Phyvac: A Python Module for Highly Flexible HVAC System Simulation, and Fault Dataset Generation as an Application Example

Shohei Miyata\textsuperscript{1}, Shanrui Shi\textsuperscript{1}, Yasunori Akashi\textsuperscript{1}, Takao Sawachi\textsuperscript{2}

\textsuperscript{1}Department of Architecture, School of Engineering, The University of Tokyo, Tokyo, Japan
\textsuperscript{2}Building Research Institute, Ibaraki, Japan

Abstract
Simulations are indispensable for heating, ventilation, and air conditioning (HVAC) systems in smart buildings. There is an increasing demand for detailed simulations in applications such as fault detection and diagnosis (FDD) and model predictive control (MPC). Various simulation programs have been developed for HVAC systems. However, simulation programs that can handle flexible system configurations and controls using a single programming language are still lacking. Therefore, the authors created a Python package called “phyvac” for HVAC system simulation. Phyvac was designed to incorporate controls with a high degree of flexibility and calculate their operating behaviors. A dataset of faulty chiller plant behaviors was generated as an example of a phyvac application.

Highlights
• Highly flexible HVAC system simulation too “phyvac” was developed.
• Phyvac is Python-based and open-source software.
• Some components were validated in a comparison to other simulation programs.
• A dataset of the faulty behaviors of a chiller plant was generated as an example of a phyvac application, and then compared with an LBNL FDD dataset.

Introduction
Recently, simulations have become indispensable for heating, ventilation, and air conditioning (HVAC) systems in smart buildings. There is an increasing demand for detailed simulations in smart building applications such as fault detection and diagnosis (FDD) and model predictive control (MPC). Various simulation programs for HVAC systems have been developed, including HVACSIM+ (Park et al., 1985), TRNSYS (Beckman et al., 1994), EnergyPlus (Crawley et al., 2001), IDA ICE (Kalamees, 2004), Hysopt (Riet et al., 2016), and Modelica Buildings Library (Wetter et al., 2014). Programs developed in Japan include the LCEM tool, ACSES/Cx, BEST, popolo, and ENe-ST (Ono et al., 2017). The primary purpose of these programs is to calculate the energy consumption. Although some programs can include detailed controls, it is necessary to understand the program in depth, modify the code, and occasionally write new code using multiple programs to incorporate controls with a high degree of flexibility. This requires significant ingenuity (Granderson, 2020).

Therefore, the authors utilized some of the features of the object-oriented language Python to develop a package called “phyvac,” which consists of consisting of equipment, piping and duct (branch), and control models, to facilitate the simulation of advanced controls in HVAC systems. To calculate the operational behavior, the physical state of the equipment must correspond to the control state. Therefore, convergence calculations are required for the flow rate calculation, making it possible to express feedback control, such as proportional-integral (PI) control. This feature has the advantage of incorporating faulty conditions into FDD research. In addition, regarding optimal control, phyvac can perform calculations with a resolution that allows it to function as an emulator. For example, in MPC research, phyvac can generate data for predictive models. Optimization programs and predictive models are built using third-party Python modules, and phyvac can then perform emulations for verification, with all of these operations performed consistently in Python code. One advantage of phyvac is that it uses Python, which has a rich machine-learning library, making it possible to apply it to various machine-learning and deep-learning algorithms.

In addition to an overview of phyvac as a whole, this paper presents component and flow balance calculations, the results of a verification of the equipment model through a comparison with other programs, and a case study of the calculation of fault conditions in a chiller plant as an example of the use of phyvac. Existing open fault data sets are also referenced to analyze the validity of the calculation results.

Phyvac overview
The name “phyvac” is a portmanteau of “py” and “hvac,” referring to HVAC simulation using Python, which was coined as the name for the program.

Figure 1 presents an overview of the simulation flow when using phyvac to compute the control setpoints, flow rate, pressure, temperature, and power consumption at each discrete time step of an HVAC system. First, inputs were provided, including the outdoor conditions, occupancy schedule, and internal heat generation. Control logic was used to regulate the operation of each device based on the room temperature and the control state from the previous time step. The flow rate and air volume were then calculated using the flow balance calculation, which depended on the states of the controlled devices. Subsequently, the inlet and outlet temperatures of the

Proceedings of the 18th IBPSA Conference
Shanghai, China, Sept. 4-6, 2023
https://doi.org/10.26868/25222708.2023.1370
equipment and power consumption were calculated using the calculated airflow and flow rates. Finally, the room temperature was determined based on the air supply conditions.

