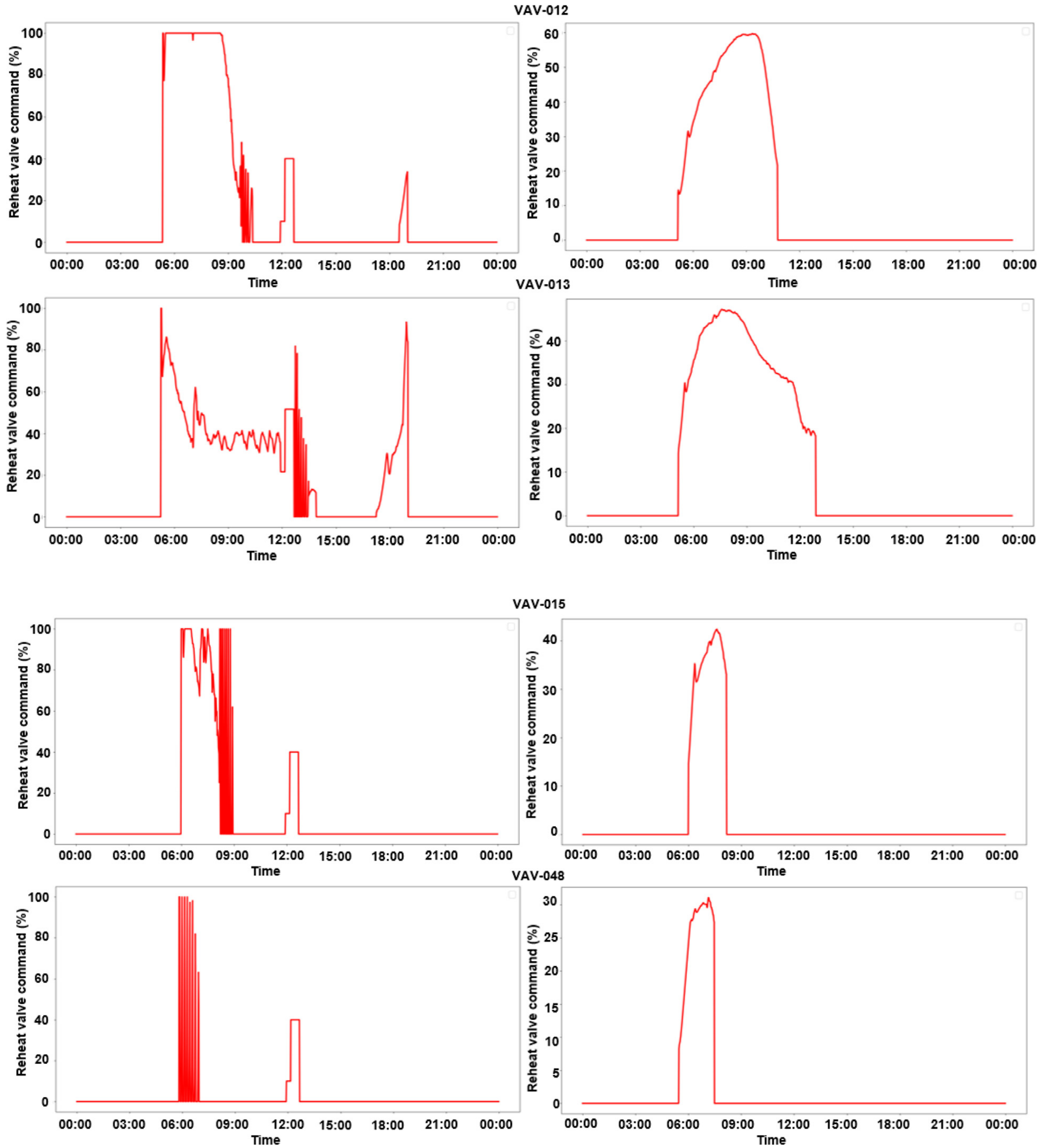


Fig. 11 (*continued*)

implement functionality on BACnet Trend Object trending intervals in FDD tools to allow dynamic changes to a sampling rate for a subset of points, to enable short-term testing and avoid network congestion. This should be achieved using universal protocols such as BACnet that are not proprietary to a specific software platform.

