This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

Construction of a Digital Twin Framework using Free and Open-Source Software Programs

Karan Shah, T. V. Prabhakar, Sarweshkumar C. R., Abhishek S. V., Vasanth Kumar T

Abstract **– We present a novel Digital Twin Framework for a portable table-top sized temperature-controlled system constructed entirely using free and open-source software programs. By utilizing tools such as Robotic Process Automation (RPA) and a PLC program, the entire process was automated to achieve a true Digital Twin. Further, a mathematical model was constructed for the temperature-controlled system. The data from the physical model and the mathematical model was then compared and a what-if scenario was analyzed to predict and prevent a potential failure in the physical system. Additionally, a simulation model was created to visualize the results.**

I. INTRODUCTION

With the advent of Industry 4.0, design manufacturing practices and automation have become smarter by getting a boost from the Internet of Things (IoT) and the remote accessibility. Digital Twin (DT) works as an exemplar for the Industry 4.0 and shows the ability of a digital system to work harmoniously alongside a physical system. A DT framework effortlessly allows for data communication and integration between a physical system and a digital system in either direction. Moreover, it can run parallel to a physical system and help in the recognition of unexpected upcoming failures. It computes the real time data and adds the predictability aspect which saves cost, manpower and time. However, development of a digital twin technology is difficult, and it comes with an investment cost. This paper focuses on the affordability and accessibility of the DT technology by developing a DT system entirely using free and open-source software programs.

The primary goal of this paper is to arrive at a suitable framework to construct DTs using open-source software programs. Subsequently to building the DT, our goal is to perform predictive analysis.

Our key contributions are as follows:

a) We built from scratch an IoT enabled table-top and portable isolated temperature controlled chamber.

b) We constructed a mathematical model of the system using first principles.

c) We constructed a Digital Twin using free and open-source software.

d) We conduct a "what-if" analysis from the data and perform an action in real time on the physical system.

II. EXPERIMENTAL SETUP

In order to successfully establish a DT environment, real time decision-making along with rapid analysis is an important requirement. Further, the entire process requires to be automated in order to differentiate digital twin from a digital shadow where manual data flow is sufficient. Therefore, three major elements are incorporated in the experimental setup where automatic data flow with real time application is ensured: (a) the Physical Model, (b) the Simulation Model, and (c) the Automation Framework.

A. Application and Physical Model

In this paper our goal is to build a table top sized, portable temperature-controlled chamber. The operational temperature range is between 20 and 40 °C. This system comprises of an isolated chamber that maintains the chamber temperature between a given range for a specific time period. A DT of such a system allows to monitor and predict the operational challenges to better control the physical system in real time. It adds a layer of safety and provides the ability to run what-if scenarios to predict the behavior of the physical system.

1) Hardware: The physical model is an electro-mechanical system of an isolated chamber with an analog temperature sensor [1] that regulates the temperature inside the chamber. Our system comprises of a peltier chip inside the chamber with hot side contributing to the rise in chamber temperature. A heat sink with an attached fan is connected to the peltier chip using a thermal adhesive. Similarly, there are two peltier chips on top of the chamber with their cold sides accounting for the decrease in chamber temperature. An aluminium plate is attached to these peltier chips using a thermal adhesive.

Our system also consists of an additional thermal sensor [2] that maps out a 4 x 4 temperature region inside the chamber. A cylindrical aluminium object is placed in the 4 x 4 sub region to accurately map out a reference point. This reference point is established to validate the results from the digital system.

Figure 1 shows the hardware for the physical model. Towards the left, the analog temperature sensor attached to the top plate along with the fan, sink and a peltier chip attached to the bottom plate is shown. Also the thermal sensor along with the reference object is shown. Towards the right, the cold side hardware consisting of the peltier chips, heat sinks and fans are shown.

