TRANSIENT REDUCED-ORDER CONVECTIVE HEAT TRANSFER MODELING FOR A DATA CENTER

A Ph.D. Dissertation
Presented to
The Academic Faculty

by

Rajat Ghosh

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2013

TRANSIENT REDUCED-ORDER CONVECTIVE HEAT TRANSFER MODELING FOR A DATA CENTER

Approved by:

Dr. Yogendra Joshi, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. S. Mostafa Ghiaasiaan
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Satish Kumar
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Karsten Schwan
College of Computing
Georgia Institute of Technology

Dr. Roshan Vengazhiyil
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: Nov. 14, 2013

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor Prof. Yogendra Joshi for his unrelenting support and insightful guidance during my doctoral studies. I would also like to extend my deepest gratitude to my dissertation committee, comprising Prof. Karsten Schwan, Prof. S. Mostafa Ghiaasiaan, Prof. Satish Kumar, and Prof. Roshan Vengazhiyil, for their valuable comments on my research. I am really thankful to all my present and former colleagues in the CEETHERM Data Center Laboratory and the Microelectronic and Emerging Technologies Thermal Laboratory (METTL) for the valuable technical inputs on my dissertation. Particularly helpful to me during this time were Vikneshan Sundaralingam, Junwei Li, Dr. Emad Samadiani, Dr. Vaibhav Arghode, and Dr. Pramod Kumar.

I would like to acknowledge the assistance I received from Woodruff school academic professional including Dr. Wayne Whiteman, Dr. Jeffrey Donnell, Ms. Glenda Johnson, Ms. Trudy Allen, and Ms. Regina Neequaye. Special thanks to several Georgia Tech services such as Strigerette transportation, Library, and Communication Center.

Finally, I am deeply indebted to all my former teachers at Georgia Tech, IIT Kharagpur, St. Xavier’s College, Kolkata, and Sanskrit Collegiate School, Kolkata for their dedicated effort and attention in educating me. Last but not the least; I am extremely grateful to my parents and my sister for their sacrifices and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

List of Symbols[1]

q_{CRAC}	Heat transfer from CRAC unit
$\varepsilon_{\mathrm{CRAC}}$	CRAC effectiveness
C	Specific heat
$T_{\mathrm{air,hot}}$	Air temperature in CRAC return side
$T_{\mathrm{air,cold}}$	Air temperature in CRAC supply side
$T_{\mathrm{CW,cold}}$	Chilled water temperature in CRAC inlet side
$T_{\mathrm{CW,hot}}$	Chilled water temperature in CRAC outlet side
$\dot{m}_{air}$	Mass flowrate of air
$\dot{m}_{CW}$	Mass flowrate of chilled water
ΔP	Pressure drop
$T_{\mathrm{CW,cold}}$	Chilled water temperature in CRAC inlet side
f	Moody friction factor
L	Tube length
D	Tube hydraulic diameter
ρ	Density of water
A	Tube cross-section area
η	Pump efficiency
$\dot{E}$	Pump power

[1] Unless specified otherwise, all units are in S.I systems.

W_{chiller}	Chiller compressor work input
Q_{chiller}	Chiller heat load
Q_{DC}	Data center compute heat load
Q_{BCW}	Building chiller pump heat load
Q_{CRAC}	CRAC heat load
q_{CT}	Heat transfer from cooling tower
h_{ao}	Enthalpy of exhaust air in cooling tower
h_{ai}	Enthalpy of inlet air in cooling tower
ρ_{air}	Density of air
h_{swi}	Enthalpy of saturated air
T_i	Parametric temperature data
T_0	Parametric-average of data
T_i^*	Parametric component of data
In	Independent variable
D_i	Dependent parametric variable
ψ	POD modes
R	Covariance matrix
u	Eigenvector of R
λ	Eigenvalue of R
σ	Singular value
k	Number of retained POD modes
B	Coefficient matrix
b	POD coefficients

T_{data}	Temperature data
T_{POD}	POD-based temperature prediction
$E_{\text{Prediction}}$	Prediction Error
$E_{\text{Analytical}}$	Analytical Error
$\Delta T_{\text{Scale}}^{\text{Measurement}}$	Temperature measurement scale
c_0	Arbitrary constant for a priori error of POD/Interpolation scheme
(c_1, c_2, c_3, c_4)	Arbitrary constants for a priori error of POD/Extrapolation scheme
μ	Mean
σ	Standard deviation
I	Unified decision-making index
t	Time
(x, y, z)	Spatial location
$\overline{Q}$	Normalized heat load
$\overline{t}$	Normalized time
Q_{en}	Heat load ensemble
t_{en}	Time ensemble
Q_{int}	Interrogation heat load
t_{int}	Interrogation time

List of Abbreviations

DC	Data center
POD	Proper orthogonal decomposition
CDC	Cloud data center
CRAC	Computer room air conditioning unit
RDHx	Rear door heat exchanger
PDU	Power distribution unit
UPS	Uninterrupted power supply
NTU	Number of transfer unit
COP	Coefficient of performance
CT	Cooling tower
BCW	Building chilled water
CFD	Computational fluid mechanics
C.E.P.	Captured energy percentage
IT	Information technology

SUMMARY

A measurement-based reduced-order heat transfer modeling framework is developed to optimize cooling costs of dynamic and virtualized data centers. The reduced-order model is based on a proper orthogonal decomposition-based model order reduction technique. For data center heat transfer modeling, the framework simulates air temperatures and CPU temperatures as a parametric response surface with different cooling infrastructure design variables as the input parameters. The parametric framework enables an efficient design optimization tool and is used to solve several important problems related to energy-efficient thermal design of data centers.

The first of these problems is about determining optimal response time during emergencies such as power outages in data centers. To solve this problem, transient air temperatures are modeled with time as a parameter. This parametric prediction framework is useful as a near-real-time thermal prognostic tool.

The second problem pertains to reducing temperature monitoring cost in data centers. To solve this problem, transient air temperatures are modeled with spatial location as the parameter. This parametric model improves spatial resolution of measured temperature data and thereby reduces sensor requisition for transient temperature monitoring in data centers.

The third problem is related to determining optimal cooling set points in response to dynamically-evolving heat loads in a data center. To solve this problem, transient air temperatures are modeled with heat load and time as the parameters. This modeling framework is particularly suitable for life-cycle design of data center cooling infrastructure.

The last problem is related to determining optimal cooling set points in response to dynamically-evolving computing workload in a virtualized data center. To solve this problem, transient CPU temperatures under a given computing load profile are modeled with cooling resource set-points as the parameters.

CHAPTER 1
INTRODUCTION

The goal of this dissertation is to develop a measurement-based parametric model for rapid assessment of data center temperatures to optimize energy usage during dynamic events which are often triggered by time-varying computing loads, fluctuating cooling resource allocations, and power outages.

Background and Motivation

With economic and societal shift from paper-based to digital information management, data centers (DCs)—computing infrastructure facilities that contain equipment used for data processing, data storage, communication, and networking—have become an indispensable cyber-physical system for e-commerce, communication, trading, and other daily activities. Triggered by increasing demand for data processing and storage, the DC industry has been growing rapidly over last decade. This demand is driven by several factors, including but not limited to:

- Growth of internet communication and entertainment
- Growth of e-commerce, online banking, and electronic trading
- Shift from paper-based to electronic record storage
- Adoption of satellite navigation and electronic shipment tracking
- High performance scientific computing

During the past decade, increasing growth in DC operations has led to significant growth in DC energy usage. In fact, DCs are consuming more than 2% of world electricity production [1-3]. The increase in energy usage has several important implications, including:

- Increased energy bills for business and government
- Increased greenhouse gas emission
- Increased capital cost for expansion of DC capacity and construction of new DCs
- Increased strain on the existing power grid to meet increased electricity demand

To avoid these adverse consequences, there is a growing interest in exploring opportunities for improving DC energy efficiency. Figure 1 shows the schematic layout of a typical raised-floor data center. Data center components can be divided into three categories: data-processing IT equipment such as volume servers, network modules, routers, storage disks; cooling hardware systems such as computer room air conditioning (CRAC) units, read door heat exchanger (RDHx) units, and power conversion hardware such as uninterrupted power supply (UPS), power distribution unit (PDU).

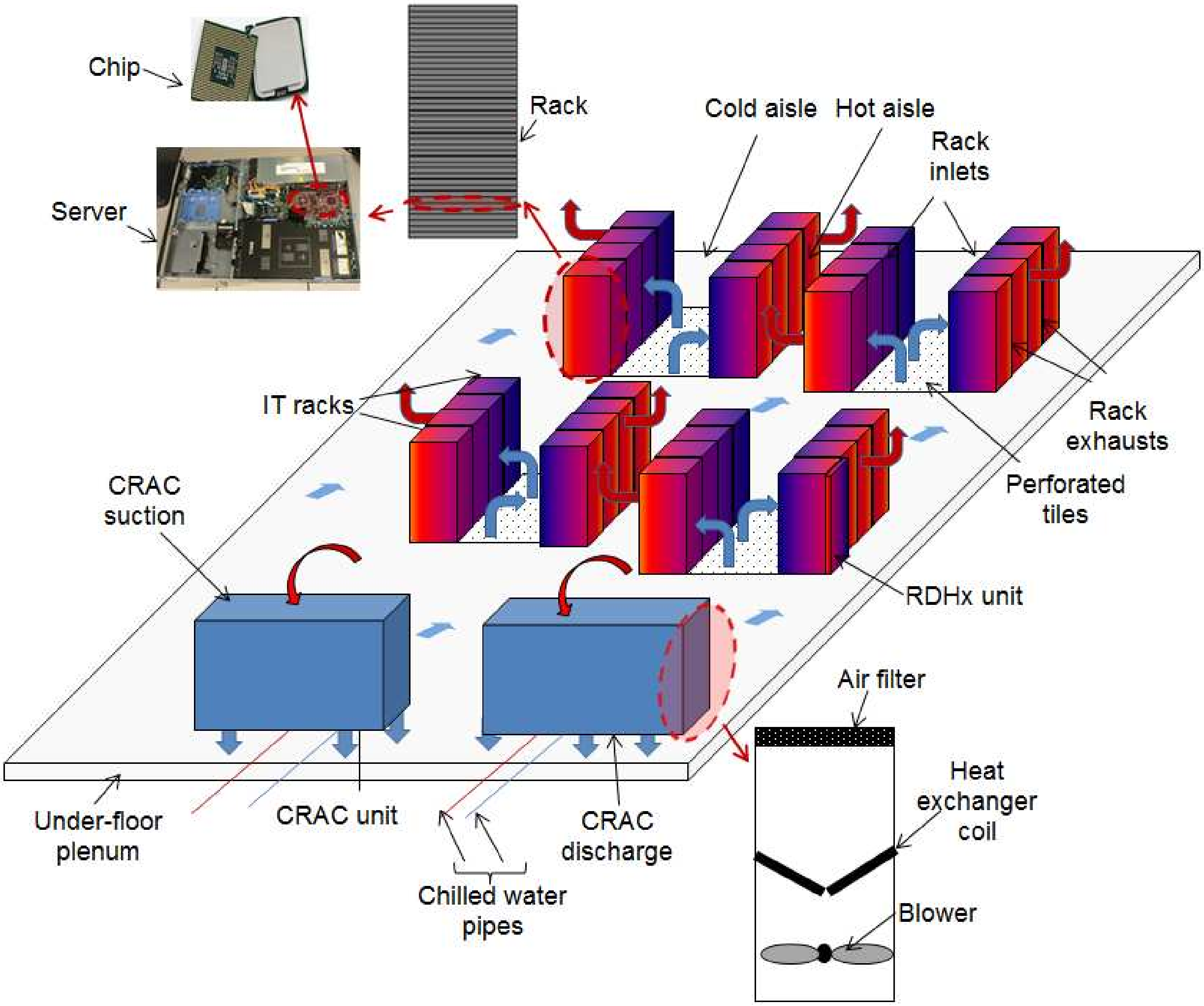


Figure 1: Schematic representation of a raised-floor data center layout

Except for some transmission loss, data center energy is consumed by its IT equipment or by its cooling infrastructure, shown in Figure 2. In fact, benchmarking studies [4] reveal that the energy required for DC cooling is 30-40% of the overall data center energy budget.

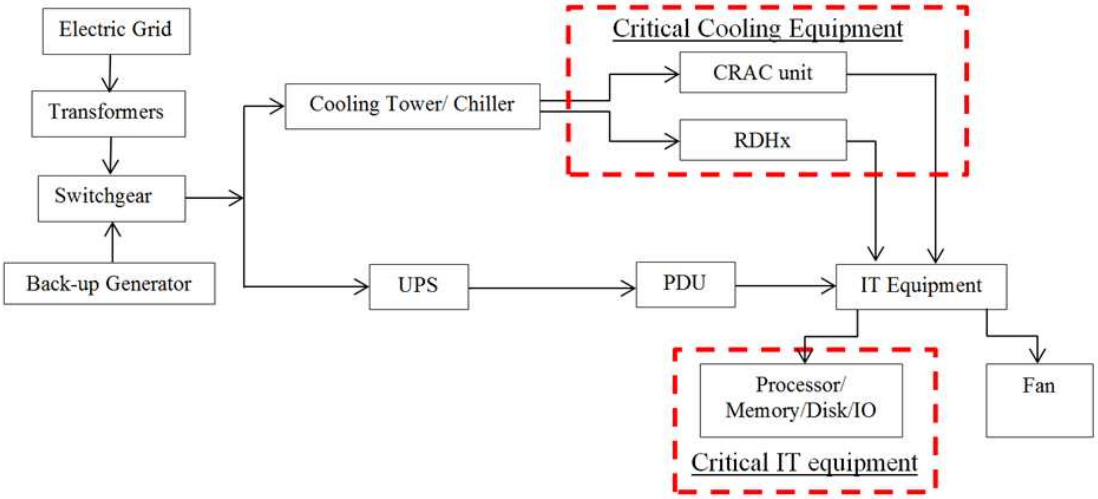


Figure 2: Power flow diagram for a typical data center

Table 1 shows peak component power consumption for a typical server in data center. A typical large scale internet data center (IDC) houses thousands such 1-U servers, driving data center electricity consumption to a few MW. In fact, data centers in the United States consumed more than 80 billion kWh electricity in 2011. Because the electrical energy consumed in a data center is released in the form of heat, cooling contributes a major fraction (30%-55%) of the total data center energy consumption.

Table 1: Component peak power consumption for a typical 1-U server [5]

Components	**Peak Power (W)**
CPU	80
Memory	36
Disk	12
Peripherals	50
Motherboard	25
Fan	10
PSU losses	38
Total	251

Typical DC cooling hardware systems are cooling tower(s), building chiller(s), chilled water pumps, computer room air conditioning (CRAC) unit blowers, rear door heat exchangers (RDHx), and server fans. Figure 3 shows the cascade refrigeration cycle used in a data center. Data center cooling takes place in three levels: first at the CRAC level and the RDHx level, then at the refrigeration chiller level, and finally at the cooling tower level. A CRAC is basically an air-to-water cross flow heat exchanger. The liquid side of the heat exchanger is coupled with the evaporator of a vapor compression cycle inside the building chiller. The condenser of the chiller unit is coupled with the liquid side of the cooling tower. The cooling tower is basically a ducted water-to-liquid heat exchanger which removes heat from warm water, taking heat from the chiller condenser to the environment. Figure 4 shows a typical chip-to-environment heat flow diagram for a typical air-cooled DC. The energy-absorbing components of this cooling scheme are server fans, CRAC blowers, building chiller pumps, chiller compressor, cooling tower pumps, and cooling tower blowers. Figure 5 shows the percentage break-up of the overall cooling energy to various cooling hardware. It shows that top energy-intensive cooling hardware units are the refrigeration chiller (46%) and the CRAC blower (28%). On the other hand, server fans consume 8% of overall DC cooling energy.

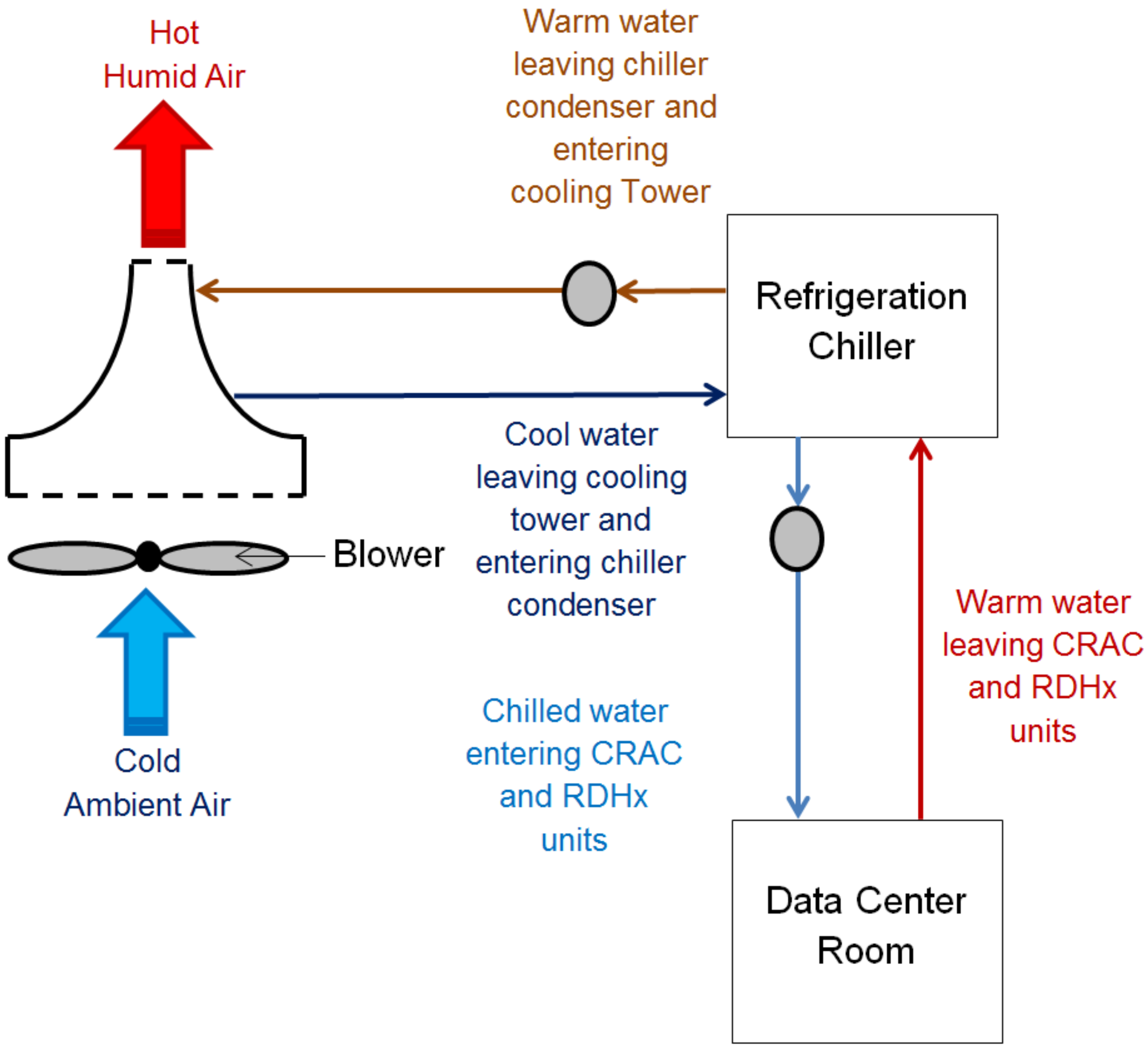

Hot
Humid Air
Warm water leaving chiller condenser and entering cooling Tower
Refrigeration Chiller
Cool water leaving cooling tower and entering chiller condenser
Blower
Cold
Ambient Air
Chilled water entering CRAC and RDHx units
Warm water leaving CRAC and RDHx units
Data Center Room

Figure 3: Cascade refrigeration cycle for data center cooling

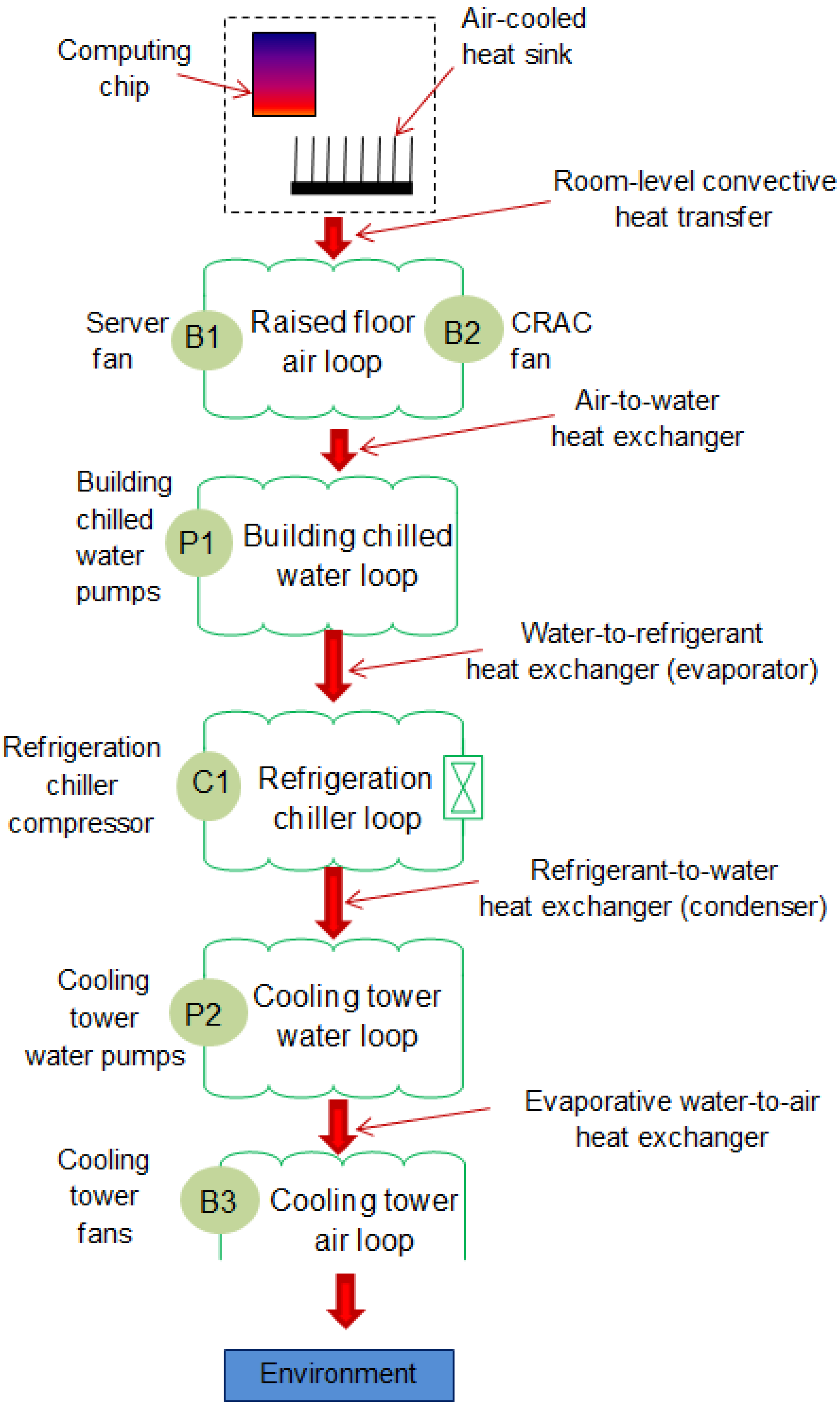

Computing chip
Air-cooled heat sink
Room-level convective heat transfer
Server fan
B1
Raised floor air loop
B2
CRAC fan
Air-to-water heat exchanger
Building chilled water pumps
P1
Building chilled water loop
Water-to-refrigerant heat exchanger (evaporator)
Refrigeration chiller compressor
C1
Refrigeration chiller loop
Refrigerant-to-water heat exchanger (condenser)
Cooling tower water pumps
P2
Cooling tower water loop
Evaporative water-to-air heat exchanger
Cooling tower fans
B3
Cooling tower air loop
Environment

Figure 4: Data center heat transfer model

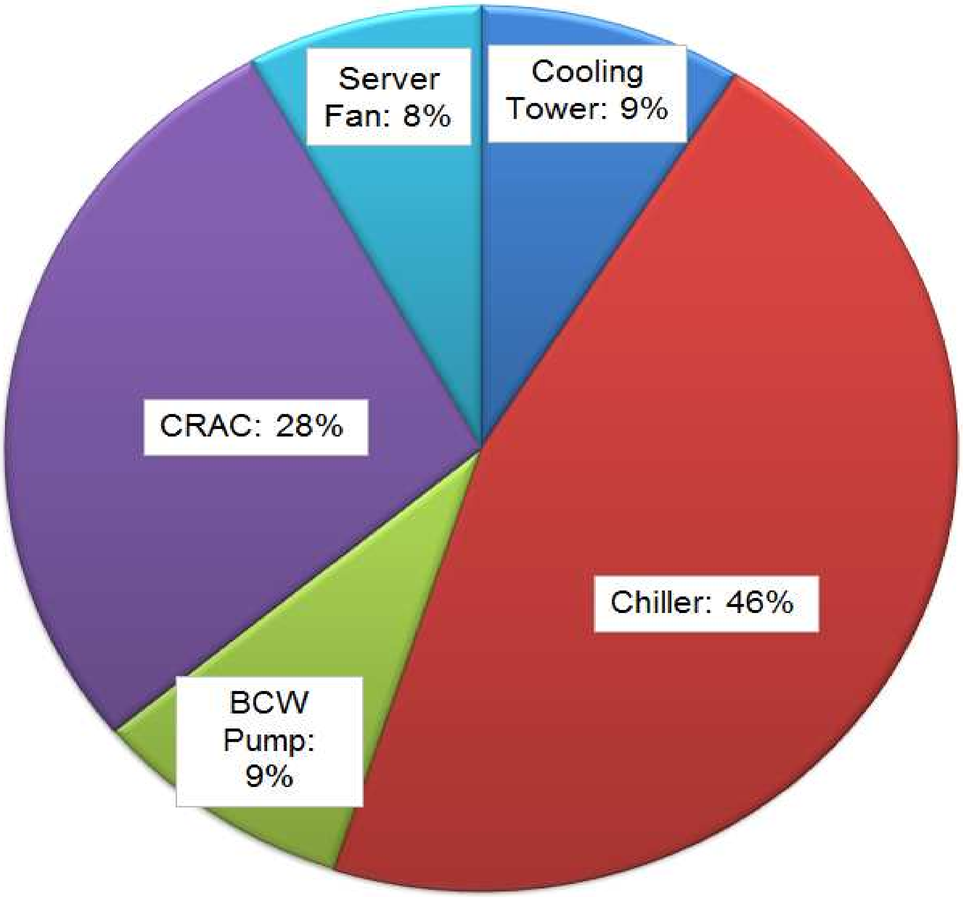


Figure 5: Percentages of cooling energy spent on various cooling hardware [6]

The CRAC unit is basically a cross-flow air-to-liquid heat exchanger. CRAC cooling is controlled by the proportional-integral-derivative (PID) controller. Often, CRAC supply air temperature or return air temperature is used as the control variable. Figure 6 shows CRAC heat transfer model. The heat transfer from hot air to chilled water is given by:

$$\begin{gathered} q_{\mathrm{CRAC}} = \varepsilon_{\mathrm{CRAC}} \times C_{\min} \times \left(T_{\mathrm{air,hot}} - T_{\mathrm{CW,cold}}\right), \\ C_{\min} = C_{air}. \end{gathered} \tag{1.1}$$

The formulation in Eq. (1.1) can be fully specified by:

$$\left.\begin{array}{l} \varepsilon_{\text{CRAC}} = \left(1/C_r\right)\times\left(1-e^{-C_r\times A}\right), \\ \left.\begin{array}{l} A=\left(1-e^{-\text{NTU}_{\text{CRAC}}}\right), \\ C_r = C_{\min}/C_{\max}, \\ NTU_{\text{CRAC}} = UA_{\text{CRAC}}/C_{\min}. \end{array}\right\} \end{array}\right\} \tag{1.2}$$

The work done by chilled water pump can be derived by the thermodynamic analyses:

$$\begin{gathered} \dot{m}_{air}c_p\left(T_{air,hot}-T_{air,cold}\right)=\dot{m}_{CW}C\Delta T_{CW} \\ \Rightarrow \dot{m}_{CW}=\frac{\dot{m}_{air}c_p\left(T_{air,hot}-T_{air,cold}\right)}{C\Delta T_{CW}} \\ \Rightarrow T_{air,cold}\uparrow\Leftrightarrow\dot{m}_{CW}\downarrow. \end{gathered} \tag{1.3}$$

It means as the supply temperature set-point increases, the building chilled water flow rate decreases. Physically, it makes sense because higher supply temperature means less thermodynamic work by the cascade refrigeration cycle.

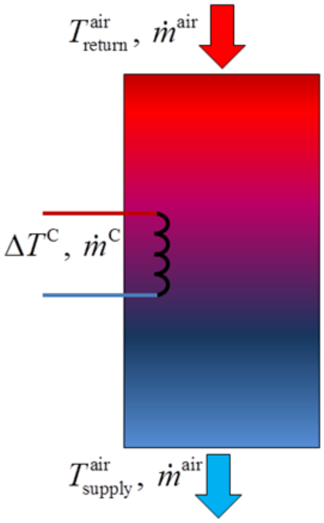

$T_{\text{return}}^{\text{air}}$, $\dot{m}^{\text{air}}$
ΔT^{C}, $\dot{m}^{\text{C}}$
$T_{\text{supply}}^{\text{air}}$, $\dot{m}^{\text{air}}$

Figure 6: CRAC heat transfer model

The pressure drop across the liquid side of the CRAC unit changes quadratically with flowrate:

$$\Delta P = f\,\frac{L}{D}\frac{\dot{m}_{CW}{}^{2}}{2\rho A^{2}}, \tag{1.4}$$

the friction factor, f can be determined from the Moody's chart.

The pump work is given by:

$$\left.\begin{aligned} \dot{E} = \frac{\Delta P \dot{m}_{CW}}{\eta\rho} = f\,\frac{L}{2A^{2}\eta D\rho^{2}}\left(C_{1} - C_{2}T_{\text{air,cold}}\right)^{3} \\ \Rightarrow \dot{E} \propto \left(1 - C_{3}T_{\text{air,cold}}\right)^{3}. \end{aligned}\right\} \tag{1.5}$$

RDHx units are shell-and-tube heat exchangers fitted at the rear end of the computing racks. They precool hot exhaust air from the server rack before it returns back to the CRAC unit. The cooling provided by RDHx units is controlled by the pressure differential in the liquid side. Figure 7 shows the RDHx heat transfer model.

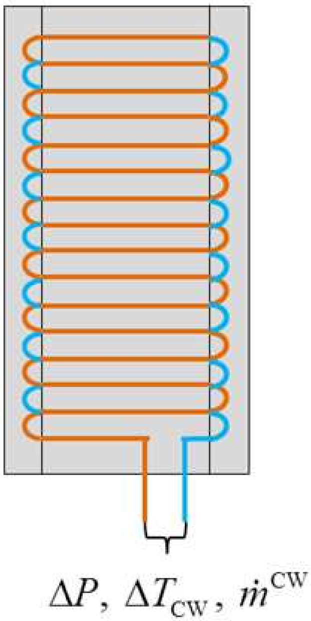


Figure 7: RDHx heat transfer model

The system-level energy balance for an RDHx unit gives:

$$\dot{m}_{CW} = \frac{\dot{m}_{air} c_p}{C \Delta T_{CW}} \left(T_{air,in} - T_{air,out} \right). \tag{1.6}$$

On the other hand, the flowrate is proportional to the pressure differential: $\dot{m}_{CW} \propto \Delta P$. Therefore, the pump work:

$$\left. \begin{aligned} \dot{E} = \frac{\Delta P \dot{m}_{CW}}{\eta \rho} = \left(\frac{2DA^2}{\rho f L \eta^2} \right)^{0.5} \left(\Delta P \right)^{1.5} \\ \Rightarrow \dot{E} \propto \left(\Delta P \right)^{1.5}. \end{aligned} \right\} \tag{1.7}$$

The RDHx pump work increases with (3/2) power of the pressure difference set-point.

The refrigeration chiller is the most important cooling hardware in terms of energy consumption. It acts as a hub integrating cooling hardware systems in the data processing room (coupled with the evaporator side) and the cooling tower (coupled with the condenser side). Figure 8 shows a schematic of the cascade refrigeration cycle used for DC cooling. The purpose of this cycle is to extract waste heat from the DC room and dump it to the cooling tower. The associated cost to run this cycle is compressor power input. The compressor power can be expressed as the ratio of chiller heat load to COP (coefficient of performance of chiller):

$$W_{\text{chiller}} = \left(Q_{\text{chiller}} / \text{COP} \right). \tag{1.8}$$

The chiller heat load is basically the sum of the DC room heat load, the CRAC blower power consumption, and the building chiller power consumption:

$$Q_{\text{chiller}} = Q_{DC} + Q_{BCW} + Q_{CRAC}. \tag{1.9}$$

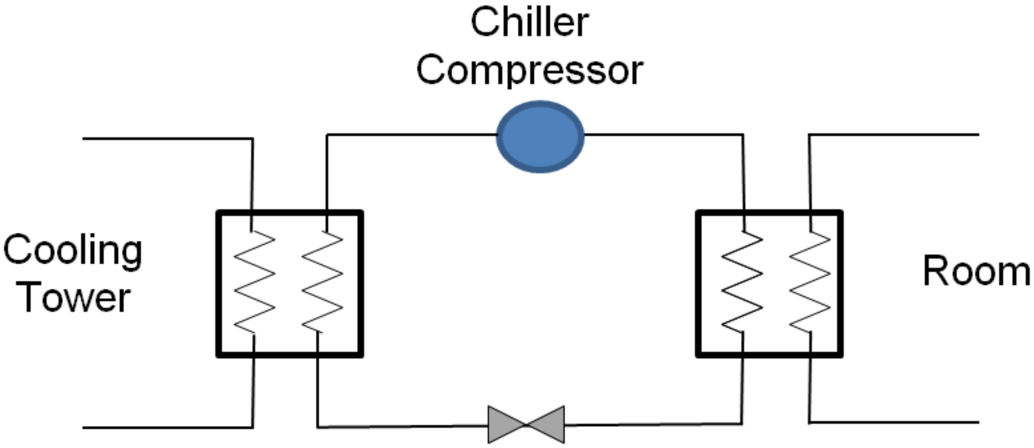


Figure 8: Schematic of cascade refrigeration cycle

The cooling tower is the environment-facing component of a DC cooling infrastructure. Figure 9 shows the thermodynamic model of a cooling tower. Heat removed by a cooling tower is given by:

$$\left.\begin{aligned} q_{CT} &= \left(h_{\mathrm{ao}} - h_{\mathrm{ai}}\right) \times \left(flow_{\mathrm{CTA}} \times \rho_{\mathrm{air}} \right), \\ h_{ao} &= \left[\varepsilon_{\mathrm{CT}} \times \left(h_{\mathrm{swi}} - h_{\mathrm{ai}} \right) \right] + h_{ai}. \end{aligned}\right\} \tag{1.10}$$

The details of the cooling tower heat transfer model can be found in [7].

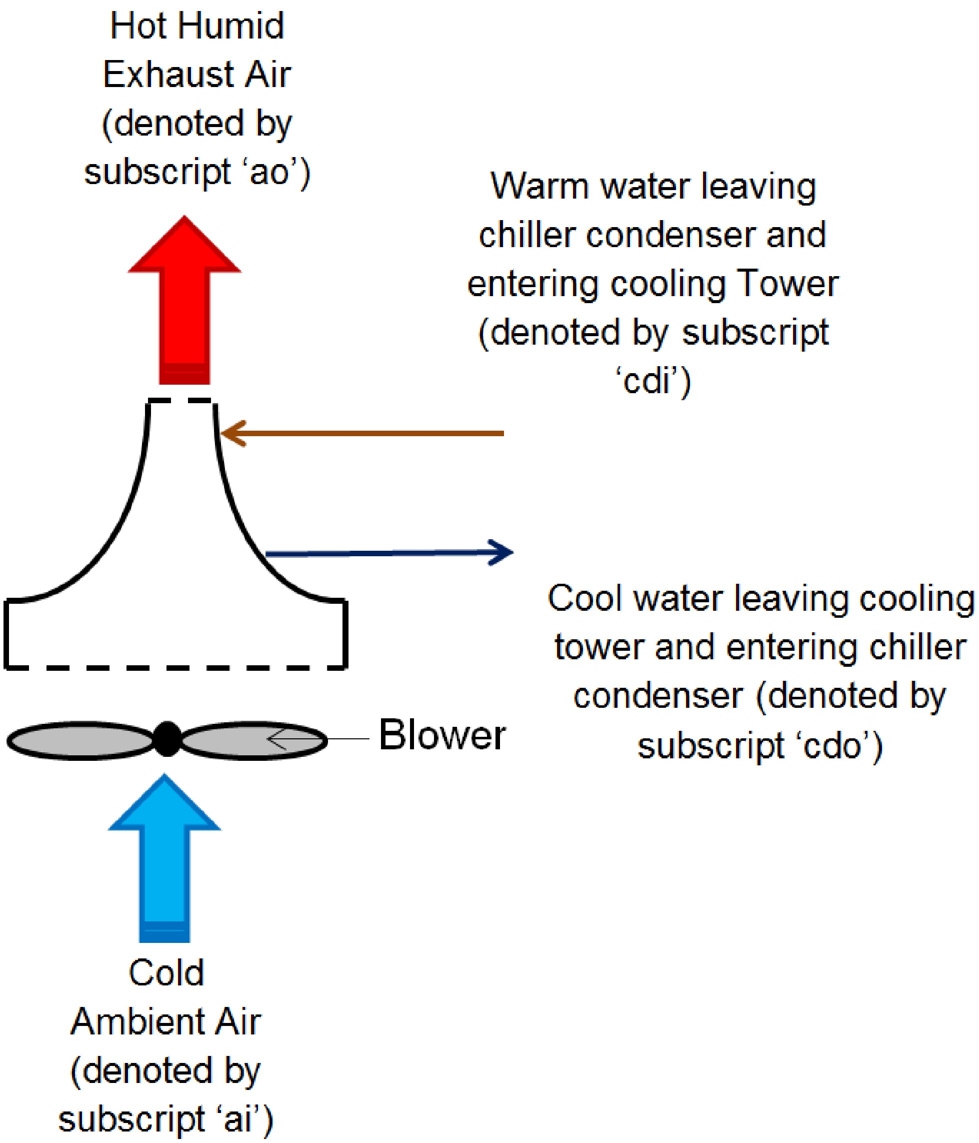

Hot Humid Exhaust Air (denoted by subscript 'ao')
Warm water leaving chiller condenser and entering cooling Tower (denoted by subscript 'cdi')
Cool water leaving cooling tower and entering chiller condenser (denoted by subscript 'cdo')
Blower
Cold Ambient Air (denoted by subscript 'ai')

Figure 9: Thermodynamic model of cooling tower

The efficiency of the overall cooling system is strongly related to the efficiency of the data center room cooling which in turn depends on the data center air temperature [8]. Therefore, one potential strategy for improving energy efficiency of a data center is to monitor its temperature closely and to modulate its heating, ventilation, and air-conditioning (HVAC) set points (e.g. CRAC thermostat set point) accordingly. To implement that strategy, this dissertation aims to develop a measurement-based parametric modeling framework that can efficiently monitor data center temperatures.

Thermal Management of Data Center Room

A data center is a multi-scale thermal system, spanning from chip-level (10^{-3} m, 0.01 s), followed by server-level (10^{-2} m, 0.1 s), rack-level (1 m, 1 s), and finally to aisle/ room-level (10 m, 10 s). Figure 10 shows the different building blocks of a data center along with its spatial and temporal scales.

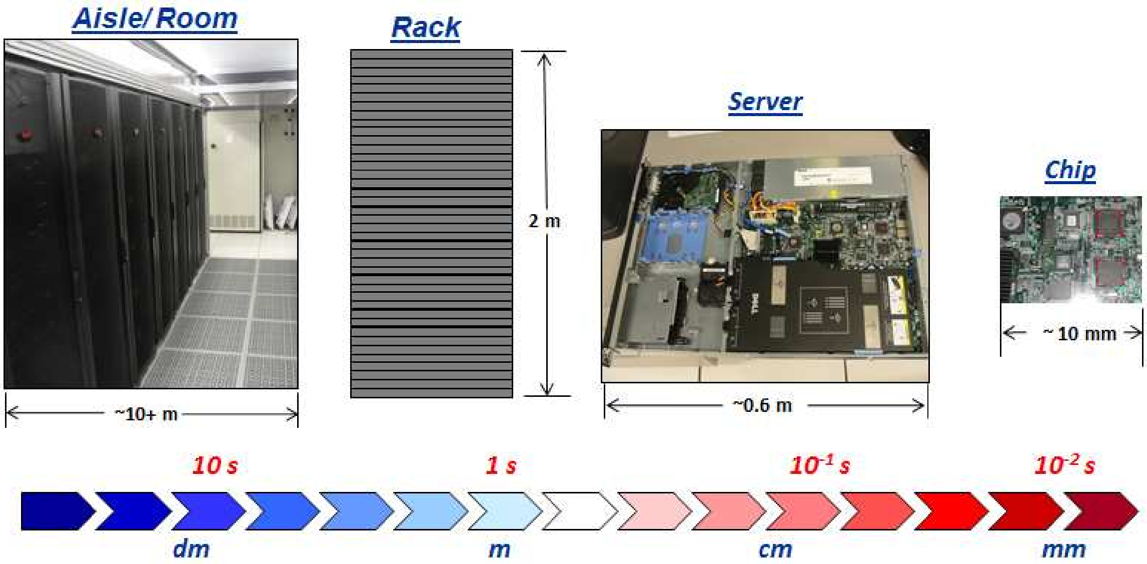


Figure 10: Multi-scale nature of data center temperatures

Due to inherent differences in transport processes, different scales pose different thermal challenges for a data center designer; therefore, each of them demands unique cooling hardware or strategies. Although liquid cooling-based strategies are gaining significant attention in recent years, most data centers employ forced-convective horizontal air cooling. In this context, Table 2 shows various cooling hardware used in a multi-scale data center.

Table 2: Cooling hardware in a multi-scale data center

Level	**Dominant Transport Processes**	**Cooling Hardware**
Chip	Conduction	Heat Sink, TIM, Spreader
Server	Turbulent Convection, Conduction	Server Fan, Heat Pipe, Cold Plate
Rack	Turbulent Convection	Fan, Rear Door Heat Exchangers, CRAC, In Row Cooler
Aisle	Turbulent Convection	CRAC

Most DC rooms are air-cooled—they use an array of computer room air conditioning (CRAC) units which operate air-liquid heat exchangers and air handlers, rack rear door heat exchangers (RDHx) which operate a shell and tube heat exchanger, and in row cooler for room/ aisle level cooling. The guidelines for air-cooled data centers specifying dry-bulb air temperature and relative humidity levels at the inlets of IT equipment have been the focus of the American Society of Heating, Refrigeration and Air Conditioning Engineers (ASHARE) TC 9.9 committee. The TC 9.9 committee suggested classification of data centers based on allowable server inlet air dry-bulb temperature and humidity. Inlet air temperature control is important because too high inlet air temperature increases chip leakage power and server failure rate. On the other hand, lowering inlet air temperature below the dew point leads to condensation of air moisture. As far as humidity is concerned, too high humidity leads to condensation of air moisture. On the other hand, too low humidity leads to electrostatic discharge of server. Therefore, temperature and humidity control of a data center is a critical problem. Based on 2011 ASHARE Guidelines, the allowable inlet air temperature for an A1 data center is between 15 and 32 ^{0}C while the relative humidity is between 20 and 80%. Higher elevations demand a de-rating of the maximum dry bulb temperature by 1 ^{0}C for every 300 m above an elevation of 900 m up to a maximum elevation of 3050 m.

A CRAC unit is an air-water heat exchanger installed with centrifugal blowers and air filter pads. The liquid side of the heat exchanger in a CRAC includes the evaporator of a vapor compression refrigeration cycle, integrated with the building chiller, the chilled water distribution pump, and the compressor. The CRAC supply/ return air temperature and humidity are controlled by a PID controller module installed inside the CRAC controller section. For controlling air temperature and humidity, a CRAC is installed with an electric heater (s), a cooling coil (s), and a humidifier (s). The CRAC PID controller can increase or decrease cooling coil chiller water flow rate to modulate air temperature. On the other hand, the PID controller can increase or decrease heater current flow to modulate air temperature. For humidity control, the PID controller can change the chiller water flow rate through the humidifier. Rear door heat exchangers (RDHx) [9] are typically installed at the rear end of the computing racks. These are shell and tube heat exchangers. They include copper tubing with circulating chilled water which cools hot exhaust air. The cooling effect produced by an RDHx is controlled by the chilled water flow which is controlled by the PID controller integrated with the building management system.

Data centers typically employ an alternating cold aisle/ hot aisle-based forced-convective cooling strategy. Figure 11 shows different convective airflow schemes for an air-cooled data center. Figure 11(a) depicts the underfloor plenum supply and room return scheme. In this scheme, pressurized cooling air from the CRAC unit travels through the underfloor plenum before entering the cold aisle through perforated tiles. Thereafter, cooling air is driven into servers via server fans. While moving though servers, cooling air extracts heat from high-temperature server components such as heat sinks, power modules, and memory modules. For the server racks installed with an RDHx, the hot exhaust air from the servers cools down in two-stages: first, by the RDHx and then, by the CRAC heat exchanger. In the room return scheme (as shown in Figure 11(a)), rack exhaust air returns to the CRAC through the room. As opposed to that in the underfloor plenum supply and ceiling return scheme (shown in Figure 11(b)), hot rack exhaust air returns to the CRAC unit through overhead ceiling ducts. On the other hand in the ceiling supply room return scheme (as shown in Figure 11(c)), cooling air from the CRAC enters into the room via overhead vents. The cooling air extracts heat from the servers, and the resulting hot returns to the CRAC unit via room vents. A detailed study of different airflow schemes is documented in [10].

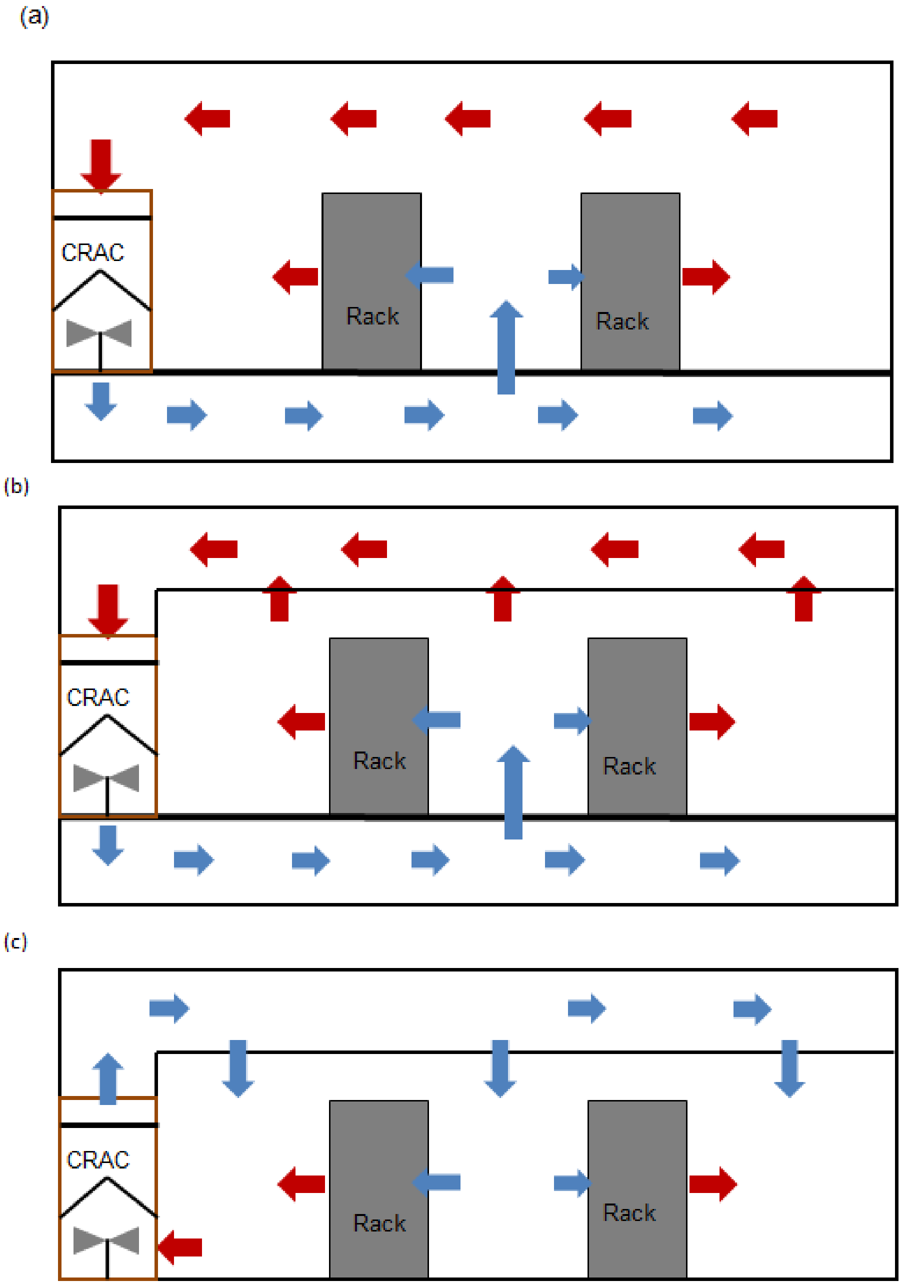


Figure 11: Different airflow schemes in an air-cooled data center. (a) Plenum supply room return scheme. (b) Plenum supply ceiling return scheme. (c) Ceiling supply room return scheme.

