Data-driven fault tolerant predictive control for temperature regulation in data center with rack-based cooling architecture

Kai Jiang, Shizhu Shi, Chuan Hu, Masoud Kheradmandi, Souvik Pal, and Fengjun Yan

***Abstract*—Recently increasing mount of data centers (DCs) have been constructed for the information and communications technology (ICT). Meanwhile, more power consumptions are required to support the operation of DCs, where nearly thirty percent of the consumptions is used for cooling system. Compared with traditional computer room air conditioner (CRAC) in DCs, rack mountable cooling unit (RMCU) has attracted more interests thanks to the energy efficiency, scalability and flexibility. However, the control design of RMCU is always obstructed by the physics modeling and some occasional faults in the system. Inspired by such problems, data-driven fault tolerant predictive control based on RMCU is proposed to regulate temperature in DC. In this work, multiple locally linear models in form of auto-regressive exogenous (ARX) model are selected to describe the strongly nonlinear system. Besides, the multiple sub-models are identified through partial least square (PLS) and corresponding training data is partitioned by fuzzy c-means (FCM). Then, on the basis of developed data-driven model, a predictive controller considering actuator faults is designed to manage the temperature in DC. Finally, real experimental results are presented and demonstrate the superior performance of proposed controller.**

***Index Terms*—Data center, temperature regulation, data-driven model, predictive control, fault tolerant control**

I. Introduction

With the growing demand of novel ICT, more DC facilities are built to maintain the operation and more energy consumption is required accordingly [1]. It was reported that the power consumption of DCs was nearly 1.3 % of the total global electricity in 2010, and 30% of the consumption in DCs was supplied to cooling system [2]. Therefore, appropriate cooling architectures and control strategies are needed to improve the energy efficiency in cooling system. Generally, there are three cooling structures in DC including room-based, row-based and rack-based cooling. Here room-based cooling means the cold air is supplied to the computer room directly and cool down the whole servers; row-based cooling denotes that several cooling units are mounted between the racks and each of them delivers cold air to chilling a row of servers; rack-based cooling indicates the cooling unit is installed within a single rack [3]. The schematic diagram of three different cooling architectures is shown in Fig. 1. Nowadays room-based cooling architecture is widely used in DCs due to the simplicity of manufacturing and handling, while the problems of cold air bypass and hot air recirculation in room-based cooled DCs deteriorate the energy efficiency significantly [4]. Therefore, row-based and rack-based cooling architectures are developed and become prevalent, in which shorter and more predictable airflow paths are able to address the above problems and reduce the pow consumption [5].

Except for the development of cooling architecture, control design of regulating temperature in DCs is also a critical and challenging issue [6]–[8]. Firstly, the DC system is a nonlinear multi-input-single-output (MISO) system, and thus advanced controller is necessary to ensure the temperature security. Nevertheless, the complicated airflow and heat transfer in DC further increase the difficulty of modeling and control. Therefore, many researchers are attempting to explore efficient control strategies to overcome the drawbacks, and some relevant works about cooling system control have been carried out so far.

The initial control strategy used in cooling system is on-off control, which means the cooling devices would be turned off or on when the temperature below or above the set point. This on-off control is the simplest control in cooling system, while the frequently on-off would lead to temperature instability and reduce the lifetime of cooling equipments [9]. Proportional-integral-derivative (PID) control is another effective technique for cooling system control in DCs. For instance, Baptiste et al. designed a PID controller for DC optimization in [10], where the controller was mainly utilized to manage the air speed. Self-tuning PID control was proposed to control CRAC in DCs as well [11]. In this work, the PID control was adopted to regulate temperature automatically and a fuzzy logic method was used to tune the parameters in PID controller. Although PID control had achieved good effectiveness in cooling system, the problem of large temperature oscillation was still a threat for server security and probably resulted in energy inefficiency. Lately, model predictive control (MPC) in DC cooling system has attracted a great deal of interests. Decentralized MPC was employed to control the blower speed and supplied water temperature in multiple CRAC units [12]. This method was able to meet the constraints of temperature in each zone and minimize the pow consumption of CRAC at the same time. A novel controller based on MPC was proposed for cooling system in [13], in which indirect fresh air was utilized and the pow consumption was reduced by 30%. Besides, the authors in [14] investigated DC control in cyber-physical perspective through MPC. This work considered computational network, thermal network and the effect of electricity market, and MPC was used to implement the global optimization in DC.

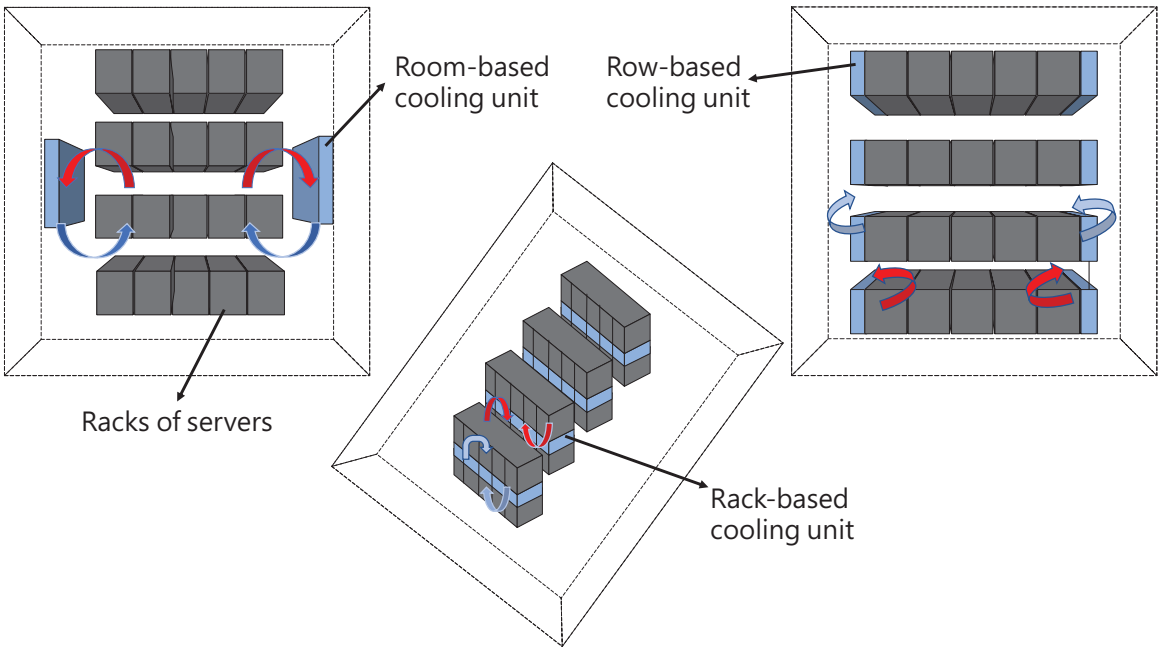

Fig. 1. Schematic diagram of DC with room-based, row-based and rack-based cooling architecture

.