At the other end of the spectrum, some control loops have a very slow response time. For example, we measured a few hours of response time for some reheat valves – zone air temperature control loops. In these cases, the active tests must be implemented over a period of several hours, which increases the likelihood of external factors adding noise and invalidating the results. For example, outside air conditions, solar gain, and variations in external system inputs (e.g., hot water temperature, supply air temperature) can be significant driving factors behind zone air temperature changes over several hours. In addition, longer tests may have a negative impact on occupant comfort or cause equipment to operate outside of normal ranges if they are conducted when the building is occupied. This is a more fundamental problem than that of data sampling frequency, that will need more investigation from a scientific perspective (i.e., how to account for exogenous variables during tests) as well as a practical perspective (i.e., how to use unoccupied times to perform such tests).

Another challenge in generalizing the presented approach is that of dealing with more complex control loops (e.g., a CO used in multiple places or in a cascade PID control loop). Fig. 2 shows a basic PID control loop in which the PID controls one CO that only

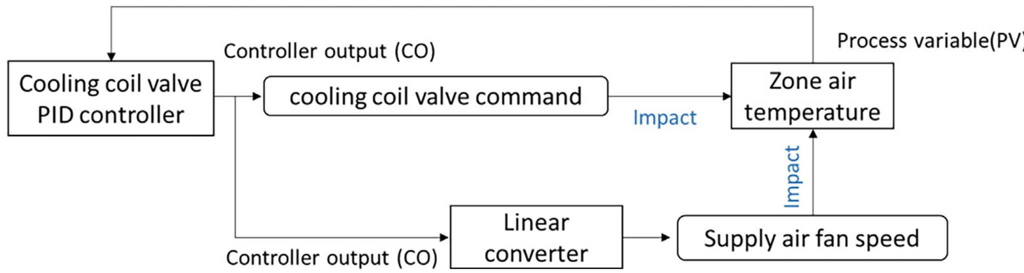


Fig. 12. Block diagram of the multiple control variables control loop in fan coil based on an actual system at the test site.

affects one process variable, but in control systems for HVAC, more complex control loops are often used. One example is control loops that use CO in two places to affect the PV (Fig. 12). Fig. 12 shows a sample fan coil zone temperature controller. The output of the fan coil PID controller is used in two places: (1) to adjust the cooling coil valve position to impact the zone air temperature (PV), and (2) as input to a linear converter that adjusts supply air fan speed, which also impacts the zone air temperature. Similarly, in cooling tower controls with staging, the output of the cooling tower leaving temperature controls is used (1) to adjust the fan speed to impact the cooling tower leaving temperature (PV) and (2) as an input to the cooling tower staging controller and to determine whether one or two towers were operating, which also dramatically impacts the cooling tower leaving temperature. The correction of control hunting faults for these complex PID control loops is more challenging. The relationship between the value of the PID parameters and the targeted CO is not one-to-one, as it is affected by other factors. All the variables that will be affected by the PID controller output need to be added into the observation list to ensure there is no adverse impact. Another example of a complex control loop is a cascade PID control loop in which two PID controllers are used in series; that is, the output of the first controller provides the inputs for the second controller. While some progress has been made in developing techniques for tuning these loops [24], how to do this automatically, external to the controller remains an open area of research. Considering all the challenges stated above, AHU cooling/heating valve – temperature control loops that are single loops with response time at time scales of minutes are the next good and impactful control loops to apply the developed fault correction algorithm.

5.2. Integration and engineering effort

To be truly scalable, these new auto-correction routines should be easy to integrate into existing mechanical systems and BAS installations at each site. The first step in integrating the control hunting routine is to verify the implementation of the PID algorithm within the BAS/controller logic.

During our field test, we spent significant time understanding the details of controllers, including controller outputs and process variables, PID equations, and tuning parameters. The names of controller outputs and process variables varied between BAS vendors and controllers' models and vintages, and depended on the specific sequence of operation used, which was often custom-coded on site during installation. Manual inspection of the control code was necessary, as it was difficult to identify programmatically the role and meaning of these variables outside of the BAS interface. In addition, we realized that PID equations and tuning parameters can be mathematically presented in different ways in different manufacturers' controllers, as there are no industry-wide standards to represent them. Eq. (1) shows the traditional PID control equation in which the proportional gain K_p, integral time T_i, and derivative time T_d are the three PID parameters that can be adjusted. Instead, in the field testing of this study, the PID equation was in parallel form (Eq. (6)) in which K_p and K_{iT} are the two PID parameters that can be modified. Further, in a Johnson Controls Inc. (JCI) unitary controller controller,[6] used in another campus building, the PID equation was in a different form as shown in Eq. (10), and the proportional band *(PB)* and integral time T_i were the adjustable tuning parameters.