When modeling only the plant, the temperature of the return chilled water or hot water was calculated, assuming that the entire heat load was processed on the secondary side.

The time step could be set arbitrarily, and calculations were performed at 5 s intervals for variable air volume (VAV) control sequences and 1 min intervals for simplified control sequences in the recommended plant system in previous research (Nomura, et al., 2021) (Miyata et al., 2023).

To ensure that a simulation program can be used with confidence by a wide range of users and to create a thriving community in an open development environment, open source code (OSC) licenses are often applied. The Berkeley source distribution (BSD) license was selected for phyvac and phyvac is available on the GitHub website (website 1).

Figure 1: Phyvac simulation flow.

HVAC system modeling in phyvac
HVAC components and controls
Phyvac takes advantage of the object-oriented features of Python to create equipment and control components as classes, which are functions that can be replicated and utilized as a single program. The main components currently included in phyvac are listed in Table 1. These components include equipment such as chillers, control logic including proportional-integral-differential (PID) control, and functions for calculating air conditions, all of which are essential for modeling the operation of an HVAC system. A brief overview of the models for each component is presented below. Application programming interface (API) documentation is provided for each component, showing the model parameters, input/output information for functions in the model, and a code example (Figure 2).

- Chiller
  The chiller model includes the calculation of the operating point from the curves of the load ratio and coefficient of performance (COP) based on the condenser water temperature, as shown in Figure 3. Different curves were

![Figure 1: Phyvac simulation flow.](image)

![Figure 2: Example of API document](image)

<table>
<thead>
<tr>
<th>Components</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chiller</td>
<td>Chilled/condenser water flow/inlet temp., chilled water outlet temp. setpoint</td>
<td>Chilled water outlet temp., power</td>
</tr>
<tr>
<td>Air source heat pump (ASHP)</td>
<td>Chilled water flow/inlet temp., chilled water outlet temp. setpoint</td>
<td>Chilled water outlet temp., power</td>
</tr>
<tr>
<td>Water to water heat exchanger</td>
<td>Water flow, inlet temp.</td>
<td>Outlet temp.</td>
</tr>
<tr>
<td>Cooling tower</td>
<td>Condenser water flow, inlet temp., fan speed, outdoor condition</td>
<td>Outlet temp., power</td>
</tr>
<tr>
<td>Air Handling Unit (AHU)</td>
<td>Air/water flow/inlet temp./humidity</td>
<td>Air/water outlet temp./humidity</td>
</tr>
<tr>
<td>Pump/Fan</td>
<td>Water/air flow, frequency speed</td>
<td>Pressure gain, frequency speed</td>
</tr>
<tr>
<td>Valve/damper</td>
<td>Water/air flow, valve/damper opening</td>
<td>Pressure loss</td>
</tr>
<tr>
<td>Branch</td>
<td>Air/water flow</td>
<td>Pressure difference (loss or gain)</td>
</tr>
<tr>
<td>PID control</td>
<td>Setpoint, measured value</td>
<td>Control variable</td>
</tr>
<tr>
<td>Staging control</td>
<td>Flow/heat, waiting time</td>
<td>Operating number</td>
</tr>
<tr>
<td>Moist air function</td>
<td>Air variables</td>
<td>Other air variables</td>
</tr>
</tbody>
</table>

Table 1. Examples of components in phyvac.
prepared for different chilled water outlet temperatures. The model of the air-source heat pump is similar.

In the chiller model, it is assumed that the chilled water outlet temperature will remain at the setpoint when the load ratio is less than 100%. The model allows for an increase in the outlet temperature only when the load ratio exceeds 100%.

- Cooling tower
  The cooling tower is based on the EnergyPlus model. The total heat exchange between the air and condenser water is simulated using the equations proposed by Merkel (1925). The main parameters are the product of the overall heat transfer coefficient and heat exchange area \((UA)\), which are adjusted to calibrate the model for different equipment (Eq. 1). The air volume and fan power consumption are assumed to be proportional and cubically proportional to the fan speed, respectively.

\[
d\dot{Q}_{\text{total}} = \frac{U dA}{\varepsilon_p} (h_s - h_a) \tag{1}
\]

- Pump or fan
  The pump model calculates the pressure and power consumption based on the flow rate and frequency speed using the flow-head characteristic curves and flow-efficiency curves. Conversely, the flow rate can be estimated from the pressure. The curves are obtained by approximating performance test data and equipment characteristics. Currently, the model can handle up to third-order polynomial curves; however, the approximation process must be conducted carefully because a curve generally needs to be convex upward within the calculation range to obtain appropriate solutions.