At last the simulation indicated this method could achieve great cost saving. As discussed above, MPC is extensively applied in DCs and obtains satisfactory effect. However, most of the existing works concerning MPC in DCs are based on physics models, which is quite complicated and time-consuming. Moreover, inaccurate physics models owing to lots of modeling assumption usually lead to the decrease of control performance.

To tackle the problems, a data-driven predictive control is put forward for DC with rack-based cooling architecture. In order to simulate the nonlinear characteristics in DC system, the data-driven model is established on multiple linear ARX models which is identified via PLS. Besides, the algorithm of FCM is utilized to partition the training data for multiple modeling. Finally, appropriate weights derived through membership matrix in FCM are used to combine the multiple sub-models together. According to such data-driven model, predictive controller is developed to implement the temperature regulation in DC. Additionally, fault tolerant strategy is creatively considered in the data-driven control to deal with three typical actuator faults (aging of fans, failure of fans and valve deactivation) in rack-based cooling architecture.

The rest of this article is organized as following. Section II presents the fundamental of DC and the rack-based cooling system. Then the methodology of data-driven modeling is introduced in Section III. The detailed development of fault tolerant predictive control and experimental results in real system are provided in Section IV and Section V separately. Finally, conclusion is depicted in VI.

II. Fundamental of DC with Rack-based Cooling Architecture

Cooling system plays a significant role in DCs to regulate temperature and further protect servers. Currently, there are three types of cooling methods in DCs, which contains air cooling, liquid cooling and phase-change-material cooling [15]. Considering the operational feasibility, cooling security and equipment cost, the method of air cooling is widely adopted in DC cooling system. In light of cooling architecture, room-based, row-based and rack-based cooling units are regarded as three typical architectures. As mentioned above, row-based and rack-based cooling system are more energy-efficient over room-based cooling system. Besides, in row-based and rack-based cooling systems, rack-based cooling system is more suitable to DCs with stand-alone high-density racks due to the flexibility and scalability. Thus, the detailed investigation relating to rack-based cooling architecture is conducted in this work.

The main concept of rack-based cooling system in DCs is to extract the heat generated by servers out of rack through the heat exchange between chilled water and hot air. The airflow from hot aisle to cold aisle is driven by several fans and goes through a heat exchanger filled with circulating chilled water, then the heat of hot air would be taken away via the chilled water. The circulating water is supplied and cooled through a chiller outside. The whole system is depicted in Fig. 2, and the rack mountable cooling unit (RMCU) is also represented in Fig. 3, which comprises fans, heat exchanger, valve, power source and some pipes. Moreover, the location of RMCU in the rack could be adjusted based on the demand of customers.

According to Fig. 2 and Fig. 3, we can figure out the

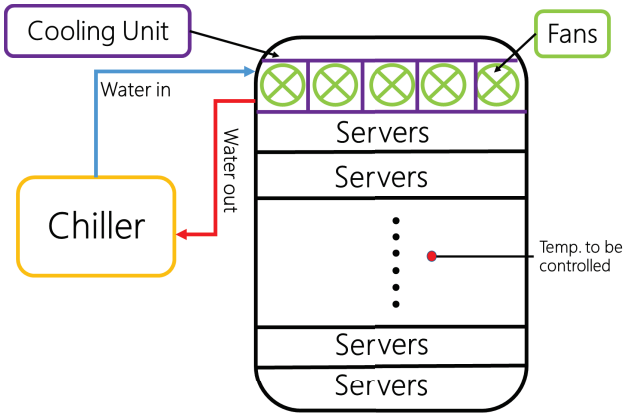

Fig. 2. Detailed configuration of DC with rack-based cooling architecture.

thermal control system in DC clearly. The output or the control objective is the temperature in rack. The inputs are the air flow rate controlled by fans and the water flow rate controlled by valve. In addition, the temperature of inlet water is considered as measured disturbance, which is determined by the chiller system in the building.

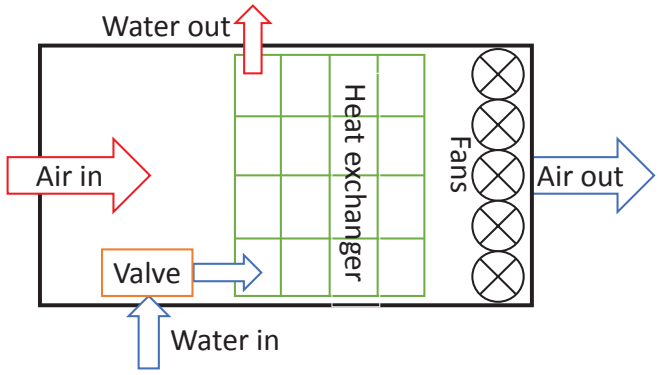

Fig. 3. Internal structure of rack mountable cooling unit.

III. Data-Driven Modeling

Although the individual rack is just a small part of DCs, the thermal dynamics inside including heat conduction, convection, and radiation are quite complicated and thus difficult to describe through a physics model. Nevertheless, the different types of servers in the rack have different thermal characteristics, which would further increase the modeling difficulty. Therefore, a data-driven model is developed in this program to tackle the modeling problem and used for further predictive control.

