Autonomous real-time Model Updating within Digital Twin Frameworks for Thermal systems

by

Braden Robert Priddy

Bachelor of Science
University of South Carolina 2022

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Mechanical Engineering

Molinaroli College of Engineering and Computing

University of South Carolina

2024

Accepted by:

Austin Downey, Director of Thesis

Kerry Sado, Reader

Ann Vail, Dean of the Graduate School

Prepared under ONR Contract # N00014-22-C-1003

Abstract

As engineering systems increase in scale and complexity in the era of the Fourth Industrial Revolution, data-driven solutions will become essential in enabling the next generation of these systems. One of the trending tools that can aid in this transition is digital twins. As physical systems degrade throughout their life cycles, their behavior also changes. Digital twins use data assimilation to continuously update virtual models to represent the current state of their physical counterparts. A reliable digital twin can be leveraged by a system operator to perform diagnostics, optimize, and tests without ever needing the physical system. However, implementing effective digital twins involves overcoming challenges such as ensuring model accuracy and minimizing latency between the physical system and its virtual representation. This work proposes an updating scheme that utilizes real-time sensor data and a particle swarm optimization algorithm to update model parameters for continuous virtual model calibration within a digital twin framework. The PSO algorithm iterates through different multi-physics model configurations to reduce the discrepancy between the physical and virtual spaces. All computations are performed on edge devices, aligning with real-time constraints for high-performance applications that require on-site data processing. To evaluate the performance of this methodology, it was implemented on two electro-thermal systems designed to emulate the power and energy systems of a naval ship. Results demonstrate that the updating scheme can effectively update a digital twin in a reasonable amount of time, guarantee a higher level of accuracy, and adapt to external changes in its physical counterpart. This work aims to provide a novel model updating scheme that operates within a digital

twin frame to boost system resilience and adaptability. This approach sets the stage for more robust and autonomous applications across various engineering fields as they evolve alongside emerging technological demands.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

As threats against Navy vessels increase in size and complexity, modernizing its fleet is imperative. To address this issue, the United States Navy recognized the predicament. It shifted its focus towards fully electric ships, which promise enhanced efficiency, increased sustainability, and versatility in combat scenarios. While this transition is promising, it presents significant engineering challenges requiring data-driven solutions to manage the ship's systems and subsystems. An example is the development and implementation of new high-power laser weapons and radar systems. These systems will produce a large amount of waste and, if not properly managed, can result in downtime. If a ship were to overheat while carrying out a mission, it could leave the vessel vulnerable to adversaries and jeopardize the safety of its crew. To enable these new naval systems, virtual models have emerged as pivotal tools for thermal management. Virtual models can enable naval operators to simulate system behavior under various conditions, facilitating system assessments and informed decision-making without needing the physical system. However, as a physical system undergoes degradation or is affected by operational actions during its lifecycle, its virtual representation will fail to capture the current state of the system. Due to this problem, an important question arises: how will the virtual representation be updated to reflect the changes in the physical system over its life cycle? A digital twin can help resolve these issues by utilizing sensor data from the physical system to update its virtual representation. While the concept of digital twins has been around since the early 2000s, they were not widely investigated until recently due to the hardware and software limitations of the time. A digital twin comprises two key elements: the virtual models and simulations that form a virtual representation and a feedback loop between that virtual representation and its physical counterpart. Since the development of digital twins is still in its infancy, there is no unified way to construct this feedback loop within a digital twin framework. The method proposed in this work leverages a population-based algorithm to continuously evaluate permutations of model parameters to link virtual and physical spaces together.

Particle Swarm Optimization (PSO) has gained significant attention for its effectiveness in real-time parameter estimation. PSO is a meta-heuristic algorithm that mimics the foraging behavior of birds or fish. This technique has shown considerable success in finding the global minimum of a given search space quickly and accurately, making it especially suitable for updating model parameters in real-time.

In this work, PSO is the foundation of a generic model updating scheme, providing real-time updating for virtual representatives of liquid-cooled thermal loops. The updating scheme first initializes particles on a search space with random positions. These positions represent different permutations of various model parameters. The particles will continuously iterate through model parameters to reduce the error between the sensor data of the physical system and the simulation data of its virtual representation. Once the optimal parameters are found, the model is deemed calibrated.

In the following sections of this paper, two investigations are performed to evaluate the robustness of the model updating scheme. Each investigation will outline the methodology employed in developing the updating scheme, present the investigated physical system, discuss the experimental validation results, and provide a conclusion. This work aims to contribute valuable knowledge to the field of digital twin technology by providing a generic model updating scheme, facilitating the advancement of complex and dynamic next-generation naval systems.

Chapter 2

Real-time thermal data assimilation for power electronics at the edge

2.1 Abstract

Physical systems are often far too complex for a virtual model to fully encapsulate the behavior of a system. Requiring the user to fine tune parameters of the virtual model over the course of a system's life cycle. Data assimilation addresses this problem by continuously updating the model with sensor data to construct a personalized model of the physical system. This personalized model is known as a digital twin. Digital twins of ship systems can provide insight into the future state of the ship to enable operators to make informed decisions to increase health of the ship. However, there are a few key challenges that need to be overcome while updating a model. The first problem is reducing latency between the physical system and the digital twin. While ensuring that the digital twin has enough time and data to update. The second problem is verifying the model is an accurate representation of the physical system. This paper proposes a methodology that uses real-time sensor data and a particle swarm optimization algorithm to update model parameters for an instrumented thermal loop developed as a stand-in for liquid-cooled power electronics. The swarm of particles represent different configurations of a multi-domain model that constitutes the digital twin of the thermal loop. All computations are done on the edge to emulate a real world system. Results demonstrate that the particle swarm algorithm can reliably update a digital twin of the thermal loop as external changes are made to the system

(radiator turned on and off) with a root mean squared error of under 0.35 °C over the whole system. All models are updated in real-time with a maximum compute time of 38.4 s; demonstrating the proposed methodologies applicability for real-time data assimilation within a digital-twin framework.

2.2 Introduction

Model-driven solutions play a critical role in the development of next generation autonomous and semi-autonomous naval systems. Digital twins are one tool investigated by researchers to link physical and virtual spaces [33]. By using knowledge of the physical system and processing real-time (online) sensor data, digital twins can increase the efficiency of a system while providing the operator/user an accurate representation of the system. One of the main benefits of having a digital twin is its look ahead capabilities [26]. With the ability to accurately predict the behavior of a system, a user can make informed changes to the system [41]. However, problems arise as the physical system ages or changes over its life cycle and the model is no longer an accurate representation of the system. To overcome this challenge, the model's parameters must be continuously updated to ensure that the model is an accurate representation of the physical system.

Real-time model updating is a cornerstone of digital twin technologies and has been demonstrated by various researchers [35]. For example, researchers have used the particle swarm optimization algorithm for quick and easy parameter estimation of a resistance capacitance (RC) thermal model. The results demonstrate the approximation of RC parameters by the particle swarm optimization method took only 1.8 s to 10 minutes, depending on the resistance capacitance configuration, and a model estimation error of +1.2 °C the junction temperature in the steady state. Additionally, the script was able to be used by staff with low technical qualification, allowing anyone to update the models thermal properties [6]. Other researchers have taken advantage of particle swarm optimization method and implemented the algorithm into real-time model updating. The particle swarm optimization algorithm has been implemented into a bridge's structural health monitoring system. The group created a finite element model of a large composite plate composed of several materials working together to support a bridge. They demonstrated that the particle swarm optimization can be used to update parameters in real time of the finite element model. By updating this model continuously, a digital twin of the composite plate structure was created. When the base finite element model was compared to the model tuned by particle swarm optimization, the error between the simulation and experimental data was reduced from 7.67% to 0.12%. The increase in model accuracy will aid in predicting the deterioration of the structure [34]. Moreover, the particle swarm optimization algorithm has a low computational cost, making it desirable when dealing with real-time constraints. This idea is explored further in research done on the system health monitoring on naval ship structures. Where a mixture of physics-based and data-driven models utilize the particle swarm optimization algorithm to identify the probability of failure in a cantilever beam. A variety of different particle swarm optimization hyper-parameters were explored to find the global minimum while also considering real-time constraints [30]. The development of real-time model updating will aid the management of next-generation structures and systems.

All the previously mentioned examples utilized particle swarm optimization as the parameter updating technique. Particle swarm optimization is a meta-heuristic algorithm that utilizes simple mathematical rules to minimize error of a given cost-function. In a study comparing the most popular meta-heuristic algorithms, differential evolution and particle swarm optimization had the lowest costs on 30 different benchmark tests. While differential evolution did get closer to the global minimum than particle swarm optimization, it did it much slower [1]. In the context of real-time model updating particle swarm optimization is the best option. For this reason it was chosen as the parameter updating algorithm in this work.

In this work, a digital twin of a liquid cooled thermal loop intended to mimic that used to cool a power converter in an electric ship application was developed. The digital twin was represented by a physics-based model being updated continuously in real-time on a edge computing device. Particle swarm optimization is used to select the candidate model parameters for testing. The physical system for this study is a recirculating liquid thermal loop cooling a simulated electrical load; where an air-cooled radiator serves as the final heat sink. The loop is instrumented with thermocouples, flow sensors, and pressure sensors for data collection. To initiate changes in the experiment, the radiator fan speed and the liquid flow valve were adjusted during runtime. The digital twin in this work consists of a physics-based model of the experiment that is updated in real-time using an error-minimization technique. All computations are done on an Windows edge-computing device with a Intel i5-7200U processor collecting data from the physical experiment in real-time.

While the collection of data happens continuously, the particle swarm algorithm updates the model using a sliding window. Here, the particle swarm optimization algorithm waits for a designated amount of time. The time-series data collected during this time is known as a window. The particles then attempt to fit the outputs of the physics-based model to the acquired window, until a new window is collected. The selection of an appropriate data window length is important.If the window is too small, the particles may not have sufficient time to find the global minimum in the search-space, thus producing a digital twin that never fully converges. Conversely, a large window will increase latency between the physical system and the digital twin. Rendering the model useless to the user, especially in highly dynamic systems. A key challenge in this work is ensuring that the updated model is accurate and represents the current behavior of the system (the thermal loop). If the model is not an accurate representation of the system, no optimizer will be able to fit a digital twin to the physical system.

The contributions of this work are two-fold, first a numerical approach for the updating of thermal models within a digital twin for power electronics is presented. Second, an experimental validation is carried out demonstrating that the proposed method can update a digital twin within a reasonable time period. By adjusting the model's parameters to minimize the RMSE error between experimental sensor data and the simulation data.

2.3 Materials and Methods

This section presents the verification of the model and methodology behind the particle swarm optimization algorithm.