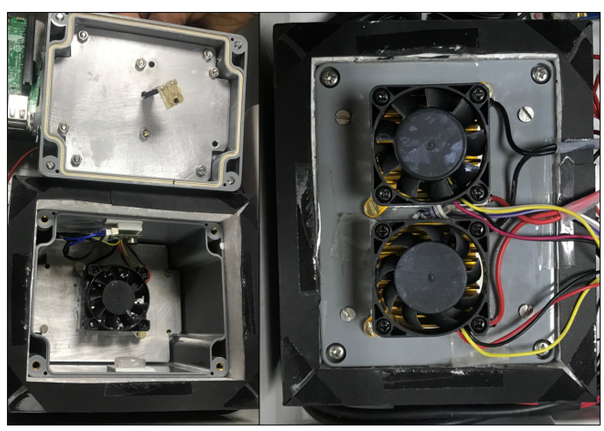

Figure 1: Chamber from inside and outside

2) Open-Source Software Program: For our physical system, we utilize the services of two open-source software programs. The first one is the OpenPLC (open-source Programmable Logic Controller) [3]. This program is used to process the analog input from the temperature sensor in order to automate the physical system. A Raspberry Pi mini computer is used to communicate with the OpenPLC to further connect with the second open-source program, Eclipse Ditto [4]. Ditto is an open-source cloud platform that focuses on the back end scenarios by providing Web APIs to simplify working with the connected devices. Eclipse Ditto along with a web application acts as a tool for remote monitoring and remote controlling of the physical chamber.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

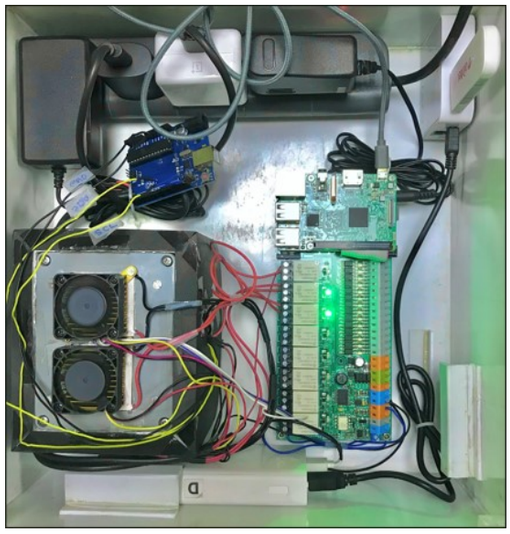

Figure 2: Entire physical system hardware

Figure 2 shows the OpenPLC hardware with the Raspberry Pi module alongside the chamber. A power supply shown along with an arduino board is used to capture the input from the thermal sensor. A USB dongle with Internet connectivity is used for hosting the web server and creating a local network.

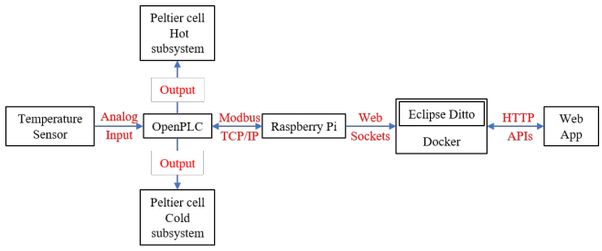

Figure 3: Overview of the physical system setup

Figure 3 shows how each component in the physical system communicates with one another. Several industrial protocols such as Modbus/TCP are used. Also, Web Sockets are used to provide real time sensor data into the DT.

3) Automation Control Loop for the Physical Model: Figure 4 shows the control loop for the physical model controlled by the OpenPLC program. The process begins with the user input in terms of a desired temperature. 20 to 40 °C is set as the executable range. If the desired temperature falls in this range, the analog temperature sensor will start measuring the current ambient temperature inside the chamber. If the ambient temperature is above the desired temperature, the cold subsystem gets turned on. If the first case is not true and the ambient temperature is below the desired temperature, the cold subsystem will be turned on and the hot subsystem will be turned off. Here, an offset of - 0.2 and + 0.1 is used to maintain a threshold with respect to the desired temperature. The process of ambient temperature measurement and comparison with the desired temperature is repeated continuously at 50ms intervals.