The framework of data-driven model is based on ARX model, which is an efficient linear discrete model to represent the relationship between inputs and outputs [16]. The typical ARX model is formulated as following. As seen in Eq. 1, the output at specific step is a function of historical outputs, current inputs, historical inputs, constant factor and modeling bias. $x(k)$ in the equation denotes output vector at kth step, $u(k)$ stands for input vector, and $b(k)$ means the modeling bias. The parameters α_i and β_i are system coefficients. σ represents the constant factor. N_x and N_u express the total number of time lapse in historical outputs and inputs.

$$x(k) = \sum_{i=1}^{N_x} \alpha_i x(k-i) + \sum_{j=0}^{N_u} \beta_j u(k-j) + \sigma + b(k), \quad (1)$$

The algorithm of PLS is utilized for parameter identification in ARX model in this work, which is a popular multivariate statistical method of data-driven modeling. The main procedure of PLS comprises two steps: one is data preprocessing and the other one is model identification [17], [18]. The data preprocessing is conducted through the approach of principle component analysis (PCA) that has the ability of projecting the latent variables in high-dimension to low-dimension, and parameter identification is carried out by a typical method of regression called ordinary least square (OLS). Owing to the superiority of PCA, the method of PLS is quite suitable to handle the modeling issue with high correlation. The main equations in PLS are expressed as following.

$$X = T_X P^{\mathrm{T}} + E_X, \quad (2)$$

$$Y = T_Y Q^{\mathrm{T}} + E_Y, \quad (3)$$

where X and Y represent inputs and outputs, T_X and T_Y stand for principle component of inputs and outputs respectively, E_X and E_Y denote the model residuals of X and Y, P means the matrix of loadings of X and Q is the matrix of loadings of Y. Then, to find the relationship between inputs and outputs, a function linked two principle components is required:

$$T_Y = ST_X, \quad (4)$$

where S is the parameter of two principle components. At last, by applying Lagrange multiplier, all the parameters are derived and the final relationship of inputs and outputs is demonstrated as below:

$$Y = \theta X + b(k), \quad (5)$$

in which θ is defined as the model parameters. Besides, if the objective function is organized as the form of ARX model, model parameters can be depicted as: $\theta = [\alpha_1 \cdots \alpha_{N_y} \quad \beta_1 \cdots \beta_{N_u} \quad \sigma]$ and $X = [x(k-1)^T \cdots x(k-N_y)^T \quad u(k)^T \cdots u(k-N_u)^T \quad 1]^{\mathrm{T}}$.

As the thermal dynamic in DC is a quite complex process, only a linear ARX model is insufficient to represent the system. Additionally, the complicated system is later found to be locally linear in short time series. Thus, multiple local linear models are put forward to approximatively simulate the nonlinearities in DC system. In order to obtain multiple models, the dataset for training is divided into some clusters and then the model parameters are calculated on the basis of each cluster. In this article, the approach of FCM is used for clustering, which is a typical method of clustering and widely adopted for data processing, image segmentation

and so on. The idea of FCM is to obtain the partitioned data and corresponding membership matrix by minimizing a objective function that is the sum of weighted absolute distances between cluster centers and each data point [19], [20]. It is noteworthy that the membership matrix would be utilized for later combination of the identified ARX models. The mathematical equation of objective function is depicted as follows.

$$J_{FCM} = \sum_{i=1}^{G}\sum_{j=1}^{D}(\mu_{j,i})^q||x_i - c_j||. \tag{6}$$

where x_i means the data point and the mount of data items is G. c_j denotes the center of clusters and the total number of clusters is D. $\mu_{j,i}$ expresses the membership matrix and it means the degree of x_i belonging to the jth cluster. q is defined as a scaling factor on each fuzzy membership. Moreover, some constraints should be considered within the minimization of objective function, which is described as following:

$$\begin{aligned}
&\mu_{j,i} \in [0,1], \quad for \quad 1 \le i \le G, 1 \le j \le D,\\
&\sum_{j=1}^{D}\mu_{j,i} = 1, \quad for \quad 1 \le i \le G, 1 \le j \le D,\\
&0 < \sum_{i=1}^{G}\mu_{j,i} < G, \quad for \quad 1 \le i \le G, 1 \le j \le D.
\end{aligned} \tag{7}$$

After setting the initial value of membership matrix $\mu_{j,i}$ and cluster centers c_j, the fuzzy clustering is carried out through a method of iterative optimization which is worked for updating the membership matrix and cluster centers. The equations are shown as below:

$$\mu_{j,i} = \frac{1}{\sum_{k=1}^{D}(\frac{||x_i - c_j||}{||x_i - c_k||})^{2/(q-1)}}, \tag{8}$$

$$c_j = \frac{\sum_{i=1}^{G}(\mu_{j,i})^q x_i}{\sum_{i=1}^{G}(\mu_{g,i})^q}. \tag{9}$$

The terminal condition for iteration is $\max_{ij}|\mu_{j,i}(s+1) - \mu_{j,i}(s)| < \epsilon$ or $s < N_{set}$, where ϵ is defined as a termination criterion, s means the iterative step and N_{set} is the set point of terminal steps.

As the total sub-models may contribute to the future prediction, it's better to identify them simultaneously [21]. The detailed computing trick is represented as follows:

$$Y = [\theta_1 \quad \theta_2 \cdots \theta_D] \times [U_1 \odot X \quad U_2 \odot X \cdots U_D \odot X]^{\mathrm{T}}, \tag{10}$$

in which $\odot$ means Hadamard product, and U stands for an augment of membership matrix. The extended matrix is shown as below:

$$U_j = \begin{bmatrix} \mu_{j,1} & \mu_{j,1} & \cdots & \mu_{j,1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{j,G} & \mu_{j,G} & \cdots & \mu_{j,G} \end{bmatrix}_{G \times D}. \tag{11}$$

Then based on the operation introduced above, all the parameters θ and sub-models could be derived through PLS. In addition, to represent the total nonlinearities in DC system, all the sub-models should be combined into one equation through appropriate weights which are related to the membership matrix mentioned above. The entire modeling diagram is described in Fig. 3, and the final integrated ARX model is shown as below:

$$\begin{aligned}
x_k = \sum_{d=1}^{D}\omega_d[\sum_{i=1}^{N_x}\alpha_{i,d}x_{k-i} + \sum_{j=0}^{N_u}\beta_{j,d}u_{k-j} \\
+ \zeta_d\lambda + \sigma_d].
\end{aligned} \tag{12}$$

Specifically, in this work x_k stands for the temperature in the rack, u_k denotes the control inputs including air flow rate and water flow rate, and ω_d means the weight of each sub-models. Besides, temperature of input water and workload of the whole rack are considered as measurable disturbances and defined as vector λ. ζ_g represents the coefficient vector of disturbances and σ_d is constant factor. It should be pointed out that modeling bias $b(k)$ is eliminated during the calculation of PLS. The schematic diagram of developed data-driven modeling is described in Fig. 4.