2.3.1 Thermal Loop Experiment

The physical system modeled in this work is a thermal Loop. It is a simple thermal loop outfitted with a centrifugal pump, heating plate, air-cooling radiator and an expansion tank, with sensors placed throughout the experiment, shown in Figure 2.1. Each sensor is connected to a National Instruments data acquisition system (NI-

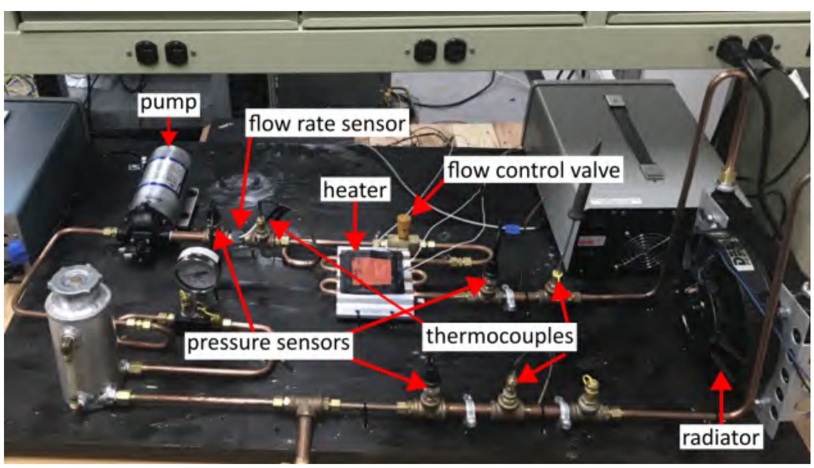

Figure 2.1 Labeled picture of tabletop thermal loop experiment.

DAQ) to acquire the temperature, pressure, and flow rate measurements. This NI-DAQ then transmits the data to the x64 edge-computing device running alongside the experiment. The particle swarm optimization algorithm running on the computer uses this data to update the digital twin. Figure 2.2 shows how the sensors are linked to the particle swarm optimization algorithm. The centrifugal pump pumps water throughout the copper piping. The temperature and flow rate at the exit of the pump is recorded by a thermocouple and a turbine flow sensor directly after the pump. Water then flows through a manually controlled valve and into the heating plate. The water absorbs heat from heating plate with a rise in liquid temperature. Thermocouples are placed on top of the heating plate, as well as in the copper pipe after the heating plate. The radiator is a fan that cools the water passing through the pipe. The heating plate, radiator and control valve are controllable parameters that are changed during the runtime. The final component of the thermal loop is the expansion tank, this ensures that the pressure of the loop is kept at the desired level for safe operation during the experiment.

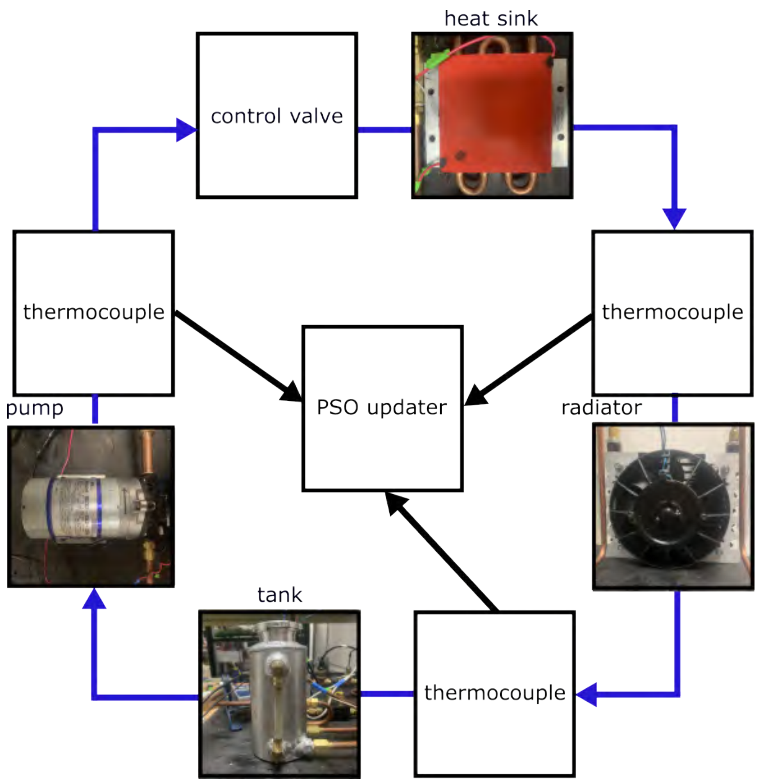

Figure 2.2 Diagram of the thermal loop experiment.

2.3.2 Model Verification

The physics-based model has numerous parameters that can significantly change the outcome of the model. There are two different methods through which these model parameters were determined. The first way is by solving for them directly. Parameters such as the cross-sectional area, mass, and length of the pipe can easily calculated, but some of the parameters cannot be easily solved for. The second way was using experimental data to characterize component thermal resistances, masses, convection coefficients, and other properties. Multiple experiments involving varying run times, valve openings, heat from the heating plate, and radiator fan speeds were performed. A grid search was conducted on each of these tests to find the optimal model parameters.

2.3.3 Particle Swarm Optimization Online Updating

Figure 2.3 illustrates the data flow of the proposed algorithm. The edge device acquires a window of sensor data and the particles' positions are initialized. The particles' positions represent the model's radiator fan speed and the valve opening. The window of time series data is not a sliding window. Instead, the window is a user defined amount of time. The data for each window never overlaps with the prior or succeeding window. Once the window is acquired by the edge device, the main loop of the algorithm begins. The newly acquired window is held and the model is updated by the particle swarm for the duration of that window. Meanwhile, new data is collected for the next window.

When a new window of sensor data becomes available, temperature data from the heating plate is obtained. After the model's heating plate temperature is set, the particles positions are updated for the duration of the window. First, a particle's position is configured to be the model's valve opening and radiator fan speed. Then the model is ran for a simulation time equal to the duration of the window. After the simulation, the acquired temperature data is evaluated by a cost function. The cost function calculates the root mean squared error (RMSE) between the simulation data and the acquired sensor data. The particle's cost is then used to calculate the velocity to update the particle's next position, as shown in equation 3.4.

$$X_i^{t+1} = V_i^{t+1} + X_i^t \tag{2.1}$$

If a particle has a high-cost relative to the other particles, its next velocity and position will change dramatically. On the contrary, if a particle's cost is low, its next velocity and position will remain largely unaffected. The velocity is influenced by three components, shown in equation 2.2. The first component is Inertia wV_i^t. The particle's velocity is inherited from the previous step and influences the particles next position. The second component is the cognitive component $r_1\phi_1(P_i - X_i^t)$.

The cognitive component is the difference between a particle's personal best position and its current position. This difference is multiplied by a user defined acceleration constant ϕ_1 and a uniformly random number r_1, between zero and one. The final component is the social component $r_2\phi_2(P_g - X_i^t)$. The social component finds the difference between the particle's current position and global best position. Then the difference is multiplied by a user defined acceleration constant ϕ_2 and a uniformly random number r_2, between zero and one [16].

$$V_i^{t+1} = wV_i^t + r_1\phi_1(P_i - X_i^t) + r_2\phi_2(P_g - X_i^t) \tag{2.2}$$

The particles' positions will continue to be updated until a new window of sensor data is available. Once a new window is available, the model with the lowest cost is returned to the user and the particles' global cost and personal costs are reset. A random number between [-1, 1] is added to each of the particles' positions, before a new window of sensor data is assessed. This is an essential step to continuously update the model of a dynamic physical system. When the physical system changes the optimal model parameters to reach a global minimum also change. Introducing another degree of randomness at the beginning of a new window will prevent the particles from getting trapped at the local/global minimum of a previous window. The calibrated digital twin can then be used to gain insight on the behavior of the system. The variables used in this work for the particle swarm optimization are:

- X_i^t: Position of particle i at time t
- V_i^t: Velocity of particle i at time t
- P_i: Personal best position of particle i at time t
- P_g: Global best position found by any particle in the swarm at time t
- w: Inertia weight, damping the impact of the previous velocity

- ϕ_1: Cognitive coefficient, controlling the influence of the personal best position
- ϕ_2: Social coefficient, controlling the influence of the global best position
- r_1, r_2: Random values in the range [0,1] used to introduce stochasticity in the velocity update equation

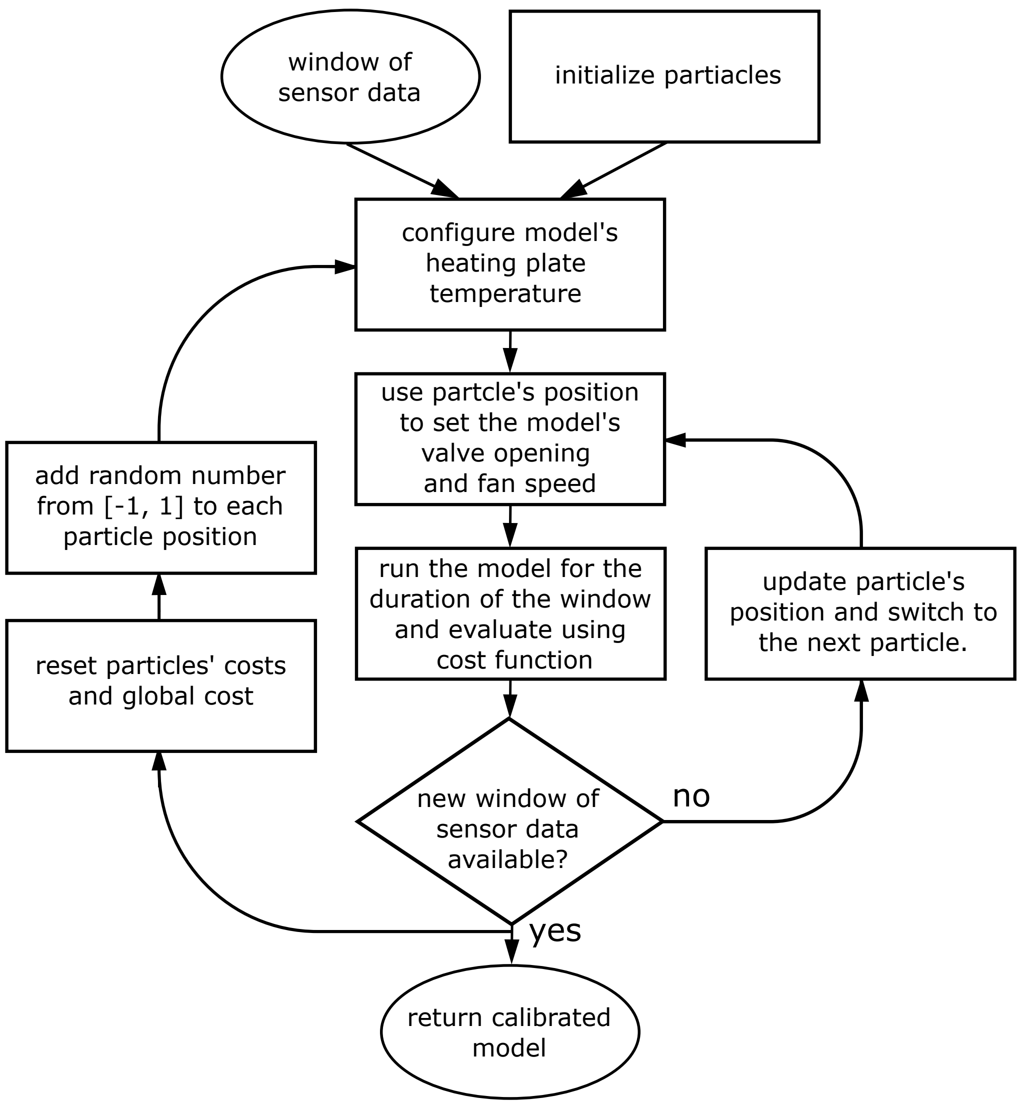

Figure 2.3 Particle swarm optimization digital twin calibration flow chart.