Eqs. (2) and (3) can be used to transform each flow head (pressure) curve based on the frequency speed.

\[
H = f_s(G) \tag{2}
\]

\[
H = f_a \left( \frac{G}{SPD} \right)^{SPD^2} \tag{3}
\]

The consumed electrical power can be calculated using the following formula, where the fan model is similar to the pump model.

\[
pw = \frac{g \rho G H}{\eta} \tag{4}
\]

- Valve or damper
  In the valve class, the relationship between the flow rate and pressure loss with respect to the degree of valve opening was modelled. The relationship between the flow rate and valve opening degree, which results in a constant pressure loss before and after the valve, is called the inherent flow characteristic. Typical inherent flow characteristics include linear and equal-percentage characteristics (Figure 5).

Because valves with equal-percentage characteristics are commonly used in building HVAC systems owing to their controllability (Kako, 1957), phyvac valve models with equal-percentage characteristics were used. The ratio between the minimum and maximum controllable flow rates when the pressure losses before and after the valve are kept constant is called the inherent rangeability, \(R\). Additionally, \(Cv\) is the flow coefficient, and \(Cv_{\text{max}}\) is the flow coefficient when the valve is fully open. For a valve with equal-percentage characteristics, flow coefficient \(Cv\) for valve opening degree \(S\) is expressed as Eq. 5. Here, \(S_{\text{max}}\) represents the degree of a fully open valve (100%). When the above equations are rearranged into valve pressure loss \(\Delta P_v\), Eq. 6 is obtained. Because dampers may have different characteristics from valves, a method for approximating the pressure loss at the target flow rate and damper opening by approximating the experimental data was also proposed (Nomura et al., 2021).

\[
Cv = Cv_{\text{max}} R \left( \frac{S}{S_{\text{max}}} \right)^{1-1} \tag{5}
\]

\[
\Delta P_v = 1743 \frac{p_w G^2}{\rho Cv^2} \tag{6}
\]

- Branch
  The branch class solves the relationship between the flow rate and pressure difference between adjacent nodes, where the nodes are the branching and merging points of a piping or ducting system. The Darcy–Weisbach equation (Eq. 7) is used to calculate the pressure loss in the branch.

\[
\Delta p = \frac{\rho v^2}{2 \lambda} \tag{7}
\]
The Python composition feature within the branch class is used to incorporate instances of individually defined pumps, fans, valves, and dampers. The section on flow-balance calculations details the specific usage of the branch class.

- **PID control**

  PID control is one of the most common feedback control methods. PID control consists of a proportional (P) action that corrects the magnitude of the deviation (the difference between the target value and measured value of the controlled object) based on its size, an integral (I) action that removes the offset, and a differential (D) action that improves the response by reflecting changes in the deviation (Eq. 8). Noise and time delays exist in HVAC systems; therefore, PI control, which ignores the differential term, is often used. Consequently, the phyvac model discretizes the PI control (Eq. 9). Because an actual system and its simulation have different time intervals, the proportional gain and integral time, which are PID model parameters, may differ from those of the actual system.

  \[
  u(t) = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (8)
  \]

  \[
  u_n = K_p \left( e_n + \frac{1}{T_i} \sum_{i=0}^{n} e_i \Delta t \right) \quad (9)
  \]

- **Staging control**

  Based on variables such as the flow rate or load conditions, the number of pumps and chillers in operation can be adjusted as required. For flow-rate-based staging control, the number of pumps or chillers is increased when the flow rate exceeds a certain increase threshold (g12) for a fixed period of time and decreases when the flow rate falls below a certain decrease threshold (g21) for a fixed period of time (Figure 6). This fixed period of time is known as the “waiting time” and can be adjusted as necessary. It is possible to prevent frequent increases and decreases in operation by incorporating a gap between the increase and decrease thresholds, thereby improving system stability. Staging control in phyvac incorporates both thresholds and waiting times as model parameters.

  The control design for an HVAC system may be unique or incorporate complex algorithms such as deep learning. Therefore, phyvac assumes that other detailed control logic is directly written in Python code in the main program.