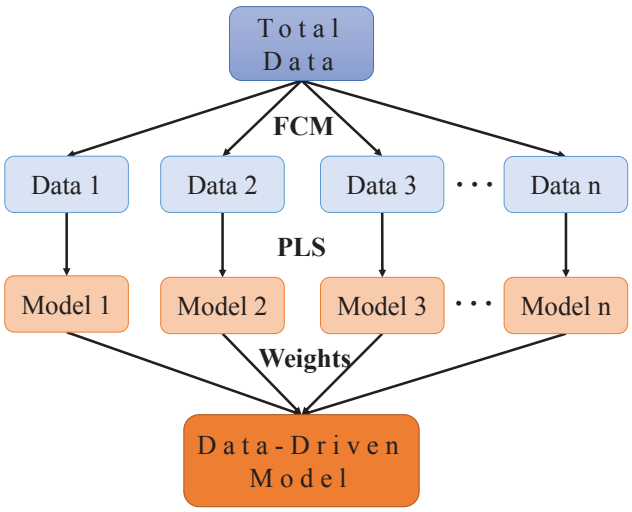

Fig. 4. Schematic diagram of developed data-driven modeling.

IV. Fault Tolerant Predictive Control

To accurately regulate the temperature in DCs, many control strategies such as on-off control, PID control and MPC are employed. Particularly, MPC is widely used for DC management due to the ability of dealing with time series prediction, constrained optimization and nonlinear systems [22]–[26]. In our previous work [27], MPC control is also utilized to implement the temperature regulation based on data-driven model. However, during the real industrial operation, we found that some occasional actuator faults would deteriorate the performance of data-driven predictive controller and further threaten the security and reliability of entire system. Consequently, in

order to mitigate the effects of actuator faults and ensure the safety of DCs, a novel fault tolerant part is considered in the MPC framework. At last, three experimental cases demonstrate the effectiveness and practicability of proposed controller.

A. Model predictive control

MPC is known as an advanced process control where the control actions are acquired by predicting the future states in a finite horizon and optimizing corresponding objective function iteratively [28]. It is this predictive characteristic based on system model that brings superior control performance [29]. Nevertheless, the ability of dealing with state and input constraints is another strength of MPC, and thus it is widely applied for practical industry [30]–[32].

First, we transform the proposed data-driven model to a general discrete form (Based upon the validated experiments, N_x is selected as 1, N_u is set as 0 in the data-driven model):

$$x_k = Ax_{k-1} + Bu_k + d, \tag{13}$$

where x is temperature, $u = [F_a, F_w]^{\mathrm{T}}$ represents the vector of air flow rate and water flow rate, A and B are the coefficient, and d denotes the disturbance. The main challenge of MPC is to calculate the inputs sequences through constrained optimization of cost function at each sampling time. The procedure could be formulated as below:

$$\begin{aligned} \min \quad & J = \sum_{k=1}^{M} (x_k - x_{ref})^{\mathrm{T}} \Upsilon (x_k - x_{ref}) + \Delta u_k^{\mathrm{T}} \Phi \Delta u_k, \\ \text{s.t.} \quad & x_k = Ax_{k-1} + Bu_k + d, \\ & u_{k,min} \le u_k \le u_{k,max}, \\ & \Delta u_{k,min} \le \Delta u_k \le \Delta u_{k,max}, \end{aligned} \tag{14}$$

where J means the cost function, M is the prediction horizon, $\Delta u_k = u_k - u_{k-1}$ indicates the change of inputs in each step, x_{ref} is the reference value of state, Υ and Φ are weighting parameters, $u_{k,min}$ represents the lower bound of inputs and $u_{k,max}$ is the upper bound of inputs.

The optimizing procedure above is just for one step. And at this time-step, once the optimization process yields a finite control sequence, the first control action in this sequence will be applied to the system. Then the same procedure would be repeated continuously with new measurements for the following sampling instant.

B. Fault tolerant control

As mentioned above, some actuator faults during the real industrial operation usually degrade the performance of our developed controller considerably. Therefore, fault tolerant control (FTC) is proposed in this work to compensate for the adverse influence induced by actuator faults, and further ensure the operational safety of DC system. Basically, FTC could be categorized into active FTC and passive FTC [33]–[35]. These two methods have different design philosophies, but have the same control objective [36]. Active FTC indicates that the designed controller could reorganize the control structure online according to the system failures. Besides, a fault diagnosis scheme is necessary in the active FTC to offer the real-time faults of the system [37], [38]. On the contrary, passive FTC is able to dispose of the system faults without any faults information and control structure adjustments [39]. This kind of controller is fixed and insensitive to failures because of the pre-designed redundancy and robustness, and thus it's named as passive FTC.

The actuators in this system are direct current fans and electrical motorized ball valve, which are utilized to control the air flow rate and water flow rate respectively. And based on the feedbacks from engineers, there are three common actuator faults including fans aging, fans failure and valve deactivation. Generally, it's difficult to deal with three different actuator faults simultaneously in a fixed control structure, therefore, active FTC is employed in this work instead of passive FTC. The detailed methodologies are shown as following.

*1) **Fault model**:* In order to model the actuator faults, a fault factor is proposed in the previous data-driven model:

$$x_k = Ax_{k-1} + B\rho_k u_k + d, \tag{15}$$

where $\rho = diag(\rho_a, \rho_w)$ is the fault factor vector which denotes the ratios between the real actuator responses and control commends from controller. If the actuator responses are equal to the control commends, it means the system is healthy and $\rho = I$; if the actuators lose functionality completely $\rho = 0$.

*2) **Fault identification**:* To compensate for the faults and work out appropriate control actions in real time, it is essential to do the fault identification. The detailed procedure of fault identification in this article is online calculation or estimation of fault factor. The formula of fault factor is presented as below:

$$\rho_i = \begin{cases} \frac{\nu_{act}}{\nu_{com}}, & \text{if } |\nu_{act} - \nu_{com}| \ge \delta \\ 1, & \text{if } |\nu_{act} - \nu_{com}| \le \delta, \end{cases} \tag{16}$$

in which ν_{act} means the real actuator action, and ν_{com} is defined as the control commend. δ represents the threshold for fault diagnosis which is determined by repeated experiments. In general cases of fault identification, the real actuator information are usually estimated through input observers such as Kalman filter, sliding mode observer, moving horizon estimation and so on. However, for real industrial applications, the customers have high demands of information accuracy and reliability, especially the input and output data. Consequently, several physical sensors are employed to measure the actuator actions in this program. Nevertheless, for the measurement of fan speed, it is quite convenient to use the fan's own internal hall effect sensor. And the water flow rate is measured through the system-provided flow meter. These two approaches of obtaining actuators information are all more reliable than input observers without increasing costs.