2.4 Results and Discussion

To test the ability of the particles to calibrate the digital twin to the thermal loop. The experiment was run alongside the digital twin for two and a half hours while changes to the experiment were made periodically. Before the edge device began sampling data, the pump was turned on and power was supplied to the heating plate. After a couple of seconds, the edge device began acquiring data. The experiment was ran uninterrupted without any changes for 30 minutes. Until the radiator was turned on and the water began to cool. After another 30 minutes, the temperature of the water converged to about 29°C. Then the radiator was turned off and the control valve was adjusted to be 50% open. Finally, the power to the heating plate was turned off and the radiator was turned back on until the temperature converged. The edge device stopped acquiring data and the experiment was turned off. Figure 2.4 shows the results from the particle swarm optimization algorithm running alongside the experiment for two and a half hours.

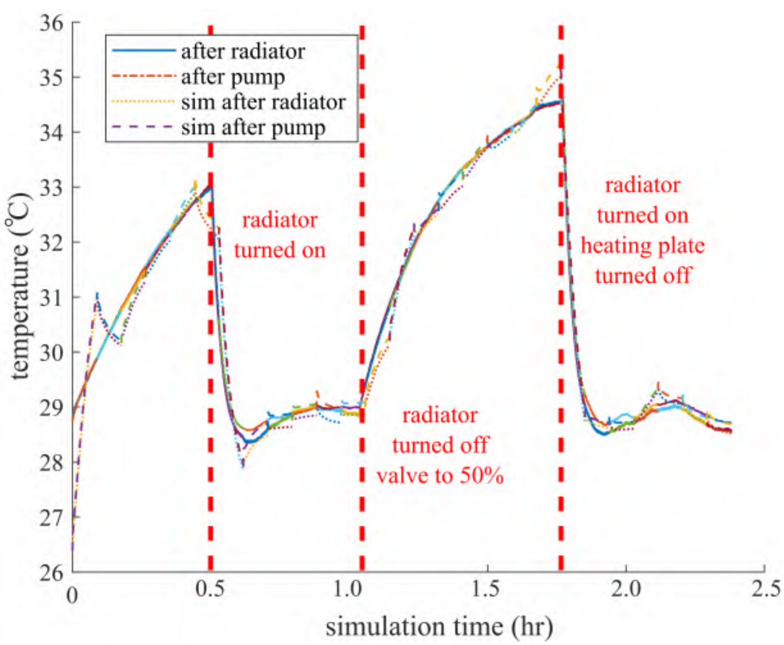

Figure 2.4 Particle swarm optimization updating a digital twin every 5 minutes using online temperature data.

The swarm consists of five particles updating on a five-minute window. Each particle updated an average of five times per window. Initially, the particles' representation of the thermal loop experiment results in an unfitted model. This is to be expected, as all the particles are initialized with random radiator fan speeds and valve openings. The starting temperature of the Simscape model is unknown and assumed to be ambient temperature. These initial guesses skew the starting parameters of the particles, producing an inaccurate model initially. However, the information gained from the previous window is then used to improve the initial guess on the next window. The recovery from bad guesses is shown in Figure 2.5.

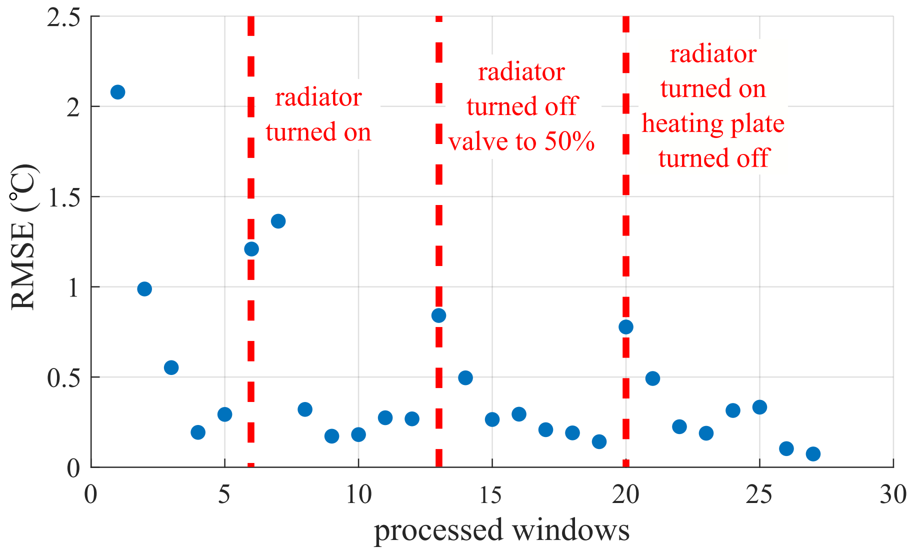

Figure 2.5 Particles positional improvement over time.

Table 2.1 Performance metrics of the concatenated windows.

thermocouple location	SNR (dB)	RMSE (°C)	MAE (°C)
after pump	39.60	0.323	-0.031
after radiator	39.07	0.342	0.070

The particles' parameters are also able to recover from changes during the thermal loop experiment. As the components are turned on and off, the forecast of the model changes. This results in more initial bad guesses, as the information gained from the last window skews the results of the latest window. Despite the changes throughout the experiment, the particles can recover in about two windows of data.

An investigation into the consistency of the simulation times was performed and reported on in Figure 2.6. For this test, 1000 simulations were run on the same five-minute window. The recorded results demonstrate an average runtime of 21.76 seconds and a standard deviation of 2.8 seconds. As expected from a windows OS the timing distribution is heavily skewed to the left with large outliers. The max simulation time was 38.4 seconds. The max time limiting factor for how many particles can be used, while ensuring each particles' parameters converge.

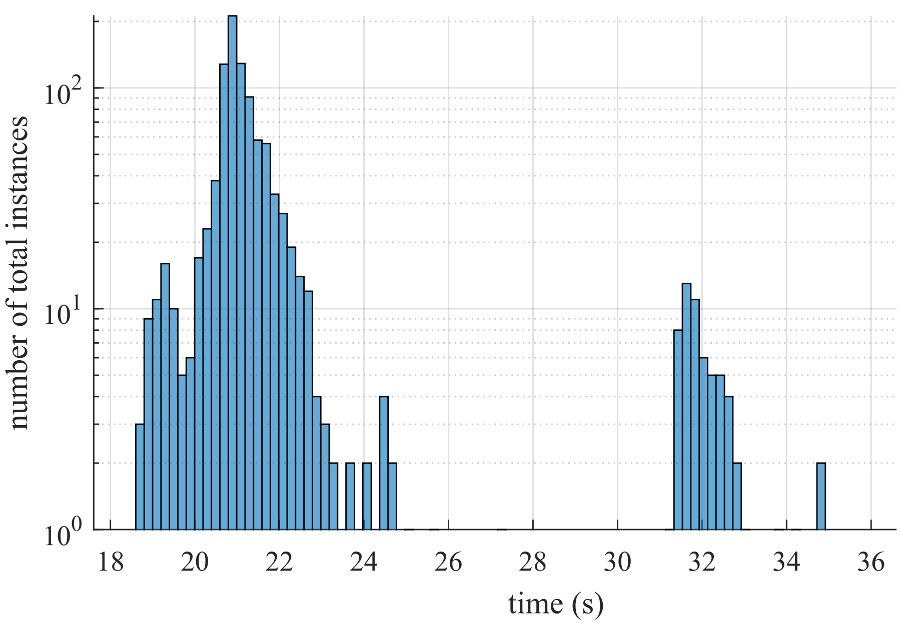

Figure 2.6 Logarithmic timing distribution of a 1000 simulations.

Table 2.2 1000 simulation timing report.

metrics	time (s)
mean	21.76
standard deviation	2.8
max time	38.4

2.5 Conclusion

As naval systems become increasingly complex, the need for data driven solution will become essential in maintaining the overall health of a ship. In this research, a thermal loop experiment was modeled in Simscape to act as a digital twin. The model was updated by using online sensor data to inform a particle swarm algorithm. Where five particles would find the optimal radiator fan speed and valve opening to fit the model to a window of temperature data. The results demonstrate the ability of the particle swarm to return an accurate representation of the experiment every five minutes in the form of a digital twin.

2.6 Acknowledgment

This work was supported by the Office of Naval Research under contract nos. N00014-20-C-1106, N00014-22-C-1003, and N00014-23-C-1012. The support of the ONR is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Navy.

Approved, DCN# 543-1868-24 DISTRIBUTION STATEMENT A on 05/30/2024. Approved for public release: distribution unlimited.

Chapter 3

Autonomous real-time model updating within digital twin frameworks for thermal systems

3.1 Abstract

The next generation of naval high-power weapons and systems will significantly strain current naval ship cooling systems. Throughout the lifecycle of a system this strain will alter its behavior and its virtual representation will become outdated as the system components age. Digital twins are a trending tool that can help alleviate this stress by providing ship operators with prognostication and strategic planning capabilities. To enable these complex power systems throughout their lifecycles, data-driven solutions for model updating will become essential. This paper investigates the application of a digital twin framework to enhance the performance of a multi-physics model. The digital twin framework comprises an updating scheme, a physical testbed designed to emulate the cooling system of a ship, and a multi-physics representation of that system. The updating scheme leverages a swarm of particles and online sensor data to evaluate permutations of model parameters to update the digital representation periodically. Two scenarios were applied to evaluate the performance of the digital twin framework. Results demonstrate that the digital twin framework can adapt to system changes in real-time and improve the accuracy of a static virtual model by more than 90%.

3.2 Introduction

At the beginning of the 21st century, the Navy expressed interest in switching its fleet to fully electric ships. Since then, threats against its fleet have only increased in scale and complexity [12]. Consequently, modernizing the power and energy systems on ships has become more imperative than ever to support new combat systems and high-power laser weapons [21]. New high-power weapons dissipate a substantial amount of heat, which, if not properly managed, could necessitate the shutdown of power and energy systems for indeterminate periods. Virtual models are tools for thermal management that operators use to simulate system behavior under various conditions without the need for the physical system. However, the virtual model of the system will no longer be accurate as the physical system changes whether due to degradation or operator actions. Digital twins (DT) can address this issue by assimilating real-time sensor data to continuously update a virtual model of the cooling system, linking virtual and physical spaces.

A digital twin is a collection of dynamic digital models that faithfully represent an existing physical system or subsystem, known as the Physical Twin (PT) [28]. A digital twin comprises two key elements: a virtual representation and a feedback loop between the virtual and physical counterparts[22]. The digital twin is continuously updated using real-time sensor data to accurately replicate and adapt to the behavior of the physical twin during its operational state [27]. While the concept of digital twins has been around since the early 2000s, their implementation was hindered by the technology of that time [37]. Recently, digital twins have gained popularity with the advent of the fourth industrial revolution, as evidenced by the increased number of publications and patents in recent years [32]. As engineering problems become more complex and improving the efficiency of existing systems becomes more challenging, digital twins have emerged as an attractive tool for performing functions such as prognostication, optimization, testing, and control [31]. For instance, a digital twin updated using fuzzy logic and operational data from a power plant has demonstrated the feasibility of creating an accurate virtual representation of the plant [4]. This approach allowed for safer training of new control room operators and studying different control schemes. Other digital twin applications include online parameter optimization to increase the efficiency of a system. The cooling water system of a district plant behavior was reproduced and helped in optimization efforts, achieving 2-3% in energy saving from the previous year [19]. Another example of increased system performance boosted by digital twins, is using online measurements from a heat pump to calibrate a model to reduce fouling and unplanned downtime [3]. Real-time health monitoring is also an area that significantly benefits from digital twin implementation. A digital twin of cooling fans achieved a 95% fault detection success rate, informing a user of a fault before it had occurred [23]. A complete digital twin implementation enables real-time monitoring, improves system reliability, enhances risk management, and increase system efficiency [2]. In naval applications, a digital twin can be coupled with other existing combat systems and electrical models to enhance the survivability of a ship. Digital twins allow operators to address dynamic changes in the system and support strategic decision-making. Most importantly, this tool can aid in the operational management of the next generation of naval electric ships and high-power weapons.