There are several problems associated with the alternating cold aisle/ hot aisle-based air-cooling strategy. Figure 12 shows a typical alternating cold/hot aisle-based airflow scheme in which the IT equipment inlets face the cold aisles and the outlets or exhausts face the hot aisles. Computer room air conditioning units (CRACs) drive pressurized cooling air into a raised-floor plenum. Unlike hot aisles, cold aisles have perforated tiles that allow cooling air to come up and get entrained into the servers via server fans. The hot exhaust air returns to CRAC units driven by the negative pressure gradient created by the CRAC blowers. Cold and hot air mixing can form local hotspots. Unnecessarily conservative CRAC set-points, established to mitigate these hotspots, and inappropriate server fan operation often lead to coolant bypass, in which cooling air directly returns back to the CRAC unit. The problem of cooling airflow management is further compounded by the introduction of hypervisor-based virtualization technologies that facilitate dynamic server load migration.

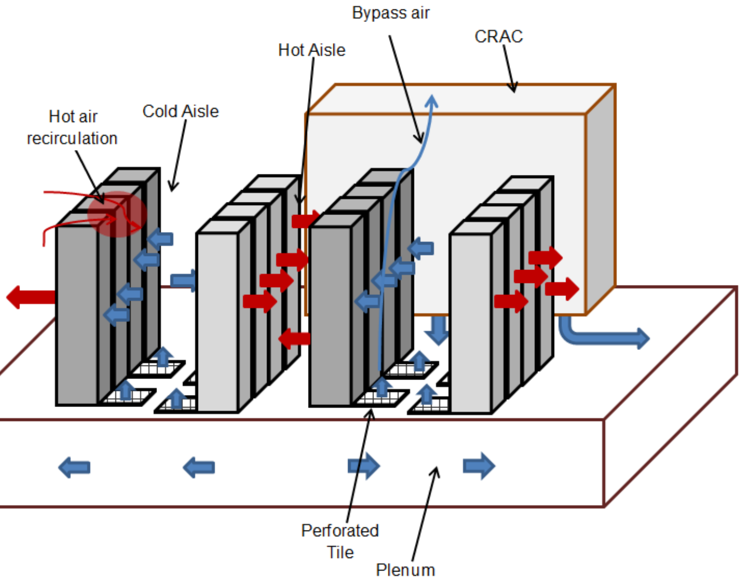


Figure 12: Problems in alternating cold aisle/ hot aisle-based forced convective cooling in data center

Thermal Modeling of Data Centers

The solution to hot air recirculation and by-pass air demands optimal design of convective air temperatures inside a DC. Different modeling strategies exist for thermal design of data centers. Figure 13 shows different thermal modeling techniques for a transient data center.

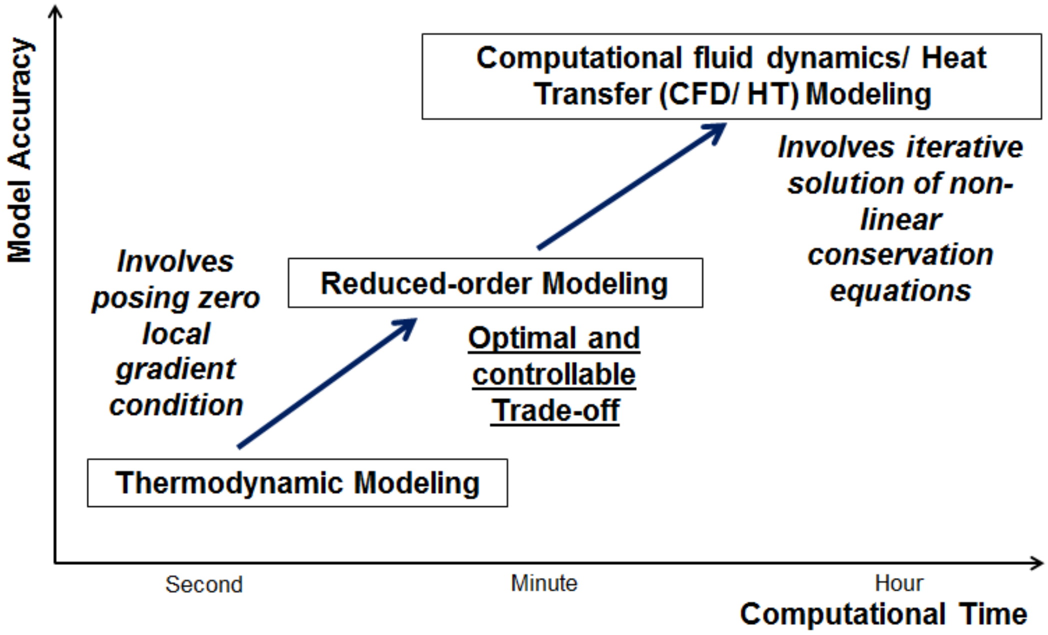

Model Accuracy
Computational fluid dynamics/ Heat Transfer (CFD/ HT) Modeling
Involves iterative solution of non-linear conservation equations
Involves posing zero local gradient condition
Reduced-order Modeling
Optimal and controllable Trade-off
Thermodynamic Modeling
Second
Minute
Hour
Computational Time

Figure 13: Different techniques for modeling transient temperatures in data centers

The most prevalent of these strategies is computational fluid mechanics/ heat transfer (CFD/HT)-based modeling. Several researchers, including Kang et al. [11], VanGilder et al. [12], Karki et al. [13], Schmidt et al. [14], Patel et al. [15], use CFD/HT-based modeling for thermal design of data centers. Furthermore, CFD/HT models are employed to data center design optimization in terms of parameters such as plenum depth, facility ceiling height, cold aisle spacing, CRAC flow rate, rack flow rate, and power dissipation

[16-19]. The literature is inundated with CFD-based data center modeling. Table 3 compiles some of the important CFD-based data center design studies in the literature.

Table 3: CFD-based studies for data center heat transfer modeling and their scopes

Author	Year	Ref.	Scope
Kang et al.	2000	[11]	Plenum Design
Schmidt et al.	2001	[20]	Plenum Design, Tile Design, Experimental benchmarking
Patel et al.	2001	[21]	Optimization, Experimental benchmarking, Alternative cooling strategies
Sharma et al.	2002	[15]	Design decision-making tool
Schmidt et al.	2002	[22]	Layout design
Rambo et al.	2003	[23]	Airflow management
Wang	2004	[24]	Minimization of hot air recirculation
Shrivastava et al.	2005	[25]	Comparative analysis of DC design schemes
Herrlin et al.	2006	[26]	Free convection effect on DC cooling
Bhopte et al.	2006	[27]	Modeling of underfloor blocking
Schmidt et al.	2007	[28]	Design decision-making

Although CFD/HT simulation produces highly accurate predations, there are several issues regarding its suitability as a dynamic design optimization tool. CFD/HT modeling invokes a cubic-time algorithm: first, it discretizes transient, non-linear (convection terms), three-dimensional (three spatial dimensions), second-order (diffusion term) partial differential equations (three momentum equations and energy equation) and three-dimensional continuity equations. Then, it solves the discretized algebraic equations via iterative techniques. The iteration-based algorithms are cubic times because it scales with cubic power of the number of spatial grid points multiplied with the number of time steps. For most data centers, the number of spatial grid points is often in the order of few millions while the number of time steps is typically in the order of a few thousands depending on the type of transient problems studied. As a result of this polynomial time nature of the iterative solution procedure, CFD/HT-based modeling is time-intensive for dynamic design optimization of a data center.

An alternative modeling strategy involves thermodynamic-based modeling of data centers [29]. The exergy-based estimation tools are useful for rapid thermal assessment of data centers. However, due to the intrinsic assumption of quasi-equilibrium processes, a data center thermodynamic model tends to lose many degrees of freedom (DOF) of a convective heat transfer process. Although the thermodynamic model predicts reasonably well for a low-density facility, it fails for a high-density facility with a complex air flow pattern. Therefore, thermodynamic models are not useful for data center thermal design optimization.

Another approach that is gaining popularity is the reduced-order model. A reduced-order model of a process is a lower-dimensional model of the high-dimensional process. Mathematically, it amounts to mapping correlated data into an uncorrelated data space. Since it is impossible to design complex geometries inside a data center, it is convenient to use physics-based assumptions to reduce the modeling effort. An example of such abstraction pertains to modeling a computing server as a box with uniform heat generation. Besides such geometric simplifications, a reduced-order model can be used to improve the parametric granularity of a data set, captured either by experiments or by simulations. These reduced-order models essentially employ a two-step data fusion approach: first, they identify the linearly-uncorrelated directions of a data set. Then, they identify the weighing functions for these directions for a new parametric point. Overall, a reduced-order model analyzes experimental or simulation data statistically and synthesizes new data points to enhance the parametric granularity of the primitive (input) data set.

The transformation of a correlated data set into a linearly-uncorrelated data set is performed via several statistical modeling techniques such as proper orthogonal decomposition (POD), fast fourier transform (FFT), non-linear Volterra theory, and harmonic balance approximation. These techniques have better computational efficiencies compared to CFD/HT-based techniques. For example, POD is a logarithmic-time algorithm while FFT is a quasilinear algorithm. Incorporation of these statistical modeling techniques improves the efficiency of the resulting reduced-order model in comparison to the corresponding full-scale CFD/model. Since the weighting functions for the uncorrelated data set for the primitives constitute a low-rank matrix, it is possible to use statistical techniques such as kriging [30] or interpolation [31] to determine the weighting factor for a new data point. Overall, a reduced-order model demonstrates better predictive accuracy than a thermodynamics model because it uses CFD/HT or experimental data as primitives. In addition to POD, there are three prevalent reduced-order modeling techniques: neural networks [32], fuzzy rule-based systems [33], and genetic algorithms [34].

Neural networks are computational models, inspired by the way human brain functions. Neural network models recognize the optimal output data by identifying the interrelation between inputs and outputs. They identify the input-output mapping using a set of interconnected nodes or neurons. Each neuron processes its inputs either from external sources or from other neurons, using following relationship:

$$A_j = f\left(\sum_{j=1}^{n} W_{ij} A_i + b_j\right). \tag{1.11}$$

Here A_i is the input from the ith neuron. A_j is the input from the jth neuron. W_{ij} represents the connecting weight between two neurons. b_j is the bias on the jth neuron. *n* is the number of input neurons. The activation function, f provides a non-linear gain to the output. The neural network models can be used as a pattern regeneration tool for a data center design optimization. It has been widely used to identify optimal facility layout to maximize the cooling air ingestion by a cluster of computing racks under the constraints of rack heat load, tile airflow, server virtualization, and rack airflow [35, 36]. In the data center neural network model, CFD-based simulation data are used as the input engine. In that context, Superposition-based models [37] , thermal zonal models [38], and PDA-CFD techniques [39] are widely used to generate rapid CFD simulation data.

Fuzzy rule-based systems use many-valued fuzzy logic for inference. Fuzzy logic is based on fuzzy set theory for which binary set membership has been extended to include partial membership ranging between 0 and 1. In fuzzy rule-based systems, each model variable is defined with a series of overlapping fuzzy sets. The mapping from inputs to outputs can be expressed as a set of IF-THEN rules which can be derived from expert knowledge or from data. Fuzzy rule-based control systems have found wide-spread applications in virtualized data center resource management [40, 41].

Genetic algorithms are non-linear search and optimization techniques inspired by the biological processes of natural selection and survival of the fittest. In a genetic algorithm-based optimization procedure, a population of candidate solutions is evolved toward the better solution space. Each candidate solution has a set of mutable properties which can be altered in the process of dynamic optimization. The thermal design of a data center often poses a constrained multi-objective optimization problem which can be solved using multi-objective genetic algorithms [42].

Based on the source of primitive input data, reduced-order models can be classified into two groups: CFD/HT-based reduced-order models and measurement-based reduced-order models. In the context of design optimization, CFD/HT-based reduced-order models have been widely investigated in the literature [43, 44]. Samadiani et al. [45] developed a reduced-order DC model from distributed sensor data. Such a measurement-based framework leverages the availability of measurement data, which is replaced by the CFD-based input data. There are several advantages of circumventing CFD modeling altogether. First, the efficiency of the modeling framework improves significantly by avoiding CFD-based modeling altogether. Also, a measurement-based data set is a better representative of the stochastic physical processes inside a DC. In addition, a reduced-order model is a suitable tool for designing a measurement-based model. In a measurement-based system, there are trade-offs between density of sensors (how many sensors), location of sensors (where to place sensors), measurement frequency of the sensors (how often to read the sensors), and other parameters involving DC business needs and reliability requirements. Therefore, a measurement-based DC modeling framework can simulate a high-fidelity, high-resolution, and near-real-time optimization tool. Nevertheless, there is paucity in the literature on measurement-based modeling for design optimization of data centers. Most of the reduced-order models for data center design use POD as the model order reduction tool.

Scope of this Dissertation

Samadiani [44], Rambo [43], and Rolander [46] have used POD for data center design optimization. Likewise, this dissertation uses POD for data center design. However, this dissertation is unique in several ways:

- It uses experimental data as the model primitive unlike other DC design study in the literature.
- The developed modeling framework is dynamic in the sense that it uses time as a model parameter.
- It uses spatial location as a modeling parameter.

CHAPTER 2 of this dissertation discusses the methodology for POD-based modeling of experimental data. CHAPTER 3 develops a temperature prognostic model for data centers. Then,

CHAPTER 4 develops a framework to improve spatial resolution of measured temperature data. CHAPTER 3 and

CHAPTER 4 are similar in the sense that they use the independent variables of the energy equation (time and space) as the model parameters. CHAPTER 5 and CHAPTER 6 are pertaining to data center design optimization problems. While CHAPTER 5 studies two-parameter predictive framework of data center air temperatures with time and rack heat loads as parameters, CHAPTER 6 deals with CPU temperature predictions under a given IT workload with cooling resource set-points (CRAC supply temperature and RDHx pressure) as parameter. Finally, CHAPTER 7 draws conclusions to this dissertation.

CHAPTER 2
POD-BASED REDUCED-ORDER MODELING

This chapter pertains to proper orthogonal decomposition (POD) and its use as a data-driven modeling tool. The efficiency and error of the framework are assessed. Proper orthogonal decomposition (POD), also widely known as principal component analysis (PCA), is a data compression algorithm that transforms a data matrix into a product of a low-rank matrix (POD modes) and a coefficient matrix (POD coefficients). POD-based data compression algorithms are widely used in video surveillance [47], face recognition [48], and bio-informatics [49]. For characterizing turbulent flow, a POD model was introduced by Lumley [50] and extended by Sirovich et al. [51]. Kosambi (1943) [52], Loeve (1945), Karhunen (1946), Pougachev (1953), Obukhov (1954) utilized similar mathematical procedures independently but with different names, such as principal component analysis (PCA), Karhunen-Loeve decomposition (KL), singular value decomposition (SVD), empirical eigenfunction decomposition. POD has been used as a parametric optimization tool for the DC infrastructure design problem. Typical parameters include rack heat load [53], CRAC Flowrate [30], and time [54].

Data-driven Meta-modeling

A model is an abstraction of the physical phenomena; a meta-model is yet another abstraction on the model. Data-driven meta-modeling means modeling of experimental data. A data-driven meta-modeling philosophy is shown in Figure 14. The output data is modeled as a function of input variable space, which can be classified into independent variable space and input parametric space. Keeping independent variable space fixed and varying parametric input variable(s), several sets of output data were generated via physical experiments. While the blue regions in Figure 14 represent those output data sets, the red regions represent predicted response surfaces. The motivation of the response surface generation is to improve the parametric granularity of the measurement data. A response surface offers several attractive values:

- It reduces experimental operating cost. First, it reduces the required number of experimental runs. Second, it allows operators to work with low quality sensors. For example, if a reduced-order model of air temperature is developed with time as a parameter, then an experimentalist can manage with inferior (higher response time) temperature sensors.

- It facilitates near-real-time decision-making. Unlike a method-driven approach such as CFD, this data-driven meta-modeling approach operates online with lower systemic latency.

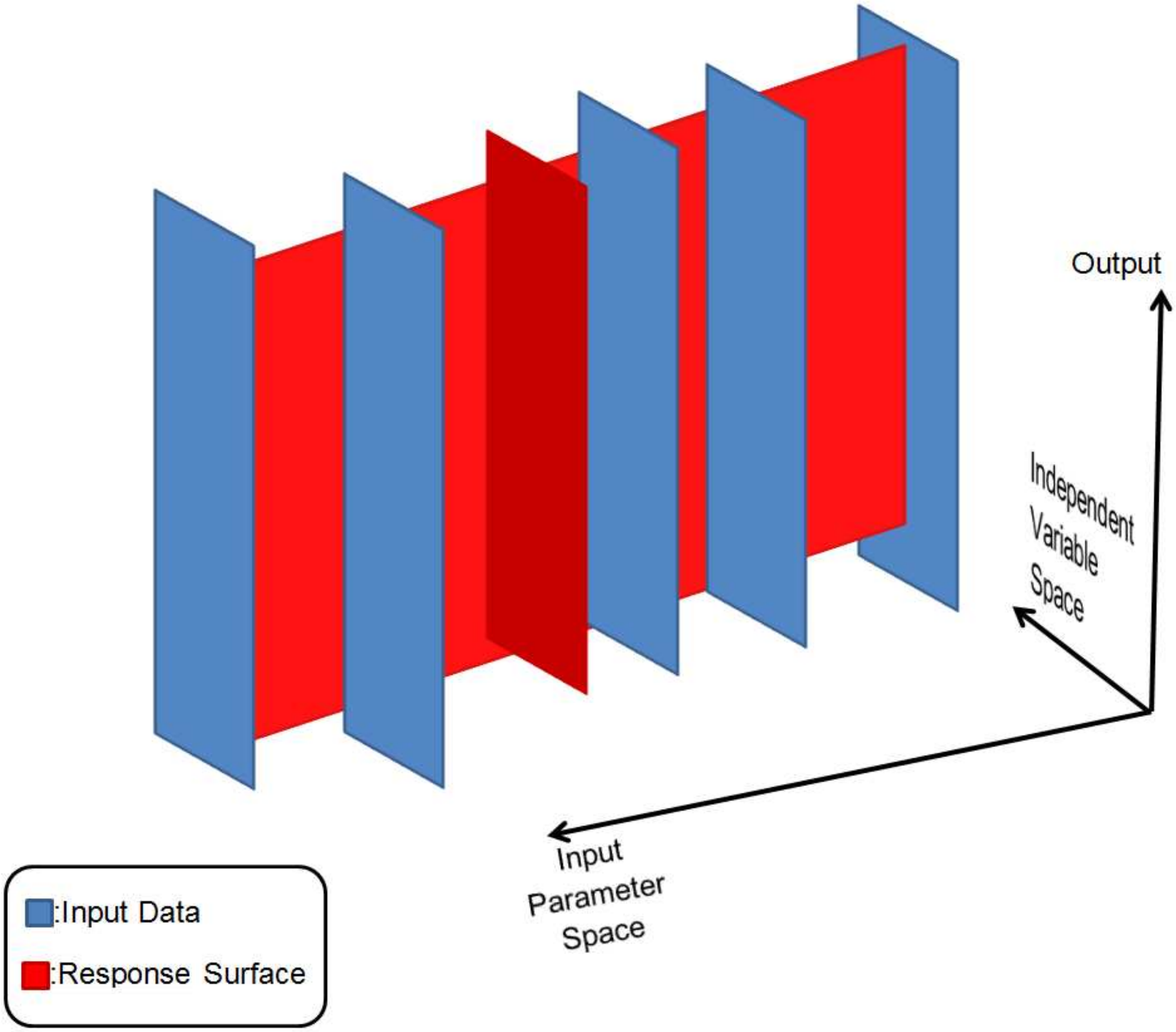


Figure 14: A data-driven meta-modeling philosophy

It can be readily observed that this data-driven prediction strategy is a best-fit subspace problem of finding a set of data points in $\Re^m$ (where *m* is the size of independent variable space) in a *n*-dimensional parametric space. A simple-minded strategy for determining new data sets is direct regression analysis of the input data. However, the computational time for such analyses is in the order $\sim O(mn)$. Therefore, it is not a suitable method to handle a dataset with large *m* and *n*. Another approach is to model the problem into a *k*-means clustering problem. This approach involves finding prediction points that minimize the sum of distances from the nearest data points. A natural relaxation to the *k*-means problem is to find the *k*-dimensional subspace that minimizes the sum of the distances of the data points to the subspace. In this context, POD is an effective tool to determine that *k*-dimensional subspace. A decomposition of any arbitrary matrix, $A = \sum_{i=1}^{n} \sigma_i U_i V_i^T$ is called the proper orthogonal decomposition if the sequence of σ_i is non-increasing, and the sets of $\{U_i\}, \{V_i\}$ are orthonormal.

POD-based Modeling

As an input, POD needs a data ensemble, $T_i(In; D_i) \in \mathrm{R}^{m\times n}$ generated from physical experiments or from numerical simulations. $(In; D_i)$ is the input to the data ensemble. While In is the independent variable field for the data ensemble, D is the dependent variable field. The subscript, *i* indicates parametric data ensemble. The ensemble is compiled over n-dimensional parametric space spanned by D_i. The row dimension, m indicates the dimensionality of the independent variable or predictor space.

The first step of a POD model is to compute the parametric-average of the data ensemble:

$$T_0(In) = \frac{\sum_{i=1}^{n} T_i(In; D_i)}{n}, \in R^{m\times 1}. \tag{2.1}$$

The parameter-dependent part of the data ensemble is modeled as:

$$T_i^*(In; D_i) = \left[T_i(In; D_i) - T_0(In)\right];\ T^* \in R^{m\times n}. \tag{2.2}$$

By using POD-based modal decomposition, T^* is expressed as the product of a low-rank matrix with corresponding weighting scalars. The low-rank matrix is the compilation of optimal basis functions, called POD modes. The weight scalars are called POD coefficients. While POD modes are independent of parameters, POD coefficients are parameter dependent.

The attractive feature of POD modes lies in their optimality in the sense that N POD modes convey more information about the data response surface than any other basis functions generated by comparable decompositions such as fast Fourier transform (FFT). The mathematical statement of the optimality is that the optimal basis functions, ψ should maximize $\left\langle \left| T^*, \psi \right|^2 \right\rangle$ with a constraint $\|\psi\|^2 = 1$. The corresponding functional for this constrained variational problem is:

$$J(\psi) = \left\langle \left| T^*, \psi \right|^2 \right\rangle - \lambda(\|\psi\|^2 - 1). \tag{2.3}$$

The necessary condition for the optimization suggests that the functional derivative of $J(\psi)$ tends to zero with all variations in $\psi + \delta\theta \in L^2([0,1]), \delta \in R$:

$$\frac{d}{d\delta} J[\psi + \delta\theta]_{\delta=0} = 0. \tag{2.4}$$

The simplification of the previous equation for a discrete data ensemble leads to the governing equation for POD modes:

$$Ru = \lambda u. \tag{2.5}$$

This is an eigenvalue equation with $R = \frac{1}{m}\left(T^*\right)^{Tr} T^*$; the superscript '*Tr*' denotes the transpose of the matrix. The eigenvalues indicate the importance of corresponding POD modes in the data response surface. Larger λ s have larger relative information contents of the data response surface. The solution of the eigenvalue equation is performed via a power method-based numerical iterative procedure:

1. Assign a random unit vector, u.
2. Iterate until it reaches convergence: $u := \frac{Ru}{\|Ru\|}$.
3. Compute the POD mode as the dyadic product of T^* and u:

$$\psi = T_d \otimes u, \;\; \psi \in R^{m \times n}. \tag{2.6}$$

The power algorithm ensures rapid convergence time. Let $\{u_i\}$ be the eigenvectors of R and let $\{\lambda_i\}$ be the corresponding eigenvalues. Let, x^k be the unit vector obtained after the k^{th} iteration. Since $\{u_i\}$ are orthonormal:

$$\|Ru^k\|^2 = \frac{\sum_i \lambda_i^{2k+2}}{\sum_i \lambda_i^{2k}}. \tag{2.7}$$

Now, by the Holder's inequality:

$$\sum_i \lambda_i^{2k} \le \left(\sum_i \lambda_i^{2(k+1)}\right)^{\frac{k}{k+1}} n^{\frac{1}{k+1}}, \tag{2.8}$$

where n is the rank of the eigenspace.

Rearrangement of the resulting inequality yields:

$$\frac{\sum_i \lambda_i^{2k+2}}{\sum_i \lambda_i^{2k}} \ge \frac{1}{n^{1/k}} \left[\sum_i \lambda_i^{2k}\right]^{1/k} \ge \frac{\lambda_1^2}{n^{1/k}}.$$

$$\Rightarrow \|Ru^k\|^2 \ge \frac{\lambda_1^2}{n^{1/k}}. \tag{2.9}$$

On the other hand, since $\{u_i\}$ are orthonormal:

$$\frac{\lambda_1^2}{n^{1/k}} \le \left\| Ru^k \right\|^2 \le \lambda_1^2. \tag{2.10}$$

This bound shows that $\left\| Ru^k \right\|^2$ asymptotically converges to λ_1^2 .

The left inequality suggests the minimum number of computational steps required for reaching a converged solution. At the k^{th} iteration the ratio of the iterative solution to the converged solution is equal to $1/n^{1/k}$. A convergence criterion is chosen as 2^{-r} such that:

$$1/n^{1/k} \sim 2^{-p}. \Rightarrow \frac{\log_2(n)}{k} \sim p. \Rightarrow k \sim \frac{\log_2(n)}{p}. \tag{2.11}$$

Since p is a machine dependence parameter, the time complexity of the Power method is on the order of $\log(n)$. This is an important property for any reduced-order modeling algorithm; it indicates that the computation time is marginally affected even if the rank of the data matrix is quite high. Such scalability is a desirable feature for a model order reduction problem.

The computational time for each POD mode is in the order of log(n). Therefore, the number of POD modes to describe a response surface within certain accuracy tolerance is a critical parameter for the efficiency of the model. Since an eigenvalue, λ_i indicates the energy content of the corresponding POD mode, ψ_i, the minimum number of POD modes required to capture a certain percentage of energy or information content of a data set is given by, k :

$$\left(\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > \text{C.E.P.} \right) \bigcap \left(\min(k) \right), \tag{2.12}$$

where, C.E.P. is defined as the captured energy percentage by k POD modes. The previous equation indicates that k POD modes can predict a response surface within certain accuracy tolerance defined by the captured energy percentage (C.E.P.).

The parametric component of the response surface is governed by the POD coefficients. The numerical algorithm for computing POD coefficients at the interrogation parametric point is described as follows:

1. Compute the complete coefficient matrix:

$B\left(D_{en}\right) = \psi^{+} \otimes \left(T^{*}\left(In; D_{en}\right)\right), B \in \Re^{k \times n}$.

The subscript “en” indicates the parameter related to the ensemble space.

2. Determine the POD coefficient, $b\left(D_{\text{int}}\right) \in \Re^{n \times 1}$ by applying multi-dimensional interpolation of B . The subscript “int” indicates the parameter related to the interrogation space. The computational steps for this multi-parameter interpolation are in the order of $\sim O(k \times n)$, which is

considerably lower than direct interpolation of the data ensemble $\sim O(m \times n)$. It is because $k < n < m$.

Another approach to compute POD coefficient is kriging [55]. Kriging is an optimal interpolation scheme based on the regression of data points according to spatial covariance values.

Finally, the parametric response surface is generated by adding the parameter-independent component and the product of POD modes and POD coefficients:

$$T_{\text{int}}\left(In; D_{\text{int}}\right) = T_0\left(In\right) + \psi\left(In\right) \otimes b\left(D_{\text{int}}\right). \tag{2.13}$$

The high-level numerical procedure for POD-based data-driven modeling is shown in following flowchart:

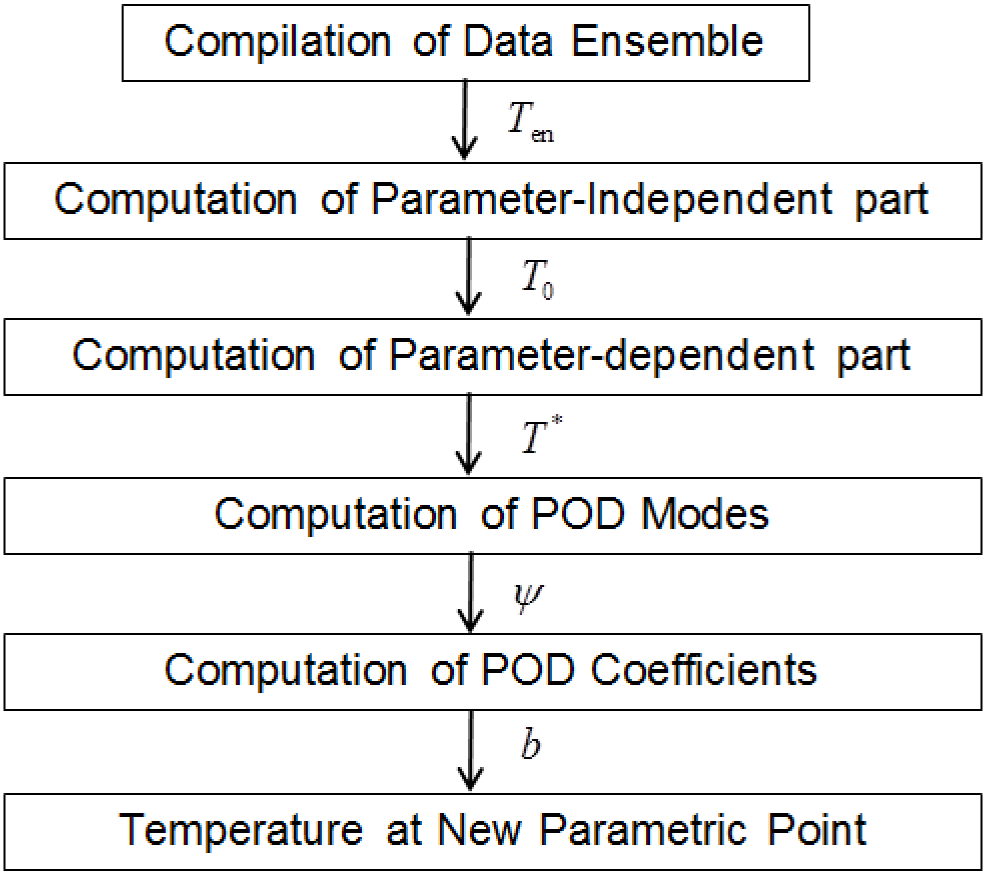

Compilation of Data Ensemble
T_{en}
Computation of Parameter-Independent part
T_0
Computation of Parameter-dependent part
T^*
Computation of POD Modes
ψ
Computation of POD Coefficients
b
Temperature at New Parametric Point

Figure 15: POD-based data-driven algorithm for modeling experimental data

Error Estimation of POD Models

As a meta-modeling technique, the accuracy of the POD-based framework is a critical design consideration. The modeling accuracy can be determined in two ways: a priori or a posteriori. While posterior error estimation is useful for assessing modeling fidelity, a priori error estimation—often analytical in nature—is a useful design capability for near-real-time POD-based controllers. The a priori error can be integrated into the control logic of the POD controller to yield high-precision reliable output. POD modeling error can be defined as the deviation of POD predictions from experimental data:

$$E_{\text{Prediction}} = \left(T_{\text{Data}} - T_{\text{POD}}\right). \tag{2.14}$$

A POD framework is reliable if it satisfies following fidelity condition:

$$\begin{gathered} E_{\text{Prediction}} \leq f\,\Delta T_{\text{Scale}}^{\text{Measurement}}, \\ \text{where} \\ f \text{ is an operater dependent scalar, numerically varies between 0 and 1;} \\ \Delta T_{\text{Scale}}^{\text{Measurement}} \text{ is the representative temperature scale of the problem.} \end{gathered} \tag{2.15}$$

The factor *f* quantified the degree of relaxation on the modeling accuracy. If *f* is equal to 1, the model is highly relaxed because the model is allowed to incur error equal to $\Delta T_{\text{Scale}}^{\text{Measurement}}$. Conversely, as *f* tends to 0, the accuracy demand from the model increases proportionally.

The analytical error can be defined as the deviation of POD predictions from the exact solution:

$$E_{\text{Analytical}} = \left(T_{\text{Exact}} - T_{\text{POD}}\right). \tag{2.16}$$

A comprehensive a priori error estimation scheme should consider both interpolation and extrapolation-based POD/regression model. The interpolation is required when the interrogation point lies within the input parameter domain, otherwise extrapolation is required. While POD/interpolation error can be determined statistically; POD/extrapolation error estimation requires functional analysis of the governing differential equation. Since this dissertation pertains to convective heat transfer modeling of data centers, POD/extrapolation error of data center air temperature is determined. The governing differential equation used in the functional analysis is the energy equation.

Analytical Error for the POD/ Interpolation Framework

For determining the analytical error of the POD/ interpolation scheme, $E_{\text{Analytical}}^{\text{POD/Interpolation}}$, a linear algebra-based analysis, as documented in section 2.3 of [56], is utilized. The important features of the analysis are outlined in this section.

Let $T^1, T^2, ..., T^l$ be snapshots and let $\zeta := span\{T^1, T^2, ..., T^l\} \in T$ with $m := \dim(\zeta)$.

Assume $\{\psi\}_{i=1}^m$ is an orthonormal basis of ζ :

$$T^j = \sum_{i=1}^{m} (T^j, \psi_i)\psi_i, \ \ for\ j = 1, ..., l. \tag{2.17}$$

The fundamental principle of reduced-order modeling is finding $d(< m)$ orthonormal basis vectors $\{\psi_i\}_{i=1}^d \in T$ such that the mean square error between the elements of the ensemble set and corresponding d^{th} partial sum is minimized on average:

$$\min_{\{\psi\}_{i=1}^d} \frac{1}{l} \sum_{j=1}^{l} \left\| T^j - \sum_{i=1}^{d} \left(T^j, \psi_i\right)_v \psi_i \right\|_1^2 , \tag{2.18}$$

$$\text{subject to } (\psi_i, \psi_j) = \delta_{ij} \text{ for } 1 \le i \le d,\ 1 \le \text{j} \le \text{i}.$$

POD error can be reformulated:

$$\min_{\{\psi\}_{i=1}^d} \frac{1}{l} \sum_{j=1}^{l} \left\| T^j - \sum_{i=1}^{d} \left(T^j, \psi_i\right)\psi_i \right\|_v^2 = \min_{\{\psi\}_{i=1}^d} \frac{1}{l} \sum_{j=1}^{l} \sum_{i=d+1}^{m} \left| \left(T^j, \psi_i\right)_1 \right|^2 = \sum_{i=d+1}^{m} \lambda_i . \tag{2.19}$$

In addition, a constant, c_0, is multiplied to the sum of the eigenvalues corresponding to the discarded POD modes to fully specify $E_{\text{Analytical}}^{\text{POD/Interpolation}}$. The arbitrary constant, c_0, quantifies the interpolation error.

For the POD/interpolation scheme, $E_{\text{Analytical}}^{\text{POD/Interpolation}}$, is given by:

$$E_{\text{Analytical}}^{\text{POD/Interpolation}} = c_0 \sum_{i=k+1}^{n} \lambda_i. \tag{2.20}$$

Analytical Error Bound for POD-based Extrapolation

For determining the analytical error of the POD/extrapolation scheme, $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$, a weak formulation-based functional analysis, as documented in [56], is used. Instead of a weak formulation-based functional analysis for the Navier-Stokes equations as conducted in [56], the analytical error for the POD/extrapolation framework requires a functional analysis of the energy equation. The governing equation for the convective air temperature field, $T(x,y,z,t)$ inside a data center is:

$$\frac{\partial T}{\partial t} - \left(\alpha + E_H\right)\nabla^2 T + \vec{u}.\nabla T = \dot{q}. \tag{2.21}$$

For the sake of simplicity, the initial condition is chosen to be independent of spatial locations: T (t=0) =T_0. The boundary conditions for air temperatures in a data center are often complicated: following [56], the boundary temperatures are chosen to be equal to zero. Both the Navier-Stokes equations and the energy equations are conservation equations; therefore, they have similar forms except the energy equation does not have the pressure gradient term like the Navier-Stokes equations. Nevertheless, the same analytical methodology [56] is used considering that the pressure gradient term does not feature in the weak formulation in [56].

$$\left(T_t, v\right) + a\left(T, v\right) + b\left(u, T, v\right) = \left(q, v\right). \tag{2.22}$$

$$a(T,v) := \alpha \int_{\Omega} \nabla T : \nabla v dx,\ b(U,T,v) = \int_{\Omega} (U.\nabla) T.v dx. \tag{2.23}$$

The determination of the analytical error, $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$, in [56] is essentially a two-step procedure: first, the estimation of the deviation between the exact solution and the numerical solution [57, 58], and second, the estimation of the deviation between the numerical solution and the reduced-order solution. The second part of the procedure is exhaustively derived in [56]. Finally, the errors determined from previous two steps are added to obtain the bound for the deviation between the exact solution and the reduced-order model solution, $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$.

The deviation between the exact solution and the POD-based prediction is:

$$E_{\text{Analytical}}^{\text{POD/Extrapolation}} \leq c_1 \left(\sigma^{-1}\left(t_m\right) k + h^p\right) + \left[\frac{l \exp(\theta)}{1-\theta}\left(c_2 + c_3 k\right) \sum_{n=d+1}^{m} \lambda_n\right]^{\frac{1}{2}}, c_4 k \leq \theta < 1. (2.24)$$

where, c_1, c_2, c_3, c_4 are arbitrary constants. $\sigma^{-1}(t\ ) = \min(1, t)$. $k :=$ Time step size.

$h^p :=$ Finite element size. $l :=$ Number of snapshots.

$\lambda_n :=$ Eigenvalues corresponding to POD modes.

With k and h^p featuring in Eq.(2.24), it is evident that the discretization of the numerical scheme is an integral part for determining $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$. By definition, a numerical solution framework involves discretization, which is essentially transforming continuous equations into its discrete counterparts. Similarly, experimental data can be modeled as a discrete sample set of the solution space of the governing equation. For an experimentally-derived discrete dataset, the time step, k, can be modeled as the time difference between two consecutive observations, and the finite element size, h^p, can be modeled as the normalized distance between two neighboring sensors. After the functional form of the analytical error, $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$, is determined, its complete specification involves a multi-dimensional optimization analysis.

Optimization Procedure Complete Specification of A Priori Error

It is apparent from Eqs.(2.20) and (2.24) that complete determinations of $E_{\text{Analytical}}^{\text{POD/Interpolation}}$ and $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$ require optimal numerical values for the empirical constants c_0 and $\left(c_1, c_2, c_3\right)$. It is obvious that the numerical values of these constants depend on the specific initial data. Therefore, the numerical values of these constants are determined via a statistical optimization procedure. The central philosophy of this procedure is that the fractional difference between $E_{\text{Analytical}}$ and $E_{\text{Prediction}}$ is optimally minimized for the different values of optimization parameter(s): c_0 for the POD/ interpolation framework, and $\left(c_1, c_2, c_3\right)$ for the POD/ extrapolation framework. The fractional difference between $E_{\text{Analytical}}$ and $E_{\text{Prediction}}$ is defined as the error functional $\left(e\right)$:

$$e = \frac{abs(E_{\text{Prediction}} - E_{\text{Analytical}})}{abs\left(E_{\text{Prediction}}\right)}. \tag{2.25}$$

For the POD/ interpolation framework, the optimization problem is:

$$\min\left[e(c_0)\right],\ c_0 \in \mathrm{R}. \tag{2.26}$$

For the POD/ interpolation framework, the optimization problem is:

$$\min\left[e(c_1, c_2, c_3)\right],\ \left(c_1, c_2, c_3\right) \in \mathrm{R}. \tag{2.27}$$

$E_{\text{Prediction}}, E_{\text{Analytical}}$, and e are multi-dimensional vectors. The minimization of e is conducted statistically: for a given c_0 or $\left(c_1, c_2, c_3\right)$, e is calculated. Thereafter, average $\left(\mu\right)$ and standard deviation $\left(\sigma\right)$ across the various dimensions of e are calculated:

$$\mu = \frac{\sum_i e_i}{\dim(e)}. \tag{2.28}$$

$$\sigma = \left(\frac{1}{\dim(e) - 1} \sum_i \left(e_i - \mu\right)^2\right)^{\frac{1}{2}}. \tag{2.29}$$

A low value of μ suggests that average values $E_{\text{Prediction}}$ and $E_{\text{Analytical}}$ are proximal to each other. On the other hand, a low value of σ suggests the difference between $E_{\text{Prediction}}$ and $E_{\text{Analytical}}$ does not deviate much from μ. A low μ together with a low σ suggests $E_{\text{Analytical}}$ tends to approximate $E_{\text{Prediction}}$ within a confidence interval determined by μ. Such an approximation will obviate the necessity of a posteriori experimental measurements for estimating the validity of the POD-based framework. T_{POD} can be directly added to $E_{\text{analytical}}$ to obtain a temperature value whose accuracy depends upon the quality of the optimization procedure. For difference values c_0 and $\left(c_1, c_2, c_3\right)$, different μ and σ can be obtained. The relative importance of μ and σ in the optimization framework can be mathematically quantified by a weighting factor, ω. To choose optimal values of c_0 and $\left(c_1, c_2, c_3\right)$, a unified decision-making index (*I*) can be modeled:

$$I = \omega\mu + \left(1 - \omega\right)\sigma. \tag{2.30}$$

For various choices of c_0 (for POD/interpolation) or $\left(c_1, c_2, c_3\right)$ (for POD/extrapolation), the choice that makes I smallest is the chosen parameter(s).

It is recognized that the computation of c_0 and $\left(c_1, c_2, c_3\right)$ by comparing the analytical error to the prediction error reduces the effectiveness of the a priori framework. However, these constants depend on a particular experimental setup and POD prediction resolution. Therefore, once these constants are determined by a benchmarking experiment for a particular experimental facility, they can be recurrently used for subsequent predictions. This method is extensively developed in [54].

An alternative approach can be developed by modeling error as:

$$e = \left(E_{\text{Prediction}} - E_{\text{Analytical}}\right). \tag{2.31}$$

In this approach, the computation of c_0 is conducted via the minimization of the inner product of e:

$$L = e' \bullet e. \tag{2.32}$$

The candidate space for c_0 is determined by the bisection method [59]. The efficiency of a numerical procedure can be defined by the number of iterations, n needed to achieve a given error, ε. For the bisection method, it is given by:

$$n = \log_2\left(\frac{\varepsilon_0}{\varepsilon}\right), \tag{2.33}$$

where, ε_0 is the size of parametric domain.

On the other hand, the analytical error for POD/Extrapolation is dependent on three arbitrary constants. One method to determine these constants is via iteration-based minimization of the decision-making index I. An alternative method is the conjugate gradient method-based optimization procedure. The ultimate purpose of analytical error is to match with prediction error:

$$E_{\text{Analytical}} \rightarrow E_{\text{Prediction}}. \tag{2.34}$$

$E_{\text{Analytical}}$ can be decomposed into two parts: one of these parts depends upon arbitrary constants and other part depends on time:

$$E_{\text{Analytical}} = F(t) \otimes g(c_i). \tag{2.35}$$

The determination of c_i can be modeled as a least-square problem:

$$F^T E_{\text{Prediction}} = F^T F g. \tag{2.36}$$

In Eq.(2.36), $F^T E_{\text{Preddiction}}$ is a column vector, $F^T F$ is a square symmetric matrix, and g is the vector with the constants c_i as elements. The constants can be determined by the conjugate gradient method [59].

Closure

This chapter developed a data-driven modeling strategy based on POD-based model order reduction. POD modes have several features useful for developing low dimensional models. First, POD modes can be computed by a logarithmic-time Power method. Therefore, the model can act as a highly efficient computing platform for design optimization. Then, the number of POD modes can be optimally chosen to control the prediction fidelity of the model. On the other hand, a priori error estimation is particularly useful for robust controller design.

CHAPTER 3
REAL-TIME DATA CENTER PROGNOSTIC MODEL

This chapter pertains to a measurement-based parametric model of data center rack inlet temperature with time as a parameter. This model improves the temporal granularity of measured temperature data in a DC. A measurement-based parametric reduced-order transient DC model can be used as a high-fidelity, high-dimensional, and near-real-time prediction tool. When time is used as a parameter, such reduced-order models can be used a real-time prognostic model. A prognostic model of rack inlet temperatures enables a real-time decision-making tool useful particularly during emergencies such as power outages. This chapter begins with the pertinent problem statement, followed by the hypothesis. The following sections are methodology, case study, results and discussions, and conclusions.

Problem Statement

Sever inlet temperature is a critical design parameter for data centers. ASHRAE TC 9.9 2011 guidelines recommend a 15-32 ^{0}C range as an allowable server inlet temperature band for a class-1 data center. Too high operating temperature leads to the risks of thermal failure of servers and compromised computational integrity. On the other hand, too low operating temperature leads to the risk of condensation on the electronic circuit board. Transient prediction of server inlet temperature is particularly important during various dynamic events such as power outages in data centers. During power outages, IT equipment is run on a cooling improvised environment. Different classes of data centers employ different dynamic cooling resource management protocols, as shown in Figure 16. It is evident that all classes of data centers follow a similar strategic pattern in response to a power outage: first, engagement of an emergency generator (t_1) and then, initiation of a cooling hardware response protocol (t_2). During these dynamic events, server rack inlet temperature prognostic is important for dynamic energy auditing. A desirable feature of this temperature prognostic model is the ability to predict in near-real-time. A real-time prognostic model is suitable for critical decision-making pertaining to thermal reliability such as whether server inlet temperature has reached an allowable threshold and advanced power management protocol (e.g. shutdown) needs to be initiated.

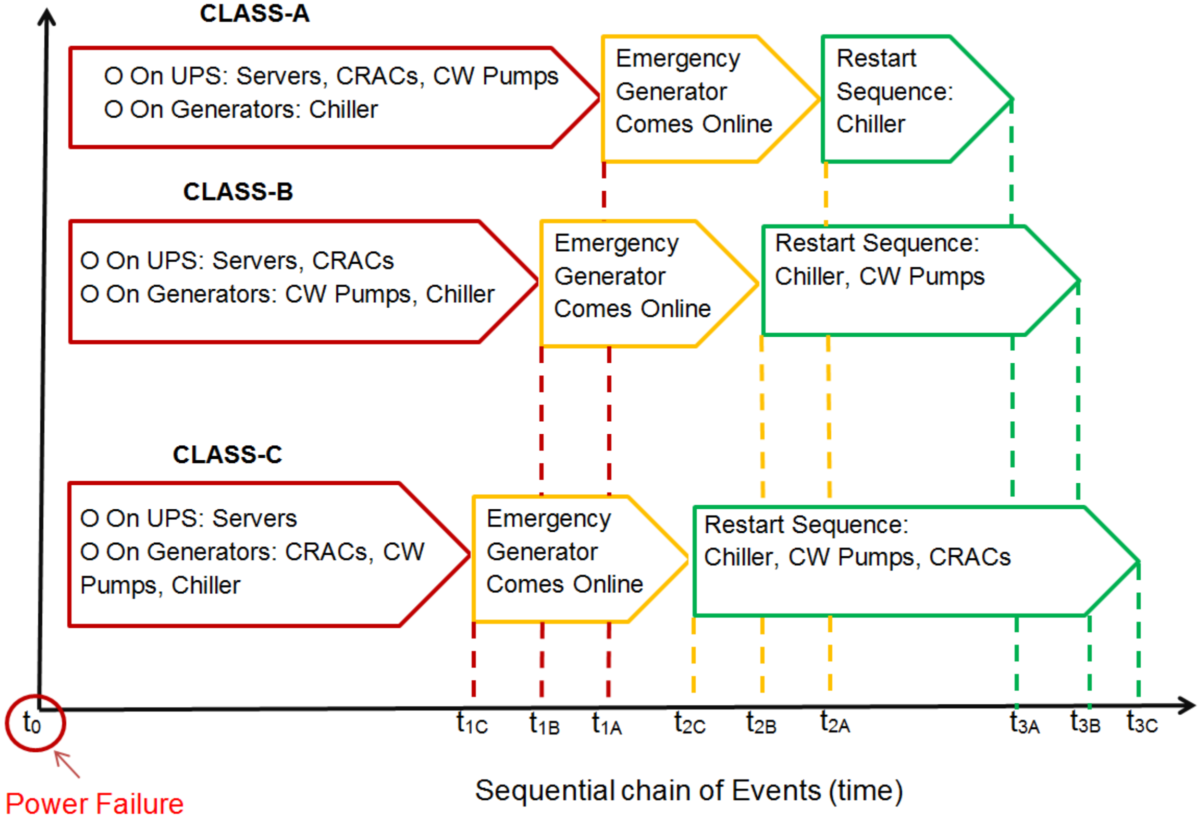

CLASS-A
O On UPS: Servers, CRACs, CW Pumps
O On Generators: Chiller
Emergency Generator Comes Online
Restart Sequence: Chiller
CLASS-B
O On UPS: Servers, CRACs
O On Generators: CW Pumps, Chiller
Emergency Generator Comes Online
Restart Sequence: Chiller, CW Pumps
CLASS-C
O On UPS: Servers
O On Generators: CRACs, CW Pumps, Chiller
Emergency Generator Comes Online
Restart Sequence: Chiller, CW Pumps, CRACs
t0
Power Failure
t1C
t1B
t1A
t2C
t2B
t2A
t3A
t3B
t3C
Sequential chain of Events (time)

Figure 16: Response protocol following a power outage in data centers (adapted from [60])

It is hypothesized that a near-real-time high-fidelity prognostic model can be developed via POD-based model order reduction and suitable regression operations. The input to the model is measured air temperature data at the rack inlet. The effectiveness of the POD-model in terms of accurate and efficient prediction of temperature data is demonstrated. The hypothesis is proved using the following case study.