$$CO(t) = \frac{100\%}{PB}\left[e(t) + \frac{1}{Ti}\int e(t)dt\right] \quad (10)$$

These equations were not expressed in the BAS or controllers but had to be retrieved from the instruction manual of each controller/BAS. Furthermore, the conversion from analog (i.e., documented in the literature) to digital PID equations (i.e., implemented in the actual controllers) introduced additional implementation choices in the BAS that needed to be considered. For this reason, the research team had to read the documentation of each controller carefully to understand the utilized PID equation and tuning parameters and develop the conversion functions accordingly. To overcome this issue, the control logic and the PID equations in each controller should be represented digitally, in such a way that they can be interpreted without human intervention. While industry is developing approaches to digitize BAS control logic [34], there is still work to do to make these systems more interoperable [3,30] and reduce the effort required to automatically correct these common hunting faults. These efforts to digitize BAS-level logic and enhance interoperability, are not getting to this level of PID control.

Another practical issue faced during our experiment was the potential need to modify the BAS control settings, making the PID tuning parameters accessible by the FDD tool through the BACnet network. This is a significant practical barrier to scalability of our approach, especially given the diversity of implementations and the variety of installed technology. To address this issue the BACnet standard should define a PID loop object using a standard equation and parameters, and require the PID parameters be exposed as objects within each BACnet-certified controller.

6. Conclusion and future work

Application of FDD tools in commercial buildings has grown and matured significantly over the past decade in the United States. Using FDD tools to correct control hunting faults has the potential to mitigate those issues cost-effectively across a great number of controllers from various BAS vendors in many buildings. This paper describes the development, implementation, and field testing of an automated control hunting fault correction algorithm in an FDD tool. Following the lambda open-loop tuning rules, the correction algorithm determines the improved PID parameters through step change tests and writes them into the BAS controllers.

[6] Johnson Controls. Unitary Controller. https://cgproducts.johnsoncontrols.com/MET_PDF/6363081.PDF

The algorithm was applied as four modules in the FDD tool: (1) detection of hunting faults, (2) management of active tests, (3) calculation of improved PID parameters, and (4) database and interface to access test results. The field test allowed the facility staff to optimize the operation of nine VAV boxes in an office building and eliminate the hunting faults, without human intervention. Facility staff has implemented the correction algorithm in all VAV boxes in the test building, inspired by the success. While this approach achieved first success in our field testing on reheat valve – DAT control loops of VAV boxes which data collection interval was adjusted to one minute, we identified several areas that need to be addressed in the future, to allow scalability to multiple HVAC equipment, buildings, BAS vendors, and FDD products. Future research recommendations for scale include the following:

- The ability to change BACnet Trend Object trending intervals dynamically from the FDD tool using BACnet functionalities[7] or equivalent approaches to enable short-term testing and more granular data collection. This would enable us to run this procedure in control loops with faster dynamics without saturating a controllers' memory.
- The ability to automatically read PID parameters in digital format and interpret their meaning in terms of alternative PID equations. This could be achieved by requiring standardization in PID loop implementation and/or documentation using a standard equation and parameters, and requiring BACnet objects to expose PID parameters for all control systems.

Additional research also should address how to correct hunting in more complex control loops, such as cascade systems and loops affecting multiple systems simultaneously.

Data availability

The research codes will be available at https://github.com/LBNL-ETA/SkySpark-based-Fault-Correction, when the opensource license is approved.

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.

Acknowledgements

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The authors wish to acknowledge Hayes Jones, Hannah Debelius, and Harry Bergmann for their guidance and support of the research.