Since the development of digital twin frameworks is still in its infancy, there is no unified way to develop and deploy a digital twin. A study evaluating 50 publications on this subject failed to reach a consensus on a universal digital twin framework while stressing the importance of this tool and the need for a generic digital twin framework[32]. As a result, further exploration into this topic is necessary, as different physical systems may require different frameworks [33]. In the case of naval vessels, the framework should work in real-time, be computationally efficient, and be relatively accurate. When constructing a digital twin framework, an important question arises:

how will the virtual representation be updated to reflect changes in the physical system?

Population-based optimization algorithms have recently gained considerable traction for updating digital twin frameworks. Data-driven algorithms can learn from a large amount of data to model the complex behavior of a system that other techniques fail to capture [10]. These meta-heuristic algorithms seek an optimal solution to a cost function by utilizing a swarm of model instances to systematically explore a search-space [38]. Swarm algorithms have shown successful implementation in updating multi-physics model parameters for heat management, such as optimizing cooling strategies in a data center [42], enhancing prediction performance of CNCMT spindle thermal error [18], digital twin controller of HVAC systems [20]. A comprehensive review of the most popular swarm optimization algorithms was done by testing them on 30 different benchmark functions [1]. It was concluded that Differential Evolution (DE) performed the best, closely followed by Particle Swarm Optimization (PSO). However, DE was the second slowest algorithm. In an environment where computation cost and speed are necessary, PSO outperforms DE. This idea is further backed by a similar study directly comparing DE and PSO [25]. While PSO model updating for thermal virtual representations of power electronics has been discussed in previous publications [36], [17], [40]. These publications focus on the performance of the digital twin rather than the speed at which it is updated. In the context of updating a virtual representation of a cooling system to form a digital twin, speed is of the utmost importance [5]. For these reasons, PSO was chosen as the model updating method.

The contribution of this work is the development of a generic digital tuning framework of a physical asset within digital twin systems. The framework works to autotune representation parameters to adapt to changes in the physical system in realtime. Ensuring the digital twin can accurately reflect the dynamic behavior of the physical system, thereby improving model accuracy in real-time applications. To demonstrate the capabilities of this framework, it was deployed onto a thermal system representative of a ship cooling system. The remainder of this paper is organized as follows. Section 2 discusses the methodology behind the proposed DT framework, the cooling system, and modeling methods. The results and discussion are presented in Section 3. The final section provides a conclusion to this work.

3.3 Materials and Methods

This section provides an overview of the digital twin framework, the cooling system of the power & energy testbed, the faithful representation, data acquisition, and updating scheme that form the digital twin framework.

3.3.1 Digital Twin Framework

The digital twin framework aims to assimilate real-time sensor data to link a physical counterpart to its virtual representation. An overview of the digital twin framework is showcased in Figure 3.1. During the operation of a system, operators control various mechanical and electrical components to meet certain objectives. Throughout the lifecycle of the system, these actions will cause the system to degrade, forever changing its behavior. These changes are captured by sensors instrumented throughout the system, and data is relayed to the digital twin via data acquisition equipment. Running alongside the physical system, the digital twin utilizes the acquired data to tailor the virtual model to its physical counterpart. To accomplish this goal, a digital twin tuning using an updating technique within the digital twin autonomously tunes parameters in the faithful representation. The tuned faithful representation outputs simulation data and is evaluated against the real-world data. The process of assessing permutations of model parameters happens continuously until the optimal parameters are found. Once the optimum is found, the virtual representation is deemed calibrated and the projected digital twin response is returned to the operator. The operator can then leverage the digital twin to perform strategic planning, system optimization, or increase operational efficiency.

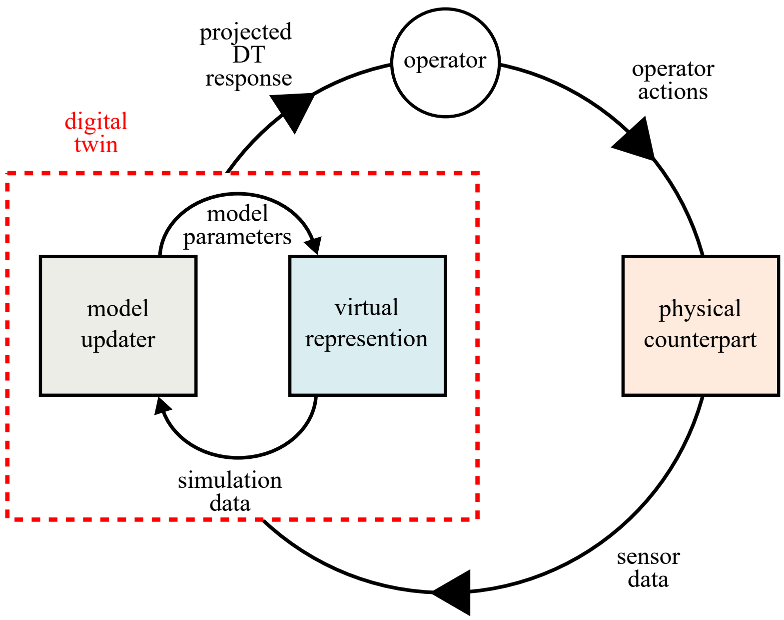

Figure 3.1 Diagram of the digital twin framework.

3.3.2 Testbed Configuration Overview

In this study, a notional shipboard power system is used as an example to validate and study the proposed digital twin framework. The power system is replicated by a testbed composed of six power electronic converters that interface with various loads as illustrated in the electrical setup diagram Figure 3.2. These converters are part of a microgrid designed to emulate the power and energy systems of a naval ship, effectively replicating the onboard power system [15]. Each module is designed to manage specific voltage inputs and outputs while supporting a load. The efficiency of these converters typically results in power losses ranging from approximately 100 to 150 W [14]. Typically, these power losses would handled by the cooling system. However, the power modules were not operational in this work and only act as heat sinks. Instead, the dissipated heat from the power modules was emulated by a water heater.

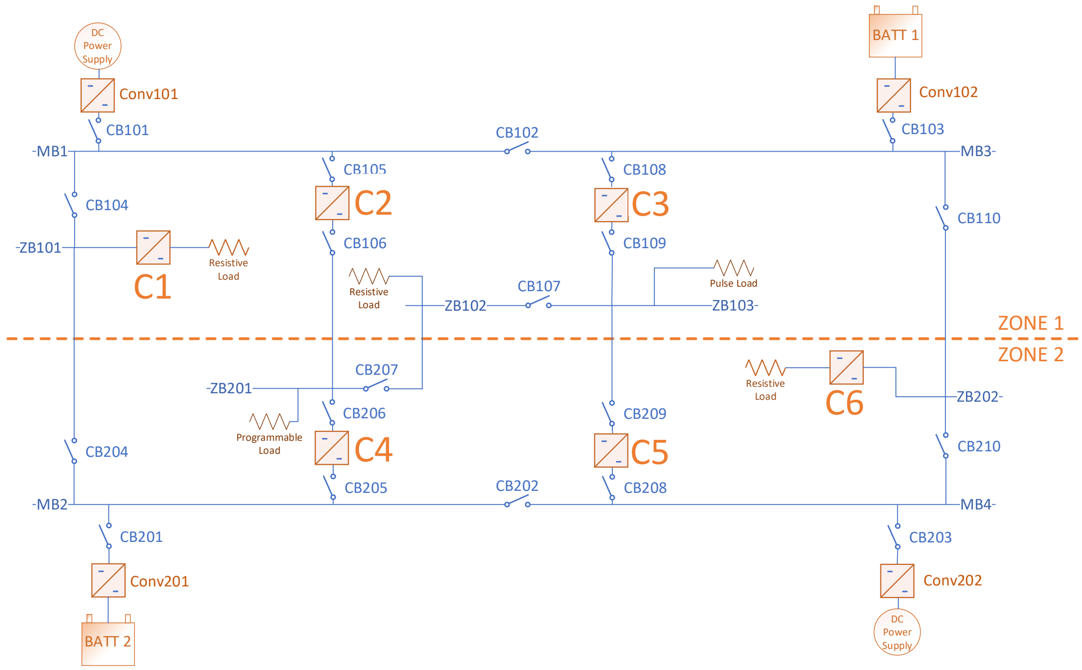

Figure 3.2 Electrical diagram of power converter system within the testbed. Note. Reprinted from “Digital Shadow-based Detection of Blockage Formation In Water-Cooled Power Electronics” by Richard Hainey, 2024, IMECE, Volume, page number. Copyright Year by "Name of copyright holder".

The testbed is outfitted with a cooling loop for heat management and designed to mimic the cooling system of a naval ship. The cooling loop composed of a submersible pump, tank, water heater, the six power modules, and a three-fan radiator, as shown in Figure 3.3. The testbed is also instrumented with nine thermocouples that record temperature data during experiments. There are four points where these thermocouples are located: six on the heat sinks of the power modules, one submerged in the tank, and one after both the heater and radiator. The direction in which the water flows through the cooling loop is shown in Figure 3.4. Distilled water is circulated throughout the loop at one gpm by a submersible pump inside a 10-gallon tank. Water first flows through a water heater coupled with a power supply, where dissipated heat produced by ship systems is emulated. This heated water then flows through the coolant plates of six power modules. While the power modules can produce heat, they were not operational during testing and only act as thermal masses. After the power modules, water enters the fan radiator and the chilled water returns to the tank. These components help form the cooling system that deploys a 40-hour load profile simulating a battle-time scenario.

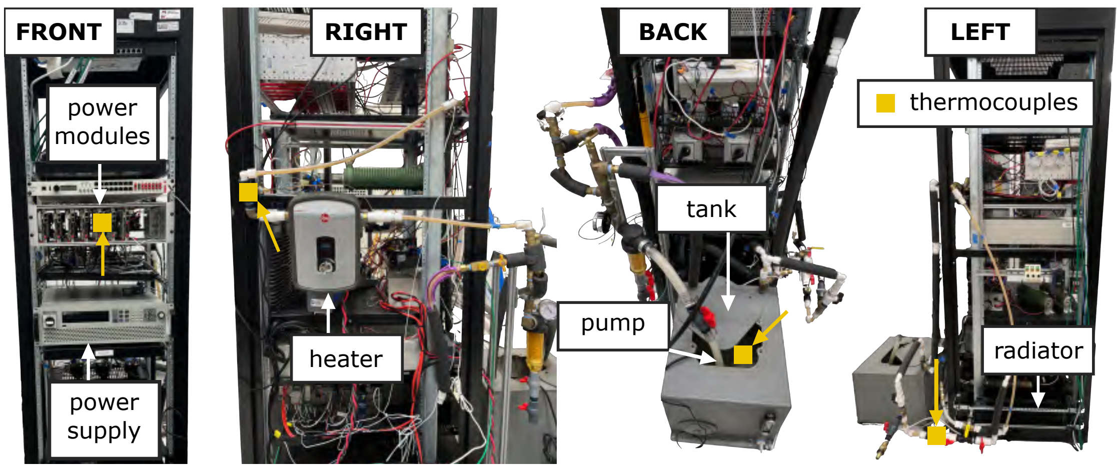

Figure 3.3 Labeled image of the testbed from the front, left, right, and back.