Case Study

This case study focuses on an impulse response of data center air temperatures to a step-change in the capacity of a computer room air conditioning (CRAC) unit. This is an experimental case study. A CRAC unit is suddenly switched off at $t = -120$ s. After remaining inactive for 2 min, the CRAC unit is powered back at $t = 0$ s. The subsequent temperature evolution is observed experimentally.

Experimental Setup

The experiment was conducted in the CEETHERM Data Center Laboratory (located in Atlanta, GA at an elevation ~1,027' (313 m)). As shown in Figure 17, the experimental setup is a data center that employs a raised floor plenum supply and overhead plenum return air flow scheme. The servers and other IT equipment are mounted in cabinets, or racks, on a raised floor. An alternating ''cold aisle'' and ''hot aisle'' configuration is employed, where the inlet side of the server faces a cold aisle, and the outlet side faces a hot aisle. The computer room air conditioning (CRAC) unit supplies pressurized cold air into the underfloor plenum. The cold air flows up through perforated tiles, and is entrained into the servers by server fans. The hot air from the server outlets is cooled by chilled water circulating in air-to-water and rear-door heat-exchangers mounted on the rear cabinet doors prior to discharge into the hot aisle. It then returns to the CRAC though an overhead plenum for further cooling to the supply temperature. Fig. 3 shows the plan view of the experimental setup, which is populated with 16 standard size server cabinets or racks of height: 2,134 mm, depth: 1,067 mm, and width: 584 mm. The racks are arranged in an 8x2 architecture with alternating cold and hot aisles. The facility has three CRAC units. However, in the present case study, CRAC-1 is the only active unit which supplies cooling air at 4.6 kg/s at its 100% capacity. Additional pertinent specifications, including the hardware housed within the racks and their power dissipations are listed in Table-1. Racks are numbered as R-I, where $\mathrm{I} = 1 - 16$.

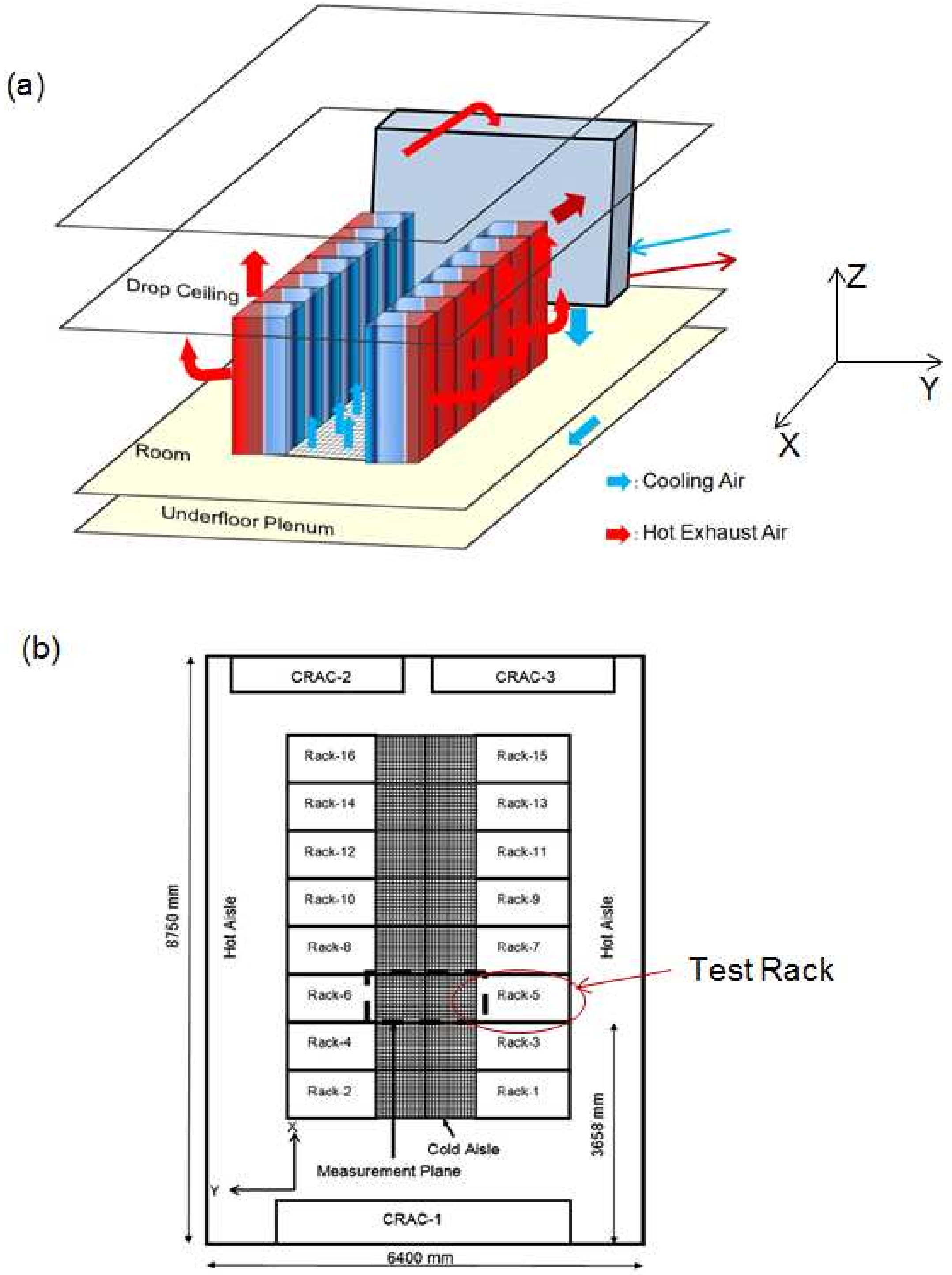


Figure 17: Experimental setup. (a) Underfloor plenum supply, front-to-rear rack flow, and drop ceiling return airflow scheme. (b) Plan view of the experimental setup. The facility has 16 racks, labeled Rack-1-Rack-16, and three CRAC units, labeled CRAC-1-CRAC-3. Racks are arranged in 8x2 alternating cold/hot aisle architecture. CRACs are

arranged in 2R configuration. The region with grids indicates perforated floor tiles in the cold aisle. Transient temperatures were measured in the cold aisle between Rack-5 and Rack-6, shown by black dotted box. Rack-5 is the test rack whose inlet temperature field is scrutinized.

Table 4: Specification of the experimental setup

Components	Specifications	Comments
Rack-1	5.2 kW	Network Rack
Rack-2	5.2 kW	Storage Rack
Rack-3	8.48 kW	IBM Blade Center
Rack-4	6.4 kW	IBM Blade Center
Rack-5	10.08 kW	IBM Blade Center
Rack-6	10.08 kW	IBM Blade Center
Rack-7	8.8 kW	IBM Blade Center
Rack-8	10.72 kW	IBM Blade Center
Rack-9	9.6 kW	IBM Blade Center
Rack-10	6.4 kW	IBM Blade Center
Rack-11	9.6 kW	IBM Blade Center
Rack-12	0	Empty
Rack-13	10.48 kW	IBM Blade Center
Rack-14	0	Empty
Rack-15	0	Empty
Rack-16	0	Empty
Perforated Tiles	610 mm x 610 mm; 56% Porosity	Passive Tile
Floor Plenum	914 mm Height	Cooling Air Supply
Room	3,048 mm Height	
Drop Ceiling Plenum	1,524 mm Height	Hot Air Exhaust

The thermometry, as shown in Figure 18, uses copper-constantan (T-type) thermocouples (TCs), made from 28 gauge (AWG) thermocouple wire (0.32 mm. diameter). A lumped capacitance analysis [61] in an air-driven convective environment (h~10 W/m^2.K) indicates a response time of the order of 1 s. As shown in Figure 18(d), each measurement domain has six grid-based thermocouple units; each comprising 21 T-type exposed junction thermocouples. Six thermocouple units are located at six different heights from the floor: 220 mm, 576 mm, 932 mm, 1288 mm, 1644 mm, and 2000 mm. As shown in Figure 18(c), 21 thermocouples are arranged symmetrically in a 2 ft. x 2 ft. (150 mm x 150 mm) square plane resulting in a distance between two neighboring thermocouples of 150 mm. Overall, there are 126 thermocouples in a measurement system. The thermometry uncertainty sources [62] include gain and offset, differential and integral non-linearity, quantization, noise, cold junction compensation, networking, acoustic noise and vibration. The measurement system is calibrated using an Omega® CL122 thermocouple calibrator (http://www.omega.com/pptst/CL120_134.html) and NIST traceable calibrated thermometer. The measurement chain calibration is conducted in the 10 ^{0}C-35 ^{0}C temperature range. With a 95% confidence interval, the average calibration error is estimated as $\left(0.49\ ^0\mathrm{C} \pm 0.19\ ^0\mathrm{C}\right)$.

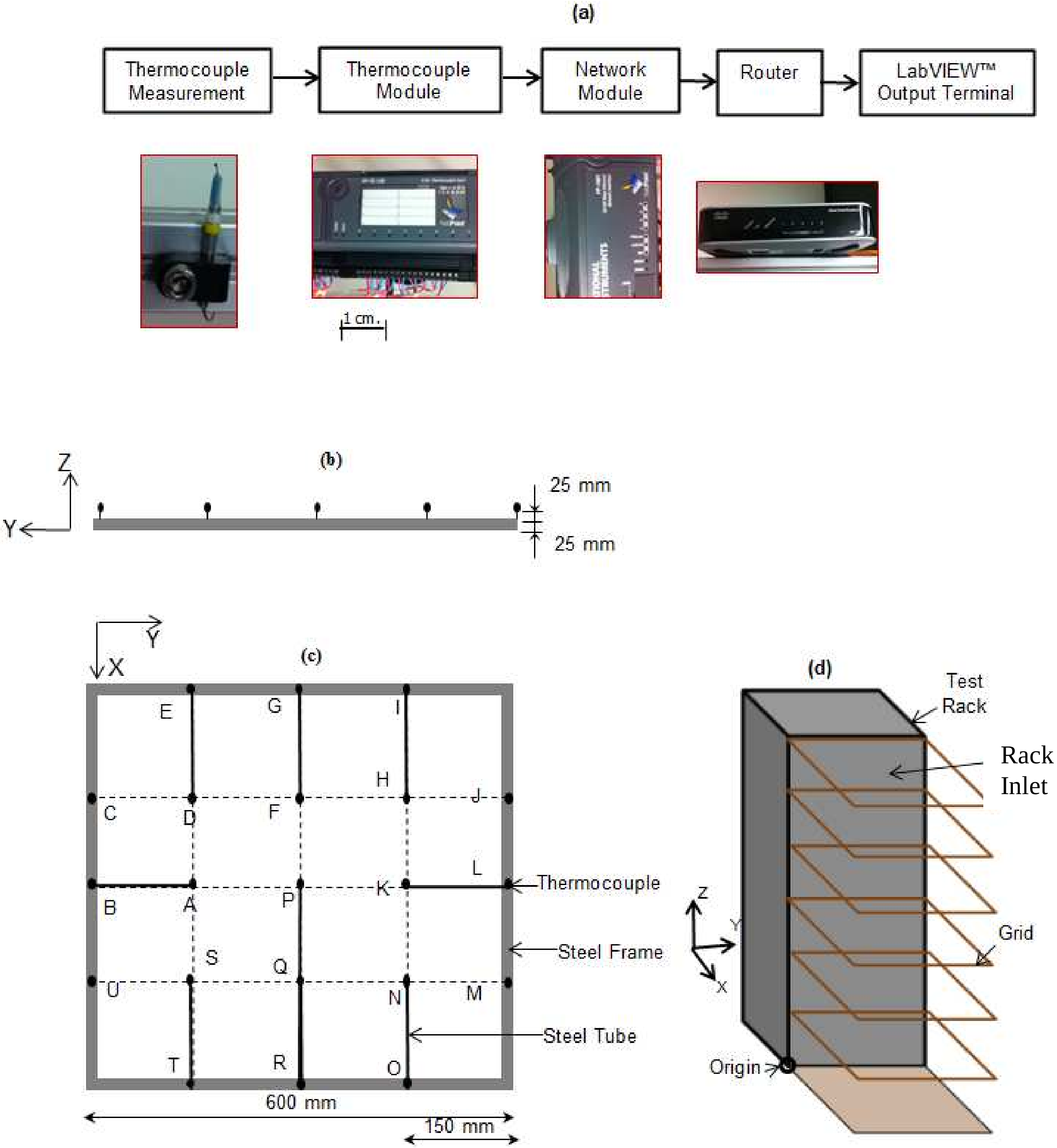


Figure 18: Details of air temperature data acquisition system. (a) Measurement chain consists of generating thermocouple-based temperature measurement data, processing at thermocouple module, processing at network module, transmitting processed data via a network router to the LabVIEW™-based output terminal. (b) Side view of the thermocouple measurement unit which is of 25 mm thickness. (c) Plan view of grid-based thermocouple measurement unit. Each unit is made of 600 mm x 600 mm steel frame and consists of 21 T-type copper-constantan thermocouples arranged in a square symmetry. The thermocouples form a grid-like structure with the distance between the

nearest neighbors being 150 mm. A, B, C,…,S, T, U are the spatial indexing of the thermocouples. (d) The six thermocouple grids are deployed at heights: 220, 576, 932, 1288, 1644, 2000 mm at the test rack exhaust.

Results and Discussion

Figure 19 shows the transient temperature response after CRAC-1 powered back at $t = 0$. The temperature responses are measured at the center of the perforated tile at the foot of the test rack. Assuming the bottom right corner of the test rack is the origin, the measurement points are: (300,300,220) mm, (300,300,576) mm, (300,300,932) mm, (300,300,1288) mm, (300,300,1644) mm, (300,300,2000) mm. After CRAC-1 resumption, cooling air enters the room through perforated tiles in the cold aisle. Although cooling airflow reduces the average air temperature in the cold aisle, the temperature reduction pattern is spatially disproportionate. As evident from Figure 19, temperature decreases gradually (~1 0C variation at 2000 mm height) near the top of the rack. This trend is attributed to hot-air recirculation near the top of the rack because of the favorable pressure gradient condition resulting from the mismatch of rack fan setting and CRAC supply set point. On the other hand, air temperature drops precipitously near the perforated tile surface—at (300,300,220) mm the temperature drop is in the order of ~10 ^{0}C, which is 10 times higher than that near the top of the test rack. This trend is consistent with the fact that the air temperature field near the perforated tile is dominated by the strong advection effect of highly-pressurized cooling air coming through the perforated tiles.

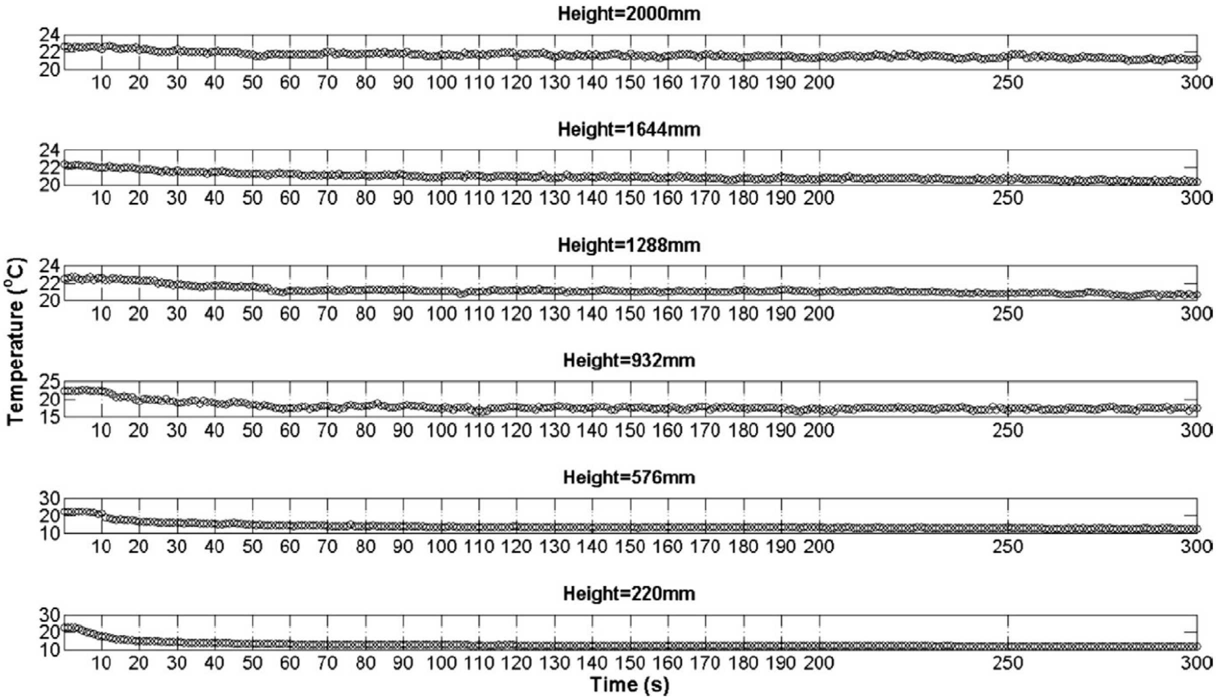


Figure 19: Transient air temperature evolutions at different heights (1,960 mm, 1,644 mm, 1,288 mm, 932 mm, 576 mm, 220 mm) at the test rack inlet. Near the top, the transient temperature variation is approximately equal to 1.5 ^{0}C and that near the bottom is about 10 ^{0}C.

Following the transient data acquisition, a temperature ensemble is constructed by taking snapshots of data at $t = 10, 20, ..., 190, 200$ s. Each snapshot compiles temperatures collected by 126 sensors in the cold aisle. Therefore, an ensemble of size 126x20 is developed. In this particular problem, the independent variable is spatial location, the dependent variable is time, and the output variable is air temperature. The functional problem statement for the response surface generation is:

$$T = f(x, y, z; t). \tag{3.1}$$

The semi-colon in Eq.(3.1) indicates the problem is parameterized in time with (x, y, z) as the independent variables.

Once the ensemble for this particular problem is compiled, the numerical procedure outlined in CHAPTER 2 is used to compute POD modes. Each POD mode is essentially an $m-$dimensional vector, with m equal to 126. There are 20 POD modes. The bar chart in Figure 20 shows the energy contents of different POD modes. It suggests the energy content for the first POD mode is more than 50% of the entire energy spectrum, and that of the second is about 10%.

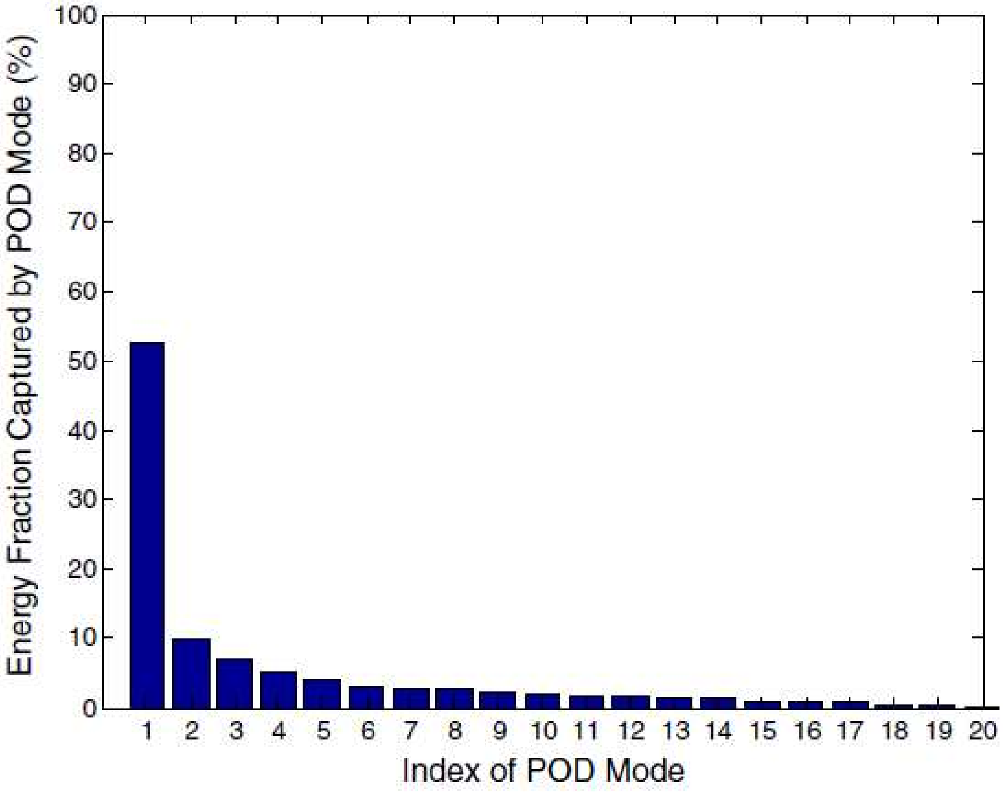


Figure 20: Relative energy contents of different POD modes

The positive skewness of the bar chart stems from the strictly non-increasing order of singular values of the ensemble. This pattern of POD modal space is leveraged to compute optimal POD basis space, as defined by Eq.(2.12). The factor C.E.P. in Eq.(2.12) depends on the accuracy requirement: a highly-accurate prediction scheme demands larger *C.E.P.* On the other hand, larger C.E.P. means higher *k* or retained POD modes. This trend is captured in Figure 21. While a crude model with *C.E.P.* =75% requires $k = 5$, a high-fidelity model with *C.E.P.* =99.5% requires $k = 18$. Between these two extremes, *k* varies non-linearly with *C.E.P.*—while *k* varies gradually between 75% - 95% *C.E.P.*, the variation becomes rather steep after the 95% *C.E.P.* limit. Since the focus of this chapter is to develop a high-fidelity prediction platform, a high *C.E.P.* =99% is chosen for the results reported in this chapter. The corresponding number of retained POD modes is equal to 17.

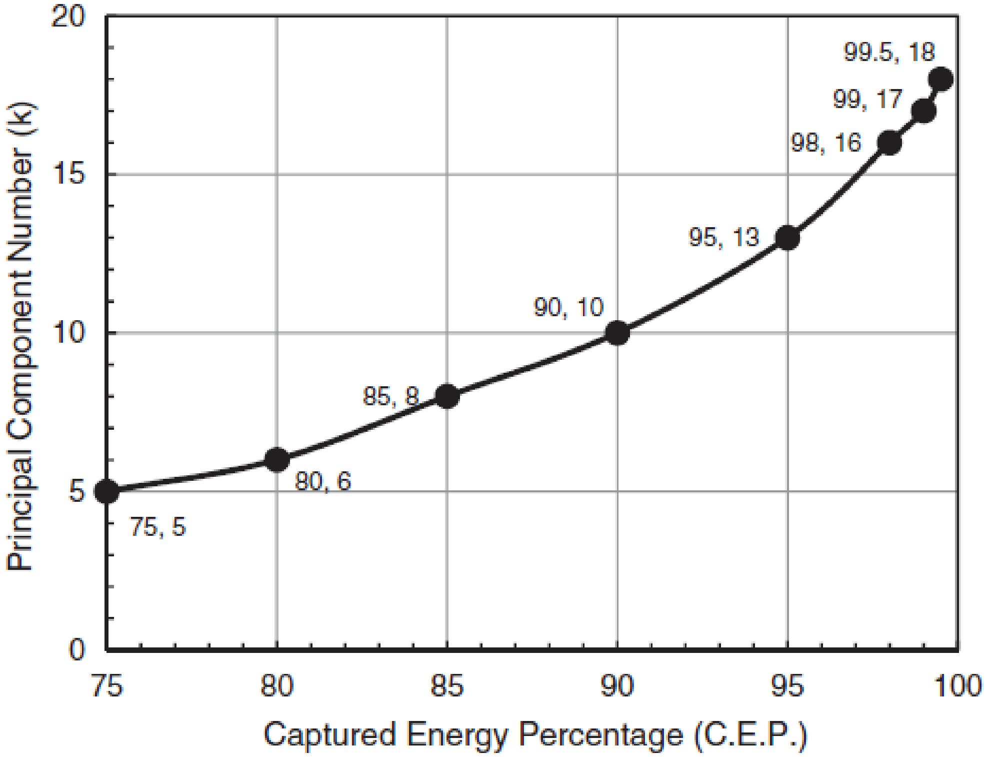


Figure 21: The variation of captured energy percentage (C.E.P.) vs. the number of retained POD modes (*k*)

Following the computation of the POD basis space, the POD coefficient vector needs to be computed. The POD coefficient vector is determined via parametric interpolation or extrapolation, depending upon the position of interrogation point in the parametric domain. If the parametric interrogation point lies within the range spanned by the parametric upper and lower bounds of the ensemble, statistical interpolation-based computation is required. Otherwise, statistical extrapolation is required.

For this case-study, the parametric space can be divided into two parametric zones: the first one is subspace spanned by $t_{\text{int}} \in (10, 200)$ s. In this zone, an arbitrary parametric point is chosen as 92 s. For this interrogation point, Figure 22 demonstrates the fidelity of POD-based temperature predictions in the spatial domain located at the test rack inlet plane. Figure 22(a) shows air temperature mapping at the test rack inlet. The black filled circles are the locations of temperature sensors. The temperature contour is produced by the Delaunay triangulation-based statistical interpolation of the measured temperature data. The POD-based algorithm is applied on the data ensemble, which is basically a compilation of transient temperature data collected at $t_{en} = [10, 20, 30, ..., 190, 200]$ s. Figure 22(b) shows POD-predicted temperature mapping at the test rack inlet at $t = 92$ s.. The POD-based temperature predictions resemble closely the measurement data. The locations of the hotspots and stratified temperature layers are correctly captured by the POD model. In fact, the deviations between experimental data and POD predictions are in the order of the calibration error, as shown in Figure 22(c). Since the error varies within a range of [-0.5, 0.4] ^{0}C and the maximum predictive uncertainty is 2.2%, the predictive framework can be considered high-fidelity with 98% confidence. Since $t = 92$ s is an arbitrary point for $t_{\text{int}} \in (10, 200)$ s, similar accuracy is expected for any interrogation point. The POD/interpolation-based prediction for the rack-inlet temperature at $t = 92$ s (intermediate to snapshots at $t = 90$ s and $t = 100$ s) requires 4 s with an Intel Core 2 Duo CPU at 2.54 GHz. This is definitely faster than an independent experiment. Therefore, it minimizes experimental data acquisition cost. On other hand, with 98% prediction accuracy, the model enables experimental data acquisition with lower grade temperature sensors. While a measurement frequency of 1 Hz demands 28 mil thermocouple sensors, a measurement frequency of 0.1 Hz requires 40 mil thermocouple wire. According to Omega© website (www.omega.com), the former costs $2 (~8%) more than the latter.

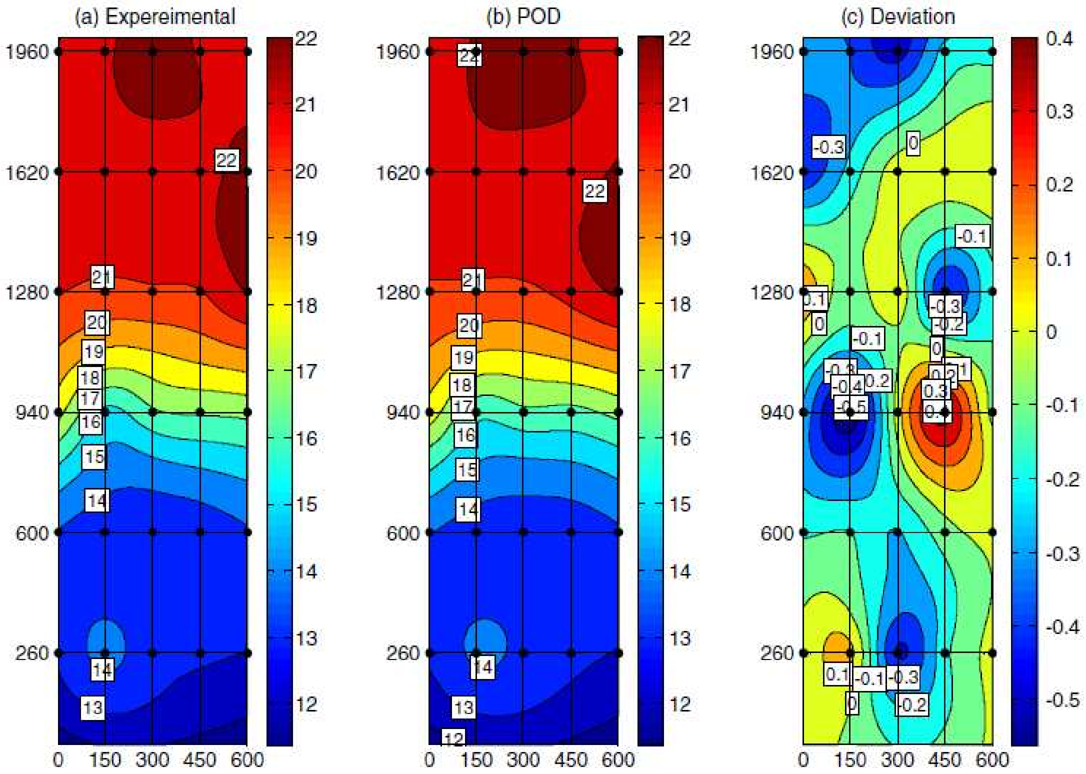


Figure 22: The contour plots for temperature distributions at the inlet of Rack-5 at $t = 92$ s. The horizontal direction of 600 mm length indicates the width of the test rack,and the vertical direction of 2000 mm length indicates the height of the rack. (a) shows experimentally-acquired temperature field. (b) shows POD-predicted temperature field. The predictions closely resemble the data. As shown (c), the absolute deviations between experimental data and POD predictions are within a scale of [-0.5 ^{0}C, 0.4 ^{0}C].

The black filled markers are the locations of temperature sensors. The contours are generated by Dalaunay triangulation.

As discussed in CHAPTER 2, an optimized analytical estimate of error bounds for the POD/ interpolation framework obviates the necessity of finding prediction error (2.14) which requires a posteriori independent experiments. This is useful for high-fidelity near-real-time controller design. As discussed in CHAPTER 2, the analytical error for the POD/interpolation framework is given by:

$$E_{\text{Analytical}}^{\text{POD/Interpolation}} = c_0 \sum_{i=k+1}^{n} \lambda_i . \tag{3.2}$$

The sum of the eigenvalues corresponding to the discarded POD modes is equal to 0.7933. The constant, c_0 is determined by two methods: the first approach is based on iterative computation of I as a function of c_0. It involves plugging in different values of c_0 and estimating values of c_0 that minimizes unified decision-making index, I as defined by Eq. (2.30). Table 5 documents different values of I for different values of c_0. It shows I reaches its minima at $c_0 = -0.01$.

Table 5: Iterative Method for Determination of c_0

c_0	**Average**	**Std. Dev.**	**w**	***I***
1.0	580.6	2683.6	0.5	1632.1
-1.0	0.728	0.057	0.5	0.3925
0.1	58.2	268.3	0.5	163.3
-0.1	0.05	0.031	0.5	0.0705
-0.75	0.529	0.057	0.5	0.293
-0.5	0.331	0.057	0.5	0.194
-0.25	0.134	0.054	0.5	0.094
-0.05	29.3	134.2	0.5	81.7
-0.01	0.06	0.055	0.5	0.0575

An alternative approach is based on the minimization of the inner product of the error vector, as defined by Eqs (2.31)-(2.32). The choice of c_0 is governed by the minimization of the inner product of the error vector. The choice of c_0 is driven by the bisection method. The chosen parametric domain is [-1,1] because the sum of the eigenvalues corresponding to the discarded POD modes is equal to 0.7933, which is in the same order as the prediction error. Figure 23 shows the bisection method-based computational procedure. The convergence criterion for this computation is set to 0.1. As predicted by Eq. (2.33), the number of iterations needed to determine the optimal value of *I* is equal to 7. Based on this criterion, the optimal value of c_0 is found to be equal to -0.01.

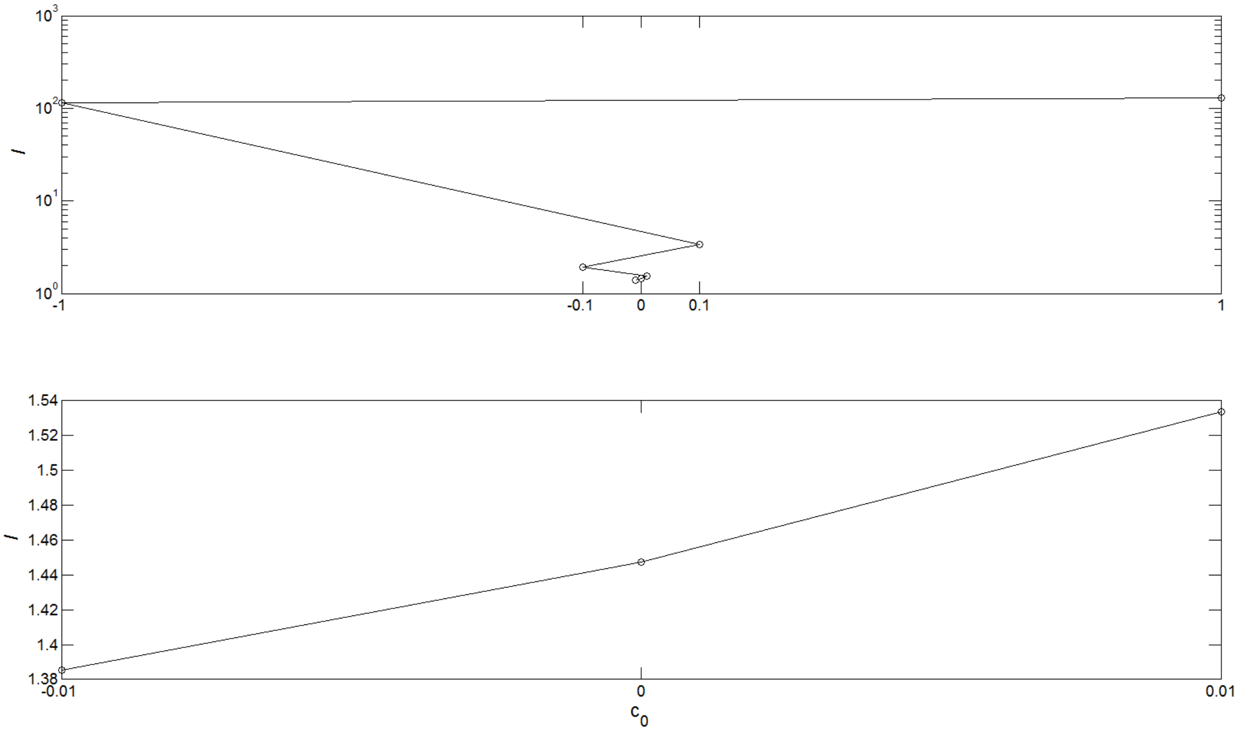


Figure 23: Minimization of Decision-making Index, *I* with respect to c_0.

For both methods, the values of c_0 are found to be identical. However, the bisection-based method is more systematic with predictable computational time.

Figure 24 shows prediction errors for the POD/interpolation scheme and corresponding analytical error estimate. As expected, the analytical error estimate remains flat in the interpolation time domain. The optimized constant, c_0 scales the magnitude of analytical error such that the absolute fractional deviation between analytical error and prediction error is minimized. Indeed, Figure 24 shows that the maximum deviation between prediction errors and analytical errors is equal to 0.1 ^{0}C. Therefore, the analytical error estimate, as shown by (3.2), can replace the prediction error within +/- 0.1 ^{0}C fidelity limit.

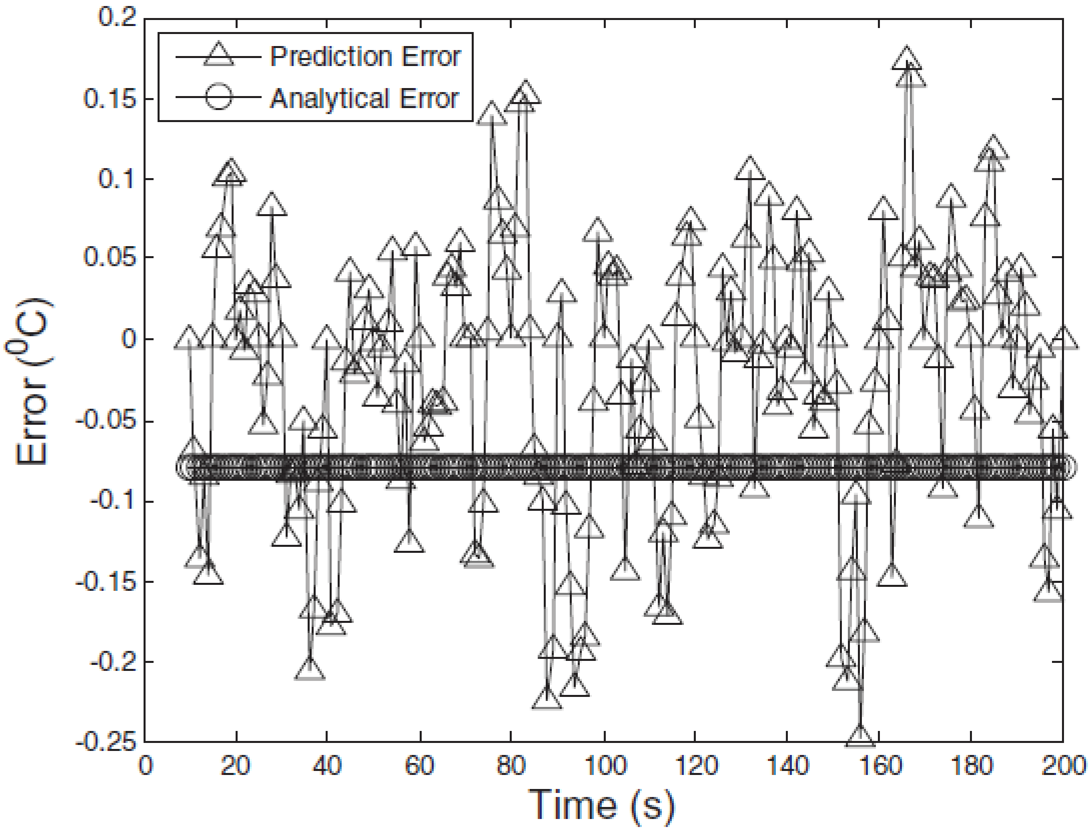


Figure 24: Analytical error bound for POD-based interpolation. The solid line with the triangular markers shows the transient deviations in prediction error between the experimentally-acquired temperature data and the POD-predicted temperature data. The solid line with the circular markers shows the analytically-determined transient deviation or error between the exact solution data and the POD-predicted temperature data.

The complementary parametric zone is spanned by time, $t \notin [10, 200]$ s. The POD coefficient computation in this zone requires parametric extrapolation in time. While the zone defined by time, $t : (t < 10 \text{ s})$ is of theoretical interest, the temporal zone spanned by time, $t : (t > 200 \text{ s})$ is of practical interest, particularly for the development of a near-real-time temperature prognostic model. It amounts to predicting new temperature data in future from the present temperature measurement in time, $t \in [10, 200]$ s. Such a capability is useful during thermal emergencies such as power outages. An arbitrary point in this parametric zone is chosen at $t = 207$ s. Figure 25 shows experimentally-measured temperature field (Figure 25(a)) at the test rack inlet at $t = 207$ s; corresponding POD predictions (Figure 25(b)); and the deviation (Figure 25(c)) between experimental data and POD predictions. A careful comparison between experimental data and POD predictions reveals moderate differences, which are reflected in the deviation of the scale of [-2.5 ^{0}C, 1.5 ^{0}C]. Indeed in some points such as (150, 1960) mm, the deviation is as high as 12.5% of the original data. Therefore, additional error analyses and suitable conditioning of POD/ Extrapolation prediction is critical for using it as a high-fidelity prediction platform.

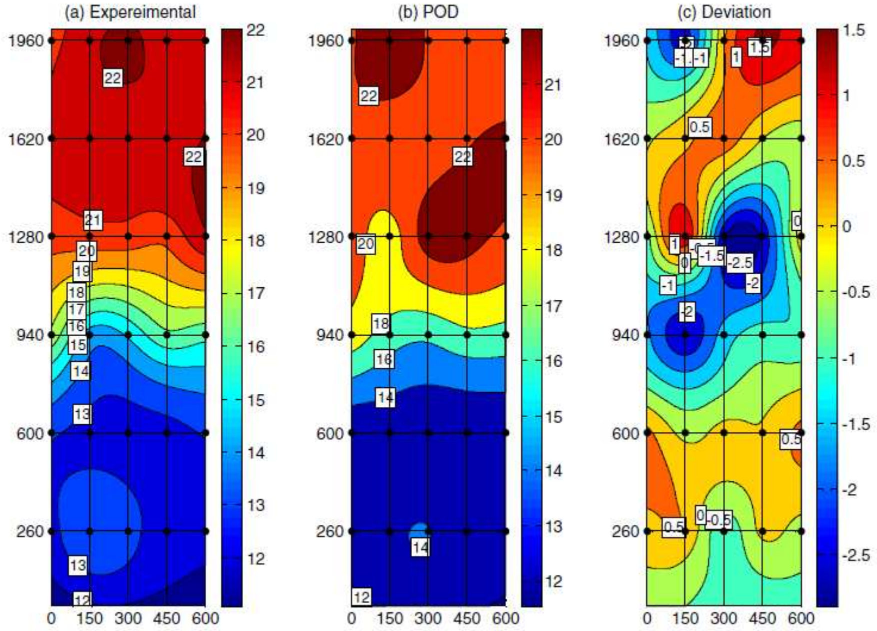


Figure 25: The contour plots for temperature distributions at $t = 207$ s at the inlet of Rack-5. The horizontal direction of 600 mm length indicates the width of the rack, and the vertical direction of 2000 mm length indicates the height of the rack, (a) shows the experimentally-acquired temperature field and, (b) shows the POD-predicted temperature field. The POD-based algorithm uses extrapolation to compute the temperatures. The temperature scales are almost identical [14 ^{0}C, 22 ^{0}C]. Indeed, as shown in (c), the deviations between experimental data and POD-predicted data are within a scale of [-2.5

^{0}C, 1.5 ^{0}C]. The black filled markers are the locations of temperature sensors. Remaining data points are produced by Delaunay triangulation.

To define a reliable extrapolation window, the scale of the temperature difference is chosen as $\Delta E_{\text{Scale}}^{\text{Measurement}} = (20-12)^0 C = 8^0 C$. This scale represents the difference between the minimum initial temperature and the temperature of supplied cooling air. Indeed, it is a characteristic of the thermal system involved in this case study. The scale factor, f, in Eq. (2.15) is arbitrary chosen to be 0.25. Based on these arbitrarily chosen parameters, the extrapolation horizon is calculated to be equal to 24 s i.e. the present scheme can extrapolate till $t = 224$ s within the specified error limit. Once E.H. is derived, the determination of $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$ requires identifying the case-specific constants (k, h^p, l, θ) and conducting the optimization procedure for identifying the arbitrary constants (c_1, c_2, c_3). The case-specific constants depend upon the experimental setup and case-specific conditions. The time-step for the POD/extrapolation framework is $k = 1$, since the extrapolation is carried out at a frequency of 1 Hz beyond $t = 200$ s. The normalized length scale (h^p) is defined as the ratio of the distance between two neighboring sensors (=150 mm in this case), and the characteristic length of the measurements system (=600 mm is the length of the square grid). Hence, it is calculated: $h^p = 0.25$. The number of snapshots included in the temperature ensemble is equal to 20: $l = 20$. The non-dimensional time, θ, defined as the time normalized against end of the transient measurement window (=300 s in this case). Hence, it is calculated $\theta \in [0.67, 0.74]$ for the derived E.H. The optimization procedure for determining arbitrary constants (c_1, c_2, c_3) is outlined in Table 6. The 'Average' column lists the average of e, and the 'Std. Dev.' column lists the standard deviation of e. The fractional difference, e, between $E_{\text{Analytical}}^{\text{POD/Extrapolation}}$ and $E_{\text{Prediction}}^{\text{POD/Extrapolation}}$ is calculated based on Eq.(2.25). The unified decision-making index (I) is calculated based on an optimization weightage, $\omega = 0.5$, which is arbitrarily assigned. As listed in Table 6, the minimum value of I is equal to 1 which corresponds to $(c_1, c_2, c_3) = (-15, 1.7, 2)$. Based on these constants, the analytical errors are estimated by Eq. (2.24). Figure 26 shows analytical errors along with prediction errors.

Table 6: The optimization procedure for the determination of (c_1, c_2, c_3). For different combinations of (c_1, c_2, c_3), the unified decision-making indices (I) are calculated. The combination $(c_1 = -15, c_2 = 1.7, c_3 = 2)$ is the best choice because it optimally minimizes I

c$_1$	**c$_2$**	**c$_3$**	**Average**	**Std. Dev.**	**w**	***I***
-10.0	1.8	2.0	17.5	13.8	0.5	15.6
-5.0	1.8	2.0	33.7	27.9	0.5	30.8
-15.0	1.8	2.0	1.5	0.7	0.5	1.1
-15.0	2.0	2.0	3.1	1.2	0.5	2.1
-15.0	1.5	1.5	4.0	4.9	0.5	4.5
-15.0	1.8	1.8	1.3	1.6	0.5	1.4
-15.0	1.8	2.0	1.7	0.8	0.5	1.2
-15.0	1.7	2.0	1.3	0.6	0.5	1.0
-16.0	1.7	2.0	2.3	3.6	0.5	2.9
-14.0	1.7	2.0	4.3	2.2	0.5	3.3

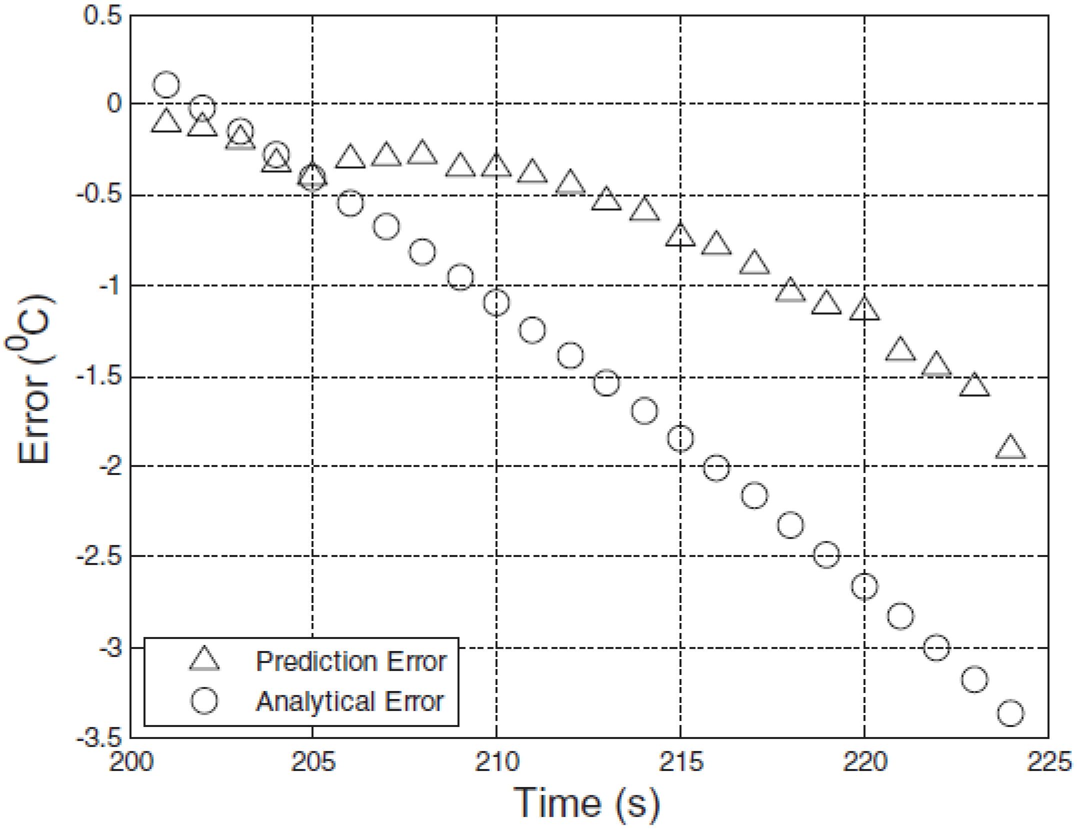


Figure 26: Analytical Error Bound for POD-based extrapolation determined by iterative procedure. The triangular markers show the transient deviations between the experimentally-acquired temperature data and the POD-predicted temperature data or prediction error. The circular markers show the analytically-determined transient deviation between the exact solution data and the POD-predicted temperature data or analytical error.