References

[1] American Society of Heating Refrigerating and Air Conditioning Engineers (ASHRAE). (2020) *Standard 135-2020—BACnetTM—a data communication protocol for building automation and control networks* Available online: https://www.ashrae.org/technical-resources/bookstore/bacnet (2020), Accessed 24th Dec 2022.
[2] K.J. Åström, T. Hägglund, Revisiting the Ziegler-Nichols step response method for PID control, J. Process Control 14 (6) (2004) 635–650, https://doi.org/10.1016/j.jprocont.2004.01.002.
[3] Bergmann H., Mosiman C., Saha A., Haile S., Livingood W., Bushby S., Fierro, G., Bender J., Poplawski M., Granderson J., & Pritoni M. (2020). *Semantic Interoperability to Enable Smart, Grid-Interactive Efficient Buildings.* ACEEE Summer Study on Energy Efficiency in Buildings 2020. https://doi.org/10.20357/B7S304.
[4] Q. Bi, W.J. Cai, Q.G. Wang, C.C. Hang, E.L. Lee, Y. Sun, B. Zou, Advanced controller auto-tuning and its application in HVAC systems, Control Eng. Pract. 8 (6) (2000) 633–644, https://doi.org/10.1016/S0967-0661(99)00198-7.
[5] Borase et al., A review of PID control, tuning methods and applications, Int. J. Dyn. Control. 2021 (2021), https://doi.org/10.1007/s40435-020-00665-4.
[6] M.R. Brambley, S. Katipamula, Beyond commissioning: The role of automation, Pacific Northwest National Laboratory, 2005, Report PNNL-14990..
[7] M. Brambley N. Fernandez W. Wang K.A. Cort H. Cho H. Ngo J.K. Goddard Final Project Report: Self-Correcting Controls for VAV System Faults Filter/Fan/Coil and VAV Box Sections Volume 20 2011 Pacific Northwest National Laboratory (PNNL): Richland, WA, USA 10.2172/1032697 No. PNNL-20452;
[8] Y. Chen, G. Lin, E. Crowe, J. Granderson, Development of a Unified Taxonomy for HVAC System Faults, Energies 14 (17) (2021) 1–25, https://doi.org/10.3390/en14175581.
[9] Chintala et al., Identification and Elimination of Hunting Behavior in HVAC Systems, ASHRAE Trans. 121 (12) (2015).
[10] E.B. Dahlin, Designing and tuning digital controllers, Instruments and Control Systems. 41 (1968) 77–83.
[11] C. Dey, R.K. Mudi, An improved auto-tuning scheme for PID controllers, ISA Trans. 48 (4) (2009) 396–409, https://doi.org/10.1016/j.isatra.2009.07.002.
[12] Doe, Buildings Energy Data Book, United States Department of Energy, Washington DC, MD, 2011, Technical Report..
[13] N. Fernandez M.R. Brambley S. Katipamula Self-correcting HVAC controls: Algorithms for sensors and dampers in air-handling units 2009 Pacific Northwest National Laboratory Richland, WA 10.2172/973449 Report PNNL-19104.
[14] Fernandez, N., M. Brambley, S. Katipamula, H. Cho, J. Goddard, & L. Dinh. (2009b). *Self-correcting HVAC controls:. Project Final Report.* PNNL-19074. U.S. Department of Energy. https://doi.org/10.2172/971101.
[15] J. Granderson, G. Lin, H. Kramer, A Primer on Organizational Use of Energy Management and Information Systems, Department of Energy, Better Buildings, U.S, 2021, https://betterbuildingssolutioncenter.energy.gov/sites/default/files/attachments/EMIS_Primer_Organizational_Use.pdf.
[16] S.H. Hwang, S.M. Fang, Closed-loop tuning method based on dominant pole placement, Chem. Eng. Commun. 136 (1) (1995) 45–66, https://doi.org/10.1080/00986449508936353.
[17] C.Y. Jin, K.H. Ryu, S.W. Sung, J. Lee, I.B. Lee, PID auto-tuning using new model reduction method and explicit PID tuning rule for a fractional order plus time delay model, J. Process Control 24 (1) (2014) 113–128, https://doi.org/10.1016/j.jprocont.2013.11.010.
[18] S. Katipamula, M.R. Brambley, L. Luskay, Automated proactive techniques for commissioning air-handling units, J. Sol. Energy Eng. 125 (3) (2003) 282–291, https://doi.org/10.1115/1.1591800.
[19] Krakow, K, I., & Lin, S. (1995). *PI control of fan speed to maintain constant discharge pressure*. United States. https://www.osti.gov/biblio/211814-pi-control-fan-speed-maintain-constant-discharge-pressure.
[20] G. Lin, H. Kramer, V. Nibler, E. Crowe, J. Granderson, Building Analytics Tool Deployment at Scale: Benefits, Costs, and Deployment Practices, Energies 15 (2022) 4858, https://doi.org/10.3390/en15134858.
[21] G. Lin, M. Pritoni, Y. Chen, J. Granderson, Development and implementation of fault-correction algorithms in fault detection and diagnostics tools, Energies 13 (10) (2020), https://doi.org/10.3390/en13102598.
[22] T. Olsen, N. Ito, Implement an effective loop tuning strategy Retrieved at: Chem. Eng. Prog. (2013) 42–48. https://www.emerson.com/documents/automation/article-implement-an-effective-loop-tuning-strategy-en-37962.pdf.
[23] Price, C. R. et al. (2015). *Effective Tuning of Cascaded Control Loops for Nonlinear HVAC Systems.* ASME 2015 Dynamic Systems and Control Conference. https://doi.org/10.1115/DSCC2015-9806.
[24] Price C. R., & Rasmussen, B. P. (2018). *Soft Implementation of Cascaded Control Architectures Using the Youla Parameterization.* 2018 Annual American Control Conference (ACC), 4652–4657, https://doi.org/10.23919/ACC.2018.8431395.
[25] M. Pritoni, G. Lin, Y. Chen, R. Vitti, C. Weyandt, J. Granderson, From fault-detection to automated fault correction: A field study, Build. Environ. 214 (2022), https://doi.org/10.1016/j.buildenv.2022.108900.
[26] Pritoni M., Lin G., Chen Y., House J., Crowe E., Granderson J. (2022b). *Market barriers and drivers for the next generation Fault Detection and Diagnostic Tools.* ACEEE Summer Study on Energy Efficiency in Buildings 2022. https://doi.org/10.20357/B7801T.
[27] Pruna, E., Sasig, E. R., & Mullo, S. (2017). *PI and PID controller tuning tool based on the lambda method.* 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2017 - Proceedings, 18–20 Oct. 2017. Pucon, Chile, 2017, https://doi.org/10.1109/CHILECON.2017.8229616.
[28] G. Qu, M. Zaheeruddin, Real-time tuning of PI controllers in HVAC systems, Int. J. Energy Res. 28 (15) (2004) 1313–1327, https://doi.org/10.1002/er.1030.
[29] J.A. Romero, R. Sanchis, P. Balaguer, PI and PID auto-tuning procedure based on simplified single parameter optimization, J. Process Control 21 (6) (2011) 840–851, https://doi.org/10.1016/j.jprocont.2011.04.003.