3.3.3 Characterization of The Multi-physics Representation

A multi-physics modeling approach was used to capture the complex behavior of the testbed cooling system. The model was constructed in MATLAB Simscape to handle non-linearities and changing operating conditions effectively. Before implementing the model into the digital twin framework, the model must be characterized to ensure that the model is faithfully representing the physical system. While the updating scheme can fit an inaccurate model to a physical system, it will reduce its accuracy of future predictions. To ensure the highest level of accuracy in the model, the cooling loop components were explored, and their parameters were found. The parameters of each component were found using three different methods: measuring, experimentally, and estimation. Measuring is the easiest method for identifying the parameters of a component. They can be found by physically measuring the dimensions of the components or by finding them in documentation/manuals provided by the manufacturer. Experimentally, it involves collecting temperature, pressure, and flow rate data for long periods. Once the experimental data is collected, the model can be calibrated. Most parameters can be found by measuring or using experimental data. Certain parameters cannot be calculated without making idealistic assumptions about the system. Model parameters such as thermal resistances, thermal conductivities, and convection coefficients are not easily solved. These parameters were either estimated using Simulink Parameter Estimation or assigned as tunable parameters for the updating scheme.

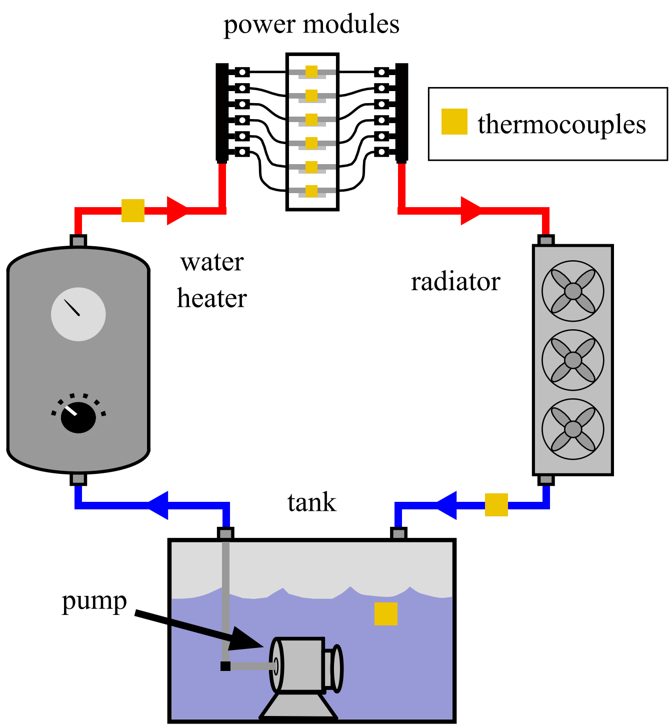

Figure 3.4 Cooling loop diagram.

3.3.4 Digital Twin Updating Scheme

The objective of the updating scheme is to continuously adjust tunable parameters and calibrate the multi-physics representation of the testbed cooling loop. To find the optimal model parameters, a meta-heuristic algorithm systematically tests permutations of model parameters to reduce the error between simulation and experimental temperatures. Particle swarm optimization is a popular stochastic optimization technique that follows simple mathematical rules to solve complex problems. This meta-heuristic algorithm is based on the foraging behavior of birds [16]. In this swarm-based algorithm, particles are initialized with random positions and velocities in a search space. In this case, the positions of the particles represent various model parameters. These particles work together to find the global minimum in the search space of a given cost function. After a particle position is evaluated, its next position is calculated by adding its velocity to its current position as shown in (3.1). Three different components contribute to the velocity of each particle. The first is the inertia component; each particle is given a random initial magnitude and direction at the beginning. To reduce the influence this random component has on future positions, a damping factor W is applied to the inertia component $W.V_i^t$. The second factor is the cognitive component; the personal best position of a individual particle will influence its own velocity $r_1\phi_1(P_i - X_i^t)$. The final factor is the social component; this updates the velocity based on the global best position of the whole swarm $r_2\phi_2(P_g - X_i^t)$. The social and cognitive components also have hyperparameters r and ϕ, which can affect the performance of the algorithm. These components dictate how the swarm navigates the search space form equation (3.2) and can be modified using the hyperparameters to suit its particular problem better.

$$X_i^{t+1} = V_i^{t+1} + X_i^t \tag{3.1}$$

$$V_i^{t+1} = W.V_i^t + r_1\phi_1(P_i - X_i^t) + r_2\phi_2(P_g - X_i^t) \tag{3.2}$$

Before implementing the PSO into the updating scheme, the challenge of handling online sensor data must be solved. While sensor data is continually relayed to the digital twin, the model is not continuously updated. Instead, the model is updated periodically using windows of time-series data. After the particles are initialized, the algorithm waits a designated amount of time until a data window is collected. The time-series data acquired during this time is referred to as a window. Then, the particle swarm attempts to fit the simulation output to the acquired window. The positions/parameters of the particles are continually evaluated until enough time has passed for a new window. Where the particle swarm will begin to fit the model to the new window of data, and the updated model is returned to the operator. The process of periodically updating the model using windows of sensor data is illustrated in Figure 3.5.

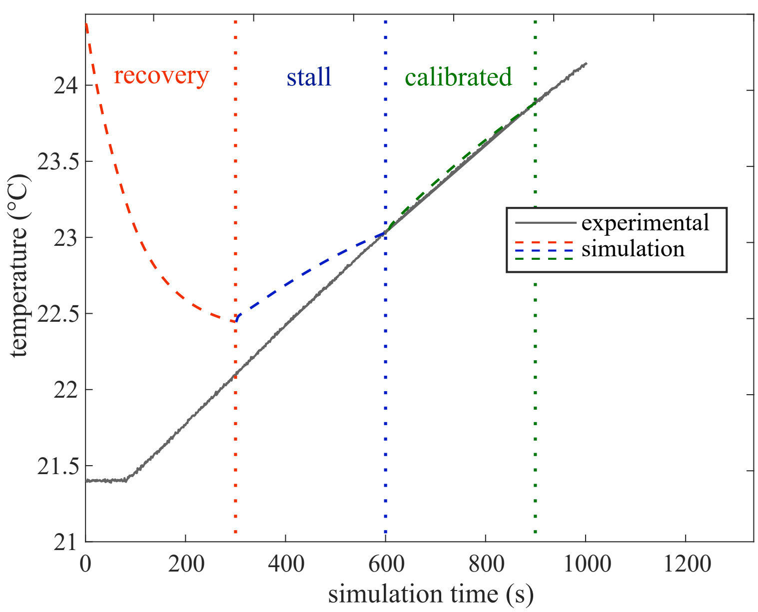

Figure 3.5 Algorithm updating the hyper-parameters and cost function according to the performance of the previous window.

The accuracy of the newly updated model is highly contingent on the final state of the previous window. To account for this problem, the hyper-parameters and cost function of the particle swarm are updated depending on the error between the initial experimental and simulation temperatures. If the error is large enough, a "recovery" window is initiated where the cost function and the PSO hyper-parameters are adjusted to encourage a much more rapid but less precise exploration of the search space as shown in (3.3). The "recovery" window only seeks to rectify the error between final temperatures so that the new window can start from an accurate state. Conversely, if the error is small, a "calibrated" window will be prompted. The cost function and the PSO hyper-parameters are adjusted to slow the particle movement to fine-tune the model parameters better as shown in (3.5). The “stall" window is a mix of both "recovery" and "calibrated" windows. The cost function is illustrated in (3.4), which is a combination of (3.3) and (3.5) to tune model parameters. These three window types reduce the calibration time of the updating scheme.

$$|T^{\mathrm{n}}_{\mathrm{exp}} - T^{\mathrm{n}}_{\mathrm{sim}}| \tag{3.3}$$

$$|T^{\mathrm{n}}_{\mathrm{exp}} - T^{\mathrm{n}}_{\mathrm{sim}}| + \sqrt{\frac{(T^{\mathrm{i}}_{\mathrm{exp}} - T^{\mathrm{i}}_{\mathrm{sim}})^2}{nSamples}} \tag{3.4}$$

$$\sqrt{\frac{(T^{\mathrm{i}}_{\mathrm{exp}} - T^{\mathrm{i}}_{\mathrm{sim}})^2}{nSamples}} \tag{3.5}$$

Choosing an appropriate window size is paramount for the performance of the updating scheme. The size of the update window is largely depends on the simulation runtime. If the window is too small, the updating scheme may not have enough time to find the optimal parameters before a new window is available. Conversely, if the window is too large, the latency between the physical system and the virtual model will reduce the ability of the digital twin to adapt to dynamic changes in the physical system. An investigation into the consistency of simulation run times was performed to determine the appropriate size for an update window Figure 3.6. The simulation was run 10,000 times on the same five-minute window to gather run-time metrics. The average run time of the simulation was 4.636 seconds, with a standard deviation of 0.634 seconds. The shape of the distribution is expected from a Microsoft Windows OS, as it is heavily skewed to the left with a few large outliers. The max simulation time was 16.925 seconds. While the max time deviates significantly from the average time, a simulation time of sixteen seconds or more only occurred twice out of 10,000 instances. As a result of this investigation, a ten-minute window was chosen.

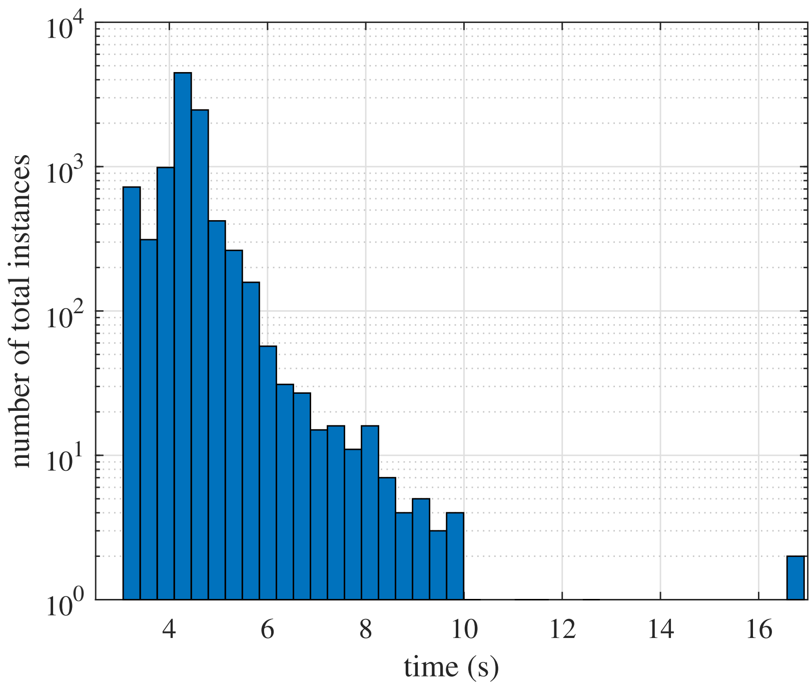

Figure 3.6 Timing distribution of 10,000 model instances.