As we can see in Fig. 3, there are five fans in the cooling unit. Thus we need to identify five fault factors of fans at the same time and combine them together. The simple equation of fault factors is shown as follows.

$$\rho = \begin{bmatrix} \rho_a \\ \rho_w \end{bmatrix} = \begin{bmatrix} 20\%(\rho_{a,1} + \rho_{a,2} + \rho_{a,3} + \rho_{a,4} + \rho_{a,5}) \\ \rho_w \end{bmatrix}. \tag{17}$$

Here, for the healthy system, all the fault factors of fans $\rho_{a,i} = 1$, and then the final fault factors of fans $\rho_a = 1$. For one of the fans fails to work, the final fault factors of fans $\rho_a = 0.8$ and it means the performance of airflow actuator decreases by 20%.

*3) **Fault tolerant MPC**:* The fault factor would be identified in each sampling instant, and the state-space model is updated accordingly. Consequently, the problem of fault tolerant MPC could be reformulated as below:

$$\begin{aligned} \min \quad & J^* = \sum_{k=1}^{M} (x_k - x_{ref})^{\mathrm{T}} \Upsilon (x_k - x_{ref}) + \Delta u_k^{\mathrm{T}} \Phi \Delta u_k, \\ \text{s.t.} \quad & x_k = A x_{k-1} + B \rho_k u_k + d, \\ & u_{k,min} \le u_k \le u_{k,max}, \\ & \Delta u_{k,min} \le \Delta u_k \le \Delta u_{k,max}. \end{aligned} \tag{18}$$

Additionally, it is also noteworthy that the fault of valve deactivation would result in the change of cost function in fault tolerant MPC. When the fault of valve deactivation occurs, the water flow rate stays as a constant and cannot be controlled any more. Thus the multi-input-single-output (MISO) system changes to a single-input-single-output (SISO) system, and the cost function should be adjusted accordingly. The updated fault tolerant MPC problem is represented as:

$$\begin{aligned} \min \quad & J^{\#} = \sum_{k=1}^{M} (x_k - x_{ref})^{\mathrm{T}} \Upsilon (x_k - x_{ref}) + \Delta u_k^{*\mathrm{T}} \Phi \Delta u_k^*, \\ \text{s.t.} \quad & x_k = A x_{k-1} + B \rho_k u_k^* + d, \\ & u_{k,min}^* \le u_k^* \le u_{k,max}^*, \\ & \Delta u_{k,min}^* \le \Delta u_k^* \le \Delta u_{k,max}^*, \end{aligned} \tag{19}$$

where $u_k^* = [F_a, \nu_{act,w}]^{\mathrm{T}}$, and $\nu_{act,w}$ is the constant measurement of water flow rate under valve deactivation and no longer a control input. Finally, by applying a switch trigger and this updated cost function, the fault of valve deactivation is addressed. To clearly represent the procedure of fault tolerant MPC in this DC system, a flowchart is provided in Fig. 5.

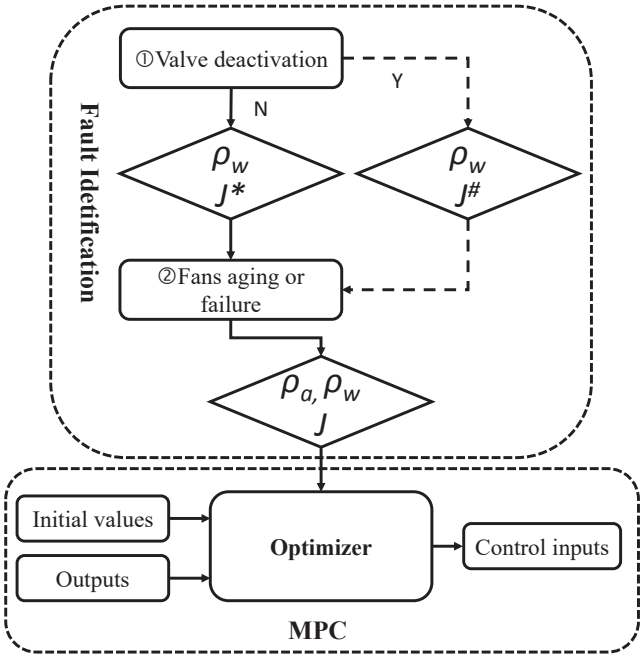

Fig. 5. Flowchart of the fault tolerant MPC algorithm in this work.

V. Experimental Results

All the experiments are implemented in a real single-rack D-C with rack-based cooling architecture which is shown in Fig. 6. The experimental DC system comprises 30 servers, cooling unit, several sensors, pipes and chilled water supply system from the building. The control algorithm is programmed in Python environment and conducted through Raspberry Pi Zero which is regarded as the master micro-controller in this work. Besides, considering reducing the computational burden of Raspberry Pi, an Arduino board (Nano) is used as the slave micro-controller whose function is to send actuators commends of Raspberry Pi and collect data from sensors. Additionally, the IT workload in this experiment is set as a constant value.

In order to obtain an efficient data-driven model, the experiment under all working condition is carried out and the relevant measurements are utilized for model training. The

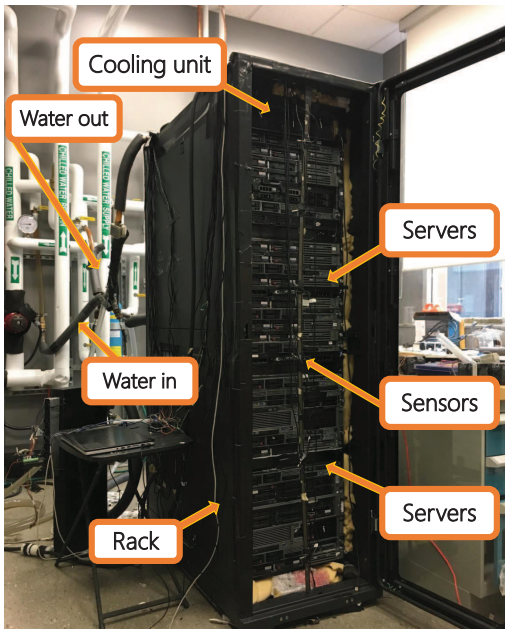

Fig. 6. Experimental DC with rack-based cooling architecture.

experiment costs nearly 80 hours and includes 10 cycles covering different air flow, water flow and IT workload. The experimental cycle is represented in Fig. 7. Then a group of data from another experiment is used to validate the performance of data-driven model, and the result of validation is depicted in Fig. 8. As we can see in this figure, the error of proposed model is roughly around 0.2 degree, which is within the acceptable boundary and further demonstrates the data-driven model is able to describe the DC system.