[7] For instance, trends log sampling frequency can be set using the "Log Interval property" in BACnet, which specifies the periodic interval in hundredths of seconds for which the referenced property is to be logged when Logging Type has value Polled [1]).

[30] Roth A., Wetter M., Benne K., Blum D., Chen Y., Fierro G., Pritoni M., Saha A., and Vrabie D. (2022). *Towards Digital and Performance-Based Supervisory HVAC Control Delivery.* ACEEE Summer Study on Energy Efficiency in Buildings 2022. https://doi.org/10.20357/B70G62.

[31] Salsbury T. (2005). A survey of control technologies in the building automation industry. *IFAC Proceedings Volumes,* 38(1) (2005), 90–100. ISSN 1474-6670, ISBN 9783902661753, https://doi.org/10.3182/20050703-6-CZ-1902.01397.

[32] J.E. Seem, A new pattern recognition adaptive controller with application to HVAC systems, Automatica 34 (8) (1998) 969–982, https://doi.org/10.1016/S0005-1098(98)00033-8.

[33] V. VanDoren, Fundamentals of lambda tuning Retrieved at: Control. Eng. 60 (4) (2013) 33–36. https://www.controleng.com/articles/fundamentals-of-lambda-tuning/.

[34] Wetter, M., Ehrlich P., Gautier A., Grahovac M., Haves P., Hu J., Prakash A., Robin D., & Zhang K. (2022). OpenBuildingControl: Digitizing the control delivery from building energy modeling to specification, implementation and formal verification. *Energy,* 238, Part A, 2022, 121501. ISSN 0360-5442, https://doi.org/10.1016/j.energy.2021.121501.

[35] Zhao, F., J. Fan, & S. Mijanovic. (2013). *PI auto-tuning and performance assessment in HVAC systems.* Paper presented at the American Control Conference (ACC), Washington, DC, June 17–19. https://doi.org/10.1109/ACC.2013.6580094.

[36] Zhou, Q., & Liu, M. (1998). An On-line Self-tuning Algorithm of PI Controller for the Heating and Cooling Coil in Buildings. Energy Systems Laboratory (http://esl.tamu.edu); Texas A&M University (http://www.tamu.edu). Available electronically from https://hdl.handle.net/1969.1/6740.