The updating algorithm utilizes windows of online sensor data and modifies the PSO cost function, and hyper-parameters morph to create a robust updating scheme. How the updating algorithm functions is shown in Figure 3.7. The algorithm starts by initializing a swarm of particles with random positions and velocities. Once enough data is collected to fill a window, the main loop begins. The same dissipated heat load profile (W) deployed onto the water heater is also set for the model, linking the virtual representation to the physical system. Once the load profile is configured, the PSO update loop begins. The values of various model parameters are tuned using the position of a particle. The configured model is then run, and the simulated temperature is evaluated against the window of experimental data using a cost function. If a new data window is not available, the next position of the particle will be calculated, and the next particle will be evaluated. This PSO update loop continues, evaluating particle positions until a new data window is collected. Once a new window is acquired, the RMSE score of the model is used to determine whether or not the model is calibrated. If the RMSE of the global best model is less than 0.1 °C, the user is flagged that the model is calibrated; if not, the model state is discarded, and the main loop continues. The cost and acceleration factors of particle swarm are reset, and a number from [-1, 1] is added to each of the positions. This element of randomness is essential to keep the particles from getting stuck at the global minimum of the previous window. Next, the algorithm updates the PSO cost function and the hyper-parameters according to the error between the experimental and the simulation initial temperatures. Once the cost function and hyper-parameters are adjusted, the following data window is configured and the main loop of the algorithm repeats.

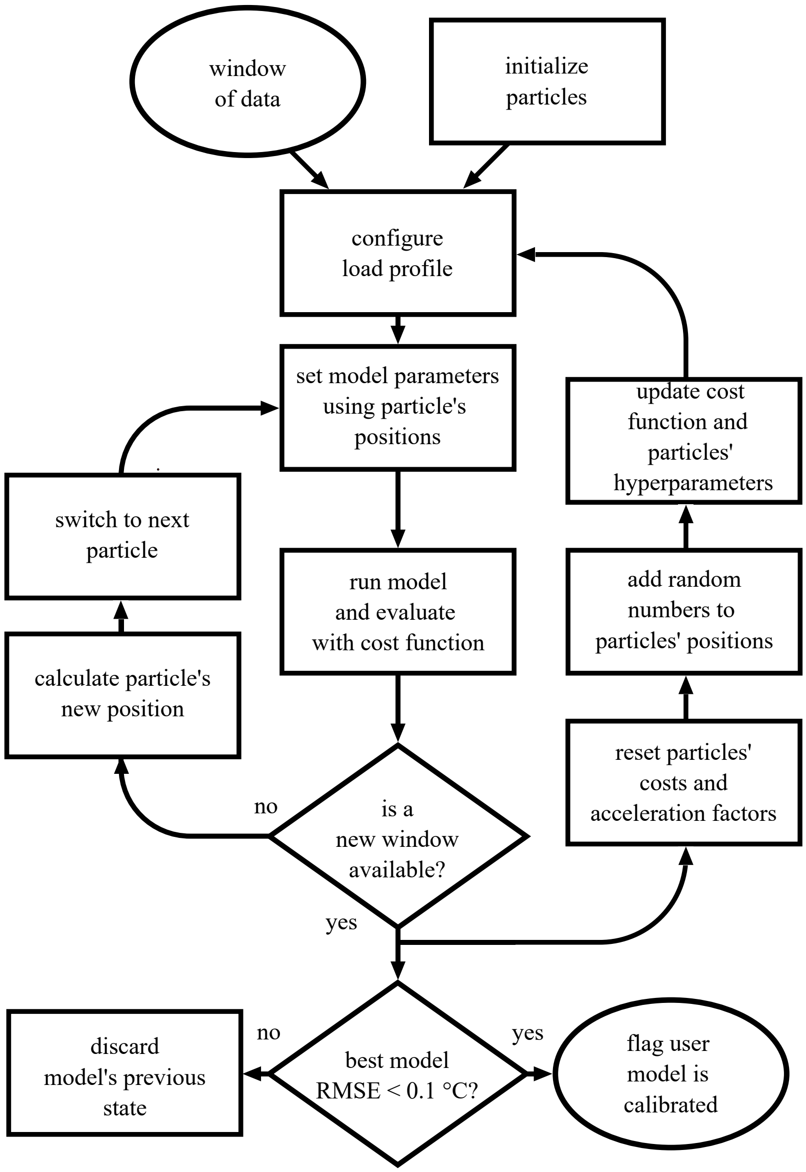

Figure 3.7 Flowchart of the multi-physics model updating scheme.

3.3.5 Test Scenario

An arbitrary ten-hour test scenario was deployed onto the testbed to gather data for model characterization. The scenario was created to replicate a battle-time scenario while keeping in mind the limitations of the testbed cooling system. Many ship systems were considered while designing a load profile. The electrical loads of the systems were calculated, and dissipated heat was dumped into the system via the power supply and water heater. However, not every system was modeled in the same way. Hotel and communication systems are air-cooled and are only modeled electrically. The water-cooled, propulsion, navigational, and directed energy systems are modeled electrically and thermally. The propulsion system is throttled throughout the test and represents the majority of the load for the duration of the test. The navigational, communication, and radar systems have a constant load throughout the test. Unlike these systems, hotel and directed energy systems are applied intermittently, depending on the time. The load of each system is combined to create a ten-hour load profile deployed onto the power supply Figure 3.8.

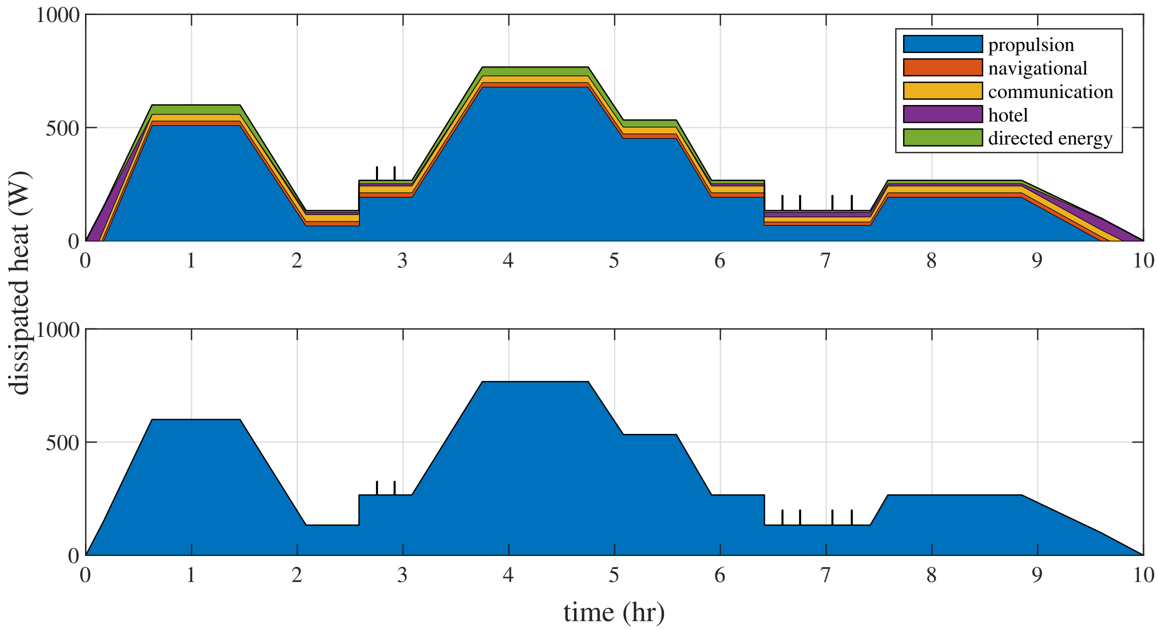

Figure 3.8 Top: Distribution of dissipated heat per ship system during a ten-hour scenario. Bottom: The cumulative dissipated heat of a ship during a ten-hour scenario.

The ten-hour load profile was repeated four times, creating a 40-hour experiment; the results are shown in Figure 3.9. During the experiment, temperature readings were taken at four points along the cooling system (heater, power modules, radiator, and tank). As previously mentioned in section 3.3.3, experimental data was used to fit the model. Results of this manually tuned model are shown in Figure 3.10. The simulation results show that the manually tuned model is relatively accurate. One significant benefit of a calibrated model is that it can be leveraged to conduct test scenarios without physically altering the testbed. In later sections, This manually tuned model will synthesize experimental data to assess the performance of the updating scheme when handling discontinuities.

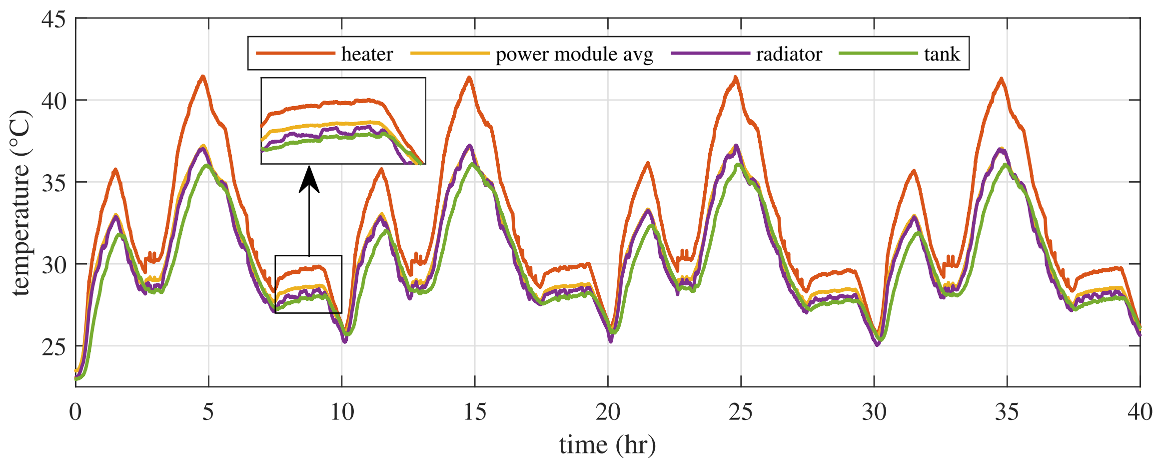

Figure 3.9 Recorded temperature at the tank, heater, power modules, and radiator, during the 40-hour experiment.

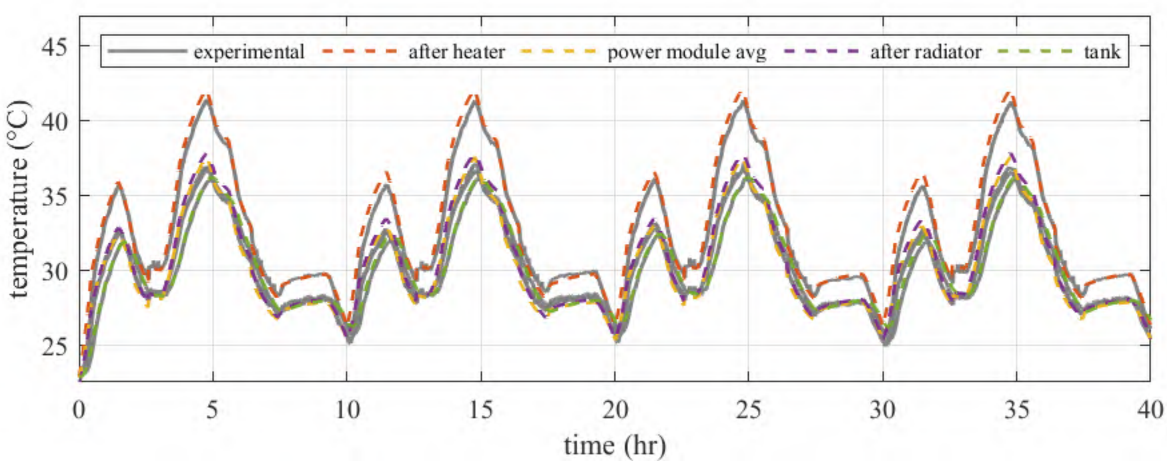

Figure 3.10 Manually tuned model simulation results of the 40-hour experiment.