The accuracy of the analytical error estimate is improved by the conjugate gradient method-based optimization procedure. The constants are determined as: $(c_1, c_2, c_3) = (3.1, -0.02, -0.02)$. Figure 27 shows analytical errors along with prediction errors.

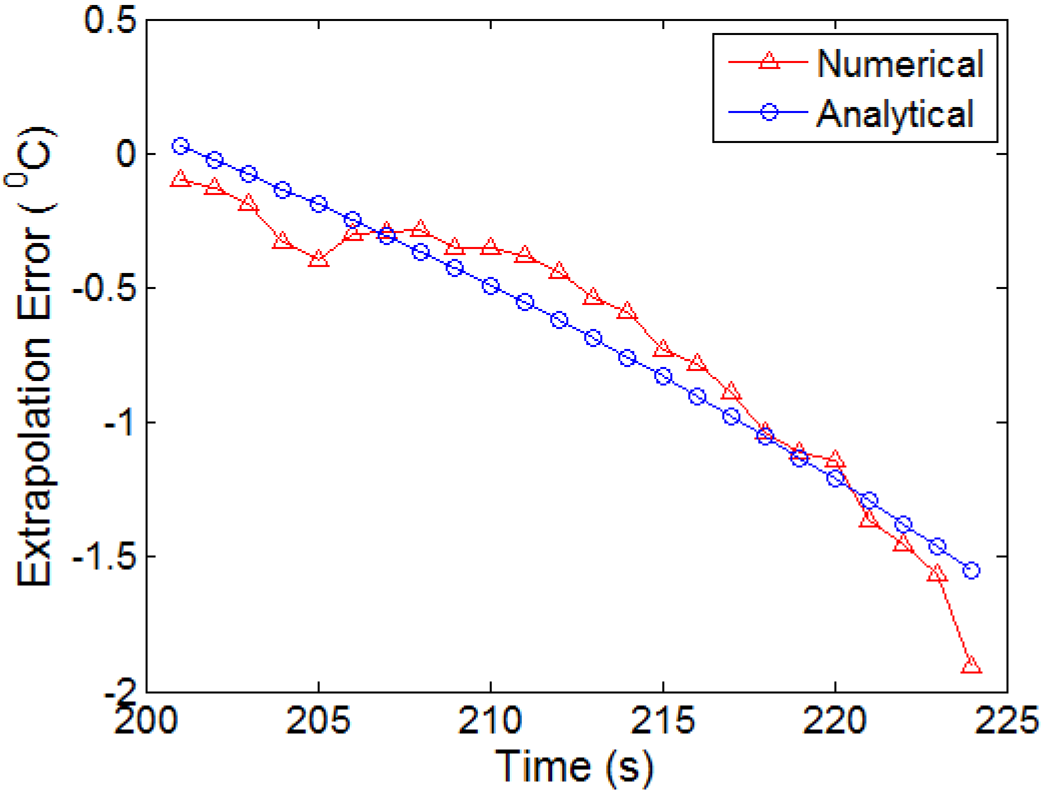


Figure 27: Analytical Error Bound for POD-based extrapolation determined by the conjugate gradient method. The triangular markers show the transient deviations between the experimentally-acquired temperature data and the POD-predicted temperature data or prediction error. The circular markers show the analytically-determined transient deviation between the exact solution data and the POD-predicted temperature data or analytical error.

Closure

Using POD-based model order reduction, a measurement-based framework is developed that improves the temporal resolution of the measured temperature data. The framework is useful as a real-time thermal prognostic tool. The effectiveness of the framework is analyzed in two time windows: one is within the measurement domain (POD/Interpolation), and another is outside of the measurement upper limit (POD/Extrapolation).

It has been shown that the POD/ Interpolation framework predicts air temperatures with 2% uncertainty. An a priori error estimate for the POD/Interpolation scheme is computed by a semi-analytical approach based on the bisection method. Determining the a priori error estimate is particularly important for robust controller design. The POD/ Interpolation framework effectively reduces the sampling frequency by 90% (from 1 Hz. to 0.1 Hz.). Such down-sampling allows low-grade temperature sensors to be used for experimental measurements. On the other hand, the POD/ Extrapolation framework predicts air temperatures with 10% uncertainty. An a priori error estimate for POD/Interpolation scheme is computed by a semi-analytical approach based on the conjugate gradient method. The POD/ Extrapolation framework is particularly useful for thermal prognostic during power outages.

CHAPTER 4
POD-BASED FRAMEWORK FOR IMPROVING SPATIAL RESOLUTION OF MEASURED TEMPERATURE DATA

This chapter pertains to a measurement-based POD framework for improving spatial resolution of measured temperature data.

Problem Statement

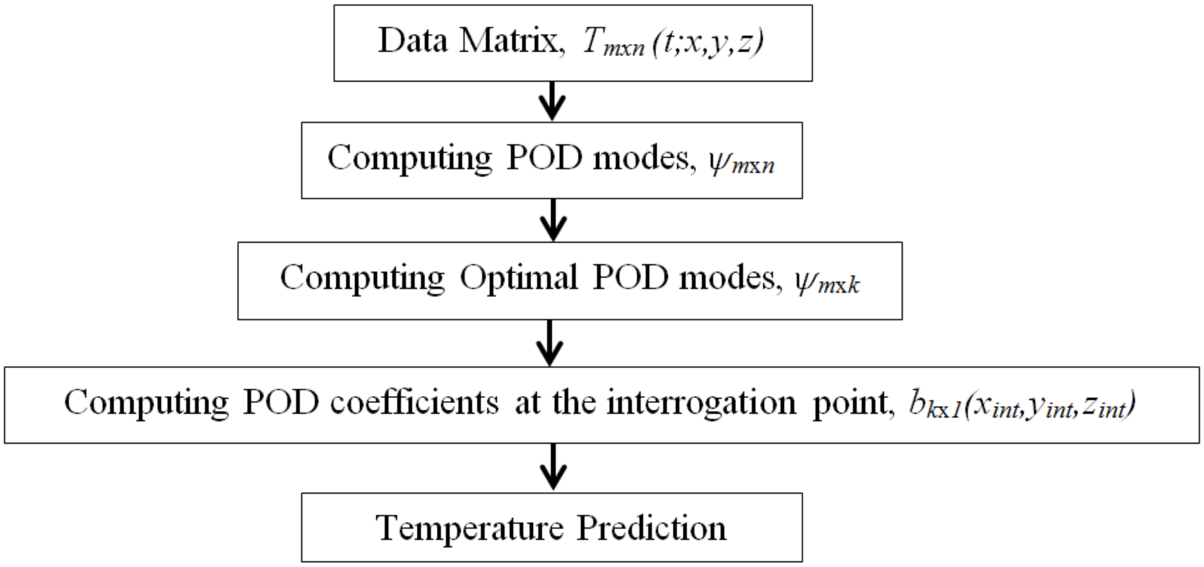
In order to avoid resource over- or under-provisioning, a real-time demand-aware cooling control system based on online temperature monitoring is required. A measurement-based monitoring framework needs to be supported by some modeling technology because temperature gradients in DCs can be quite large. For example, one might find air temperature at the corner of a server inlet differs by 10 ^{0}C compared to the center of the server inlet. Therefore, it is imperative to measure temperature data by sensors deployed at multiple strategic locations to compute reliable temperature distributions to gain meaningful insight from the real-time measurements. The design of a measurement system involves resolution of trade-offs between density of sensors, location of sensors, and their measurement frequency. This chapter focuses on developing a measurement-based technique to improve spatial resolution of measured temperature data; thereby, to utilize a given number of sensors optimally. It is hypothesized that such a measurement-based framework can be developed by using a POD-based model order reduction technique with spatial locations as the parameters. The hypothesis is explored using an experimental case-study as described in the subsequent sections. Tackling spatial locations as model parameters requires adjustment of the approach shown in CHAPTER 2.

Methodology

A measurement-based reduced-order model of transient air temperature is developed with time as the independent variable and spatial location as the parametric variable. The functional form of the temperature response surface is:

$$T = f\left(t; x, y, z\right). \tag{3.3}$$

The measurement-based reduced-order air temperature model is developed via a POD-based statistical algorithm. Figure 28 shows the proposed POD-based algorithm.


Data Matrix, $T_{mxn}(t;x,y,z)$
Computing POD modes, ψ_{mxn}
Computing Optimal POD modes, ψ_{mxk}
Computing POD coefficients at the interrogation point, $b_{kx1}(x_{int},y_{int},z_{int})$
Temperature Prediction

$$T(t; x_{\text{int}}, y_{\text{int}}, z_{\text{int}}) = \sum_{i=1}^{k} \psi_i\left(t\right)\, b_i\left(x_{\text{int}}, y_{\text{int}}, z_{\text{int}}\right)$$

Figure 28: POD-based reduced-order modeling algorithm with spatial location as the parameters. For a given time interval, the algorithm is applied on an ensemble of transient

temperatures, $T\left(t; x_i^{en}, y_i^{en}, z_i^{en}\right)$. The temperature predictions are computed for the interrogation point, $\left(x^{\text{int}}, y^{\text{int}}, z^{\text{int}}\right)$.

As shown in the flowchart in Figure 28, the data-driven algorithm consists of five major mathematical steps:

1) *Compilation of data matrix*: The data matrix compiles the temperature data. For the data-driven algorithm, the data matrix forms the corresponding problem instance: each column of a data matrix includes a transient temperature signal collected at a particular sensor location. The sampling interval of the temperature signal is Δt over a domain [0-*t*]. The infinite domain problem of live streaming can potentially be reduced to a finite dimensional problem by estimating the signal settling time, *t*, when the temperature signal reaches its steady state. Several sensors are deployed to yield a $T_{m\times n}\left(t; x, y, z\right)$ data matrix. The row rank, *m*, of the data matrix informs the length of the transient temperature signal. On the other hand, the column rank, *n*, of the data matrix informs the number of sensors deployed.

2) *Computation of POD modes*: POD modes are computed by the power method-based numerical algorithm as discussed in CHAPTER 2.

3) *Formation of optimal basis space from POD modes*: The optimal number of POD modes is computed by Eq. (2.12). The pertinent discussion is documented in CHAPTER 2.

4) *Computation of POD coefficients*: POD coefficients capture location-dependent parametric components of the response surface. The numerical procedure to compute

POD coefficients involves computation of the coefficient matrix. The rows of the coefficient matrix compile weighting factors for the corresponding POD modes. The coefficient matrix is computed by taking the dyadic product of the pseudo-inverse of the POD modes and the data matrix:

$$B = \psi^{+} \otimes T. \tag{3.4}$$

ψ^{+} is the pseudo-inverse of the POD mode matrix, ψ. B is an n-by-n matrix. The rows of B indicate the weighting factors for the corresponding POD modes. The columns of B are the characteristics of the spatial locations of the sensors. It is assumed that the coefficient vector at a new spatial location, termed the POD coefficient (b), lies in the column space of B. The literature reports various methods for the mapping: $B_{n\times n} \to b_{n\times 1}$. The most widely used method is the Galerkin projection [63]. In this method, the conservation equations (e.g. energy equation) are projected into the POD modal space and eventually discretized into a system of simultaneous linear algebraic equations. The Galerkin projection-based method is further simplified by using the flux matching approach [64]. Nevertheless, the Galerkin projection is suitable for a simulation-generated highly resolved data matrix. On the other hand, the scarcity of experimental data dictates the application of statistical methods such as spline-based interpolation [31], kriging [46]. However, the present study deals with experimentally-acquired temperature data in a measurement domain, susceptible to hot spots (abrupt change in temperature gradient). Therefore, a conditional procedure is proposed: at first, POD coefficients are generated using temperature sensors located at the boundary and the geometric center of the interrogation domain. If the resulting POD prediction uncertainty is more than a pre-assigned tolerance criterion (arbitrarily assumed to be equal to 5% in this study), a new sensor arrangement is needed. The choice of new sensor locations is guided by a physics-based reduced-order model of forced convective local airflow field. The mathematical model for POD coefficients is:

$$b_{k\times 1} = B_{k\times n} \otimes C_{n\times 1}. \tag{3.5}$$

Each element $c_i \left(\in C_{n\times 1}\right)$ indicates weighting factors for POD coefficients corresponding to different sensor locations. The essential idea of determining $C_{n\times 1}$ is based on determining isothermal zones in the air temperature field and ascribing the influence of the neighboring sensors on an interrogation location. Therefore, the determination of $C_{n\times 1}$ is domain-dependent and is discussed in detail in the results and discussion section. After $b_{n\times 1}$ is determined, an optimal POD coefficient, $b_{k\times 1}$ is extracted (ref. to Eq. (2.12)).

5) Temperature prediction: The temperature prediction at a new spatial location is given by:

$$T_{m\times 1}\left(t; x_{\text{int}}, y_{\text{int}}, z_{\text{int}}\right) = \psi_{m\times k}\left(t\right) \otimes b_{k\times 1}\left(x_{\text{int}}, y_{\text{int}}, z_{\text{int}}\right). \tag{3.6}$$

For a fidelity check, temperature data, acquired independently at the interrogation points, are compared with predictions. In this context, the prediction uncertainty is defined as the uncertainty in predicted local air temperatures. The comparison is quantified by rank correlation coefficients (ρ) and relative root mean square errors (rmse):

$$\rho = \frac{\sum_i \left(T_i^{data} - \overline{T^{data}}\right)\left(T_i^{prediction} - \overline{T^{prediction}}\right)}{\sqrt{\sum_i \left(T_i^{data} - \overline{T^{data}}\right)^2 \sum_i \left(T_i^{prediction} - \overline{T^{prediction}}\right)^2}}. \tag{3.7}$$

$$rmse = \sqrt{\frac{\sum_i \left(T_i^{data} - T_i^{prediction} \middle/ T_i^{data} \right)^2}{n}}. \tag{3.8}$$

Based on the mathematical procedure outlined, a functional algorithm can be developed on optimal utilization of available temperature sensors. For the sake of simplicity, the algorithm is applied on planar temperature data. This algorithm has two stages: the first one involves a geometry-based approach, and the second one involves a physics-based approach. The physics-based approach is invoked only when the geometry-based approach fails to satisfy a pre-assigned tolerance criterion (5% relative deviation).

The steps for the geometry-based algorithm are:

1) *Data Acquisition:* It is assumed that the number of available sensors is equal to *N*. At first, one sensor is deployed at the geometric center. The remaining sensors are distributed equally on four edges of the interrogation plane. Therefore, each edge has $\left(\frac{N-1}{4}\right)$ sensors. Of these $\left(\frac{N-1}{4}\right)$ sensors, one sensor is placed at the center of the edge. Remaining sensors are placed symmetrically with respect to the center. Given that a corner point is shared by the two edges, no sensor is placed there. The measured temperature distribution is computed via a statistical interpolation technique such as Delaunay triangulation [65].

2) *POD Mode Computation:* Optimal POD basis space is computed using the method discussed in CHAPTER 2.

3) *POD Coefficient Computation for an Interrogation Location:* The relative location of an interrogation point is determined with respect to the sensor points in the measured temperature distribution. The POD coefficient for an interrogation point is

determined by taking the average of the POD coefficients corresponding to the sensor points lying in the same isothermal zone.

4) *POD-based Temperature Computation*: The interrogation temperature is computed by Eq.(3.6). The temperature predictions are compared with the corresponding experimentally-measured data. If the percentage deviation (defined by Eq.(3.8)) is more than a pre-assigned tolerance criterion, the framework can be considered to be unreliable.

In case the geometry-based algorithm fails to satisfy the tolerance criterion, it is recommended to follow a physics-based algorithm. The steps for the physics-based algorithm are:

1. *Data Acquisition:* At first, the forced-convective flow field is estimated either by an approximation model or by a coarse-grained CFD model. Depending on the directions of the temperature gradients, the temperature field is segmented into different zones. In each temperature segment, the direction of the steepest temperature gradient is estimated. Based on that estimation, the temperature sensors are optimally distributed.

The remaining steps for the physics-based algorithm are identical to steps 2-4 in the geometry-based algorithm.

Experimental Data Acquisition

The experimental data for this case-study are acquired in the CEETHERM Data Center Laboratory (located in Atlanta, GA at an elevation ~1,027’ (313 m)). As shown in Figure 29(a), the experimental setup employs an underfloor plenum supply and an overhead drop ceiling return air flow scheme. The height of the DC room is 9 ft. (~2.75 m) with an under-floor plenum of height 3 ft. (~0.9 m), and drop ceiling height of 5 ft. (~1.52 m). Figure 29(b) shows the plan view of the experimental setup which is populated with 10 standard size server cabinets or racks of height: 2,134 mm, depth: 1,067 mm, and width: 584 mm. The racks are arranged in a 5x2 alternating cold aisle/hot aisle architecture. The facility has three CRAC units. However, for this case-study, only CRAC-1 remains active. Operating at 100% capacity, CRAC-1 supplies cooling air of 75 ^{0}F (23.9 ^{0}C) temperature at 6.7 m^3/s (~14,200 CFM) volumetric flow rate. The temperature measurement is conducted in the measurement zones shown by the white squares in Figure 29(b). These zones correspond to the cold and hot aisles of the test rack. Figure 29(c) shows a photograph of the test rack, which contains four vertically-stacked 10-U (17.5 inches~444.5 mm) server simulators. The heat load and the fan speed of a server simulator are controlled from the control unit [66] shown in Figure 29(c). The heat load switches included are: 250 W, 500 W, 1,000 W, 1,000 W, and 2,000W. The fan airflow rate can be modulated to ten different levels via a dial knob. At the full capacity, a server simulator fan supplies 650 CFM (~0.3068 m^3/s) airflow [66].As discussed in CHAPTER 3, a thermocouple grid deployed in a three-dimensional telescopic mechanism is used for the thermometry.

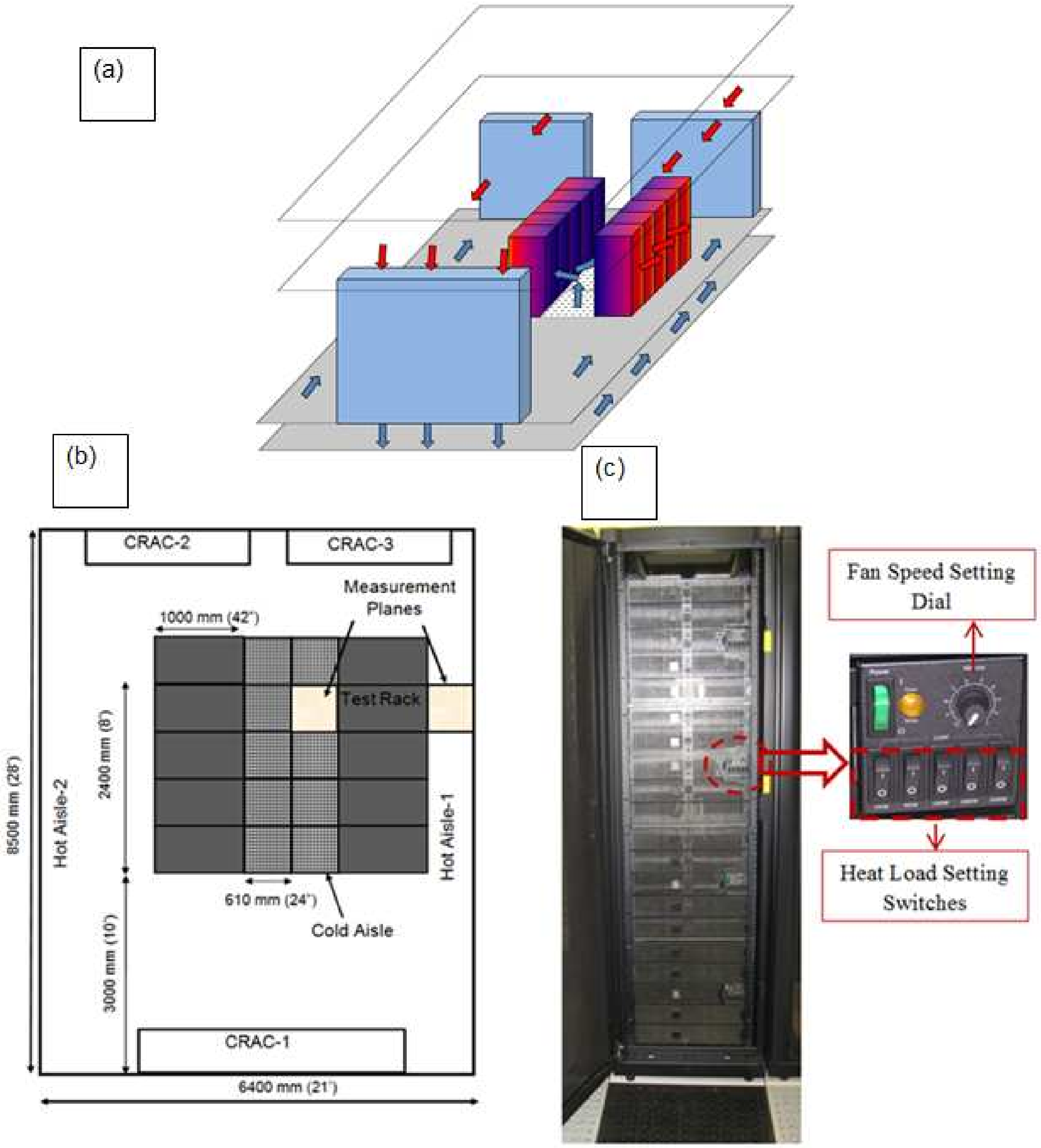


Figure 29: Experimental setup. (a) Underfloor plenum supply, front-to-rear rack flow, and drop ceiling return airflow scheme. (b) Plan view of the experimental setup. The facility has 10 racks and three CRAC units, labeled CRAC-1-CRAC-3. Racks are arranged in 5x2 alternating cold/hot aisle architecture. CRACs are arranged in 2R configuration. The region with grids indicates perforated floor tiles in the cold aisle. (c) Photograph of the test rack, which is a server simulator rack, showing fan speed setting dial and heat load control switches.

Case Study

In the present study, the test rack is suddenly switched to 20 kW power by setting each server simulator to 5000 W (this is accomplished by turning the 1000 W, 2000 W, and 2000 W switches to on). Each server simulator fan is kept at a setting which corresponds to 650 CFM (~0.3068 m^3/s) airflow [66]. The remaining racks in the facility were switched off during the experiment. The air temperature response is measured by 126 thermocouples deployed in the cold and hot aisles, as shown in Figure 30.

Results and Discussion

It is imperative that a sensor fusion algorithm be validated in measurement planes that offer sufficiently large temperature gradients. Therefore, the remaining study focuses on the hot aisle. Table 7 shows the standard deviation of measured temperature data at six different heights in the hot aisle. In Table 1 the plane at height 2,000 mm has the highest standard deviation of 3.1 ^{0}C. For prognostic-based thermal reliability modeling, this high-temperature plane is critically important because of its proximity to the fire suppression system. In addition, another measurement plane, located at 150 mm distance from the exhaust of the test rack, is chosen. The standard deviation of temperature data in this plane is equal to 3 ^{0}C. Due to its proximity to the server outlets, the temperature in this plane is very sensitive to the server IT workload variation. Therefore, rapid temperature assessment of this plane facilitates optimal dynamic cooling resource provisioning in DCs.

Table 7: Standard deviation of hot aisle air temperatures at different heights

h	**Standard Deviation**
mm	**^{0}C**
2000	3.1
1644	3.1
1288	1.6
932	1.7
576	0.9
220	1.8

Figure 30 shows the three dimensional arrangement of 126 TCs in the hot aisle. The rack exhaust is located at x=0 mm. Two temperature planes are chosen at z=2,000 mm (which includes 21 TCs) and at x=150 mm (includes 30 TCs). After identifying these two measurement planes, two independent POD-based analyses are conducted, and their respective modeling fidelities are estimated.

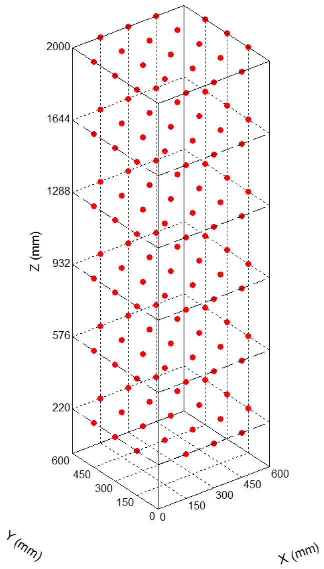


Figure 30: Sensor arrangement in the hot aisle. The filled circles indicate thermocouples.

The effectiveness of the proposed functional algorithm as discussed in the Methodology section is verified at planes: z=2,000 mm and x=150 mm. Figure 31 shows optimal sensor utilization strategies for z=2,000 mm (Figure 31(a)) and the x=150 mm (Figure 31(b)) planes. The sensor arrangement in the z=2,000 mm plane is obtained in the geometry-based algorithm. On the other hand, the geometry-based technique fails in the x=150 mm plane due to a complex airflow pattern in this plane. In this plane, the physics-based algorithm is employed.

The filled circles in Figure 31 are the locations of TCs. The data matrix compiled by the temperature signals acquired by these TCs is sufficient for the POD-based data compression algorithm to predict temperature data at the locations marked by the open circles. For the fidelity verification of the algorithm, the POD-based local air temperature predictions are subsequently compared with the corresponding experimental data. This choice of the sensor arrangement is driven by the estimation of isothermal zones in the temperature field and the minimization of sensor requisition in those zones. The choice of sensor topography has a significant impact on the subsequent POD coefficient computation (as discussed later).

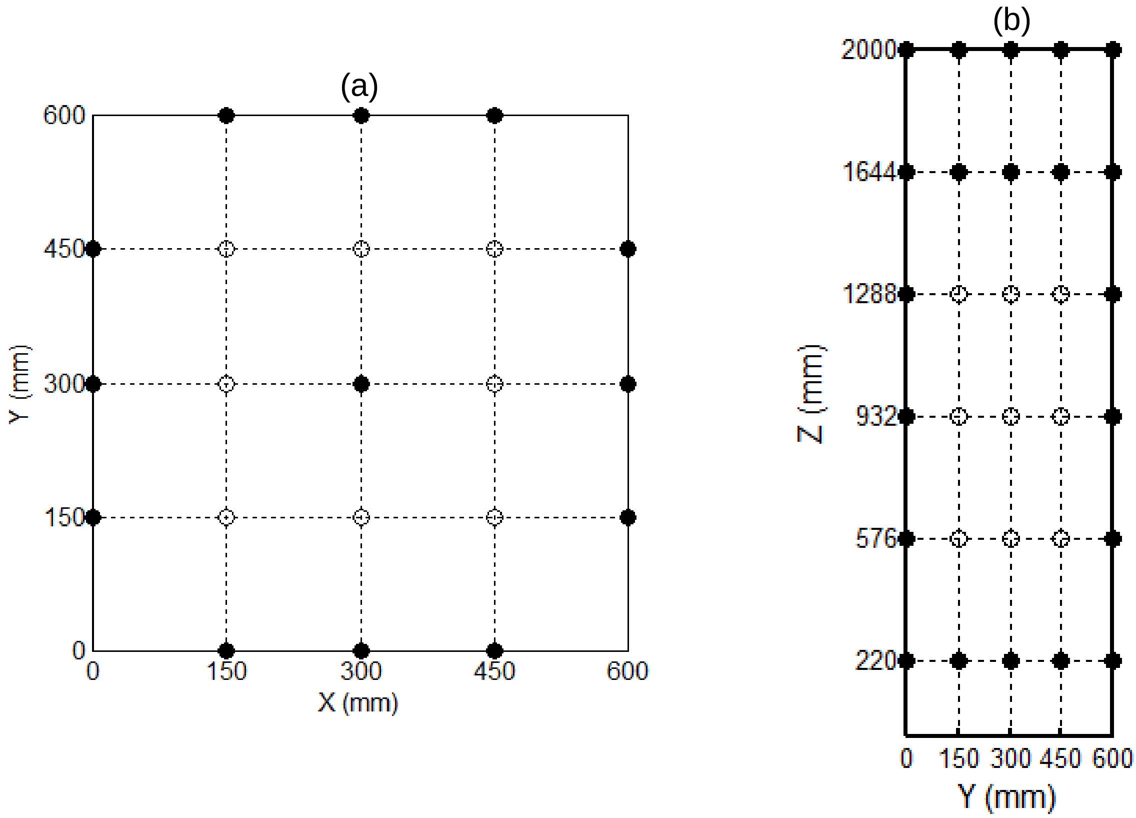


Figure 31: Two planes are identified for validation purposes: (a) z=2000 mm, a horizontal plane in the hot aisle located at near the top of the test rack. (b) x=150 mm, a vertical plane in the hot aisle located parallel to the exhaust of the test rack. The data matrix is comprised of temperature data acquired by the sensors located at the positions marked by filled black circles. The open black circles represent locations where model predictions are validated with actual sensor data.

Before fidelity verification of the data compression algorithm in different measurement planes in the hot aisle, a representative transient temperature characteristic is estimated by analyzing the transient temperature evolution at an arbitrarily chosen point located at

(150,150, 2,000) mm. Figure 32 shows the transient characteristic curve. The normalized transient temperature is defined by θ:

$$\theta(t) = \frac{T(t) - T(t = 0\ \text{s})}{T(t = 500\ \text{s}) - T(t = 0\ \text{s})}. \tag{3.9}$$

The various time instants at which θ reaches the numerical value of 0.1 (11 s), 0.2 (24 s), 0.3 (37 s),… 0.9 (208 s), 0.95 (280 s), 0.99 (475 s) are noted. These time instants will be used as signposts in the ensuing discrete transient analysis.

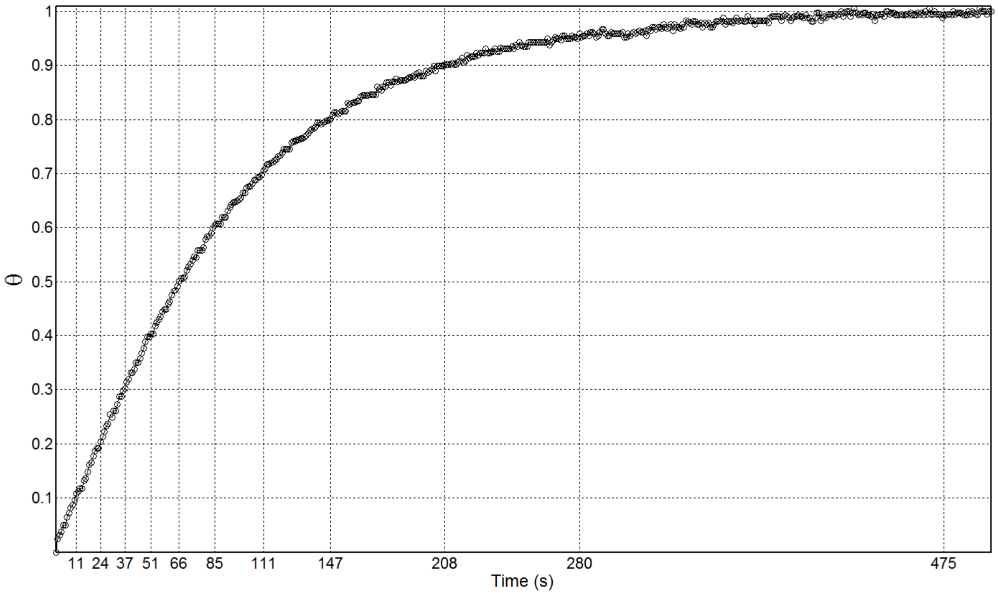


Figure 32: Normalized air temperature acquired at (150, 150, 2000) mm in the hot aisle following sudden introduction of 20 kW test rack heat load. Different time instants are identified when the response reaches (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 95%, 99%) of steady state.

Validation for Measurement Plane at z=2,000 mm

As shown in Figure 31(a), there are 13 TCs in the measurement plane located at z=2,000 mm. These TCs are arranged according to the geometry-based algorithm: a TC is placed at the center of the measurement plane. The remaining TCs are equally distributed on four edges. Each edge gets three TCs. The data acquired by these 13 TCs constitute a data matrix of size 501 x 13. The row rank, 501 corresponds to a transient domain [0-500] s at the sampling interval of 1s. Subsequent POD-based analysis yields the eigenvalue spectrum (Figure 33(a)), two optimal POD modes (Figure 33(b)) and Figure 33(c)), and corresponding POD coefficients (Figure 33(d) and Figure 33(e)). The optimality is characterized by the fact that first two POD modes capture 97% of the information/ energy of the temperature data. With the first POD mode capturing 94.96 % of the energy, this offers 84.6% data compression.

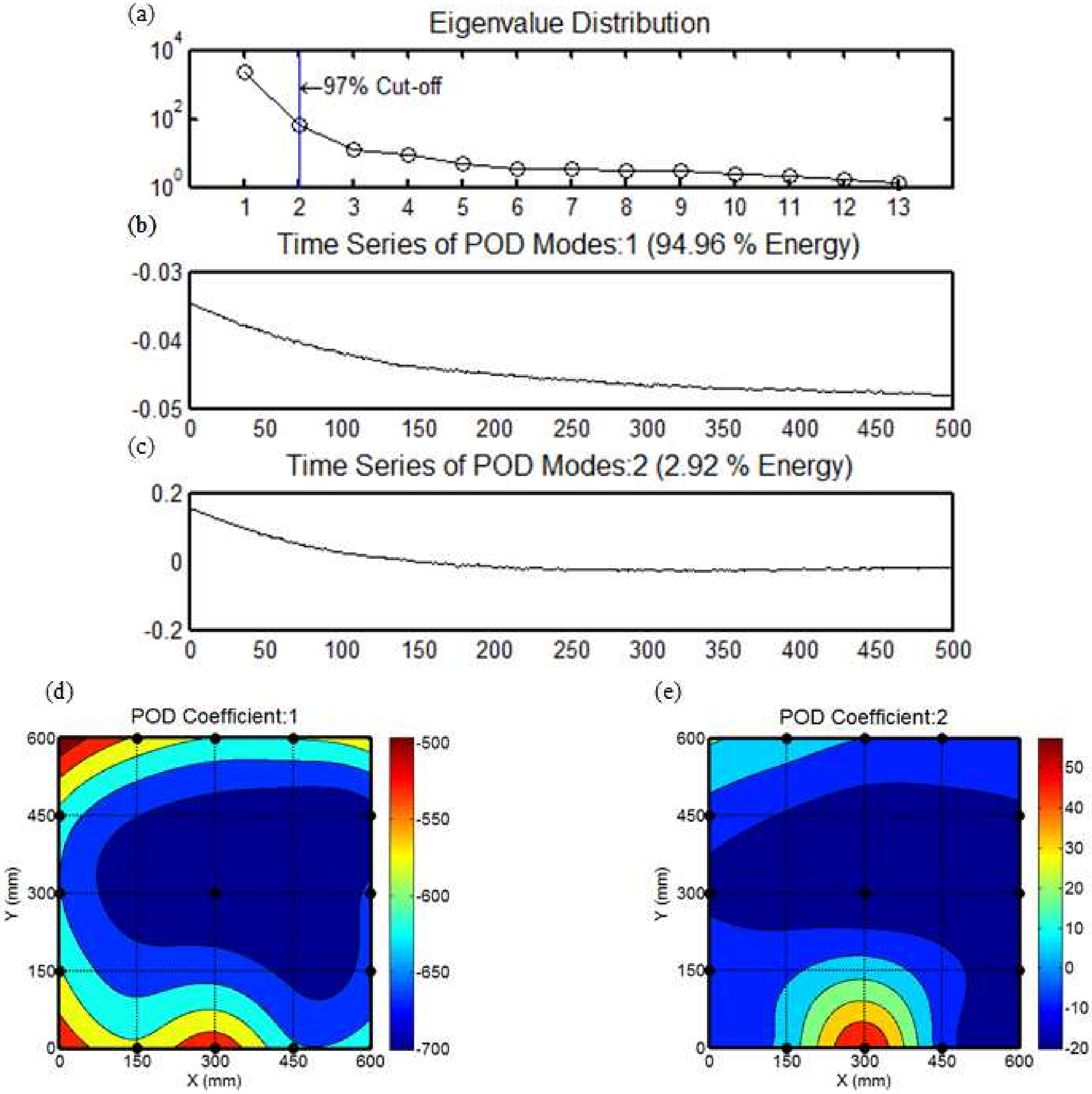


Figure 33: POD-based model order reduction for z=2000 mm. (a) shows degree of data compression provided by POD. Two out of 19 POD modes capture the coherent structure (~97%) of the data sequence. (b) shows time series for the first POD mode, which captures 94.96% of energy. (c) shows time series for the second POD mode with 2.92% of energy. (d) POD coefficient for the 1st POD mode. (e) POD coefficient for the 2nd POD mode.

As proposed by Eq.(3.5) , the POD coefficient at the interrogation location is mapped by the linear combination of two optimal POD coefficients. Hence, the parametric variation of temperature prediction is governed by the weighting scalars of two optimal POD coefficients. The convective transport processes at the horizontal plane, z=2,000 mm is characterized by upward airflow. Therefore, it can be concluded that the temperature variation in the z=2,000 mm plane is governed by local effects that can be analyzed by the geometry-based model alone. Figure 34 shows the Delaunay triangulation-based interpolation [65] of the temperature data captured by 13 sensors (shown by black filled circles). The interpolation creates different isothermal zones. It is proposed that the weighting vector, $C_{n\times 1}$ of a spatial location is governed by its position in the interpolated temperature mapping. All spatial locations in that zone have equal numerical impacts from the included sensors. Therefore, the proposed mathematical model is:

$$\begin{gathered} C_{n\times 1}^{\text{zone}} = \left(\frac{1}{m}\right)\delta_{\text{zone,sensor}}, \\ m \text{ is the number of sensors lying in the given zone.} \\ \delta_{\text{zone,sensor}} = 1, \text{ if the sensor lies in the zone.} \\ \delta_{\text{zone,sensor}} = 0, \text{ otherwise.} \end{gathered} \tag{3.10}$$

For example, the prediction at (450 mm, 450 mm) at $t = 475\ \text{s}$ depends on sensors at (300 mm, 300 mm), (600 mm, 300 mm), and (600 mm, 450 mm). So in this case, *m* will be equal to 3.

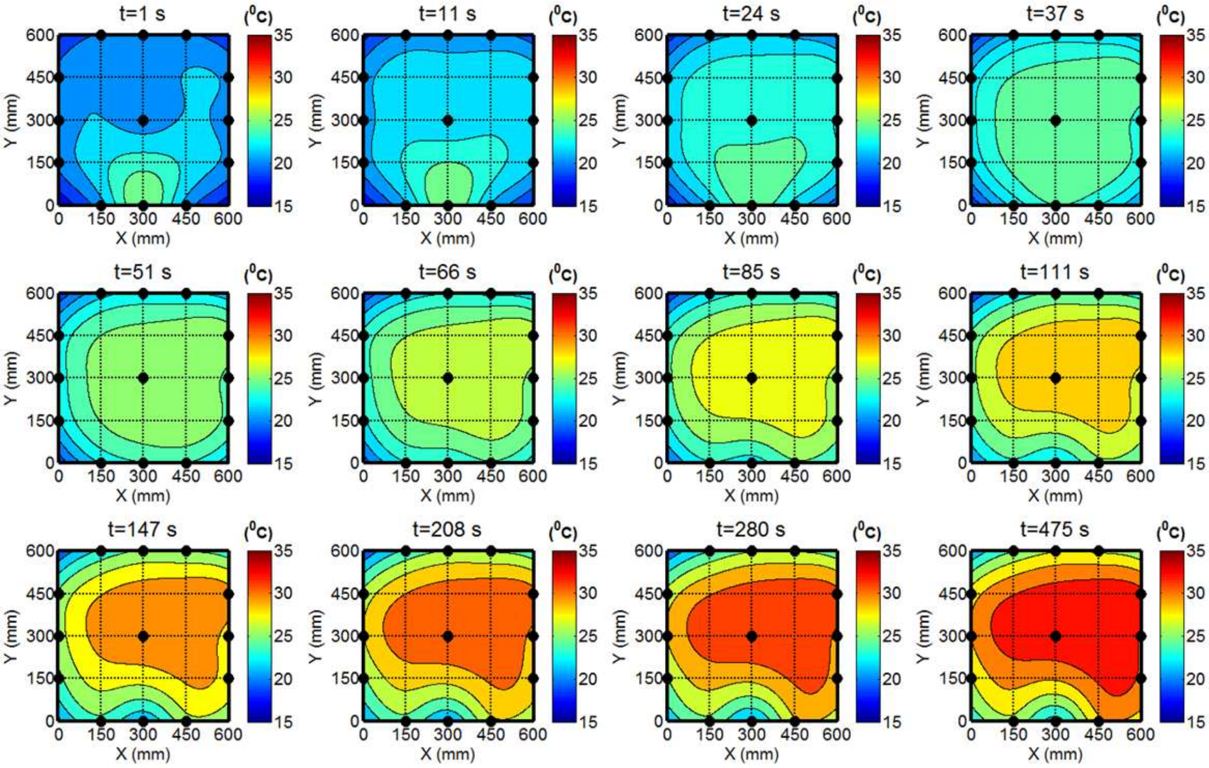


Figure 34: Transient air temperature contours at z=2000 mm. The temperature contours are generated via the Delaunay triangulation technique using the temperature data acquired by the sensors located at the points marked by black filled points. The contours identify the influence of a sensor on various spatial locations.

With the proposed model, temperature signals are computed at the interrogation points and subsequently compared with the experimental data. Figure 35 compares the experimental temperature data at eight different interrogation locations to the corresponding POD-based predictions. The comparison suggests close similarity between data and predictions. The similarity is further quantified by the corresponding correlation coefficients and relative estimated errors shown in Table 8.

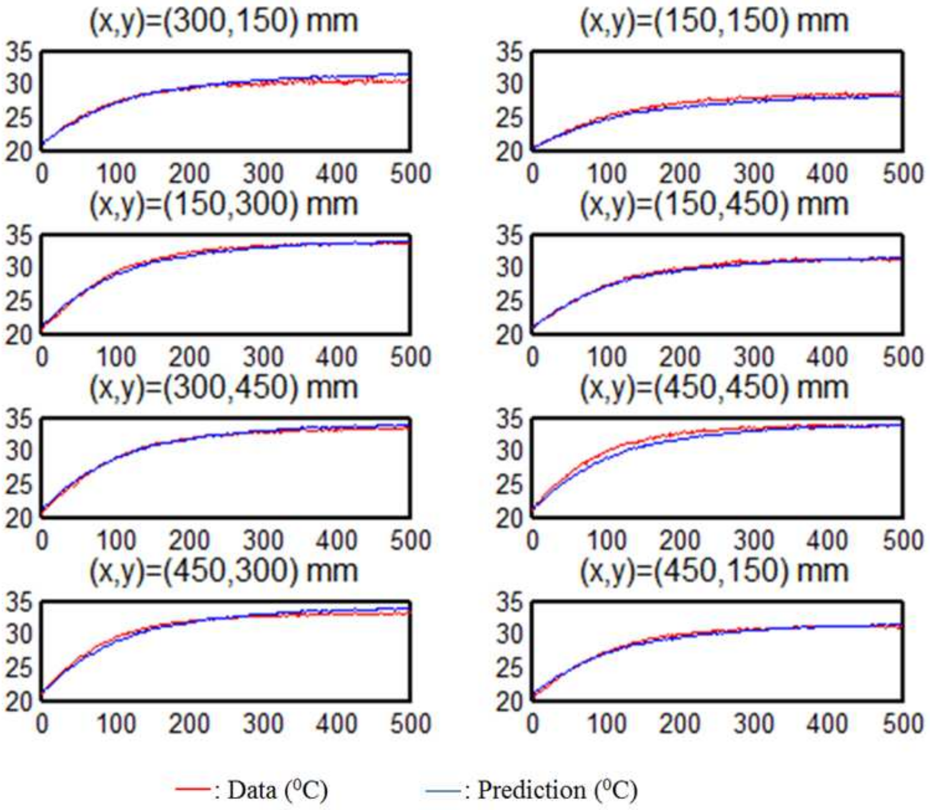


Figure 35: Data vs. POD-based predictions at different interrogation locations in the z=2000 mm plane

Table 8: Error table quantifying the deviations between transient data and predictions in z=2000 mm plane

(x, y)	**Correlation Coefficient**	**Temperature Difference**
mm		(%)
(300, 150)	0.9928	1.8
(150, 150)	0.9975	1.9
(150, 300)	0.9970	1.0
(150, 450)	0.9970	0.8
(300, 450)	0.9974	1.0
(450, 450)	0.9941	2.2
(300, 150)	0.9935	1.6
(150, 150)	0.9950	1.1

Table 8 indicates correlation on the order of 99.5% and relative estimated error on the order of 1% (maximum=2.2%). The proposed POD model is capable of predicting temperature data with 99% relative accuracy, and reducing the sensor number from 21 to 13. That amounts to 38% sensor reduction.

Validation for Measurement Plane at x=150 mm

As shown in Figure 31(b), for the temperature plane located at x=150 mm, data acquired by 21 TCs constitutes the data matrix of size 501 x 21. The POD-based analysis yields the eigenvalue spectrum (Figure 36(a)), two optimal POD modes (Figure 36(b) and Figure 36(c)), and corresponding POD coefficients (Figure 36(d) and Figure 36(e)). The optimality is characterized by the fact that first two POD modes capture 97% of the energy of the temperature data. In fact, the first POD mode captures 94.4 % of the energy. This offers 90.4% data compression. Figure 37 shows the Delaunay triangulation-based interpolation of the temperature data captured by 21 sensors (shown by black filled circles). The interpolation creates different temperature zones. However unlike the plane at z=2,000 mm, the geometry-based algorithm fails to satisfy the tolerance criterion. Alternatively, a physics-based algorithm is proposed in light of the fact that x=150 mm is a vertical plane parallel to the rack exhaust at x=0. An estimation model is developed by identifying that it has two distinct convective environments: one near the top, which is dominated by pressure gradient-driven upward airflow, and another near the bottom which is dominated by inertia-driven shear flow. Figure 38 shows a schematic representation of such a flow pattern. Following this general notion, it is assumed that the predictions above 1,288 mm are governed by the sensors near the top (Zone-1) and those below 932 mm are governed by the sensors at the two sides (y=0 and 600 mm) of Zone-2. This estimation of flow field directly influences the POD coefficient computation which is modeled as:

$$\begin{gathered} C_{n\times 1}^{\text{zone,i}} = \left(\frac{1}{m_i}\right)\delta_{\text{zone,sensor}}, i = 1, 2. \\ m \text{ is the number of sensors lying in the given zone.} \\ \delta_{\text{zone,sensor}} = 1, \text{ if the sensor lies in the zone.} \\ \delta_{\text{zone,sensor}} = 0, \text{ otherwise.} \end{gathered} \tag{3.11}$$

For Zone-1 (i=1), the number of data sensors is equal to 12, and that for Zone-2 it is equal to 9. While three prediction points lie in Zone-1 (at height 1,288 mm; marked by open circles), six prediction points lie in Zone-2 (3 at height 932 mm and 3 at height 576 mm). The choice of two heights at 1,288 mm and 932 mm is based on the standard deviation trend noted in Table 7. It can be noted that there is a sudden drop in temperature gradient from 1,288 mm to 932 mm.