As mentioned above, three kinds of actuator faults com-

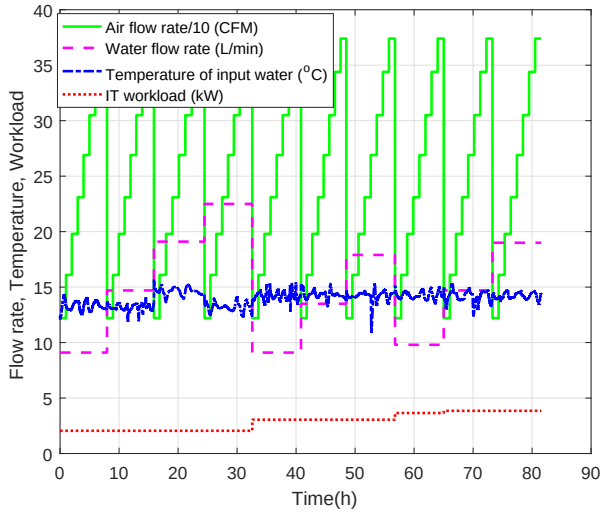

Fig. 7. Measurements under all working condition for model training.

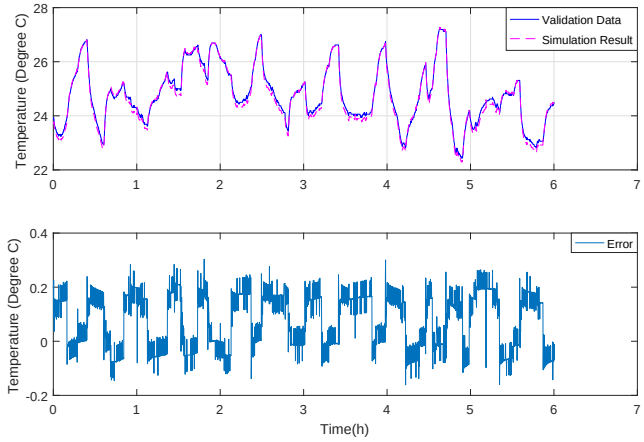

Fig. 8. Validation result of data-driven model.

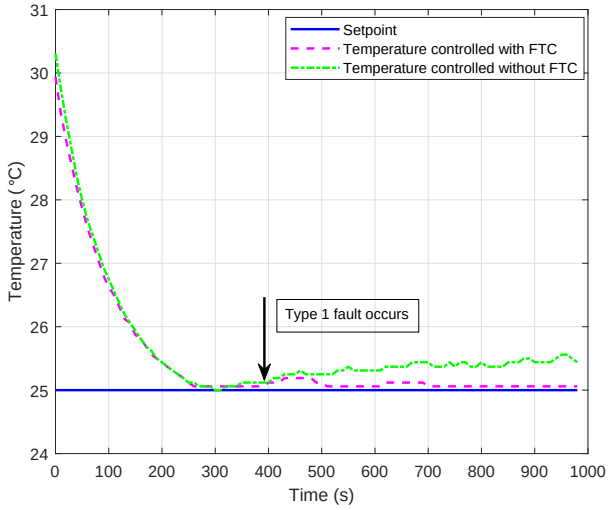

Fig. 9. The comparative performance of data-driven predictive controller with fault tolerant control and without fault tolerant control in case I.

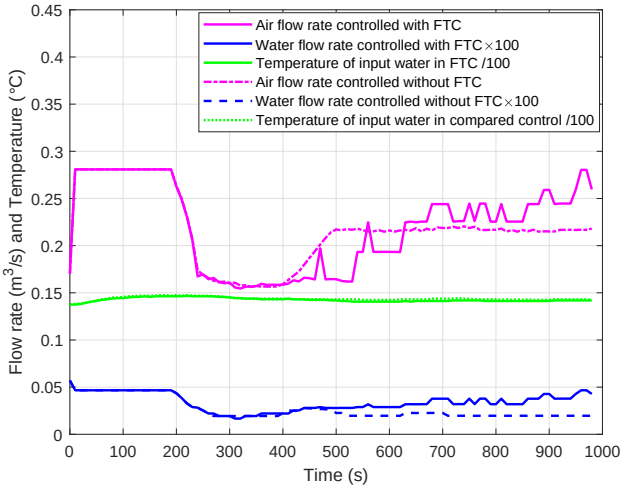

Fig. 10. Control inputs and measured disturbance in case I.

prising aging of fans, failure of fans and valve deactivation are investigated in this paper. Therefore, three groups of experiment concerning the actuator faults are carried out to test and verify the effectiveness of designed controller. The detailed experiment results are presented as follows.

A. Case I: Aging of fans

The first type of actuator faults is aging of fans, which indicates the performance of total fans degrade by a certain percent over time. In this experiment, the fault is injected at 400s and the fans' performance affected by fault is assumed to degrade by 30 percent. The comparison of results between data-driven predictive controller with fault tolerant control and without fault tolerant control are depicted in Fig. 9. Moreover, the control inputs and measured disturbance are shown in Fig. 10 as well.

As seen in Fig. 9, temperatures regulated by both of the controllers are able to track the set point properly before fault occurs, which represents the good predictive capability of data-driven model and control accuracy of developed predictive controller. When fault is injected at 400s, the data-driven predictive controller without fault tolerant control perform badly and the temperature in the rack keeps rising. On the contrary, the proposed controller with fault tolerant control works properly under fault and the temperature could still follow the set point well. Besides, the control inputs after fault occurrence described in Fig. 10 also prove the superiority of fault tolerant predictive control.

B. Case II: Failure of fans

The second type of actuator faults means that one or some of the fans stop working suddenly in cooling unit, which is one of the major faults in fans. As the other fans still work normally under this situation, the temperature in the system could be regulated by some proper control algorithms. In this case, we assume that one of the fans stops working after the system reaches the steady state and the fault is injected

at a specific point-in-time through the code in advance. The comparative experiments of data-driven predictive controller with fault tolerant control and without fault tolerant control are conducted respectively and the results are described in Fig. 11. Besides, the control inputs are represented in Fig. 12 as well.