3.4 Results and Discussion

This section investigates the robustness of the proposed digital twin updating scheme and its ability to handle physical discontinuities.

3.4.1 Updating Scheme Investigation

To understand how much the updating scheme improves the virtual model, it was directly compared against a model without updating Figure 3.11. The model was run for the first three and a half hours of the experiment without updating any model parameters. At the three-and-a-half-hour mark, the model parameters were randomized; this was done to determine if the updating scheme could recover from a bad initial state. The randomization of the model parameters causes a drastic increase in the simulation temperature. After two ten-minute windows of data, the updating scheme improves the model and returns it to its initial accuracy. After these two windows, the accuracy of the model continues to improve, exceeding the performance of the static model, best shown by RMSE percent improvement in Table 3.1.

Table 3.1 Metric results for a model continuously updated at four locations, numerical case study.

Metrics	After Heater	Average of Power modules	After Radiator	Inside Tank
mean absolute error (°C)	0.032	0.016	0.048	0.014
mean square error (°C)	0.03	0.001	0.005	0.001
root mean square error (°C)	0.051	0.024	0.024	0.021
normalized root mean squared error	0.03	0.001	0.005	0.001
percent improvement RMSE (°C)	93%	95%	96%	93%

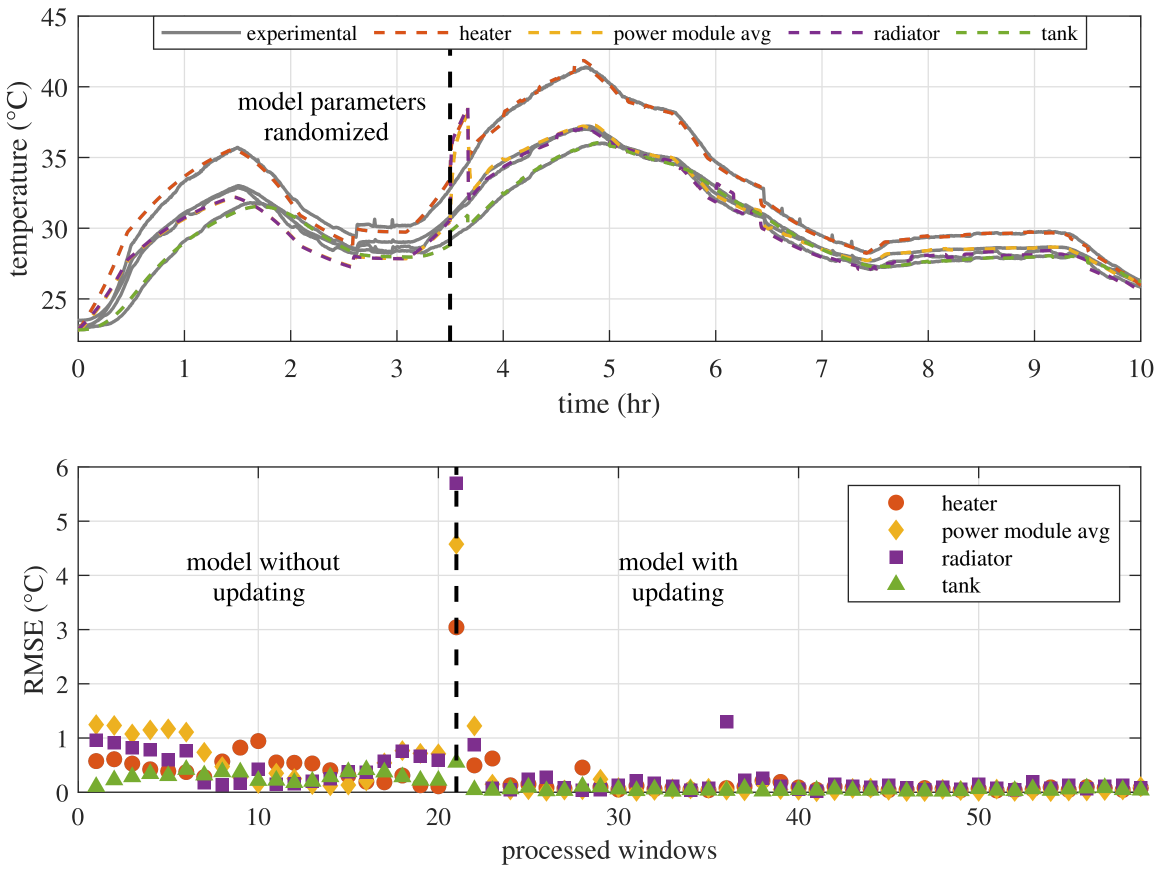

Figure 3.11 Model without updating v.s. model with updating scheme

3.4.2 Test Case Scenarios

Two scenarios were deployed to test the adaptability of the updating scheme in handling discontinuities in the physical system. Using the manually tuned model previously discussed, experimental data of two scenarios was synthesized. In each scenario the static model was ran for first three and a half hours. Then a physical change would be made to the virtual representation to adjust its behavior. The model updating scheme would then begin evaluating permutations of model instances to find the optimal parameters.

In the first scenario, the power modules dissipate heat into the system three and a half hours into the experiment. The results from this scenario are shown in Figure 3.12. After an initial struggle, The updating scheme can identify the optimal parameters after a few windows. While the virtual representation initial seems calibrated, there is large discrepancy at the four and a half hour mark. This error between the synthetic and simulation radiator temperatures can be attributed to the random nature of PSO. The random number added to the particle positions in between windows of data can sometimes be to large enough to cause needless exploration of search space. This problem can be addressed by tuning the hyper-parameters of the algorithm.

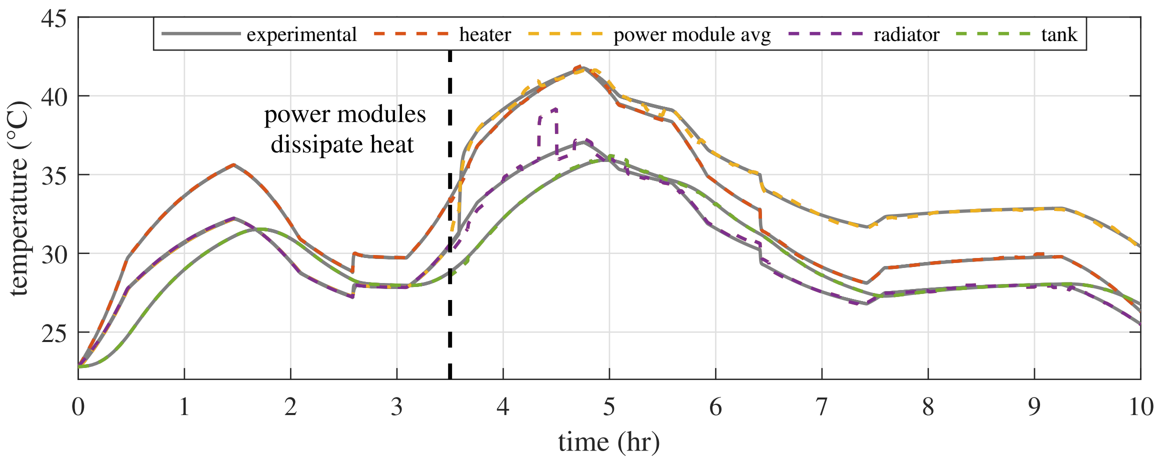

Figure 3.12 Scenario 1: Power modules dissipate heat into the cooling loop.

The second scenario is a situation where insulation on the tank is removed. Again, the updating scheme can recover after a few windows. However, when analyzing the simulation temperatures after the discontinuity, one would expect the largest error to be at the tank. However, the most significant error occurs at the heater due to the PSO cost function. The cost function is the cumulative error at each of the four points. So, the particle swarm can find model parameters that satisfy three out of the four points for the first couple of windows after the discontinuity. Only after the recovery window is initiated the swarm of particles escape the local minimum and find the global minimum, ultimately calibrating the model.

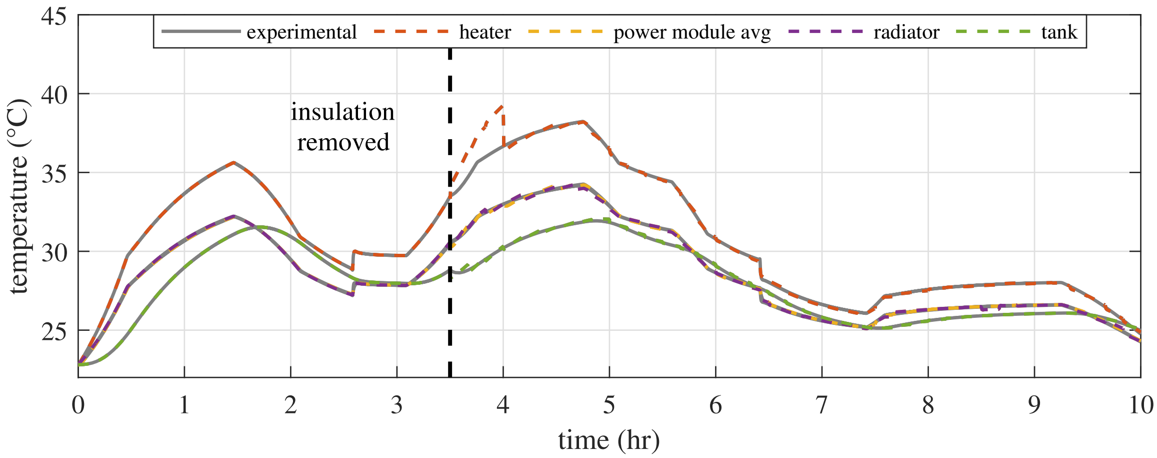

Figure 3.13 Scenario 2: Insulation removed from the tank.

3.5 Conclusion

As modern systems become increasingly complex, maintaining accurate and up-to-date virtual representations of these systems poses a significant challenge. Throughout the lifecycle of a system, discrepancies between itself and its virtual counterpart can lead to reduced performance and inaccuracies. To address this, data-driven approaches have emerged as an option for enabling real-time model updating. This paper investigates the application of a digital twin framework to enhance the accuracy of a multi-physics model. The digital twin framework comprises an updating scheme, a physical testbed designed to emulate a cooling system of a ship, and a multi-physics model of that system. The updating scheme leverages a swarm of particles and online sensor data to evaluate permutations of model parameters. Two scenarios were considered to evaluate the performance of the digital twin framework where a physical change in the cooling system would occur. Results demonstrate that the digital twin framework can adapt to system changes in real-time and improve the accuracy of its static virtual representation by more than 90%.

Chapter 4

Conclusion

As naval systems and subsystems increase in complexity, maintaining accurate virtual representations will be crucial to ensure optimal performance and reliability. This work emphasizes the importance of digital twin frameworks when addressing the challenge of sustaining up-to-date virtual models over the lifecycle. To address this problem, a novel model updating scheme within a digital twin framework is proposed, which incorporates real-time sensor data and particle swarm optimization (PSO). The updating scheme periodically updates a virtual representation using windows of data to calibrate its physical counterpart. After the virtual representation is deemed calibrated, a proposed digital twin response is returned to its user, where it can be used to optimize, test, and perform lookahead predictions of its physical system.