**Flow balance calculation**

A feature of phyvac is the principle-based calculation of the flow transport system models (flow balance calculation). The balance point is calculated by balancing the pressure loss owing to piping or ducting and equipment with the pressure boost provided by the pump or fan, assuming a steady state at each time point. This flow-balance calculation is analogous to Kirchhoff’s circuit laws for electrical circuits and follows two rules:

1. At any node, the total inflow and outflow quantities are equal.

2. In any closed loop, when the pressure is considered in one direction, the sum of the pressures is zero.

In phyvac, all the flow-balance calculations are solved using the bisection method. It was necessary to write code for the bisection method in the main program. The bisection method is a technique used to numerically solve the inverse function \( x = f^{-1}(y) \) for a given function \( y = f(x) \). However, its application to convex functions has limitations, and it is necessary to assume the range in which a solution exists in advance. If the pump characteristics and pressure loss formula can be expressed as first-order equations, the solution can be obtained using a matrix. However, because the pressure loss is a second-order equation (Eq. 7) and the pump or fan characteristic is an approximation formula of the second order or higher (Figure 4), the bisection method or Newton’s method is required to obtain numerical solutions.
The significant difference between the pressure-flow characteristics of HVAC systems and electrical circuits is that those of HVAC systems are expressed as approximations of the second order or higher.

Examples of a piping loop and its solution flow are shown in Figures 7 and 8, respectively. The coding of the flow balance calculation requires considerable effort, which is a challenge for phyvac. However, there are plans to enhance the sample code on GitHub.

**Calculation of equipment inlet temperature**

The flow rate and inlet temperature are used in each equipment model to calculate the outlet temperature and power consumption. In phyvac, the inlet temperature of each piece of equipment is calculated from the outlet temperature of the adjacent equipment in the previous time step and the current flow rate. The calculation code was written directly in Python. Miyata et al. (2023b) proposed a method for calculating the inlet and outlet temperatures of equipment based on the “feeds” relationships of Brick models. Brick models can also be used as coding assistance.

**Validation based on SHASE-G 0023-0022**

Tests were conducted to validate the simulation program’s accuracy based on the SHASE-G 0023-2022 “Guidelines for Evaluating Building Energy Simulation Tools.” This guideline is a revised edition of SHASE-G 1008-2016 (Ono, 2017), which sets various calculation conditions for individual pieces of equipment, subsystems, and entire systems, and presents examples of calculation results from multiple simulation tools. This allows newly developed programs to be compared with other programs. In this study, individual equipment tests were conducted on pumps and cooling towers.

**Pump (Case E-PMP series)**

In the equipment model test for a pump, Case 100 represented the rated condition. Cases 101–103 were used to confirm the calculation of operating conditions when the flow rate was reduced at a constant rate. Cases 110–113 were used to confirm the calculation of the changes in the head curve at 75% of the maximum speed. Similarly, Cases 120–124 were used to confirm the calculation of the changes in the head curve at 50% of the maximum speed.

The calculated power consumption showed a trend similar to the manufacturer-provided values, with an accuracy that was comparable to or even higher than those of other programs (Figure 9(a)).

**Cooling tower**

In the cooling tower tests, Case 100 represented the rated condition. Cases 101–103 evaluated the response under partial load conditions by reducing the outdoor wet-bulb temperature and condenser water inlet temperature while also decreasing the water flow rate. In addition to these conditions, Cases 110–111 included a reduced airflow to evaluate the reduction in fan power consumption that would result from changes in the condenser water outlet temperature and airflow at the air and water sides under partial load conditions.

The results obtained using phyvac showed a trend similar to that shown by the manufacturer’s specifications. However, a deviation in the condenser water outlet temperature was observed at low condenser water and air flow rates (Figure 9(b)). This could be improved by adjusting parameters such as the heat exchange efficiency.

**Fault behavior simulation**

In this section, we show how phyvac could be used to simulate faults in an HVAC system. We analyzed the calculation results of a sensor bias error in the condenser water outlet temperature of a simple chiller plant (Figure 10), consisting of chillers, cooling towers, and various pumps. Various other fault simulations of the same system are modeled in Miyata (2023a) and utilized to develop the FDD algorithm. We also analyzed a fault dataset publicly released by the Laurence Berkeley National Laboratory (LBNL) as a reference to verify the validity of the calculation (website 2).

**Fault modeling: Cooling tower condenser water outlet temperature sensor bias (+2 °C)**

In the target system, the frequency speed of the cooling tower fan is controlled to maintain the condenser water outlet temperature at a setpoint. This setpoint is calculated based on the outdoor wet-bulb temperature. Phyvac can easily implement a sensor bias fault by first defining the bias value and then incorporating it into the PID control (Figure 11, red section).