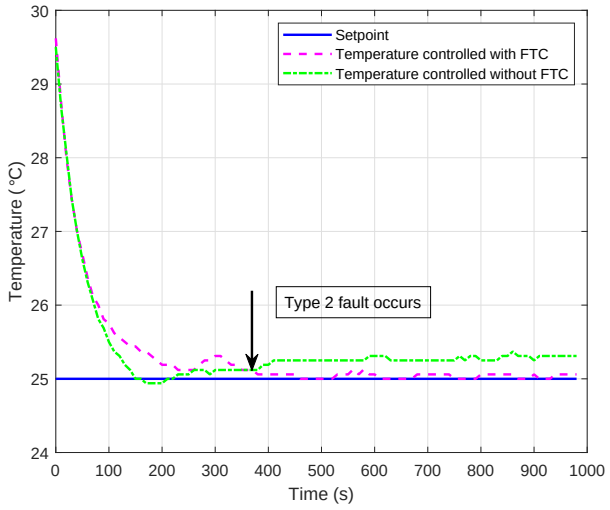

Fig. 11. The comparative performance of data-driven predictive controller with fault tolerant control and without fault tolerant control in case II.

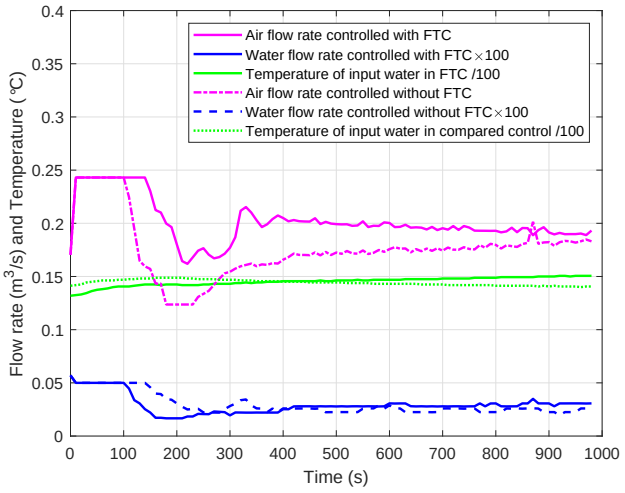

Fig. 12. Control inputs and measured disturbance in case II.

In Fig. 11, the effectiveness of controller with fault tolerant control is satisfying after the injection of fault, while the controller without fault tolerant control shows a bad adaptability to the actuator fault. Furthermore, the differences of control commends shown in Fig. 12 express the internal reason why the controller with fault tolerant control performs better. It is also noteworthy that the control performance is quite sensitive to the change of input water temperature and the temperature fluctuates greatly at the beginning period, thus the controlled temperatures in the rack overshoot and oscillate around the set point.

C. Case III: Valve deactivation

The third common actuator fault in cooling unit is the valve deactivation. The valve fault is also injected around 400s and the pre-arranged stopped water flow rate is set as 0.000333 m^3/s or $20L/min$ (it can also be set as another value). The comparison of effects between two controllers is presented in Fig. 13 and corresponding control commends are shown in Fig. 14.

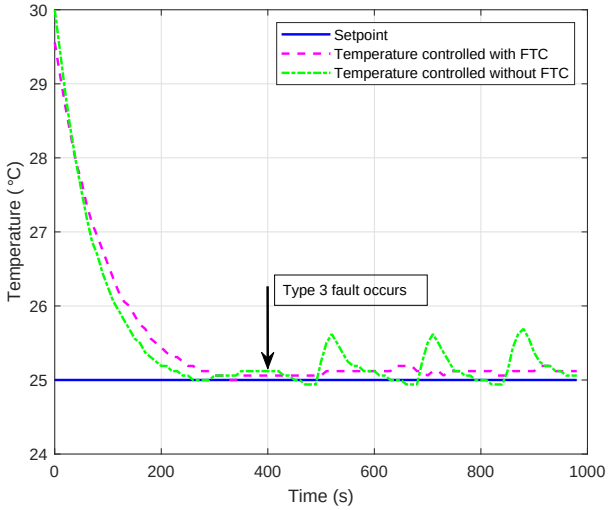

Fig. 13. The comparative performance of data-driven predictive controller with fault tolerant control and without fault tolerant control in case III.

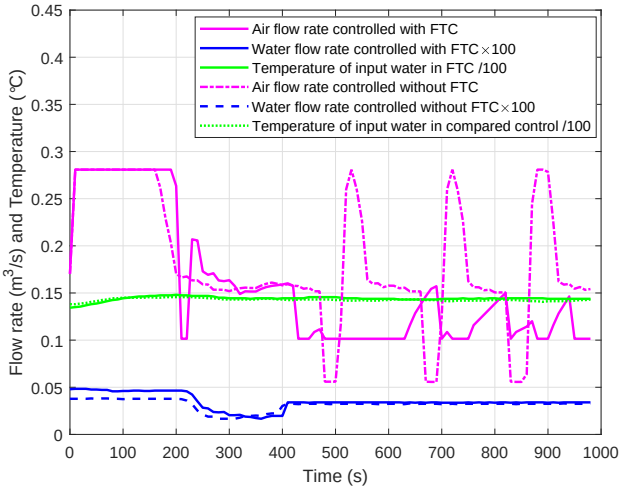

Fig. 14. Control inputs and measured disturbance in case III.

As the Fig. 13 describes, the temperatures could follow the set point precisely, and there aren't obvious overshoot and undershoot before the the injection of fault. When the valve fault occurs, the temperature managed through controller

without fault tolerant control appears a big oscillation due to the change of control structure. In addition, the control input of controller without fault tolerant control varies significantly as well. However, the behavior of controller with fault tolerant control is absolutely excellent even under the occurrence of fault owing to the timely adjustment of cost function in control algorithm.

As we can see in the three cases of experiments, the data-driven fault tolerant predictive controller behaves quite well in the respects of response time, fault tolerance, overshoot and undershoot. Nevertheless, the comparison of predictive controller without fault tolerant control further demonstrate the remarkable performance of developed controller.

VI. Conclusion

A systematical literature review is presented in this paper on recent data-driven techniques and the corresponding applications in diesel engine and aftertreatment system. Detailed introduction is divided into several aspects including data-driven modeling, data-driven estimation and fault diagnosis, and data-driven control. Furthermore, the latest applications of data-driven approaches in diesel engine and aftertreatment are summarized respectively. Finally, prospective trends, challenges and chances concerning this topic are described for the future research. Although the applications of data-driven technique in diesel engine and aftertreatment system have become a promising issue lately, there are still some unsolved and unpredictable problems for the future work. Therefore, this review is composed to figure out progresses in the field, appeal more discussions on extensions and boost relative researches.

References

[1] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew. Sustain Energy Rev.*, vol. 31, pp. 622–638, Mar. 2014.