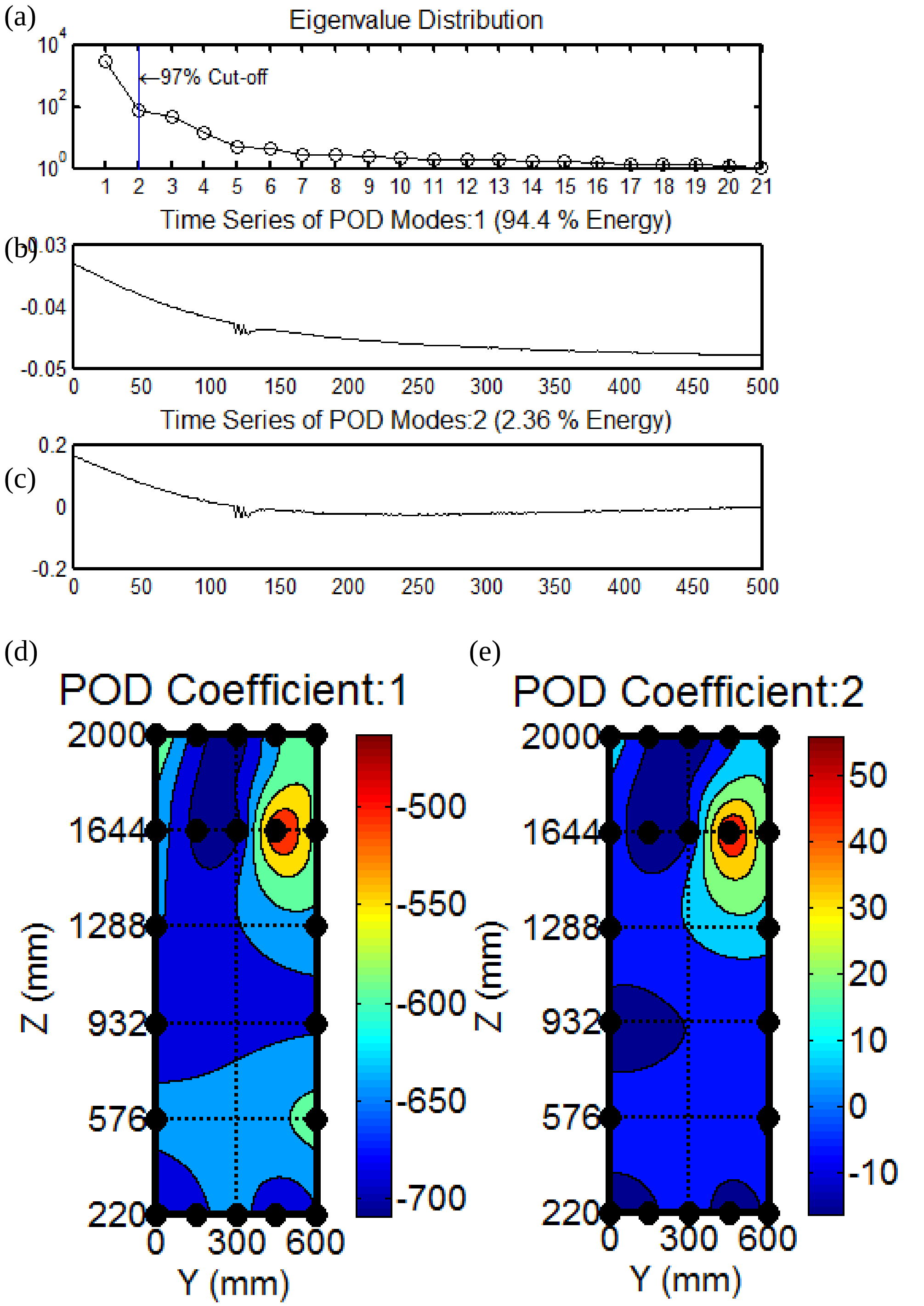


Figure 36: POD-based model order reduction for x=150 mm plane. (a) shows the degree of data compression provided by POD. Two out of 19 POD modes capture the coherent structure (>97%) of the data sequence. (b) shows the time series for the first POD mode, which captures 94.96% of energy. (c) shows time series for the second POD mode with 2.92% of energy. (d) POD coefficient for the 1st POD mode. (e) POD coefficient for the 2nd POD mode.

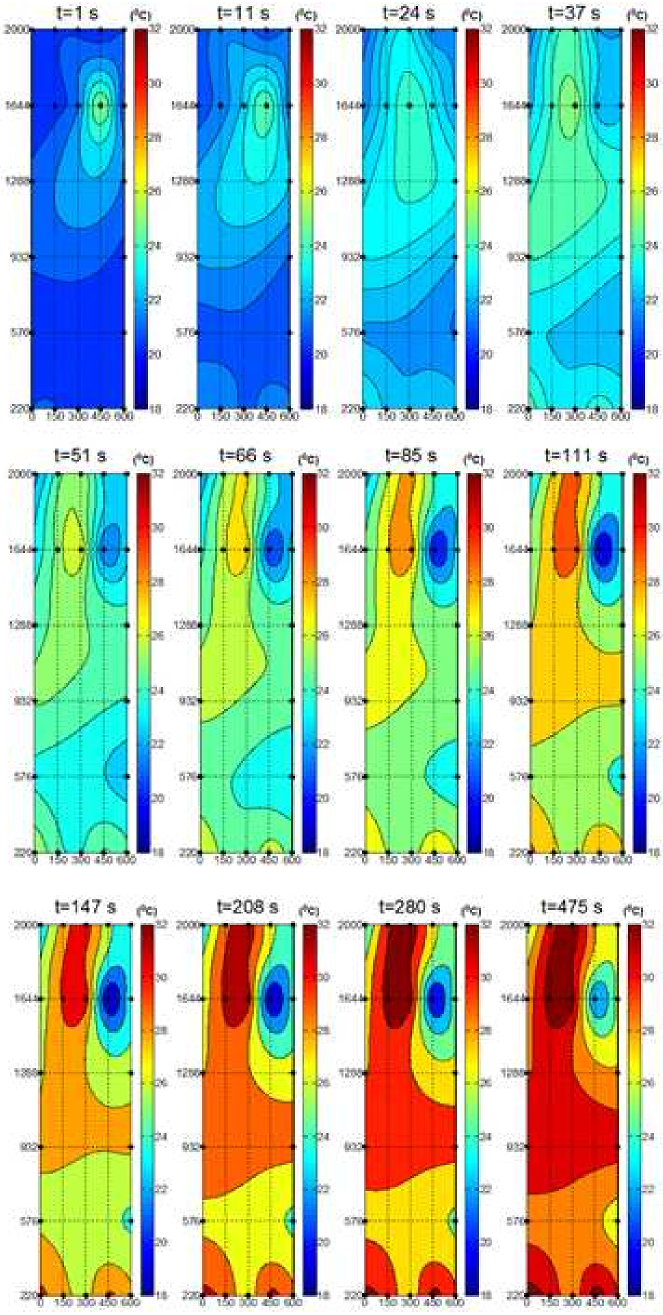


Figure 37: Transient evolution of air temperature in the plane parallel to the rack exhaust at x=150 mm. The temperature contours are generated via Delaunay triangulation technique from the temperature data acquired by the sensors located at points marked by black filled points.

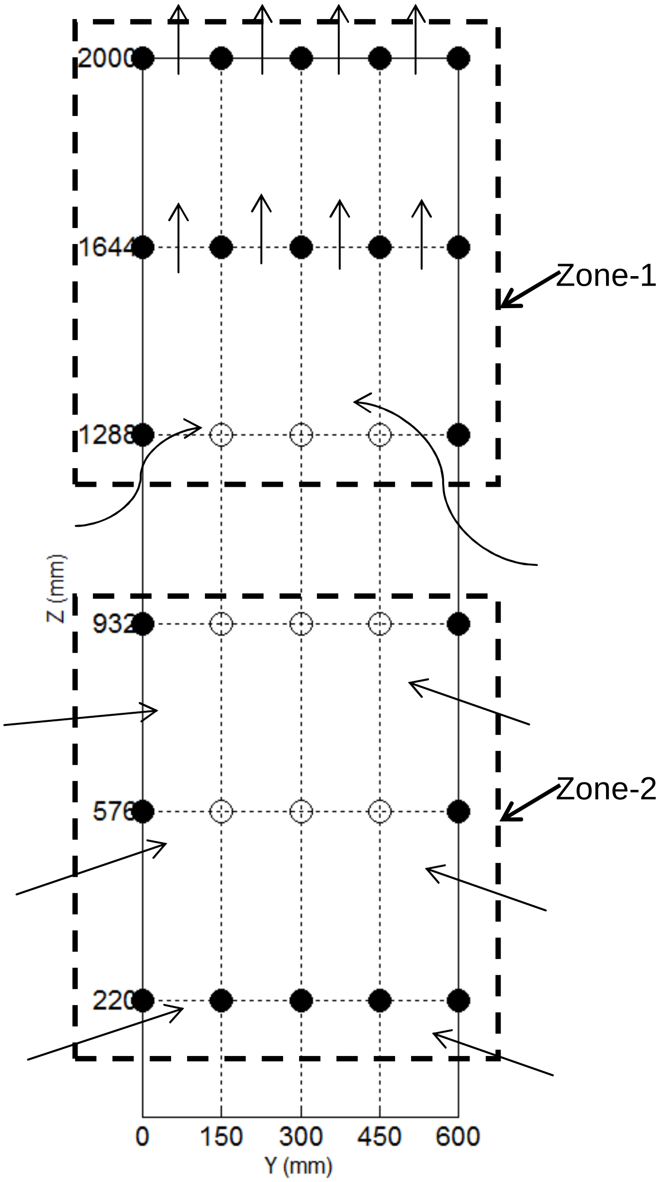


Figure 38: Zonal abstraction of forced convective temperature field in the exhaust plane (parallel to the rack exhaust at x=150 mm). Zone-1 is dominated by free shear flow directed upward to the ceiling. Zone-2 is dominated by flow entrainment from the two sides.

With the proposed model of optimal POD modes and corresponding POD coefficients, temperature signals are computed at the interrogation points and subsequently compared with the experimental data. Figure 39 compares the experimental temperature data at nine different interrogation locations to the corresponding POD-based predictions. The comparison suggests close similarity between data and predictions. The similarity is quantified by the corresponding correlation coefficients and relative estimated errors, as reported in Table 9.

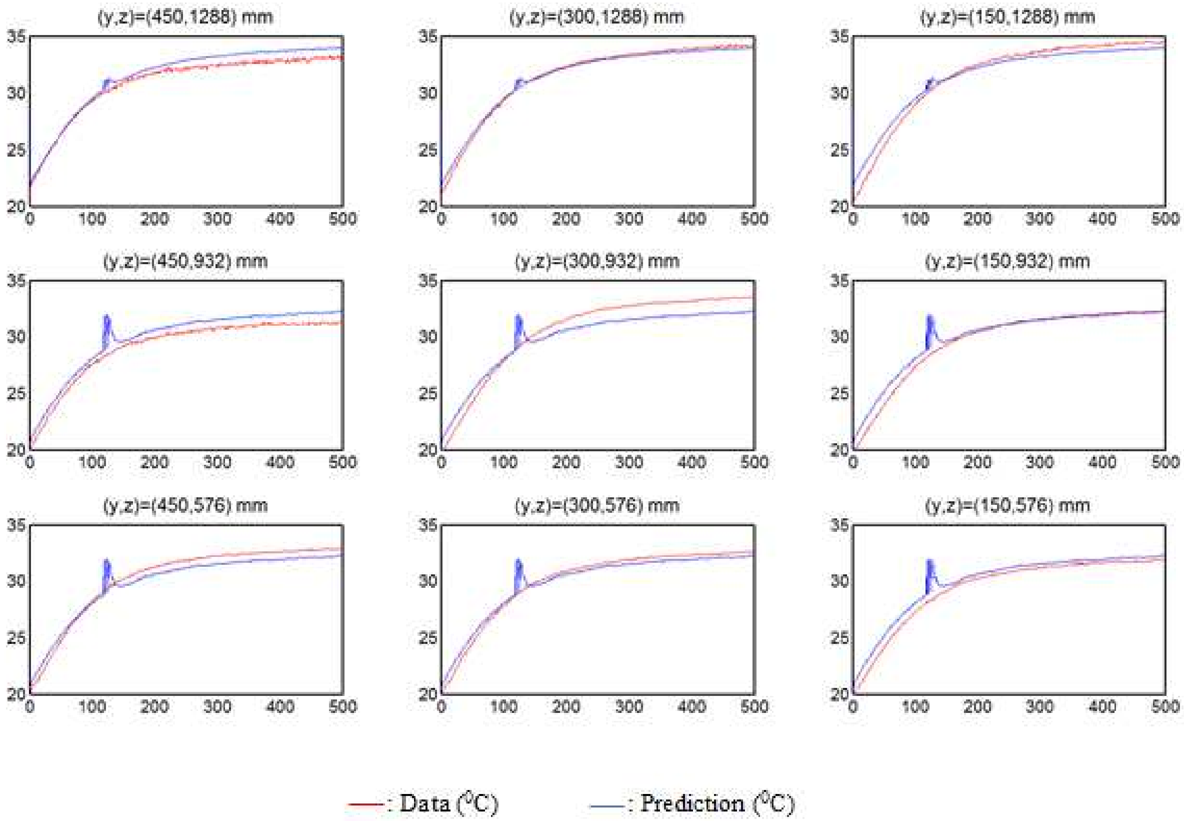


Figure 39: Data vs. POD-based predictions at different interrogation locations in the x=150 mm plane.

Table 9: Error table quantifying the deviations between transient data and predictions in x=150 mm plane.

(y, z)	Correlation Coefficient	Temperature Difference
mm		(%)
(450, 1288)	0.9968	2.1
(300, 1288)	0.9983	1.2
(150, 1288)	0.9979	2.5
(450, 932)	0.9974	2.7
(300, 932)	0.9969	3.2
(150, 932)	0.9975	2.1
(450, 576)	0.9983	1.8
(300, 576)	0.9982	1.3
(150, 576)	0.9978	2.2

Table 9 indicates that the correlation coefficient is in the order of 99.5% and the relative estimated error of the order 1% (maximum=3.2%). The proposed POD model is capable of predicting temperature data with 99% relative accuracy, and of reducing the required sensor number from 30 to 21. That amounts to about 30% sensor reduction. On a related note, Figure 39 shows few glitches in POD-based predictions between 115-130 s. These glitches are the property of the particular POD modal space, as apparent from Figure 36. Since POD modes are parameter-independent, any property of POD modes is indicative of the intrinsic nature of the measured temperature data. Therefore, the presence of those glitches certainty does not undermine the fidelity of the proposed framework.

In summary, a POD-based modeling framework is developed to solve an optimization problem with the number of temperature sensors as the decision variable and the minimization of the sensor number as the objective function. The obvious main constraint is maintaining the temperature prediction error below certain pre-assigned error limit, such as the calibration error of the measurement system. The formal description of the optimization problem is:

$$\begin{gathered}\min\ (\text{sensor number}) \\ \text{such that, deviation} < \text{error limit.}\end{gathered} \tag{3.12}$$

The proposed strategy is a measurement-based approach; therefore, the sensor pattern derived depends upon the temperature gradient, airflow pattern, rack power, and several other thermal variables. As shown in Figure 38 for this case study, the number of convective environments needed for an arbitrary data center is contingent upon its airflow pattern. For the given case study, the airflow scheme is underfloor plenum supply and overhead ceiling return. Alternative airflow schemes could, for example, be underfloor plenum supply and room return, overhead ceiling supply and room return, overhead ceiling supply and overhead ceiling return. The prediction of convective environments demands detailed CFD simulations or reduced order models. While CFD simulations are accurate, albeit computationally resource-intensive, reduced-order models are efficient, but usually have a larger prediction uncertainty. The number of zones to be employed could be estimated based on exploratory experiments, or coarse grid CFD simulations.

Indeed, an effective application of the proposed approach needs additional statistical analyses (as shown in Table 7), approximation models (as shown in Figure 38), and CFD-based analyses.

Closure

Using POD-based model order reduction, a measurement-based framework is developed that improves the spatial resolution of measured temperature data. The framework can predict temperature data with 3% uncertainty. It has been shown that the framework reduces sensor requisition by 30%.

CHAPTER 5
RAPID TEMPERATURE PREDICTIONS IN DATA CENTERS USING MULTI-PARAMETER POD

A proper orthogonal decomposition (POD)-based multi-parameter, reduced-order modeling framework that rapidly predicts air temperatures in an air-cooled data center is developed. The modeling parameters are heat load and time.

Problem Statement

The important design variables for thermal design of an air-cool data center are the heat load and time. Therefore, data center design optimization requires a dynamic framework for rapid prediction of the transient convective air temperature in response to various dynamic events resulting from time-varying IT workloads. A full-factorial design of data center temperature is useful for holistic thermal analyses of a DC facility and life cycle design of data center cooling. Theoretically, a two-variable parametric space can be divided into 2^2=4 parametric subspace, as shown by Figure 40.

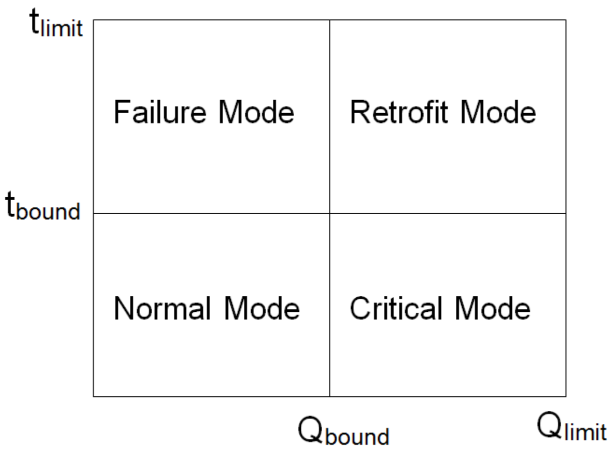


Figure 40: Parametric space for data center cooling design

For a measurement-based framework, Q_{bound} specifies the upper bound of parametric heat load. On the other hand, t_{bound} specifies the upper bound of parametric time. DC Heat load is the primary cooling design variable. Additionally, time is an important design variable because of the dynamic nature of the data center environment. In fact, heat load and time are closely coupled. DC heat load varies dynamically because of the stochastic computing demand of a DC. The parametric space bounded by $(Q_{\text{bound}}, t_{\text{bound}})$ specifies the normal operational mode, which can be characterized by some bounded dynamic heat loads. A predictive framework in this domain is important as an analysis tool for the DC cooling environment. On the other hand, the domain (Q,t) spanned by $Q \in (Q_{bound}, Q_{\text{limit}}] \cup (t < t_{\text{bound}})$ represents the critical operational mode for which transient heat load shoots up beyond Q_{bound}. This is particularly relevant during the flash crowd phenomenon when the DC login rate and associated heat dissipation increase rapidly. Q_{limit} is the maximum possible heat load that the framework can handle. In contrast, the domain (Q,t) spanned by $Q < Q_{bound} \cup t \in (t_{\text{bound}}, t_{\text{limit}}]$ represents the failure operational mode in which there is some failure in DC such as chillers pump failure etc. In this domain, a rapid thermal diagnostic is needed. t_{limit} is the maximum time beyond t_{bound} that the framework can handle. Finally, the domain (Q,t) spanned by $Q \in \left(Q_{bound} < Q < Q_{\text{limit}}\right] \cup (t_{bound} < t < t_{\text{limit}}]$ represents the retrofit operation domain. This domain is characterized by expansion in heat load capacity or changes in cooling environment. Overall, the air temperature prediction platform, with heat load and time as parameters, simulates a holistic design tool. The major problem to enable such a design optimization tool is to develop an efficient and accurate framework that rapidly predicts air temperature data at new parametric points from measured temperature data. This chapter uses POD-based model order reduction to cater to that need.

Methodology

A measurement-based reduced-order model of transient air temperature is developed with spatial location as the independent variable, while time and heat load are the parametric variables. The functional form of temperature response surface is:

$$T = f\left(x,y,z;t,Q\right). \tag{3.13}$$

To model this response surface, a data-driven algorithm is developed as shown in Figure 41. Experimental temperature data at a few selected heat loads and time instants constitute the model input space as a two-dimensional data matrix. The independent variable is the spatial location(x,y,z), which represents the row index of the data matrix. The row dimension of the data matrix is equal to the number of deployed sensors. Temperature data are collected at selected test rack heat loads, Q_{en} and time instants, t_{en}; the data are stored in different columns of the data matrix, with time as the inner variable and heat load as the outer variable. Essentially, the time index and the heat load index are condensed into the column index, j :

$$\begin{gathered} j = i_t + \left(i_h - 1\right) N_t, \\ \text{where, } i_t\text{: temporal index. } i_h\text{: heat load index.} \\ N_h\text{: number of time-varying inputs for a given heat load.} \end{gathered} \tag{3.14}$$

The number of columns represents the overall size of the parametric input space. The intrinsic assumption for the algorithm is that temperature predictions at the interrogation point $\left(Q_{\text{int}}, t_{\text{int}}\right)$ lay in the column space for the data matrix; and hence, the predictions can be expressed as a linear combination of suitable basis functions. Using the POD-based numerical procedure described in CHAPTER 2, the optimal basis functions (POD modes) for a given data matrix are determined. The optimality of the POD modes is quantified by the number of eigenvalues, which capture dominant components of the temperature field. The corresponding weighting scalars (POD coefficients) for basis functions are parameter-dependent and are determined by a suitable regression operation. The location of an interrogation point with respect to $\left(Q_{\text{en}}, t_{\text{en}}\right)$ determines the type of regression operation (interpolation or extrapolation) required. Figure 42 identifies different prediction zones in the parametric space. Table 10 specifies the regression operations needed to determine the POD coefficients at the interrogation point. POD is a model order reduction technique. The prediction vector at an interrogation point $\left(Q_{\text{int}}, t_{\text{int}}\right)$ is expressed as the product of POD modes, ψ and POD coefficients, b [54]:

$$T\left(x, y, z; Q_{\text{int}}, t_{\text{int}}\right) = \sum_{i=1}^{k} \psi_i\left(x, y, z\right) b_i\left(Q_{\text{int}}, t_{\text{int}}\right), \tag{3.15}$$

where, k is the number of principal components that capture the dominant characteristics of the interrogation vector.

Details of the numerical procedure to compute POD modes and POD coefficients are discussed in CHAPTER 2. POD decomposes the data matrix into a low-rank matrix multiplied with a suitable coefficient matrix. Based on the optimality criterion in Eq. (2.12), the model order is reduced, and the corresponding truncated POD-coefficient vector is determined. These two components are multiplied together to predict temperature at an interrogation point defined by, $(Q_{\text{int}}, t_{\text{int}})$. However, due to the data driven nature of the algorithm, the accuracy of the model needs to be controlled by a pre-defined error limit. As shown in Figure 41, if the prediction error is higher than the error limit, a secondary procedure is executed, which starts with a data matrix comprised of transient temperature snapshots for various interrogation heat loads. POD-based temperature modeling with time as the parameter is well documented in the literature [54]. The number of times the secondary procedure is invoked is quantified by a scalar named as Count. Count has significant ramifications on the data compression of the proposed reduced-order model. Together with proposed error limit, Count damps out error incurred in the primary procedure.

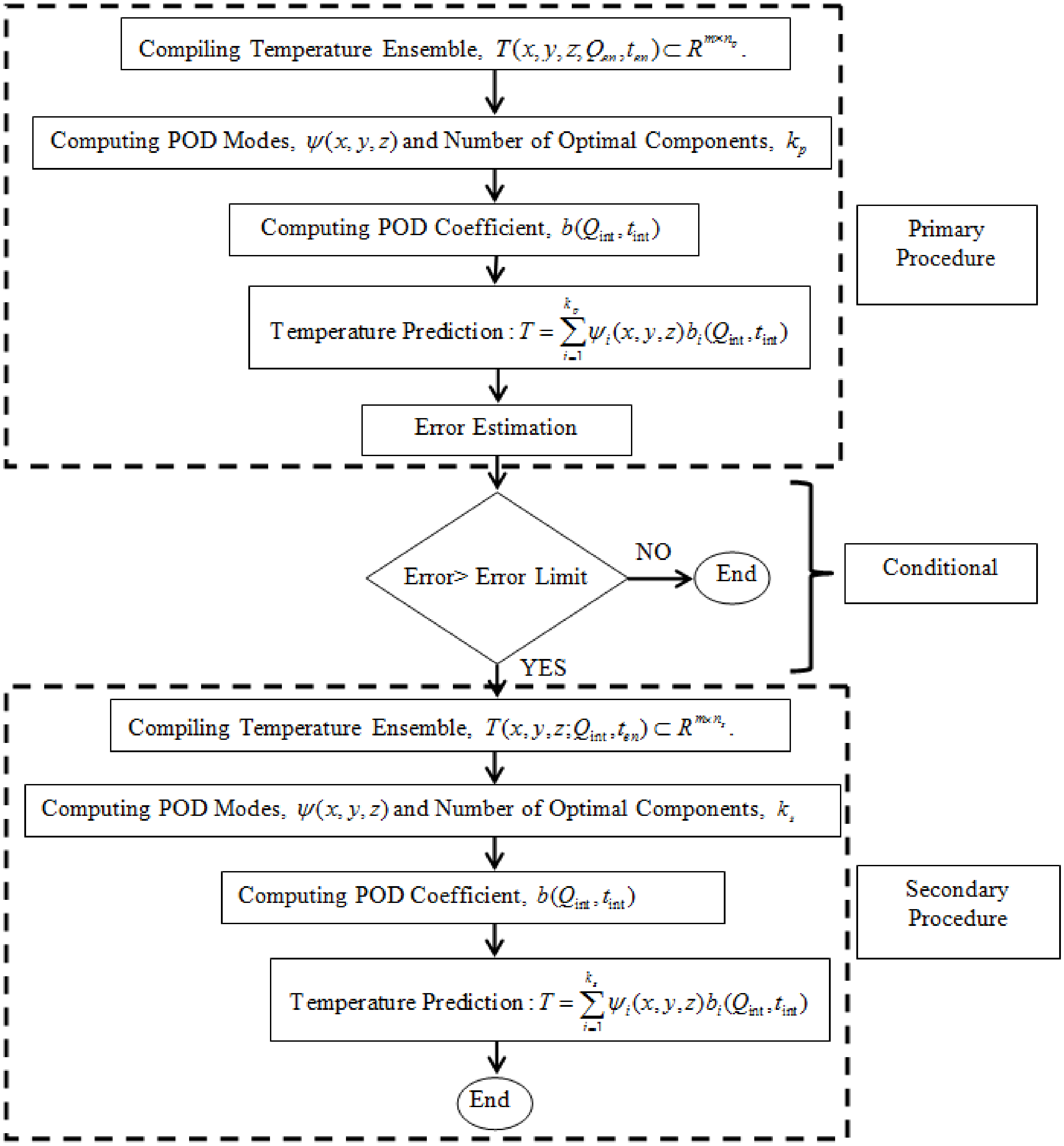

Compiling Temperature Ensemble, $T(x,y,z,Q_{en},t_{en}) \subset R^{m \times n_p}$.
Computing POD Modes, $\psi(x,y,z)$ and Number of Optimal Components, k_p
Computing POD Coefficient, $b(Q_{int},t_{int})$
Temperature Prediction : $T = \sum_{i=1}^{k_p} \psi_i(x,y,z) b_i(Q_{int},t_{int})$
Error Estimation
Primary Procedure
Error> Error Limit
NO
End
Conditional
YES
Compiling Temperature Ensemble, $T(x,y,z;Q_{int},t_{en}) \subset R^{m \times n_s}$.
Computing POD Modes, $\psi(x,y,z)$ and Number of Optimal Components, k_s
Computing POD Coefficient, $b(Q_{int},t_{int})$
Temperature Prediction : $T = \sum_{i=1}^{k_s} \psi_i(x,y,z) b_i(Q_{int},t_{int})$
End
Secondary Procedure

Figure 41: The POD-based temperature prediction algorithm is comprised of two numerical procedures, both based on computing optimal basis functions by POD. The primary procedure involves regression analyses both in heat load and time, whereas the secondary procedure involves regression analyses in time subject to the satisfaction of the condition block.

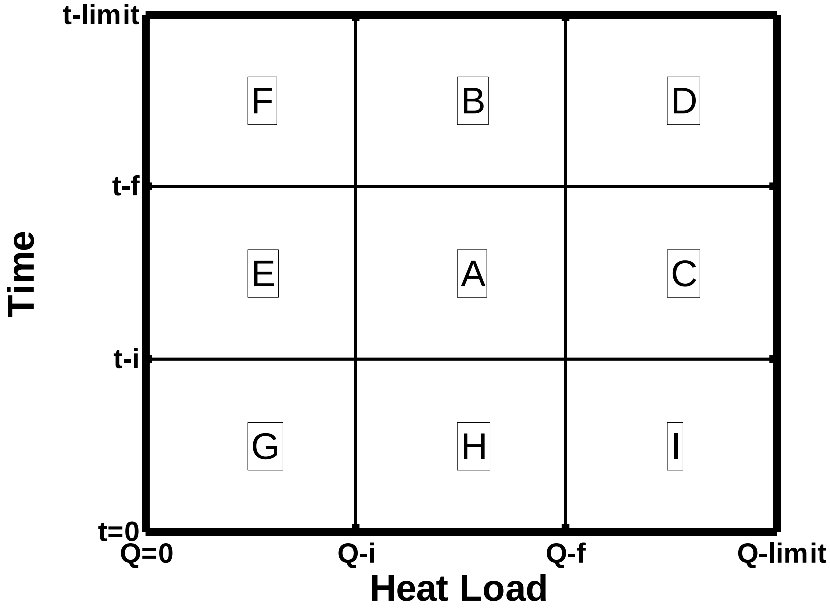


Figure 42: The interrogation space is comprised of all possible heat loads and time instants. Different zones identify whether interpolation/ extrapolation needs to be performed in heat load dimension and time dimension. Table-1 defines all the zones.

Table 10: Specification of different parametric zones in terms of different regression operations

Zone	Heat Load	Time
A	Interpolation	Interpolation
B	Interpolation	Extrapolation
C	Extrapolation	Interpolation
D	Extrapolation	Extrapolation
E	Extrapolation	Interpolation
F	Extrapolation	Extrapolation
G	Extrapolation	Extrapolation
H	Interpolation	Extrapolation
I	Extrapolation	Extrapolation

Experimental Data Acquisition

The experimental setup and measurement system are identical to that used in the study discussed in

CHAPTER 4.

Case Study

To illustrate the present approach, the heat load of the test rack is varied parametrically to 11 different levels: $Q = [4,6,8,10,12,14,16,18,20,22,23]\,\text{kW}$. For each heat load, the ensuing transient is observed by temperature measurements in the vicinity of the test rack at a sampling frequency of 1 Hz. It is characterized by two parameters:

ΔT_1 and ΔT_2. ΔT_1 is defined as the air temperature difference between the rack exhaust and the spatially-averaged incoming cooling air from the perforated tile located at the foot of the test rack:

$$\Delta T_1\left(x, y, z, Q, t\right) = T_1\left(x, y, z, Q, t\right) - T_{\text{supply}}(Q, t). \tag{3.16}$$

For a given heat load, ΔT_1 is a function of sensor height and time. In the exhaust plane, 18 sensors are deployed; therefore, ΔT_1 is an 18-dimensional vector. Figure 43 (a) shows the variation of ΔT_1 as a function of time and height from the floor. As Q increases, the variation in ΔT_1 increases: for $Q = 4$ kW, the variation of ΔT_1 is within a range of $\left[2-5\right]$ ^{0}C; that for $Q = 20$ kW is $\left[5-15\right]$ ^{0}C. Also, ΔT_1 increases with time and gradually achieves steady-state at 200 s. Therefore, the transient measurements are stopped at 250 s. On the other hand, ΔT_2 is defined as the transient difference of the spatially-averaged temperatures in the hot and cold aisles:

$$\Delta T_2\left(Q, t\right) = \left\langle T_{\text{hot}}\left(x, y, z, Q, t\right)\right\rangle - \left\langle T_{\text{cold}}\left(x, y, z, Q, t\right)\right\rangle. \tag{3.17}$$

Figure 43 (b) shows time series of ΔT_2 for different values of $Q = \left[4,\ 8,\ 12,\ 16,\ 20\right]$kW. As expected, the rise in ΔT_2 increases with Q: for $Q = 4$ kW, the rise is 2.7 ^{0}C; for $Q = 20$ kW, it is 9.6 ^{0}C. Also for all values of Q, ΔT_2 reaches a steady state before the arbitrarily determined transient measurement window of 250 s.

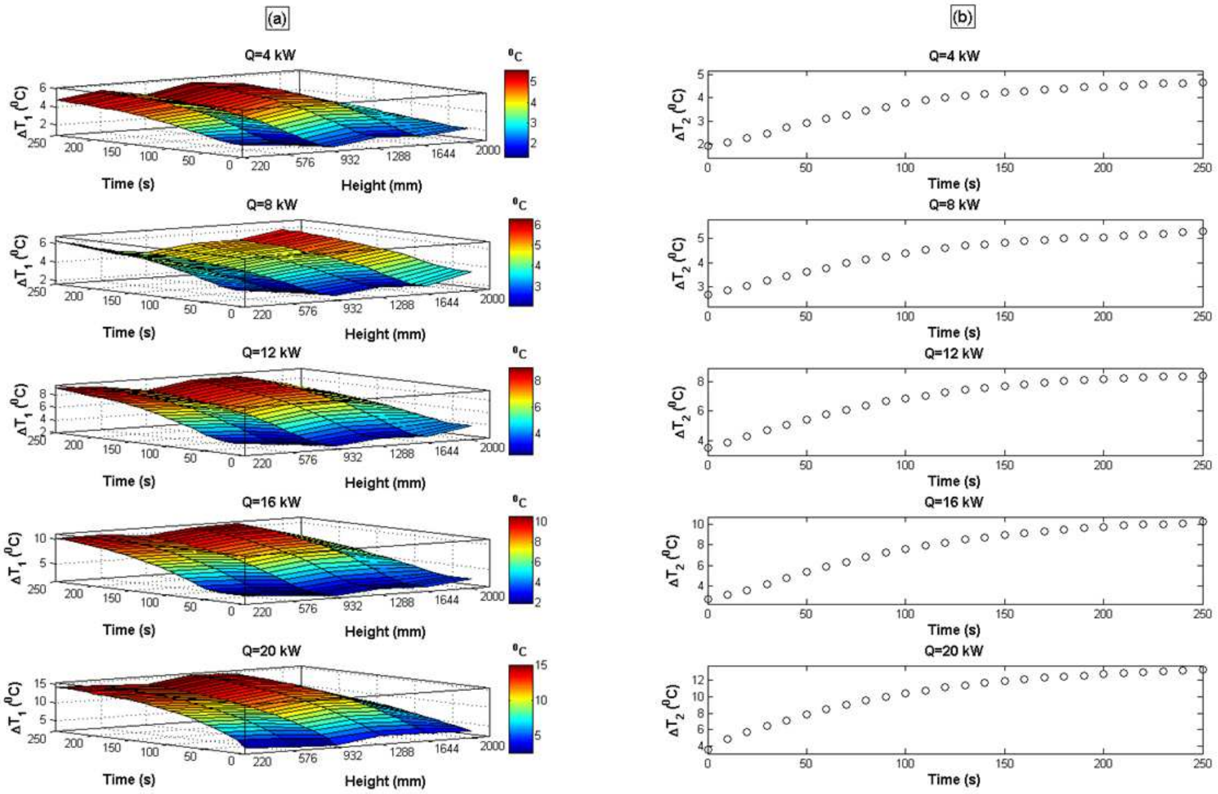


Figure 43: Impulse response of air temperature after turning on the server simulator heat loads. Five different values of rack heat load are used: $Q_i = [4, 8, 12, 16, 20]$ kW. (a) For a given value of the heat load Q_i, the corresponding surface plot shows the variation of ΔT_1 as a function of height from the floor and time. The data sampling set corresponds to six different heights from the floor: $h = [220, 576, 932, 1{,}288, 1{,}644, 2{,}000]$ mm. and 26 different time instants: $t = 0 - 250$ s at Δt=10 s. ΔT_1 is defined as the difference of transient rack exhaust temperatures to the spatially-averaged transient temperature of the cooling air coming out the perforated tile located in the cold aisle in front of the test rack. (b) For a given value of the heat load Q_i, the plot shows the time series of ΔT_2. The data sampling set includes 26 different time instants: $t = 0 - 250$ s at Δt=10 s. ΔT_2 is defined as the transient difference of average temperatures in the hot and cold aisles.

Figure 44 shows frequency responses for air temperatures collected at three different heights, $h = [1960, 1280, 600]$ mm for the maximum rack heat load of 23 kW. The Y-axis shows the gain (in dB) of rack exhaust temperature scaled by the CRAC return air temperature set point (22.9 ^{0}C~75 ^{0}F). The gain is computed by the discretized Fourier transform (DFT) using a MATLAB-based fast Fourier transform (FFT) algorithm [67, 68]. As shown in Figure 44, the scaled amplitude significantly flattens out after 0.5 Hz, implying that the minimum sampling rate to characterize the transient temperature must be at least 1 Hz, via the Nyquist-Shannon sampling theorem.

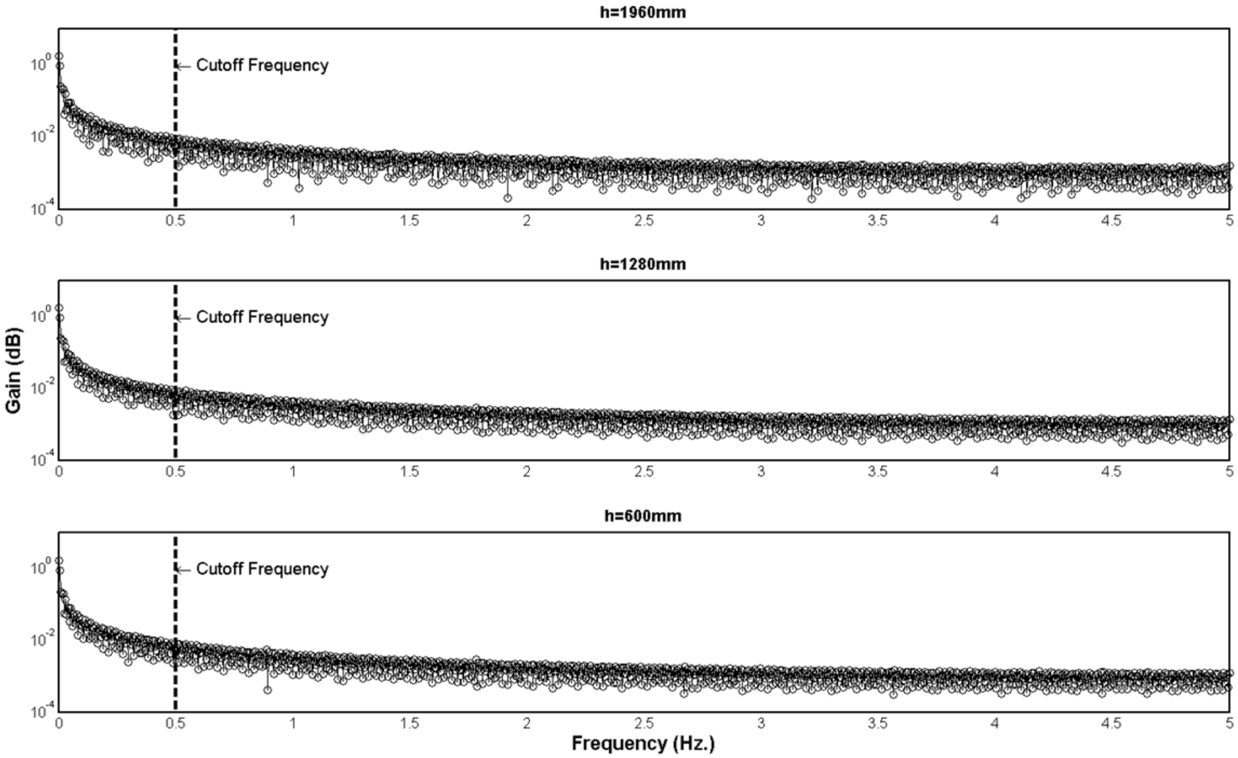


Figure 44: Frequency response of hot aisle temperature field at three different heights in the rack exhaust plane

As the first step of the POD-based framework, the ensemble set of the parameters needs to be identified. Without losing generality, it is chosen as a subset of the experimental heat load values:

$$Q_{\mathrm{en}} = \left[4, 8, 12, 16, 20\right] \text{ kW.} \tag{3.18}$$

This heat load input space defines a representative sample space of commonly-occurring dynamic heat load patterns [69] in a typical data center rack. Also, the ensemble heat load dimension reduces the input size from 11 to 5 (54% input data compression with respect to the measurement set).On the other hand, the sampling frequency for the time ensemble is reduced to 0.1 Hz:

$$t_{\mathrm{en}} = 10 - 200 \text{ s at } \Delta t = 10 \text{ s.} \tag{3.19}$$

Such a down-sampling reduces the ensemble size by 90% with respect to the measured transient data. Overall, the data for the primary procedure is organized in a 124 x 95 matrix, where 124 is the number of thermocouples deployed in the hot aisle. The column index of the data matrix is defined by (3.14). For the given problem, $N_t = 19,\ i_t = \left\{1-19\right\}, i_h = \left\{1-5\right\}$. The size of the input space is equal to 95 (=19x5) whereas that of the primitive observations is equal to 2,101 (=191x11). That amounts to 95.5% data compression. However, if the secondary procedure is invoked, the data compression in the heat load dimension is completely nullified for the sake of improving prediction fidelity. In the secondary procedure, the problem becomes single-parameter with time as the parameter. The secondary procedure amounts to 90% data compression.

Results and Discussion

The temperature ensemble is generated by applying the POD-based algorithm on distributed temperature data collected at $\left(Q_{en}, t_{en}\right)$. As shown in Figure 42, an interrogation point $\left(Q_{\text{int}}, t_{\text{int}}\right)$ can lie in nine possible regions in the $Q-t$ parametric plane. This classification is based on the position of an interrogation point with respect to $\left(Q_{en}, t_{en}\right)$ and the associated regression operation. For example, the point (10 kW, 120 s) lies in the region A, which requires interpolation both in heat load and time to compute POD-based temperature predictions. Detailed specifications of the different regions are documented in Table 10.

For the sake of scalable parametric modeling, heat load is normalized as follows:

$$\begin{aligned} \bar{Q} &= \frac{Q}{Q_{\text{base}}}, \\ Q_{\text{base}} &= 4 \text{ kW}. \end{aligned} \tag{3.20}$$

Based on this normalized definition of heat load, the set of normalized heat load snapshots is: $\bar{Q}_{en} = [1, 2, 3, 4, 5]$. This suggests that any interrogation heat load for which $\bar{Q}_{\text{int}} \in \left(1, 5\right)$ is within the heat load ensemble set, $\bar{Q}_{en}$. Similarly, time is normalized as:

$$\begin{aligned} \bar{t} &= \frac{t - t_{\text{initial}}}{t_{\text{final}} - t_{\text{initial}}}, \\ t_{\text{initial}} &= 10 \text{ s}, \; t_{\text{final}} = 200 \text{ s}. \end{aligned} \tag{3.21}$$

Based on this normalized definition of time, $\bar{t}_{en} = [0, (10/190), (20/190), \ldots, (170/190), (180/190), 1]$. This normalization suggests that interrogation time, for which $\bar{t}_{\text{int}} \in \left(0,1\right)$, is within the time ensemble set, $\bar{t}_{en}$. The normalization scheme is chosen for the sake of compactness and scalability of the parametric analysis. Based on the required regression operations, there are four (=2^2)

possible combinations: interpolation both in heat load and time (zone-A), interpolation in heat load and extrapolation in time (zones-B and H), extrapolation in heat load and interpolation in time (zones-C and E), and extrapolation both in heat load and time (zones- D, F, G, and I). The zone corresponding to $\left(\bar{t} < 0 \text{ or } \bar{Q} < 1\right)$ is ignored for its least practical significance in DC characterization. Hence, this study focuses on model prediction in zones: A, B, C, and D.

Model Prediction in Zone-A

For this case-study, Zone-A is the parametric subspace, spanned by $\left(Q_{\text{int}}, t_{\text{int}}\right) \exists \left\{\left(1 < \bar{Q}_{\text{int}} < 5\right) \cup \left(0 < \bar{t}_{\text{int}} < 1\right)\right\}$. In this zone, an arbitrary parametric point is chosen as (14 kW, 124 s); the corresponding normalized interrogation point is (3.5, 0.6). For this interrogation point, Figure 45 demonstrates the fidelity of POD-based temperature predictions in the spatial domain located at the rack exhaust plane. Figure 45(a) shows ΔT_1 computed from the temperature measurement data and its mapping at the test rack exhaust. The POD-based algorithm is applied on the measurement data. The absolute deviations between experimental data and POD predictions are also computed. From these discrete measurements/predictions, the contour surfaces are generated by Delaunay triangulation [70]. Figure 45(b) shows ΔT_1 computed from the POD-based predictions.

The computational time for the POD-based prediction is on the order of 10 s (on an Intel®Core™2.Duo CPU of specification E8200 @ 2.66 GHz supported by 4 GB RAM). Figure 45(c) shows the absolute deviation between the experimentally acquired ΔT_1 and the corresponding POD-predicted ΔT_1. For Zone-A, the error limit for the conditional block is set equal to the calibration error (0.5^0 C). The value of Count, which is defined as the number of times the secondary procedure is executed, is observed to be 4. The sensors for which the absolute error exceeds the error limit are C, F, O, and R. Interestingly, all these sensors lie along the line, $y = 150$ mm, and Figure 45(a) suggests that these locations are hotspots. The overall root mean square (RMS) deviation is observed to be equal to 0.25 ^{0}C.

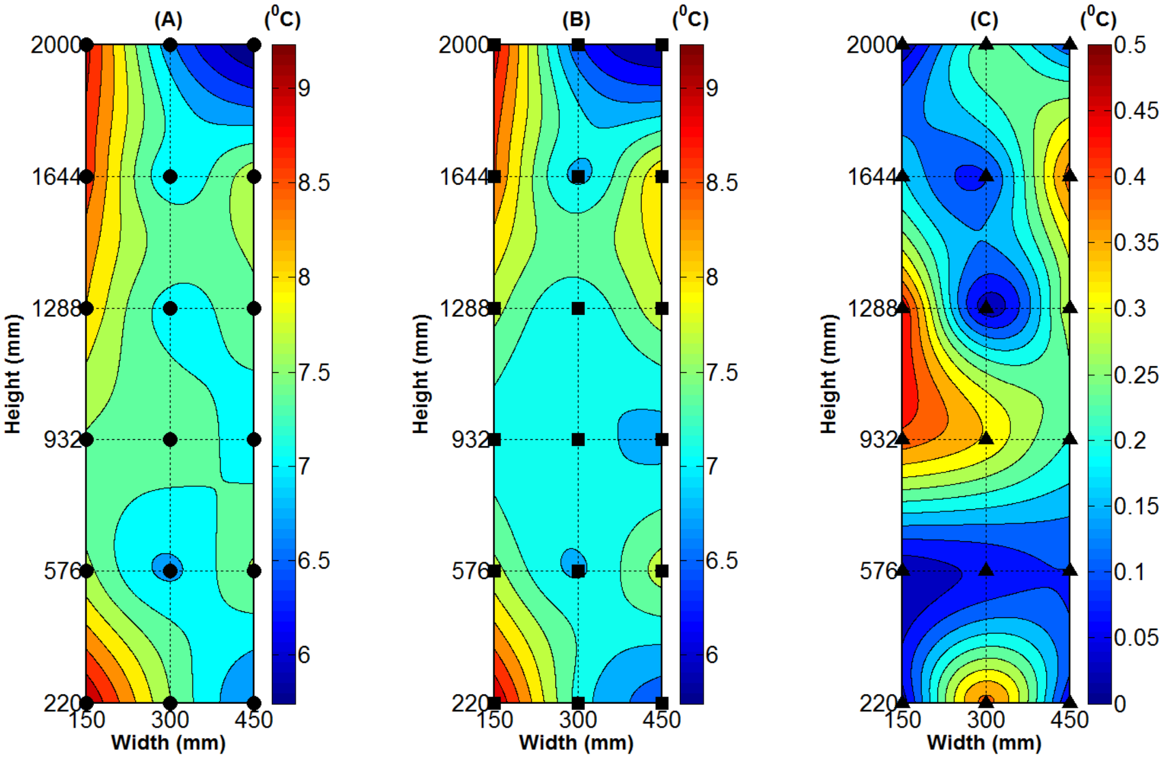


Figure 45: The temperature prediction capability of the POD-based framework in zone-A. The interrogation point is: $\left[\overline{Q} = 3.5,\ \bar{t} = 0.6\right]$. (A) Mapping of experimentally-acquired ΔT_1. Circular markers represent the location of temperature sensors. (B) Mapping of POD-predicted ΔT_1. Square markers represent the prediction points. (C) Mapping of the deviations between experimental data and POD predictions. Triangular markers show absolute deviation data. The contour mapping from the discrete data points is performed via the Delaunay triangulation. For the conditional step, the error limit assigned is 0.5 ^{0}C. The number of times the second level computation is invoked or Count is equal to 4.

Due to the semi-empirical nature of the algorithm, modeling fidelity needs to be verified for various interrogation points. Table 11 shows the prediction errors for various interrogation points defined by the first two columns $\overline{Q}$ and $\bar{t}$.