If the condenser water outlet temperature sensor in the cooling tower measures a value higher than the true value, this biased measurement affects the fan control. As a result, the fan speed increases, and the true value of the condenser water outlet temperature decreases compared to the condition without a fault. Therefore, the condenser water inlet temperature of the chillers also decreases, improving the COP of the chillers. However, the power consumption of the cooling tower fans will increase.
addition, because the condenser water inlet temperature of the chillers is used to control the condenser water bypass valves to maintain the temperature above a setpoint to protect the chillers, the opening of the valves is also affected by low outdoor wet-bulb temperature conditions. Thus, a single fault can affect various parts of the system, and phyvac can easily incorporate such faults.

**Result of faulty behavior simulation**

Simulations were conducted for two cases under real heat load and outdoor web bulb temperature conditions recorded in Sendai, Japan in 2019: a fault-free (“Unfaulted”) condition, and a condition with a fault (“Cooling tower bias+2”). The subject system was a chiller plant for a factory, which demands a heat load 24 hours a day.

The energy consumption results for both cases are presented in Table 2. Furthermore, it was confirmed that the heat loads were almost identical in both cases. The cooling tower condenser water outlet temperature and cooling tower fan speeds for a representative summer week are shown in Figures 12 and 13, respectively, and the annual condenser water bypass valve opening behavior is shown in Figure 14.

The energy consumption of the cooling towers increased because of a fault in which the value reported by the condenser water outlet temperature sensor of the cooling tower was 2 °C higher than the true value, which caused a slight decrease in the energy consumption of the chillers due to a reduction in the condenser water inlet temperature of the chillers. Figures 12 and 13 show that the true value of the temperature decreased only slightly during the representative week, because the fan speed was at its maximum value (0.5) in the target system. It should be noted that the temperature values shown are those logged by the biased sensor.

In addition, a slight change in the power consumption of the condenser water pump was observed (Table 2). The fault reduced the condenser water inlet temperature of the chiller, which led to an increase in the bypass control in winter to protect the chiller (Figure 14). Consequently, more of the flow was split between the parallel cooling tower and bypass branches, resulting in different pressure drops. However, the impact on the energy consumption was small because the heat load was relatively small in winter, and the condenser water flow rate was operated at a partial load.

**Analysis of reference dataset of the fault: LBNL fault detection and diagnostics dataset**

To validate the simulation results of the aforementioned fault behavior, an analysis of an open fault dataset was conducted (LBNL FDD dataset, website 2). This dataset includes the chiller plant shown in Figure 17, as well as a sensor bias (+2 °C) in the cooling tower leaving (outlet) temperature sensor. Although the numbers of chillers and cooling towers in this system differ from those shown in Figure 10, the basic configurations are similar. This dataset was prepared for a one year period based on weather conditions in Chicago, United States, in 2018.
and the plant was assumed to provide chilled water to a typical large office.

The annual energy consumption (Table 3), operational behavior in a representative week (Figures 16 and 17), and annual three-way valve control behavior (Figure 18) are shown from the LBNL dataset. For this dataset, the heat load values were almost the same with and without the fault (“unfaulted”).

The fault reduced the power consumption of the chiller and increased the power consumption of the cooling tower, as shown by the annual energy consumption, similar to the results listed in Table 2. The power consumption values of the chilled water primary and secondary pumps remained almost the same. However, the power consumption of the condenser water pumps decreased by 5.8%, and the total power consumption decreased because of the fault.

The trend of the control behavior during a representative summer week (Figures 16 and 17) was similar to that shown in Figures 14 and 15, with the apparent cooling tower outlet temperature increasing as a result of the fault, while at the same time the fan speed was at its upper limit. The three-way valve opening for the condenser water was also found to be larger in the winter when it was faulty than when it was fault-free (Figure 18). Because the condenser water pumps were always operated at a constant speed, the impact on the energy consumption was considered to be relatively large.

Discussion of fault behavior simulation

The above results confirmed that the simulation of fault conditions by phyvac could calculate faulty behaviors similar to the existing datasets by Modelica Buildings Library and EnergyPlus. Moreover, the propagation and magnitude of the influence differed owing to the different system configurations and detailed control logic. For example, in the phyvac case, the two cooling towers were controlled in unison, and the cooling water outlet temperature was measured at only one location. In contrast, in the reference case, cooling water outlet temperature values were measured for each of the three cooling towers, and the effect of the sensor bias at only one location was relatively small. This suggests that it is important to adequately model the system and control to appropriately represent the characteristics of the faults in the target system.