[2] H. Cheung, S. Wang, C. Zhuang, and J. Gu, "A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation," *Appl. energy*, vol. 222, pp. 329–342, Jul. 2018.

[3] H. Moazamigoodarzi, P. J. Tsai, S. Pal, S. Ghosh, and I. K. Puri, "Influence of cooling architecture on data center power consumption," *Energy*, vol. 183, pp. 525–535, Sep. 2019.

[4] T. Lu, X. Lü, M. Remes, and M. Viljanen, "Investigation of air management and energy performance in a data center in finland: Case study," *Energy and Buildings*, vol. 43, no. 12, pp. 3360–3372, Dec. 2011.

[5] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in IT server enclosures," *Int. J. Heat Mass Transf.*, vol. 127, pp. 890–900, Dec. 2018.

[6] T. D. Boucher, D. M. Auslander, C. E. Bash, C. C. Federspiel, and C. D. Patel, "Viability of dynamic cooling control in a data center environment," *J. Electron. Packag.*, vol. 128, no. 2, pp. 137–144, Nov. 2006.

[7] H. Zhang, S. Shao, H. Xu, H. Zou, and C. Tian, "Free cooling of data centers: A review," *Renew. Sustain Energy Rev.*, vol. 35, pp. 171–182, Jul. 2014.

[8] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," *Computer Networks*, vol. 101, pp. 104–112, Jun. 2016.

[9] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proc. IEEE*, vol. 100, no. 1, pp. 254–268, Jan. 2012.

[10] B. Durand-Estebe, C. Le Bot, J. N. Mancos, and E. Arquis, "Data center optimization using PID regulation in CFD simulations," *Energy and Buildings*, vol. 66, pp. 154–164, Nov. 2013.

[11] J. Deng, L. Yang, X. Cheng, and W. Liu, "Self-tuning PID-type fuzzy adaptive control for CRAC in datacenters," in *International Conference on Computer and Computing Technologies in Agriculture*, pp. 215–225, 2013.

[12] R. Zhou, Z. Wang, C. E. Bash, and A. McReynolds, "Modeling and control for cooling management of data centers with hot aisle containment," in *Proceedings of ASME 2011 International Mechanical Engineering Congress & Exposition*, 2011.

[13] M. Ogawa, H. Fukuda, H. Kodama, H. Endo, T. Sugimoto, T. Kasajima, and M. Kondo, "Development of a cooling control system for data centers utilizing indirect fresh air based on model predictive control," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 132–137, 2015.

[14] L. Parolini, B. Sinopoli, and B. H. Krogh, "Model predictive control of data centers in the smart grid scenario," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10 505–10 510, Jan. 2011.

[15] T. Ding, Z. guang He, T. Hao, and Z. Li, "Application of separated heat pipe system in data center cooling," *Appl. Therm. Eng.*, vol. 109, pp. 207–216, Oct. 2016.

[16] B. Corbett and P. Mhaskar, "Subspace identification for data-driven modeling and quality control of batch processes," *AIChE J.*, vol. 62, no. 5, pp. 1581–1601, Jan. 2016.

[17] S. Aumi and P. Mhaskar, "Integrating data-based modeling and nonlinear control tools for batch process control," *AIChE journal*, vol. 58, no. 7, pp. 2105–2119, Jul. 2012.

[18] S. Aumi and P. Mhaskar, "An adaptive data-based modeling approach for predictive control of batch systems," *Chem. Eng. Sci.*, vol. 91, pp. 11–21, Mar. 2013.

[19] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005.

[20] D. Jin, X. Bai, and Y. Wang, "Integrating structural symmetry and local homoplasy information in intuitionistic fuzzy clustering for infrared pedestrian segmentation," *IEEE Trans. Syst., Man, Cybern., Syst.*, 2019.

[21] A. Siam, C. Brandon, and M. Prashant, "Data-based modeling and control of nylon-6, 6 batch polymerization," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 94–106, Jan. 2013.

[22] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014.

[23] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[24] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic linear model predictive control with chance constraints–a review," *J. Process Control*, vol. 44, pp. 53–67, Aug. 2016.

[25] T. Gao, S. Yin, J. Qiu, H. Gao, and O. Kaynak, "A partial least squares aided intelligent model predictive control approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 11, pp. 2013–2021, Nov. 2018.

[26] M. Cannon, "Efficient nonlinear model predictive control algorithms," *Annu. Rev. Control*, vol. 28, no. 2, pp. 229–237, Jan. 2004.

[27] S. Shi, "Rack-based data center temperature regulation using data-driven model predictive control," Master's thesis, McMaster University, 2019.

[28] D. Laurí, J. A. Rossiter, J. Sanchis, and M. Martínez, "Data-driven latent-variable model-based predictive control for continuous processes," *J. Process Control*, vol. 20, no. 10, pp. 1207–1219, Dec. 2010.

[29] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven MPC design," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 577–582, Jan. 2019.

[30] M. A. Evans, M. Cannon, and B. Kouvaritakis, "Robust MPC tower damping for variable speed wind turbines," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 290–296, Jul. 2014.

[31] B. Sakhdari and N. L. Azad, "Adaptive tube-based nonlinear MPC for economic autonomous cruise control of plug-in hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 11 390–11 401, Dec. 2018.

[32] G. P. Incremona, A. Ferrara, and L. Magni, "MPC for robot manipulators with integral sliding modes generation," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1299–1307, Jun. 2017.

[33] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annu. Rev. Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008.

[34] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May. 2013.

[35] J. Lan and R. J. Patton, "A new strategy for integration of fault estimation within fault-tolerant control," *Automatica*, vol. 69, pp. 48–59, Jul. 2016.

[36] J. Jiang and X. Yu, "Fault-tolerant control systems: A comparative study between active and passive approaches," *Annu. Rev. Control*, vol. 36, no. 1, pp. 60–72, Apr. 2012.
[37] R. Wang and J. Wang, "Fault-tolerant control with active fault diagnosis for four-wheel independently driven electric ground vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, pp. 4276–4287, Nov. 2011.
[38] J. Wang, J. Zhang, B. Qu, H. Wu, and J. Zhou, "Unified architecture of active fault detection and partial active fault-tolerant control for incipient faults," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1688–1700, Jul. 2017.
[39] H. Niemann and J. Stoustrup, "Passive fault tolerant control of a double inverted pendulum-a case study," *Control Eng. Pract.*, vol. 13, no. 8, pp. 1047–1059, Aug. 2005.