Table 11: Error table for POD-based predictions in Zone-A for different interrogation points

Normalized Heat Load $\left(\overline{Q}\right)$	**Normalized Time** $\left(\bar{t}\right)$	**Max. Error**(^{0}C)	**RMS Relative Error**
1.5	0.2	0.49	4%
	0.4	0.40	2%
	0.6	0.39	1%
	0.8	0.20	1%
2.5	0.2	0.48	6%
	0.4	0.48	4%
	0.6	0.45	4%
	0.8	0.44	4%
3.5	0.2	0.12	1%
	0.4	0.49	2%
	0.6	0.46	3%
	0.8	0.45	2%
4.5	0.2	0.08	1%
	0.4	0.09	1%
	0.6	0.12	1%
	0.8	0.1	0%

For every combination of $\left(\overline{Q},\bar{t}\right)$, the third column of the error table presents the maximum error, defined as the maximum of the absolute deviations between measurement data and POD predictions. The fourth column tabulates the RMS of the relative errors. It is clear from Table 11 that the RMS errors vary within a range of [0%-6%], with an average of 2.31% and standard deviation of 1.66%.

Table 12: Sensitivity of model fidelity to ensemble sampling interval. The interrogation heat loads for these predictions are kept constant at 14 kW. A candidate space of $\Delta t = \{5\text{ s, }10\text{ s, }15\text{ s}\}$ is examined.

Normalized Time	**Scaled Time Step Size**	**RMS Relative Error**
0.2	5	6%
	10	1%
	15	5%
0.4	5	2%
	10	2%
	15	2%
0.6	5	3%
	10	3%
	15	2%
0.8	5	3%
	10	2%
	15	2%

The sampling time interval for constructing the temperature ensemble is arbitrarily assumed to be equal to 10 s, which is ten times higher than the measurement sampling interval. Since this choice is arbitrary, the sensitivity of the sampling time interval on POD predictions is analyzed in Table 12. In general, a higher sampling time interval means improved effectiveness of data compression. Nevertheless, such data compression often comes at the cost of modeling accuracy [71]. Due to the semi-empirical nature of the proposed model, three different sampling intervals of 5 s, 10 s, and 15 s (or 5, 10, 15 when they are scaled by the measurement sampling interval of 1 s) are tested for prediction fidelity at four time instants: $\overline{t} = [0.2, 0.4, 0.6, 0.8]$. The third column tabulates the RMS values of the relative errors. A sampling interval choice of 10 s yields the most accurate prediction.

Model Prediction in Zone-B

For this case study, zone-B is the parametric subspace spanned by $(Q_{\text{int}}, t_{\text{int}}) \exists \left\{ \left(1 < \overline{Q}_{\text{int}} < 5\right) \cup \left(\overline{t}_{\text{int}} > 1\right) \right\}$. This predictor space demands a temperature prognosis in time. Therefore, it is particularly pertinent to the analysis of various thermal runaway phenomena during commonly-occurring emergencies, such as grid power outages and cooling equipment (e.g. chiller pump) failures. In this zone, an arbitrary parametric point is chosen as (14 kW, 209 s); the corresponding normalized interrogation point is (3.5, 199/190). Figure 46 demonstrates the fidelity of POD-based temperature predictions in the spatial domain located at the rack exhaust plane. Figure 46(a) shows ΔT_1 computed from the temperature measurement data and its mapping at the test rack exhaust. Figure 46(b) shows ΔT_1 computed from the POD-based predictions. The computational time for the POD-based prediction is on the order of 10 s (on an Intel®Core™2.Duo CPU of specification E8200 @ 2.66 GHz supported by 4 GB RAM). Figure 46(c) shows the absolute deviation between the experimentally acquired ΔT_1 and the corresponding POD-predicted ΔT_1. For Zone-B, the error limit for the conditional block is set equal to the calibration error (0.5^0 C). The value of Count is equal to 15. The RMS deviation is equal to 0.6 ^{0}C.

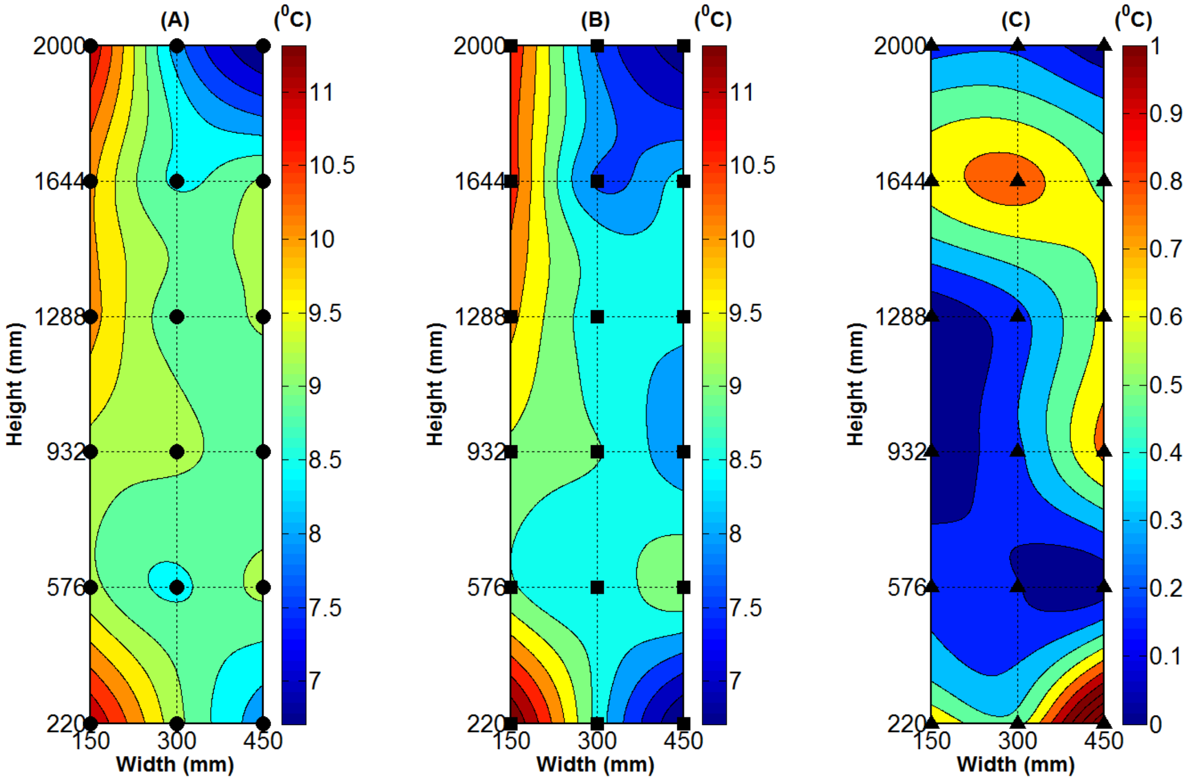


Figure 46: The capability of the POD-based framework in predicting new temperature data in zone-B. The interrogation point is arbitrarily assigned: $\left[\overline{Q} = 3.5,\ \bar{t} = 199/190\right]$. Experimental data are collected by 18 sensors shown by black circular markers arranged in the form of a 3 x 6 grid. Located at the exhaust of the test rack, the interrogation region is 300 mm (150 mm-450 mm) in width and 2000 mm (0 mm-2000 mm) in height. POD-based algorithm is applied on the measurement data. The deviations between experimental data and POD predictions are noted. (A) Mapping of experimentally-acquired temperature data. (B) Mapping of POD-predictions. (C) Mapping of the deviations between experimental data and POD predictions. The mapping from the data points is done via the Delaunay triangulation. The intermediate error limit assigned is 0.5 ^{0}C. The number of times the second level computation is invoked or Count is equal to 15.

The secondary procedure in the proposed algorithm is designed for improving prediction accuracy. However, for extrapolation, after some critical time window even the proposed secondary procedure fails to damp out the error. Therefore, it is imperative to estimate an acceptable extrapolation window. The fault-tolerance depends on the design redundancy or the tier-status [72] of a DC. A Tier-4 DC is least fault-tolerant. On the other hand, a Tier-1 DC is most fault-tolerant. For this case study, the design redundancies are quantified as the error factor, which is defined as the ratio of acceptable root mean square error to the calibration error (0.5 ^{0}C). A Tier-4 facility has an error factor numerically equal to 1, and that for a Tier-1 facility is 4. For Tier-2 and Tier-3, the error factors are 3 and 2, respectively. Therefore, an error factor equal to 3 means the tolerable RMS error limit is equal to 1.5 ^{0}C. The reliable extrapolation time window is defined as the normalized extrapolation time interval: $\left(\overline{t}_{\text{exp}} = \dfrac{t_{\text{exp}} - t_{\text{final}}}{\Delta t} : \ t_{\text{final}} = 200 \text{ s}, \Delta t = 10 \text{ s}\right)$ for which the proposed fidelity criterion is satisfied. Figure 47 shows reliable extrapolation time bounds as a function of heat load for four different DC Tiers. For different error factors, the reliable extrapolation window is computed for various values of interrogation heat load: $\left\{\overline{Q} = 1.5, 2.5, 3.5, 4.5.\right\}$. Figure 47 indicates that with increase in heat load, the reliable prediction window decreases. This facilitates prognostic-based thermal reliability modeling of a DC.

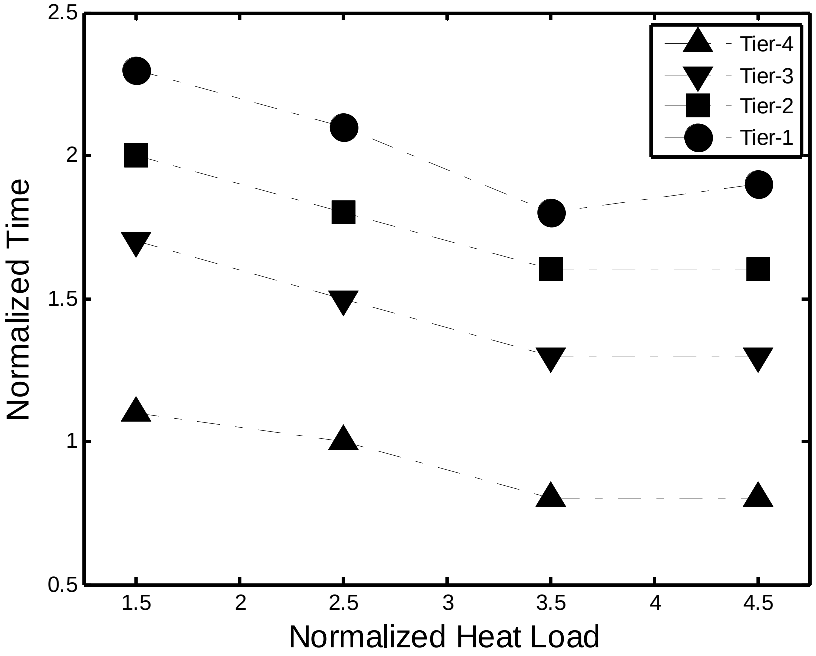


Figure 47: For four different DC Tiers, normalized extrapolation time limit $\left(\bar{t}_{\exp} = \frac{t_{\exp} - t_{\mathrm{final}}}{\Delta t} :\ t_{\mathrm{final}} = 200\ \mathrm{s},\ \Delta t = 1\ \mathrm{s}\right)$ versus interrogation heat load. DC Tiers are classified by error factor which is defined as the ratio of the tolerable RMS error to the calibration error (0.5 ^{0}C).

Model Prediction in Zone-C

For this case study, zone-C is the parametric subspace spanned by $\left(Q_{\text{int}}, t_{\text{int}}\right) \exists \left\{ \left(\bar{Q}_{\text{int}} > 5\right) \cup \left(0 < \bar{t}_{\text{int}} < 1\right) \right\}$. This predictor space demands a temperature prognosis in heat load and is particularly useful for the analysis of thermal spikes resulting from rapidly escalating DC IT loads induced by the flash-crowd effect [73]. To verify the prediction fidelity, an arbitrary parametric point is chosen as (22 kW, 124 s); the corresponding normalized interrogation point is (5.5, 0.6). Figure 48 demonstrates the fidelity of POD-based temperature predictions in the spatial domain located at the rack exhaust plane. Figure 48(a) shows ΔT_1 computed from the temperature measurement data and its mapping at the test rack exhaust. Figure 48(b) shows ΔT_1 computed from the POD-based predictions. The computational time for the POD-based prediction is on the order of 10 s (on an Intel®Core™2.Duo CPU of specification E8200 @ 2.66 GHz supported by 4 GB RAM). Figure 48(c) shows the absolute deviation between experimentally acquired values of ΔT_1 and the corresponding POD-predicted values of ΔT_1. For Zone-C, the error limit for the conditional block is set equal to four times the calibration error (0.5^0 C). Often during thermal spikes, rapid thermal characterization is desirable even at the cost of some prediction accuracy. The value of Count, which is defined as the period of time over which the secondary procedure is executed, was observed to be equal to 14. The RMS deviation is equal to 0.7 ^{0}C.

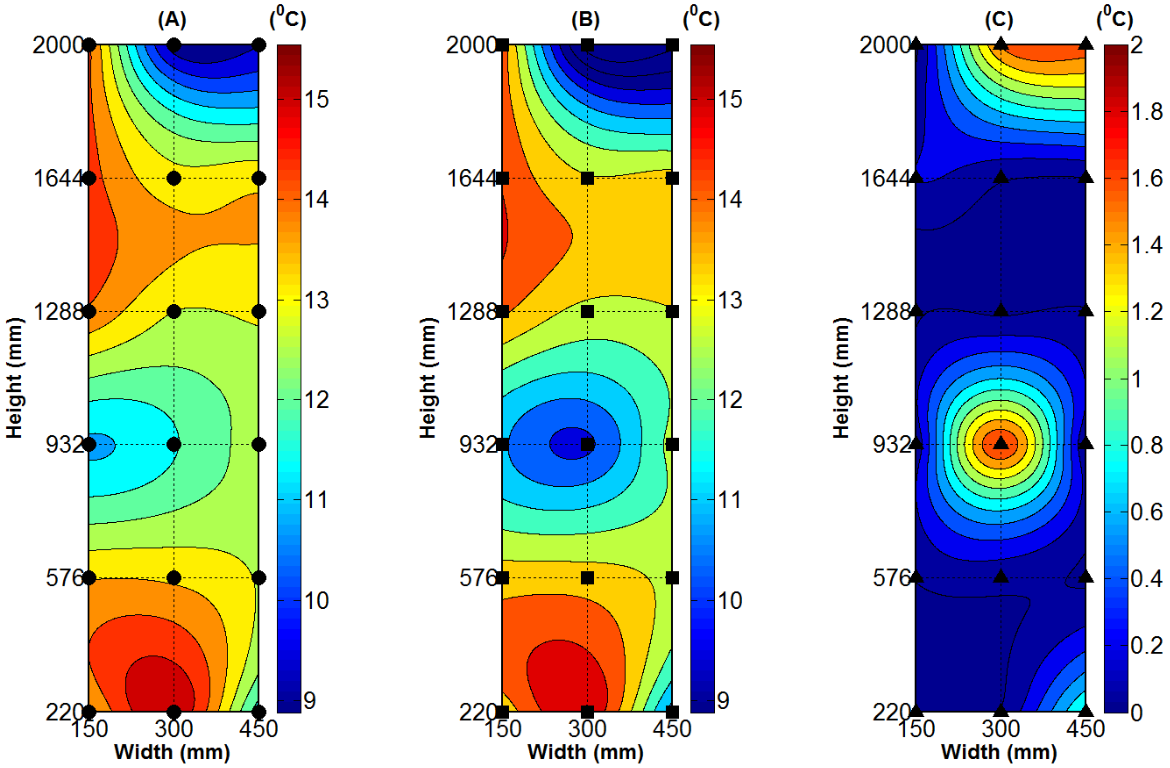


Figure 48: The capability of the POD-based framework in predicting new temperature data in zone-C. The interrogation point is arbitrarily assigned: $\left[\overline{Q}=5.5,\ \bar{t}=0.6\right]$. Experimental data are collected by 18 sensors shown by black circular markers arranged in the form of a 3 x 6 grid. Located at the exhaust of the test rack, the interrogation region is 300 mm (150 mm-450 mm) in width and 2000 mm (0 mm-2000 mm) in height. POD-based algorithm is applied on the measurement data. The deviations between experimental data and POD predictions are noted. (A) Mapping of experimentally-acquired temperature data. (B) Mapping of POD-predictions. (C) Mapping of the deviations between experimental data and POD predictions. The mapping from the data points is done via the Delaunay triangulation. The intermediate error limit assigned is 2 ^{0}C. The number of times the second level computation is invoked or Count is equal to 14.

Table 13 shows the prediction errors for various interrogation points defined by the first two columns $\overline{Q}$ and $\bar{t}$. For every combination of $\left(\overline{Q},\bar{t}\right)$, the third column of the error table tabulates the maximum error, defined as the maximum of the absolute deviations between measurement data and POD predictions. The fourth column tabulates the root mean square value of the relative errors. It is clear from Table 13 that the relative RMS error varies within a range of [5%-12%], with an average of 8.1% and a standard deviation of 2.4%.

Table 13: The prediction performance of the POD-based framework in zone-C

Normalized Heat Load $(\bar{Q})$	**Normalized Time** $(\bar{t})$	**Max. Error**(^{0}C)	**Relative RMS Error**
5.5	0.2	1.86	12%
	0.4	1.98	9%
	0.6	1.75	7%
	0.8	1.97	6%
5.75	0.2	1.62	8%
	0.4	1.91	11%
	0.6	1.99	7%
	0.8	1.74	5%

Model Prediction in Zone-D

For this case study, zone-D is the parametric subspace spanned by $(Q_{\text{int}}, t_{\text{int}}) \exists \left\{ \left(\bar{Q}_{\text{int}} > 5 \right) \cup \left(\bar{t}_{\text{int}} > 1 \right) \right\}$. Such a parametric space is particularly pertinent to the worst case analysis useful for the preliminary design of a data center. In this zone, an arbitrary interrogation point is chosen as (22 kW, 209 s); the corresponding normalized interrogation point is (5.5, 199/190). Figure 49 demonstrates the fidelity of POD-based temperature predictions in the spatial domain located at the rack exhaust plane. Figure 49 (a) shows ΔT_1 computed from the temperature measurement data and its mapping at the test rack exhaust. Figure 49(b) shows ΔT_1 computed from the POD-based predictions. The computational time for the POD-based prediction is on the order of 10 s (on an Intel®Core™2.Duo CPU of specification E8200 @ 2.66 GHz supported by 4 GB RAM).

Figure 49(c) shows the absolute deviations between the experimentally acquired ΔT_1 and the corresponding POD-predicted ΔT_1. For Zone-D, the error limit for the conditional block is set to be equal to four times the calibration error (0.5^0 C). The value of Count is observed to be equal to 14. The RMS deviation is equal to 1.02 ^{0}C.

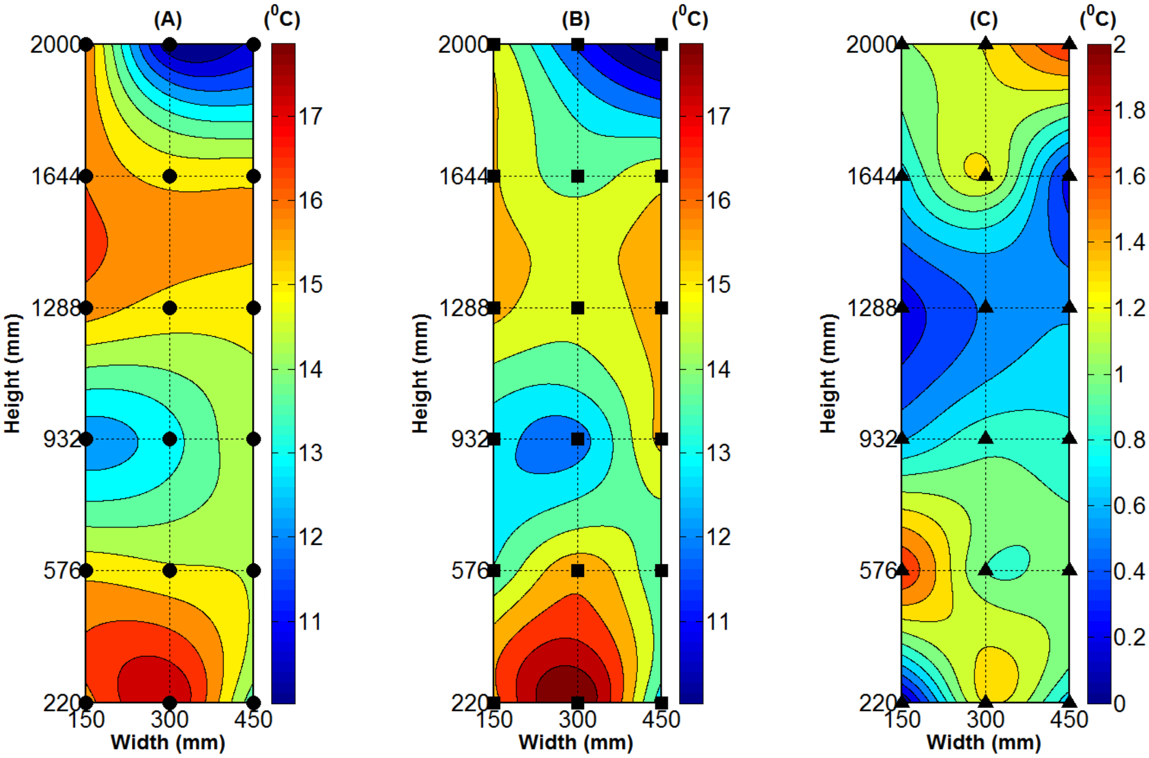


Figure 49: The capability of the POD-based framework in predicting new temperature data in zone-D. The interrogation point is arbitrarily assigned: $\left[\overline{Q} = 5.5,\ \bar{t} = 199/190\right]$. Experimental data are collected by 18 sensors shown by black circular markers arranged in the form of a 3 x 6 grid. Located at the exhaust of the test rack, the interrogation region is 300 mm (150 mm-450 mm) in width and 2000 mm (0 mm-2000 mm) in height. POD-based algorithm is applied on the measurement data. The deviations between experimental data and POD predictions are noted. (A) Mapping of experimentally-acquired temperature data. (B) Mapping of POD-predictions. (C) Mapping of the deviations between experimental data and POD predictions. The mapping from the data points is done via the Delaunay triangulation. The intermediate error limit assigned is 2 ^{0}C. The number of times the second level computation is invoked or Count is equal to 14.

POD coefficient computation in Zone-D requires extrapolation-based regression operations. Therefore, it is imperative that the model fidelity criterion be identified. The cut-off criterion is postulated as: the RMS value of relative errors is equal to 25%. Figure 50 shows the root mean variation of square relative deviations and Count with normalized extrapolation time, which is defined as: $\left(\bar{t}_{\text{exp}} = \frac{t_{\text{exp}} - t_{\text{final}}}{\Delta t} :\ t_{\text{final}} = 200\ \text{s},\ \Delta t = 1\ \text{s} \right)$.

It can be observed that for $\bar{t}_{\text{exp}} \leq 6$, the root mean square of relative error is substantially damped by the secondary procedure. For $\bar{t}_{\text{exp}} > 7$, the deviations increase rapidly to the extent that the secondary procedure fails to damp it. This is reflected in an exponential increase in root mean square values of relative errors and Count. Finally, the fidelity limit (25%) is reached at 15 s.

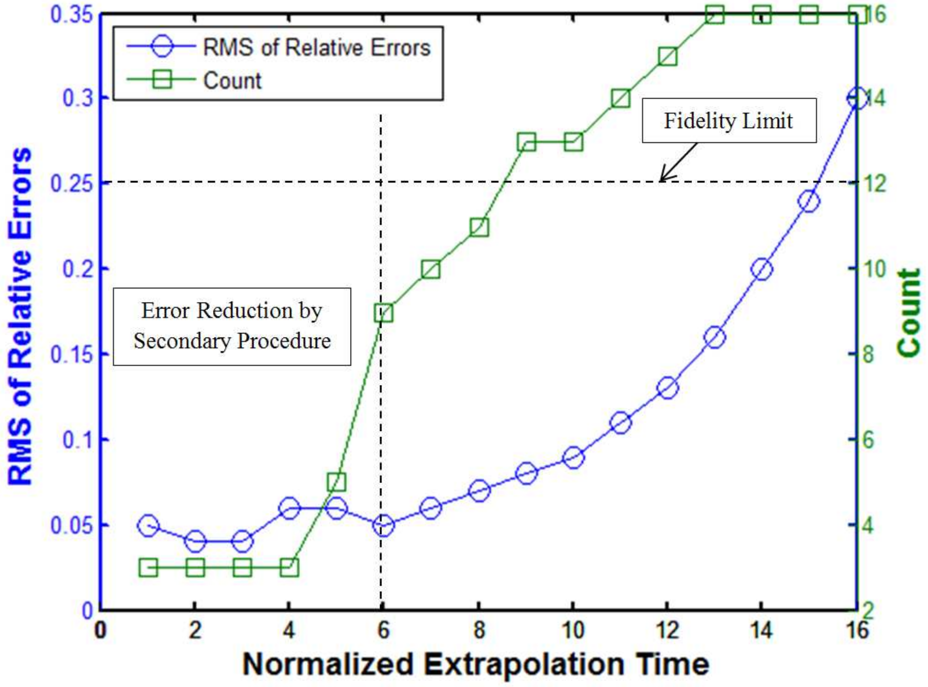


Figure 50: Root mean square of relative errors versus normalized extrapolation time for $\bar{Q} = 5.5$. All data points lie in zone-D. The second Y-axis shows the variation of Count with extrapolation time.

Coherent Structure of the Experimental Data

While the prediction fidelity of the proposed approach is established, it is worthwhile to inspect the efficiency of the algorithm. For efficient compact modeling, POD-based algorithms rely on model order reduction. For the present case study, a tolerance limit (for optimal number of POD modes) is set at 5%. For the present two-tier algorithm, two separate POD-based spectral analyses have been conducted. For an arbitrarily-assigned 5% tolerance limit, the principal component number for the primary procedure is 11 (which means (1-11/95) =88.5% order reduction), and that for the secondary procedure is 2 (which means (1-2/19) =89.5% order reduction). Figure 51 shows the POD-based mode decomposition spectrum. Figure 51(a) shows 11 optimal (dominant) POD modes for the primary procedure in the proposed algorithm. Expectedly, the first eigenvalue is the spectral radius or the maximum eigenvalue of magnitude 74. It captures as much as 75.9 % of the energy of the parametric temperature field. Figure 51(b) shows 2 optimal (dominant) POD modes for the secondary procedure of the proposed algorithm. In this case, the first eigenvalue captures 88.3% of the energy of the parametric temperature field. The POD modes essentially recognize the pattern of the dynamic temperature. Following this basic pattern, actual temperature responses are modulated by the parametric position on the interrogation point in the predictor space.

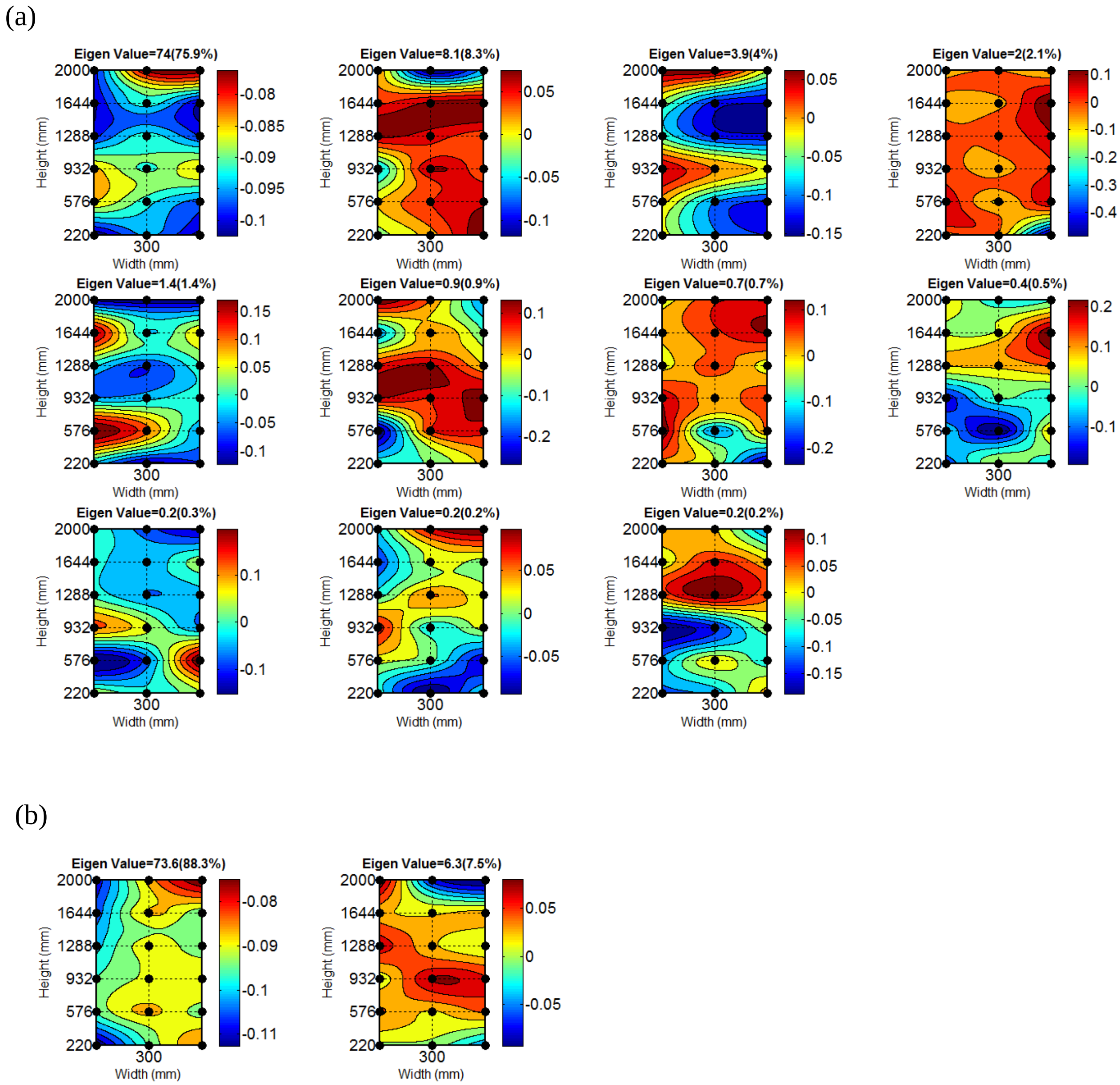


Figure 51: POD-based mode decomposition for the given data matrix. The tolerance level is set to be equal to 95%. For the primary procedure (Figure 51(a)), it takes 11 out of 95 POD modes to reach the 5% tolerance limit. On the other hand, the secondary procedure (Figure 51(b)) takes 2 out of 19 POD modes to reach the 5% tolerance limit.

Closure

A measurement-based air temperature prediction framework is developed with heat load and time as the parameters. The framework is capable of predicting air temperature in full factorial parametric space. Given that the data center cooling expenditure is directly proportional to its air temperatures, the proposed framework is suitable for life-cycle design of data center cooling systems.

CHAPTER 6
POD-BASED OPTIMIZATION FRAMEWORK FOR DYNAMIC COOLING ENERGY CONSUMPTION WITH TIME-VARYING CPU WORKLOAD

This chapter pertains to a measurement-based POD model to optimize dynamic cooling energy consumption with time-varying CPU workload.

Problem Statement

With the advent of cloud data centers (CDC), the mismatch between computing load-induced cooling demand and actual cooling supply is reducing data center energy efficiency significantly. The major cooling design problem for a CDC is its virtualized computing resources. Virtualization is creation of virtual machines that act like a real computing platform within an operating system. This application is virtual in the sense that it can be migrated rapidly to different computing nodes co-located within the same facility or even located outside the facility. The virtual machines are administered by a software application called hypervisor. The most widely used hypervisor in the industry is vSphere developed by VMware. Figure 52 shows the software stack present in a large-scale CDC. Due the stochastic nature of the application load, the computational load on a cloud data center and the associated heat load vary randomly. However, the lack of a demand-aware cooling allocation framework causes the facility to operate at the most conservative set point. That amounts to significant cooling over-provisioning, as shown in Figure 53. To avoid this wasteful cooling operation, the cooling supply curve needs to be dynamic and elastic. That is equivalent to improving cooling response time, which is a combination of controller feedback response time and cooling hardware latency. Most DC cooling hardware systems are controlled by proportional-integral-derivative controllers (PID controller). Improving their feedback response time requires efficient characterization of controller state-space. It demands an algorithm that can rapidly predict CPU temperatures for different cooling set-points. The CPU temperature is a critical decision parameter because all thermal management functionalities inside a server are based on CPU temperatures.

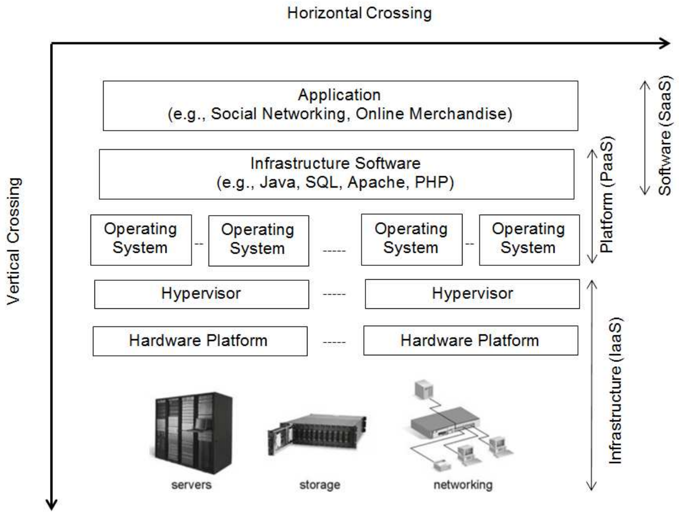


Figure 52: Functional view of a virtualized cloud data center

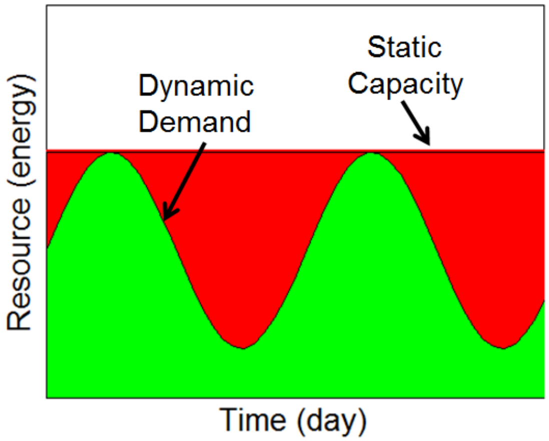


Figure 53: Mismatch in cooling demand and supply

In this chapter, a POD-based framework is developed that can rapidly predict CPU temperatures with cooling set points as the parameters. The framework is used to design an optimal cooling resource allocation system for a cloud data center. The potential cooling energy saving from the optimal controller action is also estimated.

Methodology

The proposed framework is based on a POD-based model order reduction subroutine, as discussed in CHAPTER 2. This POD subroutine improves the parametric granularity of a data matrix comprising experimental measurements. The POD subroutine offers an efficient, scalable, and reasonably accurate prediction algorithm. Figure 54 shows the flowcharts of the POD subroutine and modeling framework used in this paper. Details of the linear programming-based mathematical modeling of POD subroutine are well documented in the literature [54]. Three functional blocks in the framework are driven by a rack-level algorithm, a blade center-level algorithm, and a CPU- level algorithm. Each of these levels is applied to the data matrix complied at different length scales: blade, blade center, and rack. Each blade has two CPUs, each blade center has 14 blades with 28 CPUs, and each rack six blade centers with 168 CPUs. Figure 55 shows test rack architecture.

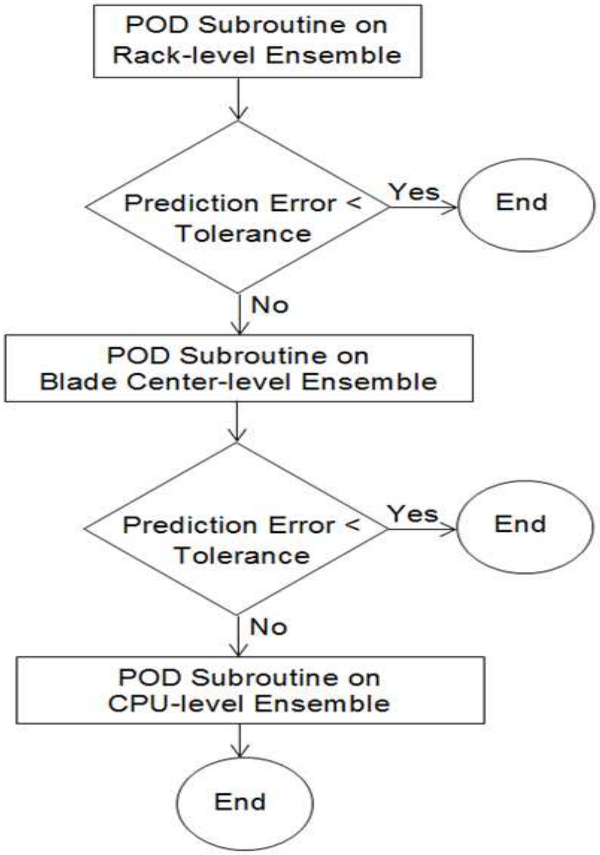

POD Subroutine on Rack-level Ensemble
Prediction Error < Tolerance
Yes
End
No
POD Subroutine on Blade Center-level Ensemble
Prediction Error < Tolerance
Yes
End
No
POD Subroutine on CPU-level Ensemble
End

Figure 54: POD-based CPU temperature prediction algorithm. Based on POD-based modal reduction, the algorithm is a three tiered statistical procedure: it starts from the ensemble of all CPU temperatures of an entire rack. If prediction error does not satisfy a certain tolerance criterion the algorithm proceeds to the blade center level ensemble. Thereafter, if the prediction error does not satisfy the tolerance criterion, the algorithm proceeds to the CPU level ensemble.

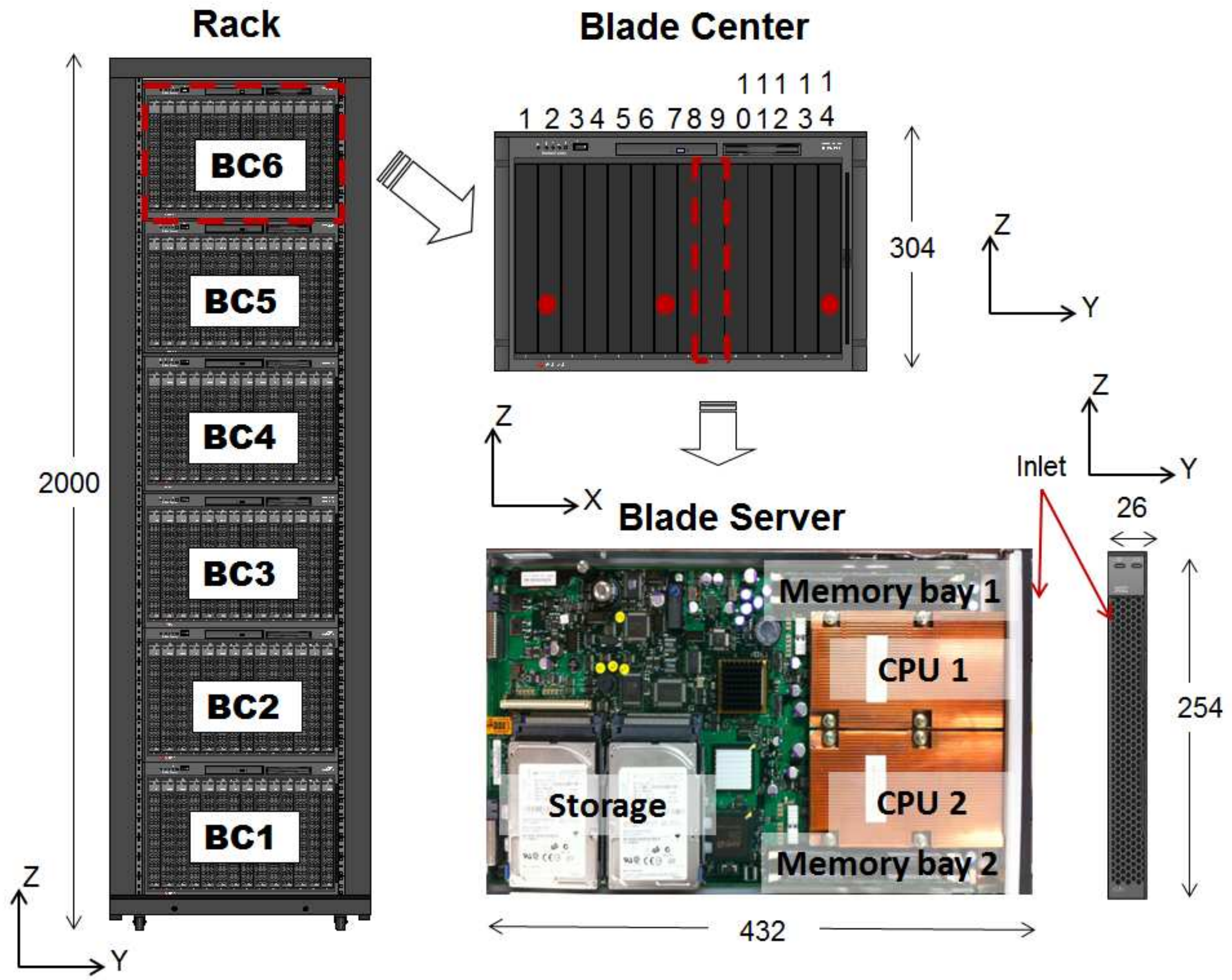


Figure 55: Details of a blade center rack (adapted from [60]). Red filled circles indicate the locations of the blades which are selected for the validation purpose. The selection is random: 2nd , 7th, and 14th blades from the left. For 2nd and 7th blade, CPU-1 temperatures are analyzed; whereas, for 14th blade CPU-2 blade temperature is analyzed.

The functional algorithm developed in this study is applied to the CPU temperature data measured from an IBM blade center rack as shown in Figure 55. The following symbols are extensively used in the following discussion pertaining to the functional algorithm. Respective numerical values for an IBM blade center with dual core server are mentioned.

Number of racks (n_racks) =1; Number of Blade Centers (n_BC)=6; Number of Blades per BC(n_blade)=14; Number of CPUs per Blade (n_CPU) =2; Number of Temperature Levels (n_T)=4; Number of Pressure Levels (n_P)=3; Total number of experimental data samples $(n\_sample = n\_T \times n\_P)$=12. Number of time samples (n_time)=44.

The algorithms for different length scales have sequential steps, described as follows:

Rack-level Algorithm

1. Compile CPU temperature data matrix, T_{data}^{rack} for the entire rack for all experimental samples. T_{data}^{rack} is a matrix of size $n\_time \times N;$ where $N = n\_CPU \times n\_blade \times n\_BC \times n\_racks \times n\_sample.$ For the present case study, $N = 2016.$
2. Compute the row-wise mean of T_{data}^{rack} to determine T_0
3. Apply Power iteration-based POD on T_{data}^{rack} to compute POD modes, ψ_{rack}, POD coefficient matrix, B_{rack}.
4. Based on 99% tolerance criteria, the principal component number is determined to be equal to 42. It means 97.9% data compression.
5. Cut ψ_{rack} and B_{rack} based on principal component number.

6. Segment B_{rack} based on CPU locations and apply bilinear interpolation on the segmented matrix to determine POD coefficient vector at the interrogation point, b_{int}.
7. The interrogation temperature is predicted as: $T_{prediction}^{rack} = T_0 + \psi_{rack} \otimes b_{\text{int}}$.
8. Determine percentage error: $e = \frac{\left|T_{data}^{rack} - T_{prediction}^{rack}\right|}{T_{data}^{rack}} \times 100.$
9. If e>tol, then go to Blade Center-level algorithm, where tol=5%.

Blade Center-level Algorithm

1. Compile CPU temperature data matrix, T_{data}^{BC} for the entire Blade Center corresponding to the interrogation CPU across all experimental samples. T_{data}^{BC} is a matrix of size $n\_time \times N;$ where $N = n\_CPU \times n\_blade \times n\_sample.$ For the present case study, $N = 336.$
2. Compute the row-wise mean of T_{data}^{BC} to determine T_0
3. Apply Power iteration-based POD on T_{data}^{BC} to compute POD modes, ψ_{BC}, POD coefficient matrix, B_{BC}.
4. Based on 99% tolerance criteria, the principal component number is determined to be equal to 42. It means 87.5% data compression.
5. Cut ψ_{BC} and B_{BC} based on principal component number.
6. Segment B_{BC} based on CPU locations and apply bilinear interpolation on the segmented matrix to determine POD coefficient vector at the interrogation point, b_{int}.

7. The interrogation temperature is predicted as: $T_{prediction}^{BC} = T_0 + \psi_{BC} \otimes b_{\text{int}}$.

8. Determine percentage error: $e = \frac{\left| T_{data}^{BC} - T_{prediction}^{BC} \right|}{T_{data}^{BC}} \times 100.$

9. If e>tol, then go to CPU-level algorithm, where tol=5%.

CPU-level Algorithm

1. Compile CPU temperature data matrix, T_{data}^{CPU} for the interrogation CPU across all experimental samples. T_{data}^{CPU} is a matrix of size $n\_time \times N;$ where $N = n\_sample.$ For the present case study, $N = 12.$

2. Compute the row-wise mean of T_{data}^{CPU} to determine T_0.

3. Apply Power iteration-based POD on T_{data}^{CPU} to compute POD modes, ψ_{CPU}, POD coefficient matrix, B_{CPU}.

4. Based on 99% tolerance criteria, the principal component number is determined to be equal to 11. It means 8.3% data compression.

5. Cut ψ_{CPU} and B_{CPU} based on principal component number.

6. Segment B_{CPU} based on CPU locations and apply bilinear interpolation on the segmented matrix to determine POD coefficient vector at the interrogation point, b_{int}.

7. The interrogation temperature is predicted as: $T_{prediction}^{CPU} = T_0 + \psi_{CPU} \otimes b_{\text{int}}$.

8. Determine percentage error: $e = \frac{\left| T_{data}^{CPU} - T_{prediction}^{CPU} \right|}{T_{data}^{CPU}} \times 100.$

The capability of high-fidelity temperature generation can be leveraged to determine the optimal cooling environment for a time-varying workload profile. The mathematical optimization problem of the optimal cooling design can be formulated as:

$$\left.\begin{array}{l}\begin{cases}\text{Maximize } T_{\mathrm{Sup}} \\ \text{Minimize } \Delta P_{\mathrm{RDHx}}\end{cases}. \\ \\ \text{Constraint: } \max(T_{\mathrm{CPU}}) < 65\ {}^{0}\mathrm{C}.\end{array}\right\} \tag{3.22}$$

The optimal cooling design offers most cost-efficient DC operation because the maximization of CRAC supply temperature under the given constraint amounts to optimizing chiller flow rate. It directly impacts 64% of data center cooling cost as discussed in CHAPTER 1. On the other hand, the minimization of rear door heat exchanger driving pressure under the given constraint optimizes building chilled water pump work which amounts to 9% of data center cooling cost. The constraint in the optimization problem specifies the reliability limit of most modern processors.

This optimization problem can be solved using a POD-based temperature signal generator with the iterative procedure, shown in Figure 56. The initial starting cooling resource set-point is determined by several factors, including the class of a data center, cooling hardware operational capability (such as set-points of RDHx). Then, the POD algorithm computes the CPU temperatures. If the maximum CPU temperature is below the critical CPU temperature of 65 ^{0}C, then the initial operating point is the optimal point. Otherwise, cooling set points are adjusted and CPU temperatures are iteratively recomputed until maximum CPU temperature goes below the critical CPU temperature.