In addition, the opposite trends of an increase and a decrease in the total energy consumption are observed in Tables 2 and 3, respectively. If a fault is defined as “a factor that adversely affects the indoor environment or energy performance,” it is difficult to interpret the free-of-fault result in the reference case where the heat load is the same but the power consumption increases over the fault case. The ideal form of fault datasets should be further discussed in the future along with the development of FDD algorithms.

Conclusion

In this paper, we first presented an overview of the Python module “phyvac” for calculating the operating behavior of an HVAC system, along with its computational logic. The phyvac equipment classes were validated based on the guidelines for evaluating simulation tools, and no major problems were found. Finally, the cooling tower condenser water outlet temperature sensor bias was

---

**Table 3: Yearly power consumption [MWh]**

<table>
<thead>
<tr>
<th></th>
<th>Chiller</th>
<th>Primary pump</th>
<th>Condenser pump</th>
<th>Cooling tower</th>
<th>Secondary pump</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unfaulted</td>
<td>384</td>
<td>149</td>
<td>258</td>
<td>18</td>
<td>436</td>
<td>1245</td>
</tr>
<tr>
<td>Cooling tower bias+2</td>
<td>378</td>
<td>149</td>
<td>243</td>
<td>26</td>
<td>435</td>
<td>1230</td>
</tr>
<tr>
<td>Impact ratio</td>
<td>-1.5%</td>
<td>0.0%</td>
<td>-5.8%</td>
<td>43.0%</td>
<td>-0.3%</td>
<td>-1.2%</td>
</tr>
</tbody>
</table>

---

Figure 15: Schematic of the chiller plant in LBNL FDD dataset (Obtained from website 2)

Figure 16: Cooling tower condenser water outlet temperature (LBNL FDD dataset)

Figure 17: Cooling tower fan speed (LBNL FDD dataset)

Figure 18: Condenser water bypass valve opening (LBNL FDD dataset)
modeled as a fault in a chiller plant and the results were analyzed and compared with the LBNL FDD dataset.

We demonstrated that phyvac can be used to construct flexible simulations of HVAC systems and that fault conditions can be incorporated as desired. Although room temperature calculations are not included in the current release, we plan to build a simulation ecosystem that includes room temperature calculations in the future.

Because this program is an OSS, it provides an environment in which functions can be expanded by voluntary participants. Therefore, the formation of a development community is expected, along with the development of functions to cover a wide range of HVAC systems that are not limited to systems for specific regions or applications. We hope that this program will be widely used to research and implement smart building applications such as FDD, optimal control, and demand response.

Acknowledgement

This study was conducted as part of “Low Energy Housing and Building with Validated Effectiveness project phase 7” established by the Institute for Building Environment and Carbon Neutral for SDGs funded by Ministry of Land, Infrastructure, Transport and Tourism in Japan. We would like to thank all of those involved in this study.

Nomenclature

\( \dot{Q}_{\text{total}} \) steady state total heat transfer [W]
\( U \) cooling tower overall heat transfer coefficient [W/(m² °C)]
\( A \) heat transfer surface area [m²]
\( c_p \) specific heat of moist air [J/(kg °C)]
\( h_s \) enthalpy of saturated air at the wetted-surface temperature [J/kg]
\( h_a \) enthalpy of air in the free stream [J/kg]
\( H \) pump head [Pa]
\( f_a \) function between flow rate and pump head
\( G \) mass flow [kg/s]
\( SPD \) pump speed ratio (0.0~1.0)
\( pw \) power [kW]
\( g \) gravitational acceleration [m/s²]
\( \eta \) efficiency [-]
\( Cv \) flow coefficient [-]
\( R \) rangeability [-]
\( S \) valve opening (0: closed, 1: fully open)
\( \rho_w \) density of water [kg/m³]
\( \rho \) the density of the fluid [kg/m³]
\( \Delta p \) pressure drop [Pa]
\( \lambda \) the Darcy friction factor
\( L \) branch length [m]
\( v \) flow velocity [m/s]
\( D \) the hydraulic diameter [m]
\( u \) control output
\( K_p \) proportional gain
\( T_i \) integral time
\( T_D \) derivative time
\( e \) error value
\( u_n \) control output at time step n
\( e_i \) error value at time step i

References


Kako, G (1957), Adjustment valves (2), Measurement 7 (7), SICE.


Website1: https://github.com/ShoheiMiyata/phyvac
Website2: https://faultdetection.lbl.gov/