Two thermal management systems were investigated to evaluate the performance of the updating scheme. The first investigation involved adjusting the radiator fan speed and valve positioning of a cooling loop during its run time. Results demonstrate that the model updating scheme could recalibrate the model every five minutes, maintaining an accurate digital twin representation. In the second investigation, a more complex electro-thermal system, designed to represent a notional shipboard, was successfully characterized. The updating scheme further improved the accuracy of the model by more than 90% when compared to a static virtual model. These outcomes validate the ability of digital twins to adapt to the current state of their physical counterparts and enhance their predictive accuracy.

The success of this methodology emphasizes the potential of digital twins in real-time system monitoring and decision-making. By maintaining accurate virtual representations of physical systems, digital twins empower operators with insights to address potential issues preemptively, thus optimizing system health and longevity. As such, integrating real-time sensor data assimilation and population-based optimization techniques within digital twins holds promise for revolutionizing system management across complex naval and industrial applications, establishing a robust foundation for the next generation systems.

Bibliography

[1] Mohd Nadhir Ab Wahab, Samia Nefti-Meziani, and Adham Atyabi, *A comprehensive review of swarm optimization algorithms*, PloS one **10** (2015), no. 5, e0122827.

[2] Ahmed G. Abo-Khalil, *Digital twin real-time hybrid simulation platform for power system stability*, Case Studies in Thermal Engineering **49** (2023), 103237.

[3] José Joaquín Aguilera, Wiebke Meesenburg, Wiebke Brix Markussen, Benjamin Zühlsdorf, and Brian Elmegaard, *Real-time monitoring and optimization of a large-scale heat pump prone to fouling-towards a digital twin framework*, Applied Energy **365** (2024), 123274.

[4] Carlos Antonio Alves de Araujo Junior, Juan Moises Mauricio Villanueva, Rodrigo José Silva de Almeida, and Isaac Emmanuel Azevedo de Medeiros, *Digital twins of the water cooling system in a power plant based on fuzzy logic*, Sensors **21** (2021), no. 20, 6737.

[5] Antonis Antonopoulos, Bill Karakostas, Takis Katsoulakos, Anargyros Mavrakos, Theodosis Tsaousis, and Stathis Zavvos, *A digital twin enabled decision support framework for ship operational optimisation towards decarbonisation*, Proceedings of Eighth International Congress on Information and Communication Technology (Singapore) (Xin-She Yang, R. Simon Sherratt, Nilanjan Dey, and Amit Joshi, eds.), Springer Nature Singapore, 2023, pp. 467–476.

[6] Sebastian Baba, Marcin Zelechowski, and Marek Jasinski, *Estimation of thermal network models parameters based on particle swarm optimization algorithm*, 2019 IEEE 13th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), IEEE, 2019, pp. 1–6.

[7] Jerry Banks, John S. Carson, Barry L. Nelson, and David M Nicol, *Discrete-event system simulation*, 3 ed., Prentice-Hall, 2001 (English (US)).

[8] Sachin Kumar Bhoi, Mohamed Amine Frikha, Gamze Egin Martin, Farzad Hosseinabadi, Sajib Chakraborty, Mohamed El Baghdadi, and Omar Hegazy, *A data-driven thermal digital twin of a 3-phase inverter using hi-fidelity multi-*

physics modelling, 2023 25th European Conference on Power Electronics and Applications (EPE'23 ECCE Europe), 2023, pp. 1–8.

[9] Alberto Broatch, Benjamín Pla, Pau Bares, and Augusto Perin, *Leveraging look-ahead information for optimal battery thermal management*, Applied Thermal Engineering **220** (2023), 119685.

[10] Hector A. Duran-Limon, Arturo Chavoya, and Martha Hernández-Ochoa, *The role of machine learning in big data analytics: Current practices and challenges*, pp. 47–74, Springer International Publishing, Cham, 2024.

[11] R.D. Geertsma, R.R. Negenborn, K. Visser, and J.J. Hopman, *Design and control of hybrid power and propulsion systems for smart ships: A review of developments*, Applied Energy **194** (2017), 30–54.

[12] Matthieu Guitton, *Fighting the locusts: Implementing military countermeasures against drones and drone swarms*, Scandinavian Journal of Military Studies **4** (2021), 26–36.

[13] Matthieu J Guitton, *Fighting the locusts: Implementing military countermeasures against drones and drone swarms*, Scandinavian Journal of Military Studies **4** (2021), no. 1, 26–36.

[14] Imperix datasheet, *PEB 8038 Half-Bridge SiC Power Module, Imperix Datasheet*, February 2023, Accessed: 2024-01-26.

[15] Andrew Wunderlich Jared Cronin, Enrico Santi and Joshua Knight, *Fast System Level Model for Digital Twin Based Optimization of Naval Power and Energy System*, IEEE Electric Ship Technologies Symposium (ESTS), 2023, pp. 1–7.

[16] James Kennedy and Russell Eberhart, *Particle swarm optimization*, Proceedings of ICNN'95-international conference on neural networks, vol. 4, ieee, 1995, pp. 1942–1948.

[17] Johannes Kuprat, Karthik Debbadi, Joscha Schaumburg, Marco Liserre, and Marius Langwasser, *Thermal digital twin of power electronics modules for online thermal parameter identification*, IEEE Journal of Emerging and Selected Topics in Power Electronics **12** (2024), no. 1, 1020–1029.

[18] Quanbo Lu, Dong Zhu, Meng Wang, and Mei Li, *Digital twin-driven thermal error prediction for cnc machine tool spindle*, Lubricants **11** (2023), no. 5, 219.

[19] Kenichi Matsuoka and David Hill, *Online optimization of cooling water system in a district cooling plant by using digital twin.*, ASHRAE Transactions **126** (2020), no. 2.

[20] Saeid-Reza Mohseni, Meisam Jahanshahi Zeitouni, Ahmad Parvaresh, Saber Abrazeh, Meysam Gheisarnejad, and Mohammad-Hassan Khooban, *Fmi real-time co-simulation-based machine deep learning control of hvac systems in smart buildings: Digital-twins technology*, Transactions of the Institute of Measurement and Control **45** (2023), no. 4, 661–673.

[21] A. Monti, D. Boroyevich, D. Cartes, R. Dougal, H. Ginn, G. Monnat, S. Pekarek, F. Ponci, E. Santi, S. Sudhoff, N. Schulz, W. Shutt, and F. Wang, *Ship power system control: a technology assessment*, IEEE Electric Ship Technologies Symposium, 2005., 2005, pp. 292–297.

[22] National Academy of Engineering and National Academies of Sciences, Engineering, and Medicine, *Foundational research gaps and future directions for digital twins*, The National Academies Press, Washington, DC, 2023.

[23] Chao-Chung Peng and Cheng-Yi Su, *Modeling and parameter identification of a cooling fan for online monitoring*, IEEE Transactions on Instrumentation and Measurement **70** (2021), 1–14.

[24] Mikel D Petty, *Modeling and validation challenges for complex systems*, Engineering Emergence, CRC Press, 2018, pp. 199–216.

[25] Adam P Piotrowski, Jaroslaw J Napiorkowski, and Agnieszka E Piotrowska, *Particle swarm optimization or differential evolution—a comparison*, Engineering Applications of Artificial Intelligence **121** (2023), 106008.

[26] Kerry Sado, Richard Hainey, Jose Peralta, Austin Downey, and Kristen Booth, *Digital twin model for predicting the thermal profile of power cables for naval shipboard power systems*, 2023 IEEE Electric Ship Technologies Symposium (ESTS), IEEE, August 2023.

[27] Kerry Sado, Jack Hannum, and Kristen Booth, *Digital Twin Modeling of Power Electronic Converters*, 2023 IEEE Electric Ship Technologies Symposium (ESTS), 2023, pp. 86–90.

[28] Kerry Sado, Jack Hannum, Eric Skinner, Herbert L. Ginn, and Kristen Booth, *Hierarchical Digital Twin of a Naval Power System*, 2023 IEEE Energy Conversion Congress and Exposition (ECCE), 2023, pp. 1514–1521.

[29] Karan Shah, T. V. Prabhakar, Sarweshkumar C. R., Abhishek S. V., and Vasanth Kumar T, *Construction of a digital twin framework using free and open-source software programs*, IEEE Internet Computing **26** (2022), no. 5, 50–59.

[30] Jason Smith, Hung-Tien Huang, Austin R. J. Downey, Alysson Mondoro, Benjamin Grisso, and Sourav Banerjee, *Multi-event model updating for ship structures with resource-constrained computing*, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2022 (Daniele Zonta, Zhongqing Su, and Branko Glisic, eds.), SPIE, April 2022.

[31] Fei Tao, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, and Ping Ji, *Digital twin modeling*, Journal of Manufacturing Systems **64** (2022), 372–389.

[32] Fei Tao, He Zhang, Ang Liu, and A. Y. C. Nee, *Digital twin in industry: State-of-the-art*, IEEE Transactions on Industrial Informatics **15** (2019), no. 4, 2405–2415.

[33] Adam Thelen, Xiaoge Zhang, Olga Fink, Yan Lu, Sayan Ghosh, Byeng D Youn, Michael D Todd, Sankaran Mahadevan, Chao Hu, and Zhen Hu, *A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies*, Structural and Multidisciplinary Optimization **65** (2022), no. 12, 354.

[34] Minh Q. Tran, Hélder S. Sousa, José Matos, Sérgio Fernandes, Quyen T. Nguyen, and Son N. Dang, *Finite element model updating for composite plate structures using particle swarm optimization algorithm*, Applied Sciences **13** (2023), no. 13, 7719.

[35] DJ Wagg, Keith Worden, RJ Barthorpe, and Paul Gardner, *Digital twins: state-of-the-art and future directions for modeling and simulation in engineering dynamics applications*, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering **6** (2020), no. 3, 030901.

[36] Andrew J. Wileman, Sohaib Aslam, and Suresh Perinpanayagam, *A component level digital twin model for power converter health monitoring*, IEEE Access **11** (2023), 54143–54164.

[37] Katherine Wilsdon, Joshua Hansel, M. Ross Kunz, and Jeren Browning, *Autonomous control of heat pipes through digital twins: Application to fission batteries*, Progress in Nuclear Energy **163** (2023), 104813.

[38] Peng-Yeng Yin, Shiuh-Sheng Yu, Pei-Pei Wang, and Yi-Te Wang, *A hybrid par-*

ticle swarm optimization algorithm for optimal task assignment in distributed systems, Computer Standards & Interfaces **28** (2006), no. 4, 441–450.

[39] He Zhang, Qinglin Qi, Wei Ji, and Fei Tao, *An update method for digital twin multi-dimension models*, Robotics and Computer-Integrated Manufacturing **80** (2023), 102481.

[40] Guanjia Zhao, Zhipeng Cui, Jing Xu, Wenhao Liu, and Suxia Ma, *Hybrid modeling-based digital twin for performance optimization with flexible operation in the direct air-cooling power unit*, Energy **254** (2022), 124492.

[41] Yi-Chen Zhu, David Wagg, Elizabeth Cross, and Robert Barthorpe, *Real-time digital twin updating strategy based on structural health monitoring systems*, p. 55–64, Springer International Publishing, 2020.

[42] TI Zohdi, *A digital-twin and machine-learning framework for precise heat and energy management of data-centers*, Computational Mechanics **69** (2022), no. 6, 1501–1516.