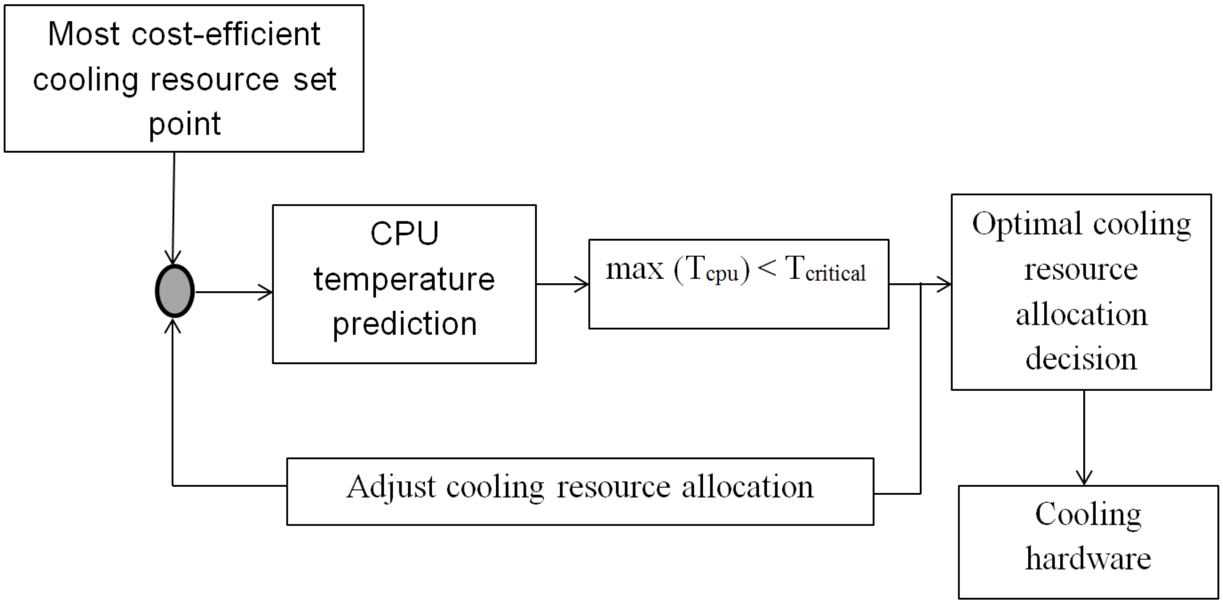

Most cost-efficient cooling resource set point
CPU temperature prediction
max (T_{cpu}) < $T_{critical}$
Optimal cooling resource allocation decision
Adjust cooling resource allocation
Cooling hardware

Figure 56: POD-based iterative procedure to compute optimal cooling set points

For this particular case-study, the cooling equipment used are the test CRAC unit and the test RDHx unit. While a CRAC unit provides room-level cooling, an RDHx provides rack-level localized cooling. Therefore, a CRAC unit is more energy-intensive compared to an RDHx unit. In light of that fact, an optimal cooling infrastructure design for a given test rack demands the first-level of cooling from the corresponding RDHx unit. The CRAC supply temperature should be modulated only when RDHx unit pressure has been pushed to its maximum level. That affects the adjustment of cooling resource set-points to identify cost-effective operation paradigm. Computationally, it means using RDHx pressure as the inner variable and CRAC supply temperature as the outer variable in the iterative optimization loop.

Experimental Setup

Figure 57 shows the experimental raised-floor DC facility (located at Atlanta, GA at elevation ~1,027' (313 m)) with three computer room air conditioning (CRAC) units. For this experimental study, rack D-5 is used as the test rack. Installed with an underfloor plenum of depth 912 mm (3') and drop ceiling height 1727 mm (5'8"), the facility height is 3048 mm (10'). For this case study, only CRAC-1 which is an APC 5 kW downflow unit (CW-140-C-KA-D) is operational. The rated cooling capacity of this CRAC unit is 140 kW (40 ton). This unit is installed with a belt-driven centrifugal fan which is rated to supply 12,200 CFM (5.76 m^3/s) cooling air. The cooling hardware (cooling coil and electric heater) inside CRAC unit is controlled by micro-processor –based PID controller with supply air temperature as the set point. The relative humidity of the supply air is maintained at 40%. 10 of 14 racks in the test facility are installed with RDHx-s (Vatte Liquid Cooling) of nominal cooling capacity 18 kW and maximum cooling capacity 24 kW. The overall cooling capacity of these RDHx-s is controlled by centralized pressure differential set point. Table 14 specifies the experimental condition. The heat load column shows average rack heat load, measured by Rack Load Tester. The tile flow column shows cooling air coming out through the perforated tile, measured by Balometer (Shortbridge ADM-860C). The rack flow column shows air drawn by rack fans, measured by Rack Load Tester.

The Rack Load Tester consists of an array of 15 x 3 sensors (45 sensors). It is placed at the outlet of the rack attached to an aluminum frame structure covered by a cloth skirt to prevent air from bypassing the sensors. Each sensor consists of a thermistor to measure temperature, and a constant temperature hot wire anemometer to measure air velocity. The sensors used were standard Accusense F900. These specification measurements are done when CRAC supply temperature is kept at 15.5 ^{0}C (60 F) and RDHx differential pressure is kept at 8 psi.

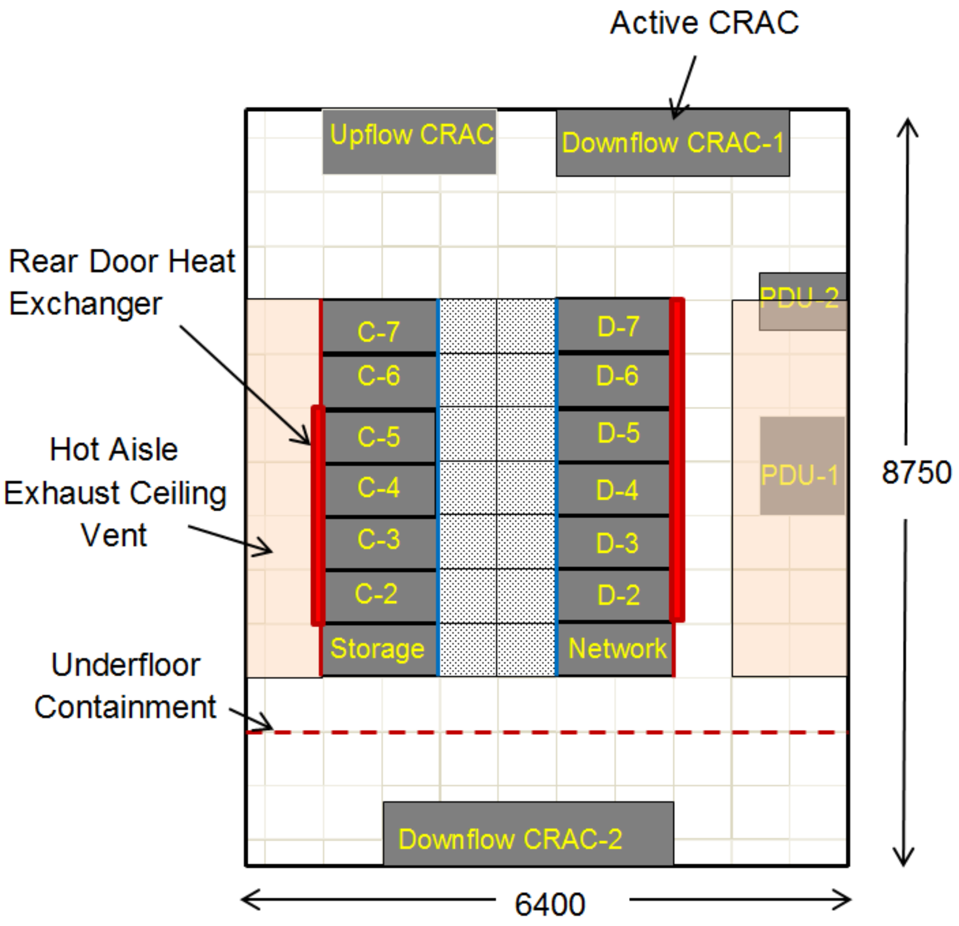


Figure 57: Details of the Experimental Setup

It can be observed that Tile Flow is 6238 CFM and Rack Flow 20278 CFM. Since Tile Flow or cooling air supply is 69.2% lower compared to Rack Flow or rack demand, the facility is severely under-provisioned.

Table 14: Specification of the Experimental Setup

Rack	Description	RDHx	Heat Load	Tile Flow	Rack Flow
			kW	**CFM**	**CFM**
			(+/-) 5%	**(+/-) 5%**	**(+/-) 5%**
C-1	Storage	No	5.2	397	1015
C-2	IBM Blade Center Rack	Yes	11.4	490	1579
C-3	IBM Blade Center Rack	Yes	11.7	533	1651
C-4	IBM Blade Center Rack	Yes	11.9	390	1617
C-5	IBM Blade Center Rack	Yes	11.8	470	1447
C-6	1-U Server Rack	No	0.0	488	1200
C-7	Empty	No	7.8	439	267
D-1	Network	No	4.5	371	1061
D-2	IBM Blade Center Rack	Yes	11.8	434	1603
D-3	IBM Blade Center Rack	Yes	11.7	377	1658
D-4	IBM Blade Center Rack	Yes	11.3	415	1724
D-5	IBM Blade Center Rack (Test Rack)	Yes	11.9	484	1716
D-6	IBM Blade Center Rack	Yes	11.9	483	1858
D-7	IBM Blade Center Rack	Yes	12.0	467	1882

In this study, rack D-5 is used as the test rack. It consists of 6 IBM blade centers. Each blade center contains 14 blade servers. Each blade has two dual-core AMD Opteron 270 processors, 4 GB of memory, and is installed with the VMware vSphere Hypervisor (ESXi) v4.1. The blades are interconnected via a Force 10 E1200 switch over a flat IP space. Each blade hosts one virtual machine installed with 64-bit Ubuntu 11.10. Since these blades are CPU-dominant in terms of power consumption, we configure those virtual machines with 4 virtual CPUs to exploit the maximum power usage. The VMware vSphere server and client software are used to manage the cloud. For the purpose of profiling, the workload in a given VM needs to be precisely controlled, which is performed by wileE benchmark [74]. It enables generation of user-defined transient CPU and memory utilization profiles for an arbitrary period of time. To emulate the real-world workload, the workload is discretized into instances of different wileE workload. The wileE benchmark can automatically perform those instances in time sequence via the use of multicast. The test rack is equipped with a PI system developed by OSIsoft. Via this PI system, the data streams generated from various sensors are transmitted to SQL database in real time. The measurement data are retrieved from this database, and subsequent analyses are performed using the framework described in the previous section. The CPU temperature data for this experiment

Figure 58 shows CPU/memory usage profiles used in this case study. The duration of each profile is 3000 s. There are four types of workloads:

Type-1

The workload is a typical load profile of an IDC. It has two fundamental components: the first one is a regular periodic component, and the other one is a discontinuous component. The latter represents a flash crowd in a data center. These flash crowd events are characterized by very high IT demand for a short duration of time. At $t = 0,$ the profile starts at 35% utilization. Then, it varies in a sinusoidal manner with 25% amplitude and 3600 s time period. The flash crowd occurs at 2400 s when the utilization profile suddenly shoots up. Within 30 s, it increases to 90% utilization. The resource utilization remains constant for 30 s between 2430-2460 s. Then, it plummets to the original periodic profile within next 60 s. From 2520 s, it continues the regular periodic profile.

Type-2

This profile simulates a square waveform with 70% amplitude and a half time period of 600 s. This particular waveform has two peaks: the first one starts at 600 s and continues till 1200 s, while second one starts at 1200 s and continues till 2400 s. The lower IT utilization point in this profile is 10%; on the other hand, the higher IT utilization point is 80%.

Type-3

This profile combines a square waveform with a sine waveform. The square waveform lasts from 0-1800 s. It has one peak between 600-1200 s with 25% amplitude. It has a lower IT utilization point of 35% and higher IT utilization point of 60%. The subsequent part of this combined waveform is a sine wave with 25% amplitude with 3600 s time period. It starts at 1800 s with 35% utilization. It subsequently reaches 10% utilization at 2700 s.

Type-4

This profile is related to an actual cloud computing service provider. This profile is characterized by a sudden jump at 280 s. While this profile has (0.24 ± 0.0126) % CPU utilization before 280 s, it shoots up to (98.38 ± 1.14) % CPU utilization after 280 s.

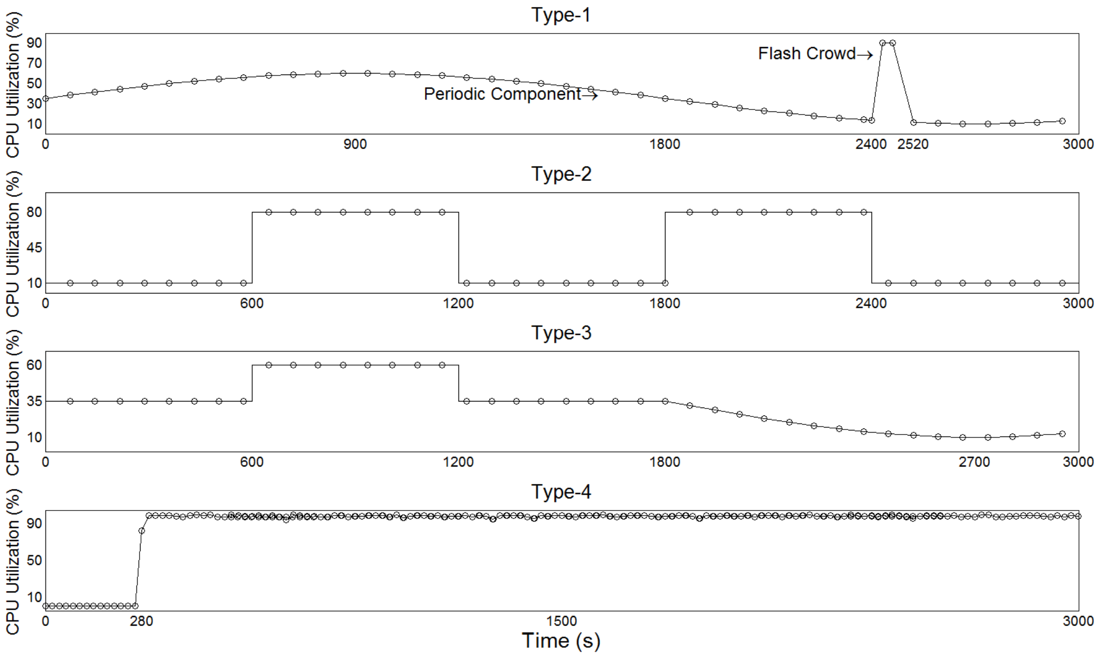


Figure 58: Simulated Load Profiles

For studying the sensitivity of the predictive framework with respect to the uncertainty in the workload pattern, a distorted profile of Type-2 waveform is developed. Figure 59 shows a Type-2 profile along with a distorted Type-2 profile.

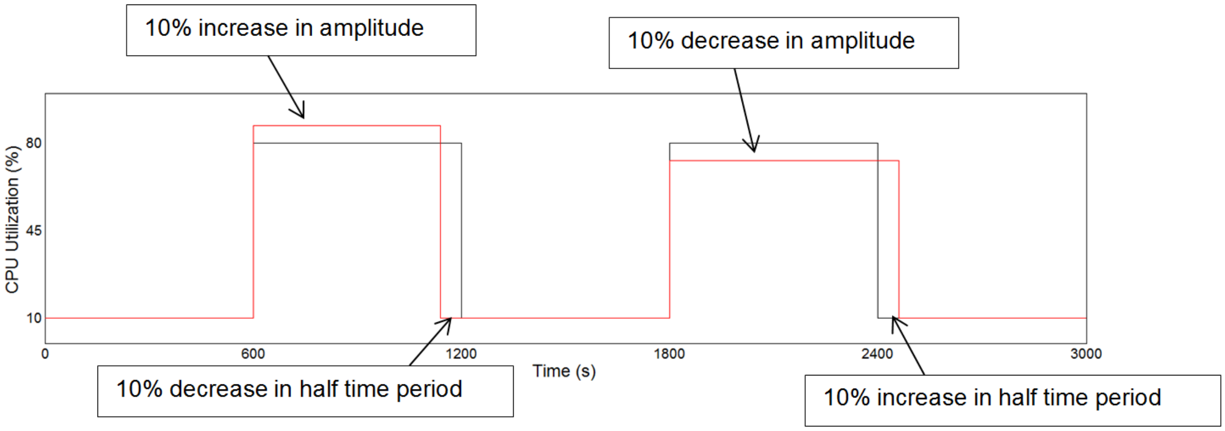


Figure 59: Distorted Type-2 Profile. The black line is the original Type-2 profile. The red line is the distorted Type-2 profile.

While the black line is the original Type-2 profile, the red line is the distorted Type-2 profile. The original Type-2 profile has been changed in four places to obtain the distorted profile. First, the amplitude of the first peak is increased by 10% to 73.5%. Second, the half time period of the first square pulse is decreased by 10% to 540 s. Therefore, the first peak finishes at 1140 s instead of 1200 s. Third, the amplitude of the second peak is decreased by 10% to 66.5%. Finally, the half time period of the second square pulse is increased by 10% to 660 s. Therefore, the second peak finishes at 2460 s instead of 2400 s.

Case Study

The POD-based framework is applied on the measured CPU temperature data to improve its parametric granularity. While CPU temperature is used as the response variable, a combination of CRAC supply temperature $\left(T_{\mathrm{sup}}\right)$ and RDHx differential pressure $\left(\Delta P_{RDHX}\right)$ is used as a predictor variable. The objective function is to improve the parametric granularity of CPU temperature data in $\left(T_{Sup}, \Delta P_{RDHX}\right)$ parametric space. This paper applies the framework on CPU temperature data collected with 12 different combinations of $\left(T_{Sup}, \Delta P_{RDHX}\right)$. The output is generated for three different prediction points. Figure 60 shows the parametric input space. There are four different levels of T_{Sup}: 17 ^{0}C, 21 ^{0}C, 25 ^{0}C, and 29 ^{0}C. These temperature points are chosen to keep this experimental study pertinent to American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) TC9.9 recommended thermal guideline. While 21 ^{0}C and 25 ^{0}C lie within the recommended range of [18 ^{0}C-27 ^{0}C], 17 ^{0}C and 29 ^{0}C lie within the allowable range of [15 ^{0}C-32 ^{0}C]. On the other hand, three different values of RDHx differential pressure are chosen: 4.0 psi (27579 Pa), 7.0 psi (48263.3 Pa), and 10.0 Psi (68947.6 Pa) from possible values between [0-12 psi]. For the rest of the paper, psi will be used as a pressure unit (1 psi=6894.75 Pa). Indicated by the red circles, three output points are chosen at the furthest possible parametric locations: (19 ^{0}C, 8.5 psi); (23 ^{0}C, 6.0 psi); and (27 ^{0}C, 5.5 psi). These output points are arbitrarily chosen and drawn from different regions of the parametric space. Therefore, it can be argued that if any framework predicts accurately in these point, it will predict accurately in the entire parametric region.

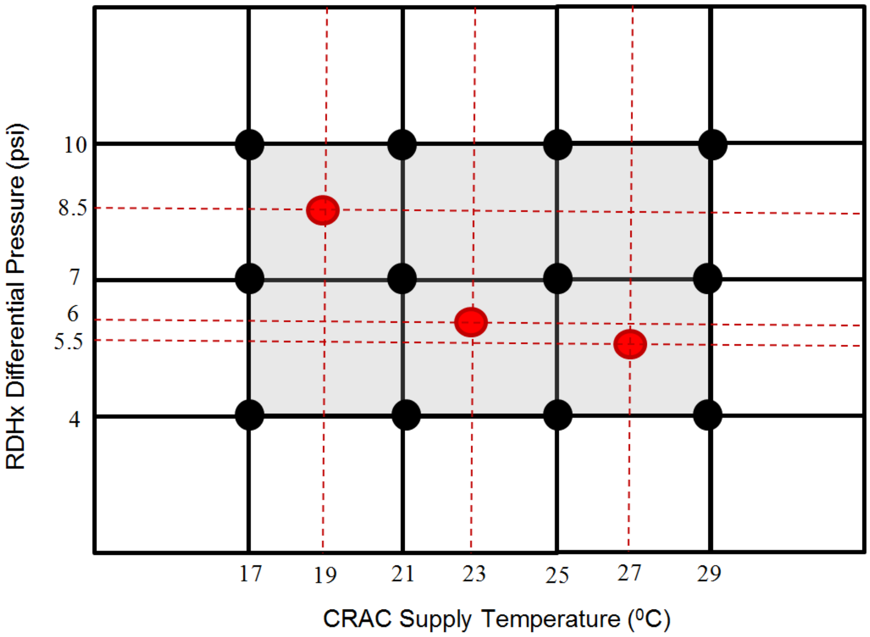

RDHx Differential Pressure (psi)
10
8.5
7
6
5.5
4
17
19
21
23
25
27
29
CRAC Supply Temperature (⁰C)

Figure 60: The interrogation space is comprised different possible combinations of CRAC supply temperature $\left(T_{\text{sup.}}\right)$ and RDHx differential pressure $\left(\Delta P_{RDHx}\right)$. Black circles indicate the parametric locations of the input ensemble. Red circles indicate the parametric locations of the prediction points.

Results and Discussions

Figure 61 shows transient CPU temperatures for six different blade centers for Type-1 workload operating at (17 ^{0}C, 4 psi). There are 28 CPUs in a blade center. Therefore, CPU temperature data are densely packed. For visualization purpose, three CPUs are picked. They are CPU1 at Blade2; CPU1 at Blade7, and CPU2 at Blade14.

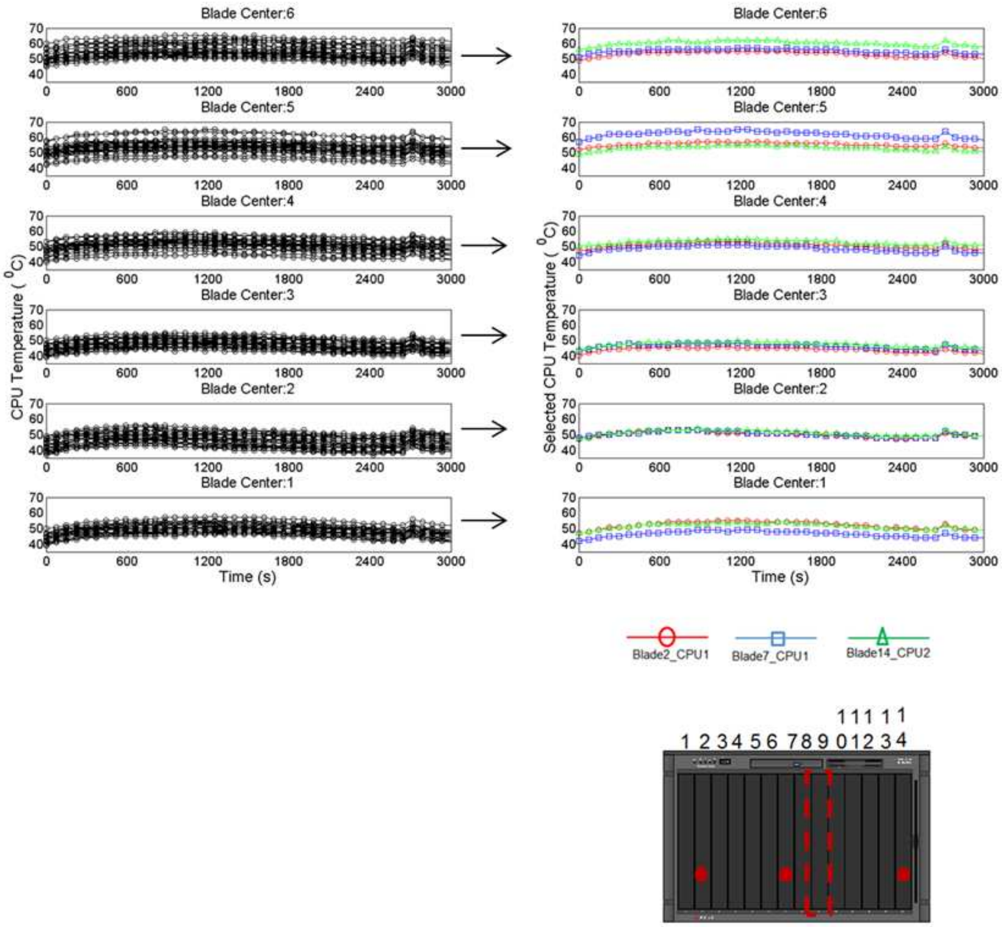


Figure 61: CPU temperature data at (17 ^{0}C, 4 psi) operating condition for Type-1 load profile. Three visualization CPUs are chosen: CPU1 at Blade2; CPU1 at Blade7, and CPU2 at Blade14.

It can be readily observed from Figure 61 that the average CPU temperature is highest near the top of the rack. Figure 62 shows average CPU temperature for each blade center. While average CPU temperature is equal to 50.4 ^{0}C in blade center 1, that is 55.4 ^{0}C in blade center 6. Higher CPU temperature near the top of the rack is caused by the warmer rack inlet temperature near the top of the rack due to hot air recirculation near the top of the rack. The effect of hot air recirculation is pronounced in this case-study because the cooling air supply in this experimental facility is severely under-provisioned. Moving down the rack, average CPU temperature decreases as the effect of hot air recirculation gradually diminishes. Nevertheless, average CPU temperature increases unexpectedly for blade center 1. This is because of the Venturi effect at the foot of the rack. Although there is a distinct trend in the vertical direction, there is no such trend in the horizontal direction. The CPU temperatures seem to undergo a random spatial variation within a blade center.

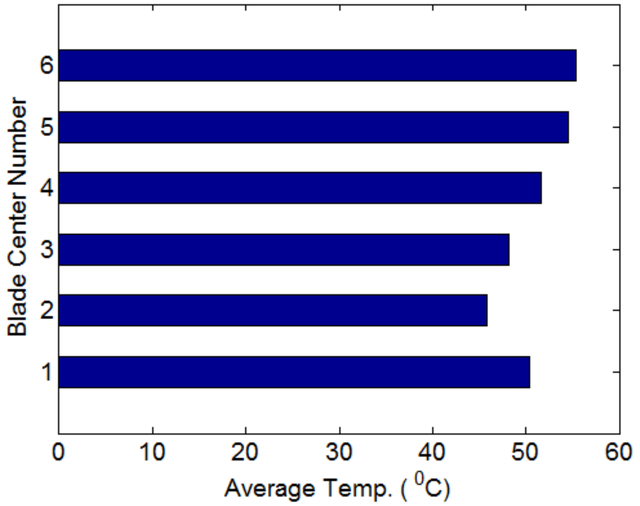


Figure 62: Average temperature for different blade centers for Type-1 workload at (17 ^{0}C, 4 psi)

Each IBM blade center has two mutually-facing centrifugal fans. Figure 63 shows transient evolution of server fan speeds. The speeds of these fans are controlled by rack inlet temperatures. Therefore, it is expected that fan speeds near the top of the rack would be higher than near the bottom. Indeed, it is observed that Blade Center (BC) 6 and BC 5 fans are operating at 100% capacity which can be explained by their larger inlet temperatures. Fans in BC4, BC3, and BC2 show a transient pattern. Fans start to operate from [95%, 85%, 60%] respectively; then, fan speed increases at around 900s. Fan speeds fall at around 2500 s. This is somewhat consistent with Type-1 load profile. Fan speed increases during peak power and flash crowd events. It falls as the amplitude of the workload decreases. Surprisingly, the BC-1 fan speed remains flat which can be explained from local cooling dominated by the Venturi effect.

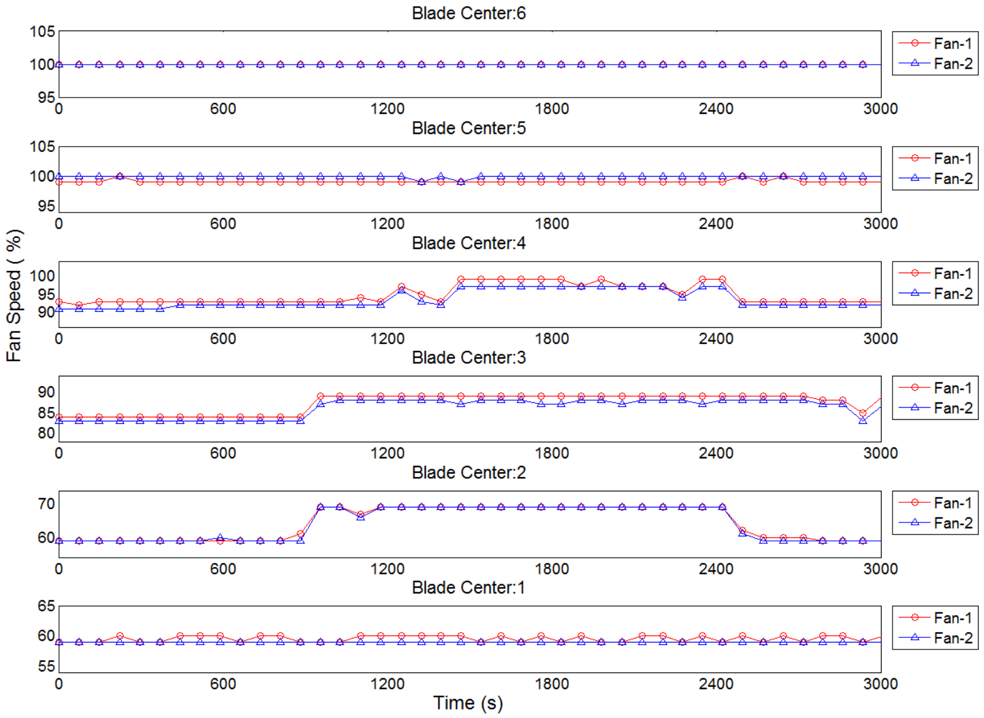


Figure 63: Fan speed variation with Type-1 workload at (17 ^{0}C, 4 psi) cooling set-points

The CPU temperatures and server fan speed show the expected behavior. A similar pattern is expected to continue for other cooling environments. The data matrix is compiled based on experimentally-measured CPU temperature data. The proposed algorithm is applied on the data matrix and the CPU temperature signals are computed. For a fidelity check, the percentage error between CPU temperature data and predictions are computed. Table 15 shows the root mean square value of time-averaged (0-3000 s with 44 time samples) error across 168 CPUs in the test rack.

Table 15: Percentage error for different workload types at different operating points

Workload Type	**Cooling Set-point**	**RMS of Time-averaged Error**	**99.7% Confidence Interval**
Type-1	(19 ^{0}C,8.5 psi)	2.39%	3.28%
Type-1	(23 ^{0}C,6.0 psi)	1.75%	2.41%
Type-1	(27 ^{0}C,5.5 psi)	2.44%	3.39%
Type-2	(19 ^{0}C,8.5 psi)	3.38%	5.34%
Type-2	(23 ^{0}C,6.0 psi)	2.57%	3.1%
Type-2	(27 ^{0}C,5.5 psi)	2.60%	4.49%
Type-3	(19 ^{0}C,8.5 psi)	3.30%	4.01%
Type-3	(23 ^{0}C,6.0 psi)	2.78%	3.33%
Type-3	(27 ^{0}C,5.5 psi)	2.16%	2.42%
Type-4	(19 ^{0}C,8.5 psi)	2.23%	2.46%
Type-4	(23 ^{0}C,6.0 psi)	2.56%	2.8%
Type-4	(27 ^{0}C,5.5 psi)	2.35%	2.81%

Table 15 shows that the maximum value of the RMS of time-averaged error for Type-1 workload is equal to 2.56%, that for Type-2 workload is equal to 3.38%, that for Type-3 workload is equal to 3.3%, and that for Type-4 workload is equal to 2.56%. On the other hand, the maximum error bound for the numerical procedure is 10%. Hence, the developed framework is accurate within a +/-10% uncertainty interval. However, as suggested by the RMS values, the framework is predicting much better than the 10% upper bound. Hence, it can be claimed that the proposed POD-based framework is capable of generating high-fidelity temperature predictions for any cooling operating points $\left(T_{\text{int}}, P_{\text{int}}\right)$ such that $T_{\text{int}} \in \left[17\ ^{0}\text{C}, 29\ ^{0}\text{C}\right] \cup P_{\text{int}} \in \left[4\text{ psi}, 10\text{ psi}\right]$.

Given that the fidelity of the prediction framework is established, the optimal controller (as shown in Figure 56) for different workload profiles can be designed. The initial starting point is (29 ^{0}C, 4 psi). This is the most cost-efficient point. Then, if the maximum CPU temperature is computed to be more than the critical limit of 65 ^{0}C, the cooling set-points are adjusted by 0.5 ^{0}C increments in CRAC supply air temperature and 0.5 psi decrements in RDHx.

Figure 64 shows optimal cooling resource allocation for Type-1 workload. Figure 64(a) shows the load profile for Type-1 workload, which is a sine waveform with amplitude 25% and period 3600 s. The first peak is reached at 900 s. At 1800 s, the waveform reaches its half-time period. These time instants are marked by dotted lines. Additionally, the beginning (2400 s) and the end (2520 s) of the flash crowd profile are also marked with dotted lines. Figure 64(b) shows the maximum, average, and minimum CPU temperature profiles for the optimal cooling set-point envelope. The optimization procedure determines the most cost-effective cooling set-points under the given constraint. Figure 64(c) and Figure 64(d) show optimal CRAC supply temperature and RDHx pressure set points, respectively. Initially, CRAC supply temperature and RDHx pressure could satisfy the temperature constraint by operating at the most cost-effective set point of (29 ^{0}C, 4 psi). With increase in CPU utilization and associated CPU power dissipation, cooling set-points need to deviate from the cost-efficient operational mode. In fact, the RDHx pressure set-point jumps rapidly from 4 psi to 10 psi between [363 s-436 s]. At 436 s, the CRAC supply temperature responds by dipping down by 0.5 ^{0}C to 28.5 ^{0}C and remains there till 581 s. Between 581 s-654 s, it increases to move back to the 29 ^{0}C set-point. Between 654 s-732 s, it decreases to 28.5 ^{0}C.Between 720 s-792 s, it plummets to 23 ^{0}C. RDHx pressure set-point, on the other hand, remains somewhat flat after 436 s except for experiencing a minor dip by 0.5 psi between 581 s-732 s. Similar dynamic adjustments of cooling set-points continue in the entire time domain based on the proposed mathematical optimization procedure, shown in Figure 56. The cooling hardware response during the flash crowd between 2400 s-2520 s is particularly interesting. Between 2415 s-2488 s, there is a steep jump in RDHx pressure set-point from 4 psi to 8 psi. On the other hand, the CRAC supply air temperature surprisingly increases from 28.5 ^{0}C to 29 ^{0}C during that time window. During the next part of the flash crowd between 2488 s-2560 s, the RDHx pressure decreases by 0.5 psi and the CRAC supply temperature remains flat. Between RDHx and CRAC, RDHx is more responsive to rack CPU utilization or power variation. It can be explained by the fact that RDHx is more tightly-coupled to a given rack. While CRAC is responsible for cooling of several racks inside the facility, RDHx is responsible for a given rack.

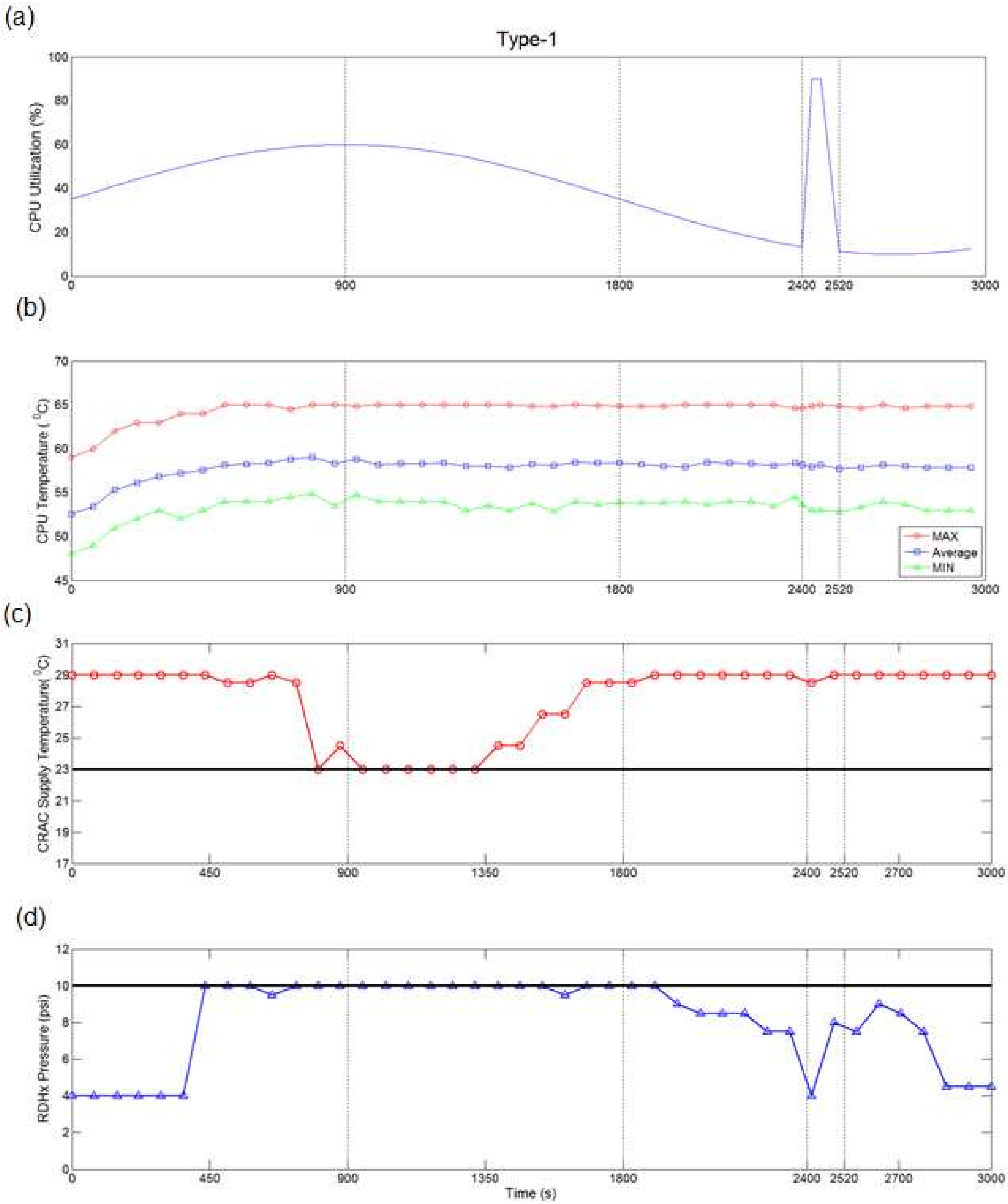


Figure 64: Optimal cooling resource allocation for Type-1 workload

The black lines in Figure 64(c) and Figure 64(d) show the most conservative set-points for CRAC (23 ^{0}C) and RDHx (10 psi), respectively. If there is no optimal control procedure, a conservative DC facility manager would operate his/her data center cooling at these points. Therefore, the developed approach has the potential to save cooling energy. Figure 65 shows the cooling energy saving potential of the developed optimization framework. The energy calculations are done by the simple thermodynamic model developed in CHAPTER **1**. Figure 65(a) shows the fraction of energy usage by the CRAC unit operating in the optimal mode to that by a similar CRAC unit operating in the conservative mode. On the other hand, Figure 65(b) shows the fraction of energy usage by the RDHx unit operating in the optimal mode to that by a similar RDHx unit operating in the conservative mode. The root mean square value of the fraction of energy saving in CRAC is equal to 51.4%. On the other hand, that value in RDHx is equal to 18.5%.

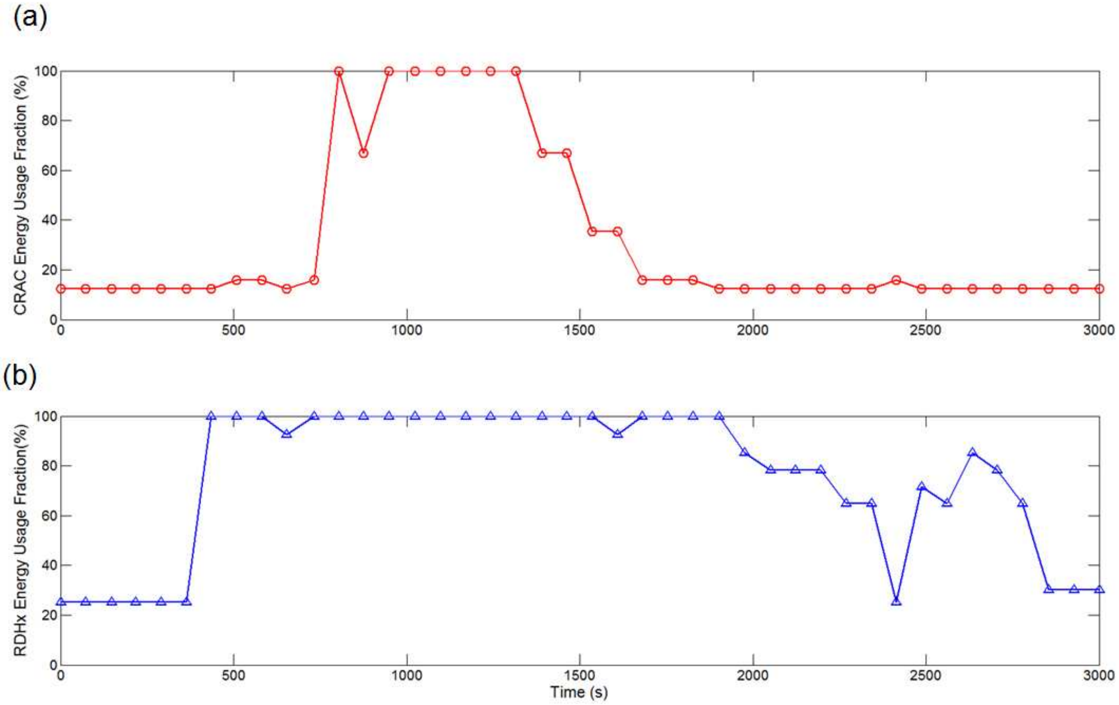


Figure 65: Cooling energy saving for Type-1 workload

Figure 66 shows the optimal cooling resource allocation for a Type-2 workload. Figure 66(a) shows the load profile for Type-2 workload, which is a square waveform with amplitude 70% and half-time period 600 s. Figure 66(b) shows the maximum, average, and minimum CPU temperature profiles for the optimal cooling set-point envelope. The optimization procedure determines the most cost-effective cooling set-points under the given constraint. Figure 66(c) and Figure 66(d) show optimal CRAC supply temperature and RDHx pressure set points, respectively. Initially, CRAC supply temperature and RDHx pressure could satisfy the temperature constraint by operating at the most cost-effective set point of (29 ^{0}C, 4 psi). With increase in CPU utilization and associated CPU power dissipation, cooling set-points need to deviate from the cost-efficient operational mode. As expected, there are major changes in cooling set-points around the discontinuities of the step profile at 600 s, 1,200 s, 1,800 s, and 2,400 s. In fact, the RDHx pressure set-point jumps rapidly from 4.5 psi to 8 psi between [512 s-584 s]. On the other hand, the CRAC supply temperature falls from 29 ^{0}C to 25 ^{0}C between [584 s-658 s]. Similar dynamic variations of cooling set-points are observed across the entire time window. There are some counter-intuitive variations in CRAC supply temperature and RDHx pressure, especially in the later parts of the two square peaks: CRAC supply temperature increases between [1026 s-1171 s] and between [2194 s-2342 s]; RDHx pressure decreases between [1026 s-1099 s] and between [2121 s-2194 s]. These changes are surprising because one would expect cooling set-points to remain flat without any change in the CPU utilization. However, these anomalous behaviors can be explained by the coordinated nature of the dynamic cooling condition with IT load: The cooling points determined in the previous time samples to these anomalous time ranges must have overprovisioned the cooling requirements and created some local cooling sources such as over-cooled server chassis body. These local cooling sources act as a thermal capacitance for CPU heat loads and modify cooling load for the dedicated hardware such as CRAC or HDHx. Due to its rapid fluctuations, Type-2 load profile is more sensitive to this thermal capacitance effect than its Type-1 counterpart.

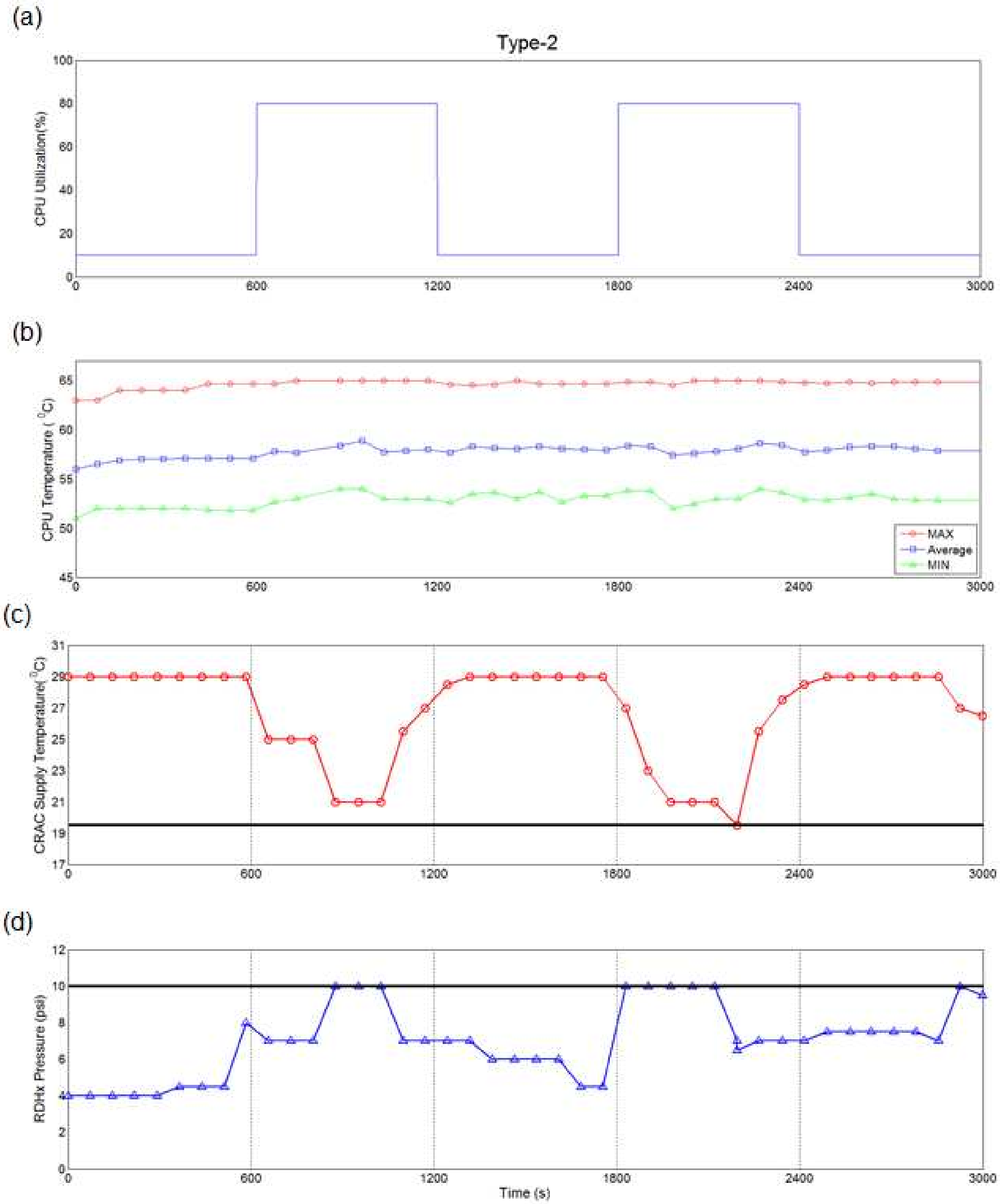


Figure 66: Optimal cooling resource allocation for Type-2 workload

The black lines in Figure 66(c) and Figure 66(d) show the most conservative set-points for CRAC (19.5 ^{0}C) and RDHx (10 psi), respectively. If there is no optimal control procedure, a risk-averse DC facility manager would operate his/her data center cooling at these conservative points. Therefore, the developed approach has the potential to save cooling energy. Figure 65 shows the cooling energy saving potential of the developed optimization framework. The energy calculations are done by the simple thermodynamic model developed in CHAPTER 1. Figure 67(a) shows the fraction of energy usage by the CRAC unit operating in the optimal mode to that by a similar CRAC unit operating in the conservative mode. On the other hand, Figure 67(b) shows the fraction of energy usage by the RDHx unit operating in the optimal mode to that by a similar RDHx unit operating in the conservative mode. The root mean square value of the fraction of energy saving in CRAC is equal to 62.7%. On the other hand, that value in RDHx is equal to 34.4%.