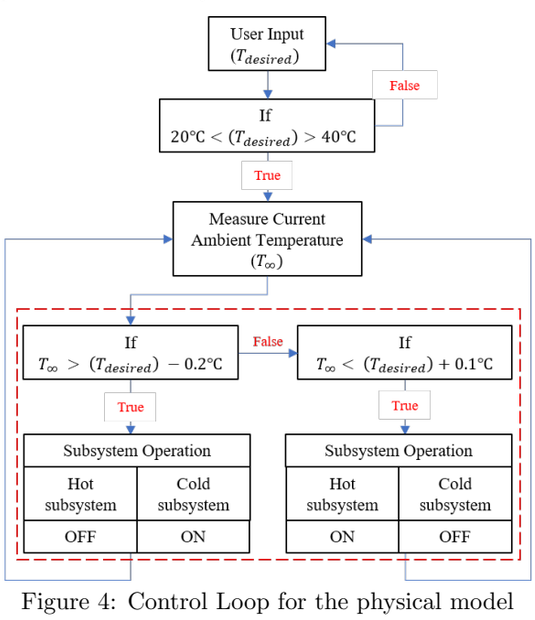

Figure 4: Control Loop for the physical model

B. Simulation Model

The simulation model consists of the Computational Fluid Dynamics (CFD) analysis of the behavior of air inside the chamber. Heat is either absorbed or released depending on which subsystem is operational at that moment.

1) Simulation Subsystems: The simulation is broken up into the cold subsystem and the hot subsystem. The hot and cold subsystems simulate the behavior of the chamber as its temperature increases or decreases, respectively. Initially, the user input and intervention is required for the first subsystem. For each subsequent subsystem, the output of the preceding subsystem will be used as the input. Also, at no given time, both the subsystems are operating simultaneously. When one subsystem turns off, the other subsystem turns on.

2) Heat Transfer Problem Setup: Both the subsystems have a conduction and a convection elements. Additionally, there will be a radiation element for the hot subsystem where the peltier chip is present inside the chamber. However, radiation was experimentally observed to be negligible, hence, it was neglected for the mathematical modeling and analysis.

For the hot subsystem, conductive heat transfer takes place between the peltier chip and the heat sink, and convective heat transfer takes place between the heat sink and the air. For the cold subsystem conductive heat transfer takes place between the peltier chips and the plate, and convective heat transfer takes place between the plate and the air. The fan disperses the air throughout the chamber.

3) Open-Source Software Programs: The list of programs with their role in the CFD simulation process is as follows:

a) FreeCAD: FreeCAD [5] is used to model the 3D CAD of the chamber. Figure 5 shows the CAD model.

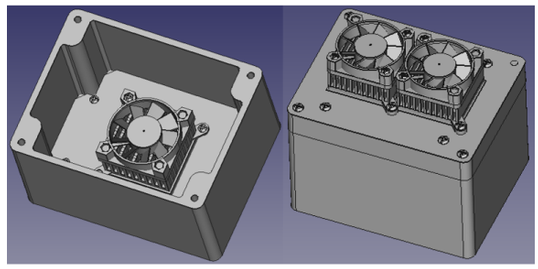

Figure 5: CAD model of the chamber from inside and outside

b) OpenFOAM: OpenFOAM [6] is used for CFD computation.

c) SimFlow: SimFlow [7] acts as a GUI for OpenFOAM. It is used to setup the CFD analysis.

d) ParaView: ParaView [8] was used to visualize the CFD results.

C. Automation Framework

Automation framework consists of three different Robotic Process Automation (RPA) workflows that can learn, mimic, and execute rule based processes to automate the operations of the digital system. UI Path Studio (community edition) [9] is used to execute these workflows.

The First workflow is designed for the simulation model where the user input initiates the CFD modeling with specific conditions. Elements such as geometry, mesh, and thermo-physical properties are preset as per the physical system. Figure 6 shows this automation flowchart.