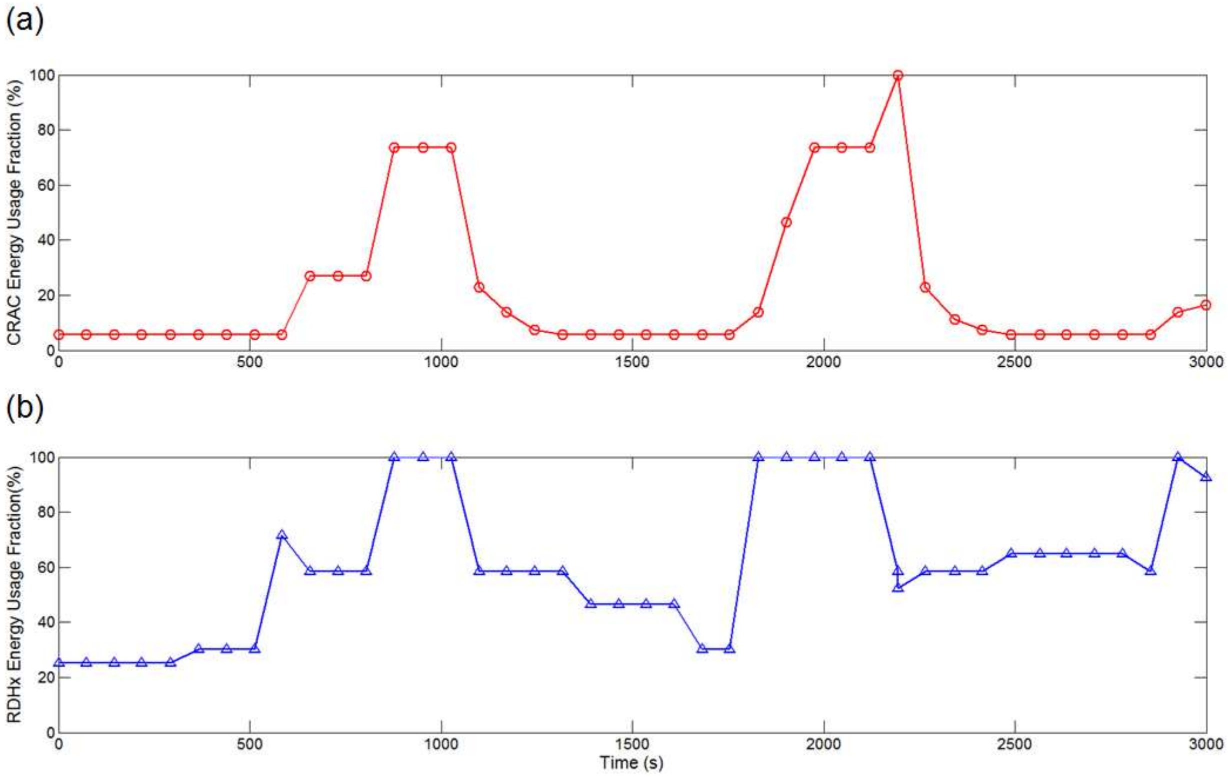


Figure 67: Cooling energy saving for Type-2 workload

Figure 68 shows optimal cooling resource allocation for Type-3 workload.

Figure 68(a) shows the load profile for Type-3 workload which is a combination of square waveform and sinusoidal waveform. The square waveform has amplitude of 25% and half-time period 600 s. In Type-3 workload profile, the square waveform lasts during [0-1800 s] with one square peak. The sine waveform lasts during [1800 s-3000 s]. It has amplitude of 25% with 3600 time period.

Figure 68(b) shows the maximum, average, and minimum CPU temperature profiles for the optimal cooling set-point envelope. The optimization procedure determines the most cost-effective cooling set-points under the given constraint.

Figure 68(c) and Figure 68(d) show optimal CRAC supply temperature and RDHx pressure set points, respectively. Initially, the CRAC supply temperature and RDHx pressure could satisfy the temperature constraint by operating at the most cost-effective set point of (29 ^{0}C, 4 psi). With increase in CPU utilization and associated CPU power dissipation, cooling set-points need to deviate from the cost-efficient operational mode. As expected, there are major changes in cooling set-points around the discontinuities of the step profile at 600 s, 1,200 s, and 1,800 s. In fact, the pattern of cooling set-point changes between [0-1800 s] remains similar to that of the Type-2. In this profile, however, the degree of changes is moderate because the amplitude of the square waveform is equal to 25% which is 64.3% lower than Type-2 profile. On the other hand, it is expected that the changes in cooling set-points will be moderate during the sine waveform in the [1800 s-3000 s] time domain. Indeed, that is reflected in the CRAC supply temperature set point, which remains flat at 29 ^{0}C. As far as RDHx pressure is concerned, it remains flat at 10 psi till 2129 s before dropping to 7 psi.

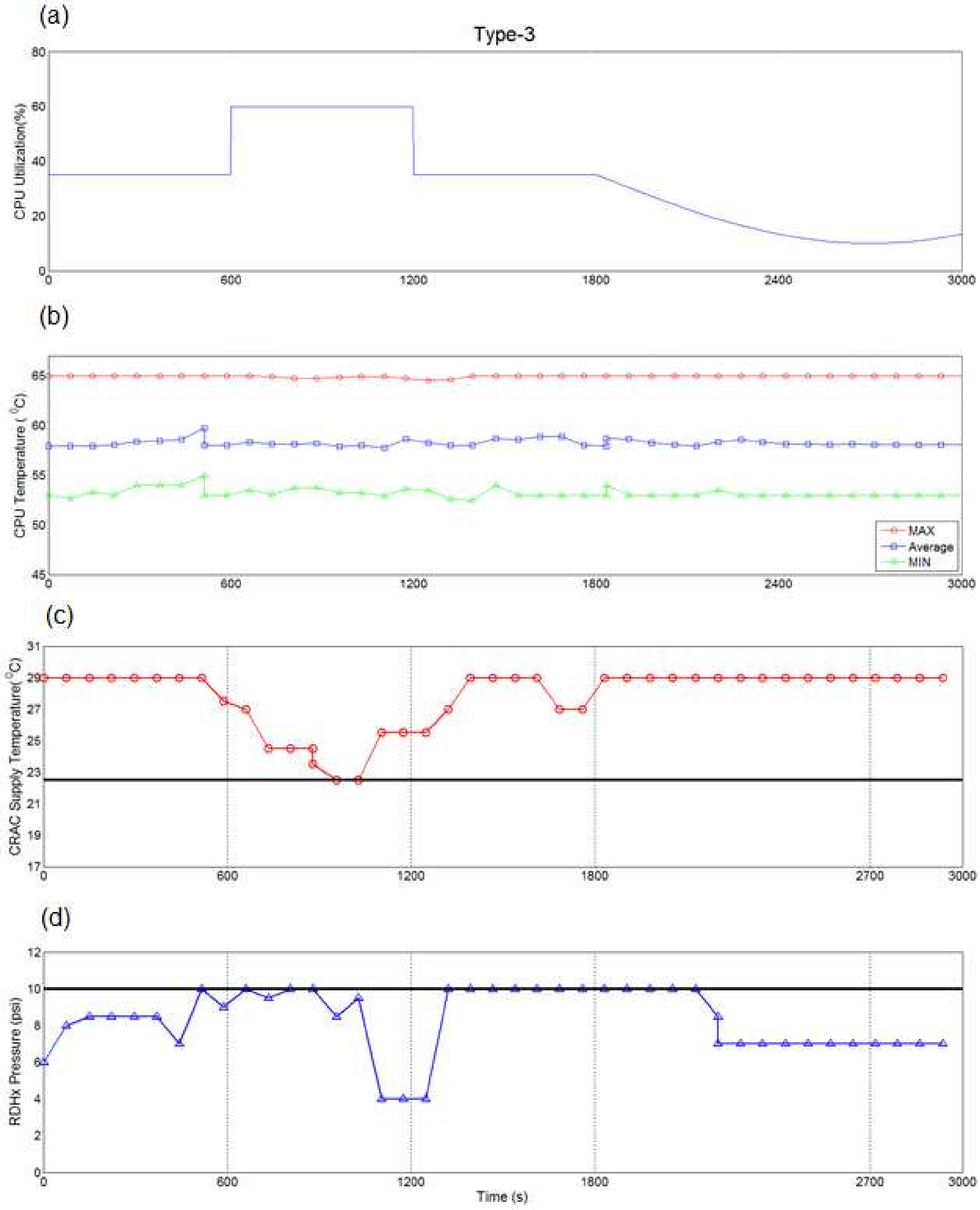


Figure 68: Optimal cooling resource allocation for Type-3 workload profile

The black lines in the Figure 68(c) and Figure 68(d) show the most conservative set-points for CRAC (22.5 ^{0}C) and RDHx (10 psi), respectively. If there is no optimal control procedure, a risk-averse DC facility manager would operate his/her data center cooling at these conservative points. Therefore, the developed approach has the potential to save cooling energy. Figure 69 shows the cooling energy saving potential of the developed optimization framework. The energy calculations are done by the simple thermodynamic model developed in CHAPTER **1**. Figure 69(a) shows the fraction of energy usage by the CRAC unit operating in the optimal mode to that by a similar CRAC unit operating in the conservative mode. On the other hand, Figure 69(b) shows the fraction of energy usage by the RDHx unit operating in the optimal mode to that by a similar RDHx unit operating in the conservative mode. The root mean square value of the fraction of energy saving in CRAC is equal to 66.2%. On the other hand, that value in RDHx is equal to 19%. It can be noted that the savings potential for the CRAC unit is significantly (more than 3 times) higher than that for the RDHx unit. This can be explained by the fact that the RDHx is closely coupled to the rack unit.

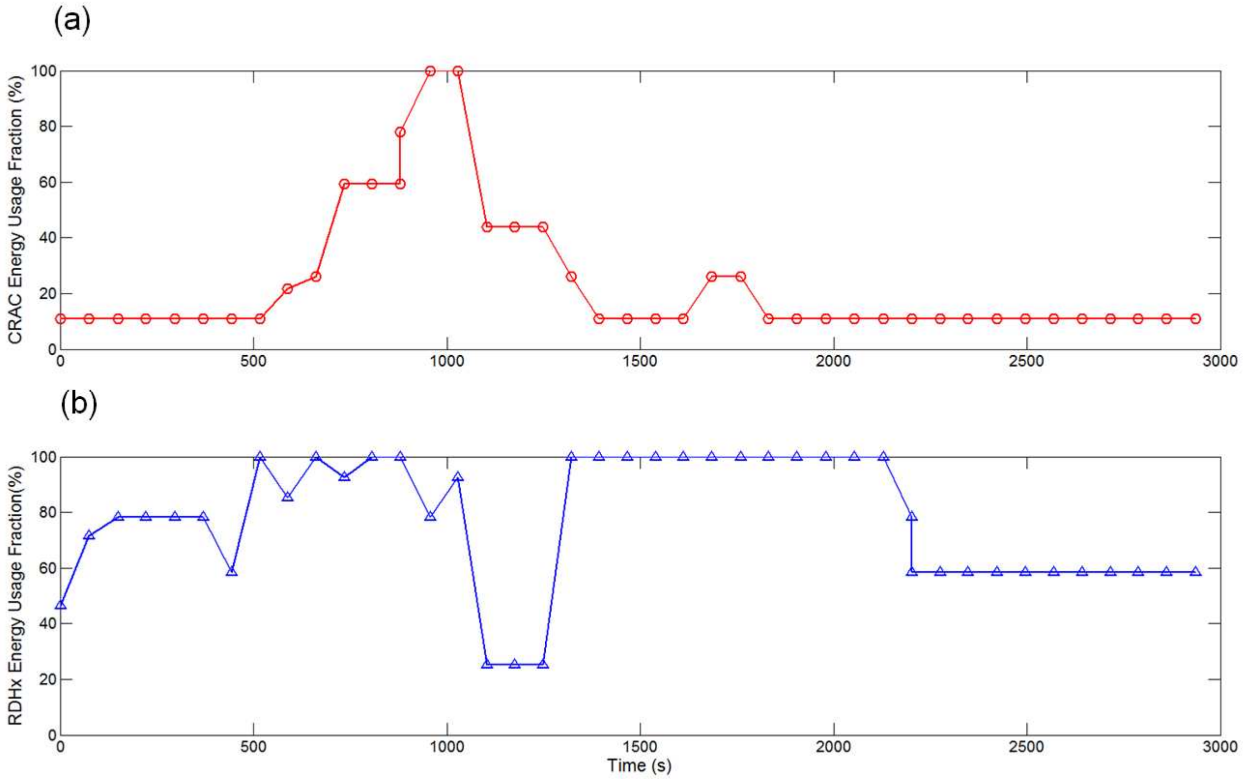


Figure 69: Cooling energy saving potential for Type-3 workload

Figure 70 shows the optimal cooling resource allocation for Type-4 workload. Figure 70(a) shows the load profile for Type-4. This profile is characterized by its sudden jump at 280 s. While this profile has (0.24 ± 0.0126) % CPU utilization before 280 s, it shoots up to (98.38 ± 1.14) % CPU utilization after 280 s. Figure 70(b) shows the maximum, average, and minimum CPU temperature profiles for the optimal cooling set-point envelope. The optimization procedure determines the most cost-effective cooling set-points under the given constraint. Figure 70(c) and Figure 70(d) show optimal CRAC supply temperature and RDHx pressure set points, respectively. Initially, the CRAC supply temperature and RDHx pressure could satisfy the temperature constraint by operating at the most cost-effective set point of (29 ^{0}C, 4 psi). With increase in CPU utilization and associated CPU power dissipation, cooling set-points need to deviate from the cost-efficient operational mode. As expected, there are major changes in cooling set-points around the discontinuities of the step profile at 280 s: while CRAC supply temperature increases from 29 ^{0}C to 20.5 ^{0}C, RDHx pressure increases from 4 psi to 9.5 psi. After 280 s, the cooling set-points encounter minor changes because the CPU utilization profile remains flat.

The black lines in Figure 70(c) and Figure 70(d) show the most conservative set-points for CRAC (18.5 ^{0}C) and RDHx (10 psi), respectively. If there is no optimal control procedure, a conservative DC facility manager would operate his/her data center cooling at these cost-efficient points. Therefore, the developed approach has the potential to save cooling energy. Figure 71 shows the cooling energy saving potential of the developed optimization framework. The energy calculations are done by the simple thermodynamic model developed in CHAPTER **1**. Figure 69(a) shows the fraction of energy usage by the CRAC unit operating in the optimal mode to that by a similar CRAC unit operating in the conservative mode. On the other hand, Figure 69(b) shows the fraction of energy usage by the RDHx unit operating in the optimal mode to that by a similar RDHx unit operating in the conservative mode. The root mean square value of the fraction of energy saving in CRAC is equal to 15.7%. On the other hand, that value in RDHx is equal to 10.5%. It can be observed that the CRAC energy saving for Type-4 workload is 4 times smaller compared to other profiles. This is due to the resource intensive nature of workload: CPUs are utilized at (98.38 ± 1.14) % load for 90.66% percentage of the time window.

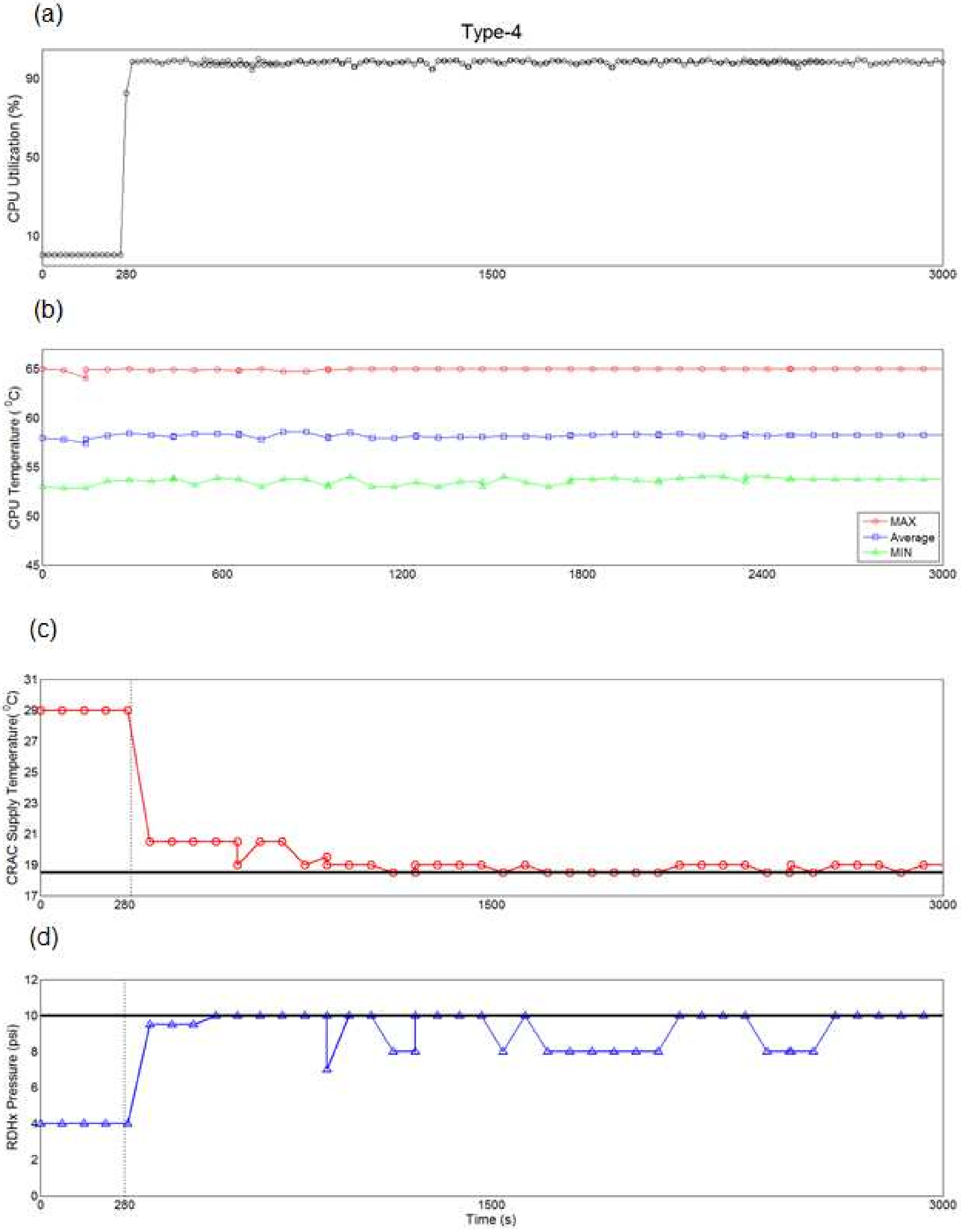


Figure 70: Optimal cooling resource allocation for Type-4 workload profile

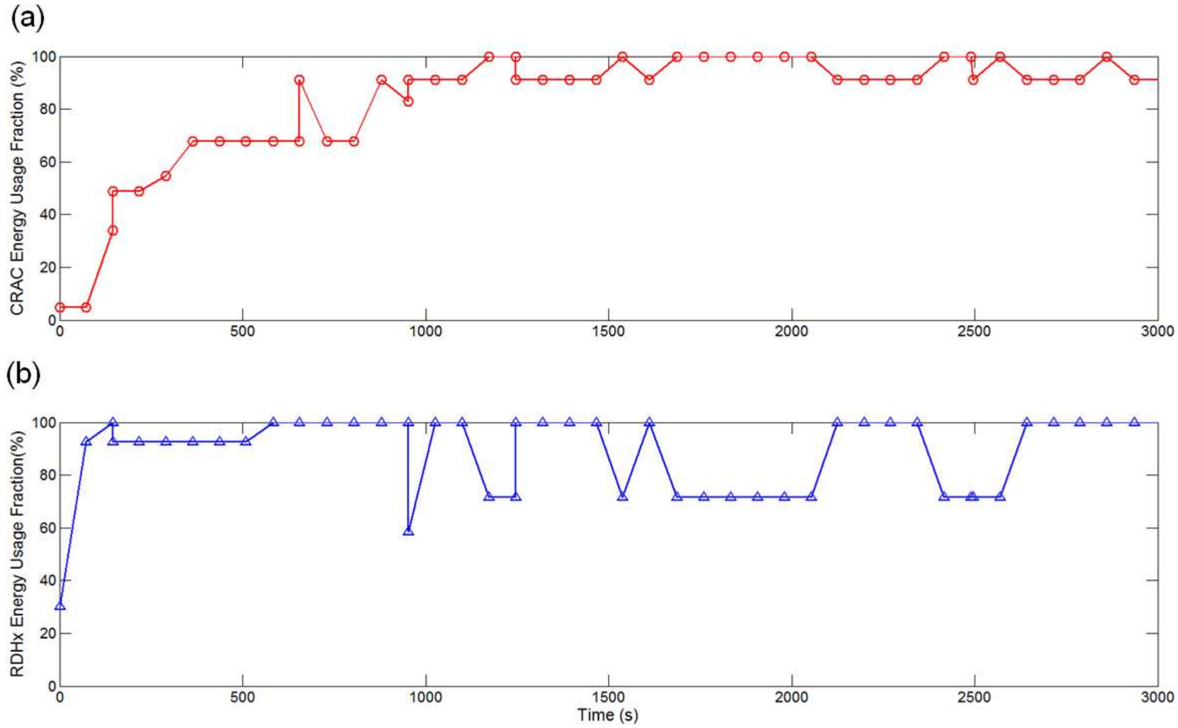


Figure 71: Cooling energy saving potential for type-4 workload

An interesting trend can be observed if the cooling power savings for different types of workloads are compiled, as done in Table 16. It can be seen that the cooling power savings are maximum for Type-2 workload with 62.7% average CRAC power savings and 34.4% average RDHx power savings. On the other hand, the cooling power savings are marginal for Type-4 workload with 13.9% average CRAC power savings and 10% average RDHx power savings. It can be inferred from this trend that savings are higher for the work-loads with higher discontinuities. Unlike Type-2 workload, Type-4 workload is very steady. Therefore, the controller does not have an opportunity to modulate CRAC supply temperature and RDHx pressure set-points. That amounts to workload-proportional cooling resource allocation which enables activity-based costing for data center cooling.

Table 16: Workload dependent cooling power saving

Workload	CRAC Power Savings	RDHx Power Savings
Type-1	51.4%	18.5%
Type-2	62.7%	34.4%
Type-3	66.2%	19%
Type-4	13.9%	10%

The proposed algorithm demonstrates high-fidelity prediction for the static workload profile. However, data center workload is stochastic in nature. Therefore, it is worthwhile to assess if the proposed algorithm can take care of uncertainty in the workload profile. In that endeavor, it is hypothesized that the POD-based analyses of the CPU temperature data generated from Type-2 workload can predict CPU temperature data of generated from the distorted Type-2 workload profile, as shown in Figure 59. The prediction fidelity is estimated for three validation points: (19 ^{0}C, 8.5 psi), (23 ^{0}C, 6.0 psi), and (27 ^{0}C, 5.5 psi). Figure 72 shows the accuracy benchmarking for temperature prediction for distorted Type-2 workload profile. It can be observed maximum percentage error is 14.5%. Also, it can be observed that the framework is particularly error-prone at the points of discontinuities (600 s, 1200 s, 1800 s, 2400 s).

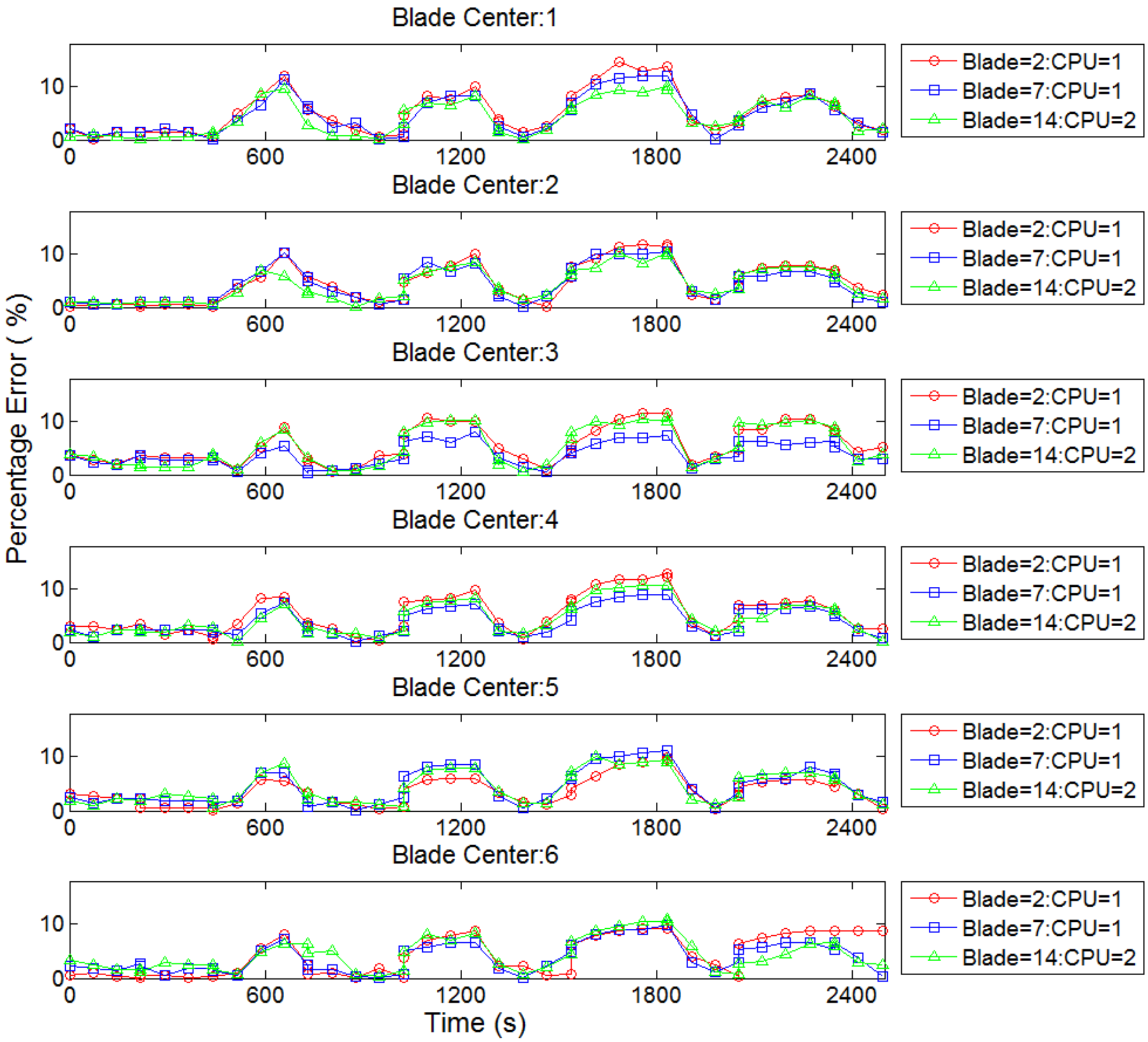


Figure 72: Accuracy benchmarking for distorted Type-2 workload at (27 ^{0}C, 5.5 psi) operating point

The root mean square (RMS) value of the time-average percentage error is equal to 4.49%. Similar analyses are conducted for cooling points (23 ^{0}C, 6.0 psi) and (19 ^{0}C, 8.5 psi).

Table 17 compiles the prediction uncertainty for different cooling set-points. It includes maximum error, RMS error, mean error, standard deviation of error, and percentage of predictions with more than 10% error.

Table 17: Prediction uncertainty for distorted Type-2 profile

Cooling Set-Point	Maximum Error	RMS Error	Mean Error	Standard Deviation of Error	Percentage of Predictions with more than 10% error
(19 ^{0}C, 8.5 psi)	19.31%	5.22%	5.15%	0.61%	12.75%
(23 ^{0}C, 6.0 psi)	14.28%	4.84%	4.82%	0.39%	10.98%
(27 ^{0}C, 5.5 psi)	14.5%	4.49%	4.46%	0.47%	6.56%

Table 18 suggests maximum prediction uncertainty for this framework is 6.98%.

Table 18: Upper limit of 99.7% (3σ) confidence interval (CI)

Cooling Set-Point	Upper-limit of 99.7% CI
(19 ^{0}C, 8.5 psi)	6.98%
(23 ^{0}C, 6.0 psi)	6.01%
(27 ^{0}C, 5.5 psi)	5.87%

It is conceded that the present version of the proposed framework can handle only relatively smoother variations in workload profiles. It is noted that the percentage errors are rapidly shooting up at the points of discontinuities. Therefore, it seems the proposed POD-based modeling framework would be of low-fidelity in case the work profile varies rapidly. To overcome that limitation, an additional parameter representing the workload variation intensity needs to be included in the POD-based formulation. This workload variation intensity would affect the heat dissipation from the computing chip and affect the CPU temperatures. Additionally, dynamic CPU temperatures would be affected by the computing chip thermal mass. The fluctuating nature of a particular workload profile can be modeled by the average time differential, θ of the workload, For a dynamic workload, $W(t)$, this factor can be defined as:

$$\theta = \frac{\int_0^t \frac{dW(t)}{dt} dt}{\int_0^t dt}. \tag{6.1}$$

Ultimately, this factor θ would affect the volumetric heat generation in the computing chip. In turn, that will affect heat dissipation from the chip and CPU temperatures. The CPU temperature can be modeled as a thermodynamic process variable.

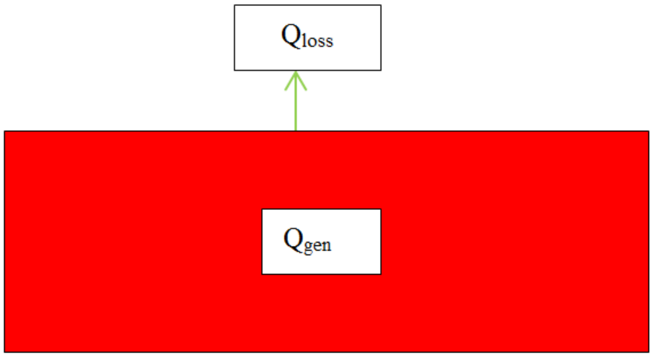


Figure 73: Thermodynamic model for CPU temperature evolution

Figure 73 shows the control volume for modeling chip heat transfer processes. The conservation of energy for this control volume yields:

$$\left.\begin{array}{c} \rho c V \dfrac{dT}{dt} = \dot{Q}_{gen} - \dot{Q}_{loss}. \\ \rho c V : \text{CPU thermal mass.} \\ T: \text{ CPU temperature} \\ t: \text{ Time.} \\ \dot{Q}_{gen} \propto \theta: \text{ The heat generation is proportioanl to } \theta. \\ \dot{Q}_{loss} = UA(T - T_{\text{sur}}): \text{ Dependent upon the cooling environment.} \end{array}\right\} \tag{6.2}$$

As discussed in CHAPTER 2, CPU temperatures can be expressed in POD modal space as follows:

$$\left.\begin{array}{r} T = T_0(In) + \psi(In) \otimes b\left(D, \theta\right). \\ In:\ \text{Independent Variables} \\ D:\ \text{Dependent Parameters other than } \theta \end{array}\right\} \tag{6.3}$$

The CPU temperatures expressed in POD modal space can be plugged back into Eq. (6.2) and POD coefficient $b(D, \theta)$ can be determined by solving Eq. (6.2) numerically.

Closure

A measurement-based framework is developed with CPU temperature as response variable and cooling set-points as the parameters. It is demonstrated that the framework is capable of generating CPU temperature data within 7% prediction uncertainty. Together with logarithmic time computational efficiency and accuracy, the framework is a useful state-space generator for designing optimal cooling control with respect to time-varying IT workload profile.

CHAPTER 7
CONCLUSIONS

A POD-based spectral algorithm is developed that can rapidly process parametric experimental data and generate new temperature predictions. The data-driven nature of the algorithm is particularly suitable for constructing a measurement-based framework. The most important feature of the proposed algorithm is its logarithmic-time computational efficiency, which makes the response for the framework near-real-time. The near-real-time predictive capability of the algorithm is utilized to solve four important problems related to a dynamic data center.

- The first problem is related to near-real-time temperature prognostic at a rack server inlet inside an air-cool data center. The framework improves the temporal resolution of measured temperature data. In fact, it allows reduction of sampling frequency by 90%. Therefore, it reduces temperature data acquisition cost. Additionally, an a priori semi-analytical error estimation framework has been developed. This error estimation framework makes the framework particularly suitable for robust cooling resource allocation controller based on server inlet temperatures.
- The second problem is related to minimizing temperature data acquisition cost in a data center. The proposed framework uses spatial locations as parameters. Therefore, it improves the spatial resolution of measured temperature data with 3% predictive uncertainty. In fact, it reduces sensor requisition at rack exhaust (with temperature data standard deviation of ~3^0C) by 33%.

- The third problem is pertaining to life-cycle design of data center cooling system. The proposed framework is a multi-parameter framework with time and rack heat load as parameters. The framework improves parametric resolution of measured air temperature data at the test rack. A full-factorial air temperature design has been carried out. It is particularly suitable for life-cycle design in the sense that different parametric zones take care of different operational domains such as normal mode, failure mode, critical mode, and retrofit mode.
- The final problem solved is related to determining optimal cooling set-points in response to time-evolving IT workload in a data center. The problem is solved using a multi-parameter framework with cooling resource set-points such as CRAC supply air temperature and RDHx pressure difference as set-points. The model improves the parametric granularity of CPU temperatures. Given that the underlying algorithm to the framework is near-real-time, the framework can dynamically check whether CPU temperatures are violating the constraint of the design optimization problem. In this case, it is the server reliability limit of 65 ^{0}C. The iterative computing starts at the most cost-efficient cooling set-points and continues until the constraint is satisfies. Since cooling resource allocation set-point is near-real-time, the optimal cooling resource set-points can be updated with the workload inside a cloud computing data center. This framework has the potential to promote workload-based cooling cost in data centers.

The high-level goal of this dissertation is to save dynamic cooling energy usage in a data center. Four different technologies have been developed toward that goal. Integrated with a data archive system such as OSISoft PI system, these technologies facilitate optimal cooling resource allocation controller in a data center. Figure 74 shows the schematic of the control loop to determine optimal cooling set-point.

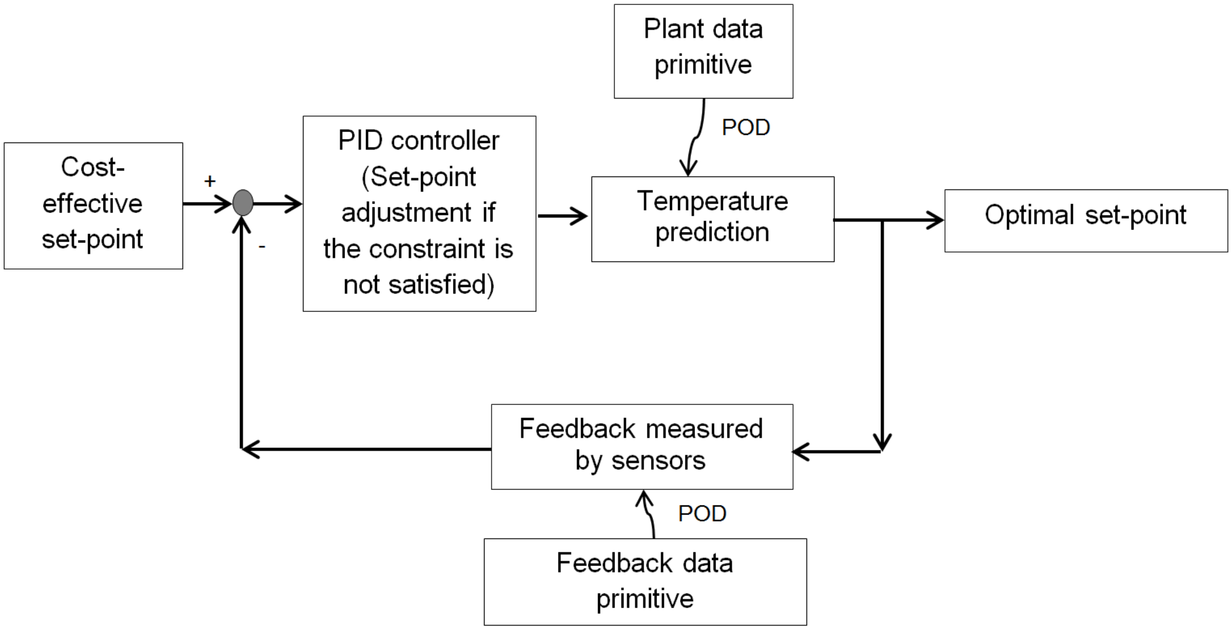


Figure 74: Real-time control loop for optimal cooling resource set-point determination by POD-based algorithm

The controller can be based on CPU temperatures, server inlet air temperatures or server exhaust air temperatures. If the controller is based on CPU temperatures, then the POD-based framework with CPU temperatures as the response variable (as developed in

CHAPTER 6) should be used as a state-space generator. On the other hand, if the controller is based on server inlet/ exhaust air temperatures, POD-based framework with server inlet/ exhaust air temperatures as the response variables (as developed in

CHAPTER 5) should be used. The feedback temperature measurement is facilitated by POD algorithm that uses time and spatial locations as parameters. A parametric framework in time improves temporal resolution of measured data, thereby it facilitates down-sampling of transient temperature measurements. That means data acquisition in the feedback system can be done by low-grade inexpensive thermal sensors. On the other hand, a data-driven framework with spatial locations as parameters improves spatial resolution of measured temperature data. Thereby, it reduces sensor requisition for feedback temperature data acquisition. Finally, CHAPTER 3 develops a priori error estimation framework which can be integrated with the POD algorithm to increase its fidelity.

REFERENCES

[1] J. Koomey, "Growth in data center electricity use 2005 to 2010," *Oakland, CA: Analytics Press. August*, vol. 1, p. 2010, 2011.

[2] L. A. Barroso, "The price of performance," *Queue,* vol. 3, pp. 48-53, 2005.

[3] R. Brown, "Report to congress on server and data center energy efficiency: Public law 109-431," 2008.

[4] S. Greenberg, E. Mills, B. Tschudi, P. Rumsey, and B. Myatt, "Best practices for data centers: lessons learned from benchmarking 22 data centers," *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings in Asilomar, CA. ACEEE, August,* vol. 3, pp. 76-87, 2006.

[5] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News,* vol. 35, pp. 13-23, 2007.

[6] R. Schmidt and M. Iyengar, "Thermodynamics of information technology data centers," *IBM Journal of Research and Development,* vol. 53, pp. 9: 1-9: 15, 2009.

[7] M. R. Stout Jr and J. W. Leach, "Cooling tower fan control for energy efficiency," *Energy engineering,* vol. 99, pp. 7-31, 2002.

[8] M. K. Patterson, "The effect of data center temperature on energy efficiency," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, 2008, pp. 1167-1174.

[9] R. Schmidt, R. C. Chu, M. Ellsworth, M. Iyengar, D. Porter, V. Kamath*, et al.*, "Maintaining datacom rack inlet air temperatures with water cooled heat exchanger," 2005.

[10] J. Rambo and Y. Joshi, "Modeling of data center airflow and heat transfer: State of the art and future trends," *Distributed and Parallel Databases,* vol. 21, pp. 193-225, 2007.

[11] S. Kang, R. R. Schmidt, K. M. Kelkar, A. Radmehr, and S. V. Patankar, "A methodology for the design of perforated tiles in raised floor data centers using computational flow analysis," *Components and Packaging Technologies, IEEE Transactions on,* vol. 24, pp. 177-183, 2001.

[12] J. W. VanGilder and R. R. Schmidt, "Airflow Uniformity through perforated tiles in a raised-floor Data Center," in *ASME Interpack*, 2005.

[13] K. C. Karki, A. Radmehr, and S. V. Patankar, "Use of computational fluid dynamics for calculating flow rates through perforated tiles in raised-floor data centers," *HVAC&R Research,* vol. 9, pp. 153-166, 2003.

[14] R. Schmidt, K. Karki, and S. Patankar, "Raised-floor data center: perforated tile flow rates for various tile layouts," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. ITHERM'04. The Ninth Intersociety Conference on*, 2004, pp. 571-578.

[15] R. K. Sharma, C. E. Bash, and C. D. Patel, "Dimensionless parameters for evaluation of thermal design and performance of large-scale data centers," in *8th ASME/AIAA Joint Thermophysics and Heat Transfer Conference*, 2002, pp. 1-1.

[16] C. D. Patel, R. Sharma, C. E. Bash, and A. Beitelmal, "Thermal considerations in cooling large scale high compute density data centers," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2002. ITHERM 2002. The Eighth Intersociety Conference on*, 2002, pp. 767-776.

[17] S. Bhopte, D. Agonafer, R. Schmidt, and B. Sammakia, "Optimization of data center room layout to minimize rack inlet air temperature," *Journal of electronic packaging,* vol. 128, pp. 380-387, 2006.

[18] M. Iyengar, R. Schmidt, A. Sharma, G. McVicker, S. Shrivastava, S. Sri-Jayantha*, et al.*, "Thermal characterization of non-raised floor air cooled data centers using numerical modeling," 2005.

[19] R. Schmidt and E. Cruz, "Cluster of high powered racks within a raised floor computer data center: effect of perforated tile flow distribution on rack inlet air temperatures," 2003.

[20] R. R. Schmidt, K. C. Karki, K. M. Kelkar, A. Radmehr, and S. V. Patankar, "Measurements and predictions of the flow distribution through perforated tiles in raised floor data centers," in *Proc of pacific Rim/ASME international electronic packaging technical conference of (IPACK), Kauai, HI,* 2001.

[21] C. D. Patel, C. E. Bash, C. Belady, L. Stahl, and D. Sullivan, "Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications," in *Proceedings of IPACK*, 2001, pp. 8-13.

[22] R. Schmidt and E. Cruz, "Raised floor computer data center: effect on rack inlet temperatures of chilled air exiting both the hot and cold aisles," in *Thermal and*

Thermomechanical Phenomena in Electronic Systems, 2002. ITHERM 2002. The Eighth Intersociety Conference on, 2002, pp. 580-594.

[23] J. D. Rambo and Y. K. Joshi, "Physical models in data center airflow simulations," 2003.

[24] D. Wang, "A passive solution to a difficult data center environmental problem," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. ITHERM'04. The Ninth Intersociety Conference on*, 2004, pp. 586-592.

[25] S. Shrivastava, B. Sammakia, R. Schmidt, and M. Iyengar, "Comparative analysis of different data center airflow management configurations," 2005.

[26] M. K. Herrlin and C. Belady, "Gravity-assisted air mixing in data centers and how it affects the rack cooling effectiveness," in *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. ITHERM'06. The Tenth Intersociety Conference on*, 2006, pp. 5 pp.-438.

[27] S. Bhopte, B. Sammakia, R. Schmidt, M. K. Iyengar, and D. Agonafer, "Effect of under floor blockages on data center performance," in *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. ITHERM'06. The Tenth Intersociety Conference on*, 2006, pp. 426-433.

[28] R. Schmidt and M. Iyengar, "Comparison between underfloor supply and overhead supply ventilation designs for data center high-density clusters," *ASHRAE Transactions*, vol. 113, 2007.

[29] S. McAllister, V. P. Carey, A. Shah, C. Bash, and C. Patel, "Strategies for effective use of exergy-based modeling of data center thermal management systems," *Microelectronics Journal*, vol. 39, pp. 1023-1029, 2008.

[30] N. Rolander, J. Rambo, Y. Joshi, J. K. Allen, and F. Mistree, "An approach to robust design of turbulent convective systems," *Journal of Mechanical Design,* vol. 128, p. 844, 2006.

[31] P. Druault, P. Guibert, and F. Alizon, "Use of proper orthogonal decomposition for time interpolation from PIV data," *Experiments in fluids,* vol. 39, pp. 1009-1023, 2005.

[32] I. Aleksander and H. Morton, *An introduction to neural computing* vol. 240: Chapman and Hall London, 1990.

[33] J. Yen and L. Wang, "An SVD-based fuzzy model reduction strategy," in *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, 1996, pp. 835-841.

[34] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM review,* vol. 45, pp. 385-482, 2003.

[35] S. K. Shrivastava, J. W. VanGilder, and B. G. Sammakia, "Data center cooling prediction using artificial neural network," in *ASME Conference Proceedings*, 2007, pp. 765-771.

[36] S. K. Shrivastava, J. W. VanGilder, and B. G. Sammakia, "Optimization of cluster cooling performance for data centers," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, 2008, pp. 1161-1166.

[37] J. W. Vangilder and S. K. Shrivastava, "Real-time prediction of rack-cooling performance," *ASHRAE transactions,* pp. 151-162, 2006.

[38] Z. Song, B. T. Murray, and B. Sammakia, "Numerical investigation of inter-zonal boundary conditions for data center thermal analysis," *International Journal of Heat and Mass Transfer,* vol. 68, pp. 649-658, 2014.

[39] J. W. VanGilder, X. Zhang, and S. K. Shrivastava, "Partially Decoupled Aisle Method for Estimating Rack-Cooling Performance in Near-Real Time," 2007.

[40] J. Xu, M. Zhao, J. Fortes, R. Carpenter, and M. Yousif, "On the use of fuzzy modeling in virtualized data center management," in *Autonomic Computing, 2007. ICAC'07. Fourth International Conference on*, 2007, pp. 25-25.

[41] J. Xu, M. Zhao, J. Fortes, R. Carpenter, and M. Yousif, "Autonomic resource management in virtualized data centers using fuzzy logic-based approaches," *Cluster Computing,* vol. 11, pp. 213-227, 2008.

[42] G. Li, M. Li, S. Azarm, J. Rambo, and Y. Joshi, "Optimizing thermal design of data center cabinets with a new multi-objective genetic algorithm," *Distributed and Parallel Databases,* vol. 21, pp. 167-192, 2007.

[43] J. D. Rambo, "Reduced-order modeling of multiscale turbulent convection: application to data center thermal management," Citeseer, 2006.

[44] E. Samadiani, "Energy efficient thermal management of data centers via open multi-scale design," 2009.

[45] E. Samadiani, Y. Joshi, H. Hamann, M. K. Iyengar, S. Kamalsy, and J. Lacey, "Reduced order thermal modeling of data centers via distributed sensor data," *Journal of heat transfer,* vol. 134, 2012.

[46] N. Rolander, "An approach for the robust design of air cooled data center server cabinets," Georgia Institute of Technology, 2005.

[47] C. K. Chang and J. Huang, "Video surveillance for hazardous conditions using sensor networks," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, 2004, pp. 1008-1013.

[48] H. Moon and P. J. Phillips, "Computational and performance aspects of PCA-based face-recognition algorithms," *Perception-London,* vol. 30, pp. 303-322, 2001.

[49] M. Hubert and S. Engelen, "Robust PCA and classification in biosciences," *Bioinformatics,* vol. 20, pp. 1728-1736, 2004.

[50] J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric turbulence and radio wave propagation*, pp. 166-178, 1967.

[51] L. Sirovich, "Turbulence and the dynamics of coherent structures. I-Coherent structures. II-Symmetries and transformations. III-Dynamics and scaling," *Quarterly of applied mathematics,* vol. 45, pp. 561-571, 1987.

[52] D. Kosambi, "Statistics in function space," *J. Indian Math. Soc,* vol. 7, pp. 76-88, 1943.

[53] E. Samadiani and Y. Joshi, "Multi-parameter model reduction in multi-scale convective systems," *International Journal of Heat and Mass Transfer,* vol. 53, pp. 2193-2205, 2010.

[54] R. Ghosh and Y. Joshi, "Error estimation in POD-based dynamic reduced-order thermal modeling of data centers," *International Journal of Heat and Mass Transfer,* vol. 57, pp. 698-707, 2013.

[55] N. Cressie, "The origins of kriging," *Mathematical Geology,* vol. 22, pp. 239-252, 1990.

[56] S. Ravindran, "Error estimates for reduced order POD models of Navier-Stokes equations," 2008.

[57] V. Girault and P. A. Raviart, "Finite element methods for Navier-Stokes equations: theory and algorithms," *NASA STI/Recon Technical Report A,* vol. 87, p. 52227, 1986.

[58] R. Temam, *Navier-Stokes equations: theory and numerical analysis* vol. 2: Amer Mathematical Society, 2001.

[59] S. C. Chapra and R. Canale, *Numerical methods for engineers*: McGraw-Hill, Inc., 2005.

[60] Y. Joshi and P. Kumar, *Energy efficient thermal management of data centers*: Springer, 2012.

[61] T. L. Bergman, F. P. Incropera, A. S. Lavine, and D. P. DeWitt, *Fundamentals of heat and mass transfer*: John Wiley & Sons, 2011.

[62] R. J. Moffat, "Describing the uncertainties in experimental results," *Experimental thermal and fluid science,* vol. 1, pp. 3-17, 1988.

[63] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry*: Cambridge University Press, 1998.

[64] E. Samadiani and Y. Joshi, "Proper orthogonal decomposition for reduced order thermal modeling of air cooled data centers," *Journal of heat transfer,* vol. 132, 2010.

[65] N. Dyn, D. Levin, and S. Rippa, "Data dependent triangulations for piecewise linear interpolation," *IMA journal of numerical analysis,* vol. 10, pp. 137-154, 1990.

[66] G. M. Nelson, "Development of an Experimentally-Validated Compact Model of a Server Rack," 2007.

[67] G. Strang, *Introduction to linear algebra*: Wellesley Cambridge Pr, 2003.

[68] M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, 1998, pp. 1381-1384.

[69] M. K. Patterson and D. Fenwick, "The state of datacenter cooling," *Intel Corporation White Paper. Available at http://download. intel. com/technology/eep/data-center-efficiency/stateof-date-center-cooling. pdf,* 2008.

[70] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Parallel Programming,* vol. 9, pp. 219-242, 1980.

[71] H. MacMahon, K. Doi, S. Sanada, S. Montner, M. Giger, C. E. Metz, *et al.*, "Data compression: effect on diagnostic accuracy in digital chest radiography," *Radiology,* vol. 178, pp. 175-179, 1991.

[72] W. P. Turner IV, J. PE, P. Seader, and K. Brill, "Tier Classification Define Site Infrastructure Performance," *Uptime Institute,* p. 17, 2006.

[73] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications,* vol. 1, pp. 7-18, 2010.

[74] B. Krishnan, H. Amur, A. Gavrilovska, and K. Schwan, "VM power metering: feasibility and challenges," *ACM SIGMETRICS Performance Evaluation Review,* vol. 38, pp. 56-60, 2011.