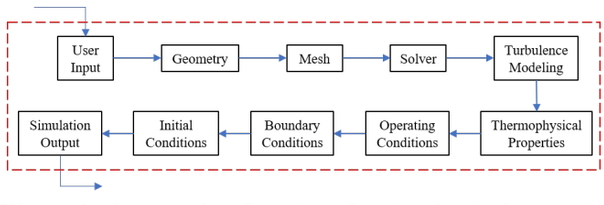

Figure 6: Automation flowchart for the simulation model (first workflow)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

The second workflow is designed to download the real time data from the Eclipse Ditto server to a Local PC and then generate variables for the digital system to process. The communication between the physical model and digital model is established in such a manner.

The third workflow is designed to send a data back to the physical system from the digital system. This workflow allows for the full cycle of DT to complete where the digital model is not only able to receive data and mirror the physical system, but also able to send data back to the physical system.

III. OVERALL SYSTEM ARCHITECTURE

Figure 7 shows the overall system architecture. The process begins with a user input in the Web App. Following this, the Eclipse Ditto Server sends an input command to execute the physical system. In real time, the data is uploaded to the server and downloaded to a local PC for the digital system to process. UI Path Studio then creates variables that hold information related to the physical system parameters. Next, the same user input command is given to the simulation and the mathematical model to arrive at the chamber temperature profile. The results of the digital model are saved in a variable in the UI Path Studio application, which then gets uploaded back to the Eclipse Ditto Server as a data file. The Web App takes over from here to further operate the physical system as per user requirements.

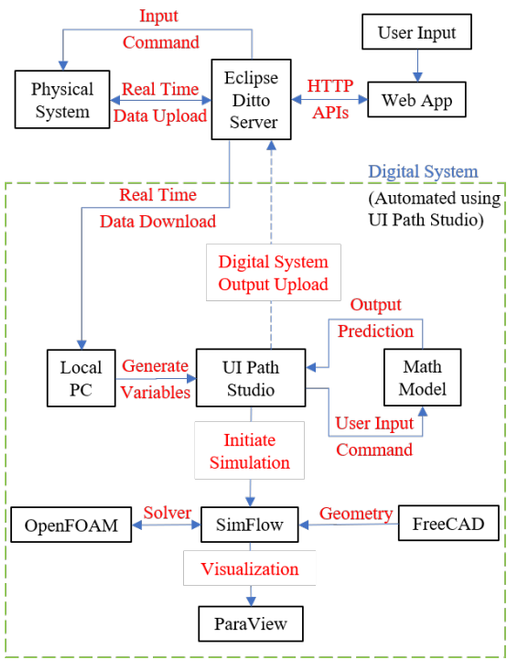

Figure 7: Overview of the system architecture

IV. TESTING SCENARIO

In order to experimentally present the functioning of the entire system, a test case with initial user input was analyzed. The initial user input consisted of the following values:

a) Initial Temperature of the Air in the Chamber: 25 °C (The air was experimentally brought to 25 °C before starting the experimentation).

b) Initial Reference Temperature: A small aluminium cylindrical object was placed inside the chamber in order to have a reference point for which exact temperature can be measured and verified. Assuming it to be in thermal equilibrium with the air inside the chamber, 25 °C value was used.

c) Desired Final Temperature of the Air in the Chamber, Upper Limit: 40 °C

d) Desired Final Temperature of the Air in the Chamber, Lower Limit: 30 °C

Here, the initial temperature of the Air is below the desired upper and lower limits, therefore the hot subsystem gets turned on first where the temperature rises to 40 °C. As the temperature would continue to rise, the hot system gets turned off and the cold subsystem gets turned on where the temperature now drops down to 30 °C.

The data from the physical system and the digital system will be compared to understand the temperature profile of different objects.

A. What-If Testing Scenario

A what-if scenario is constructed for the system where we try to achieve a final temperature inside the chamber, that is assumed to be catastrophic for the physical system. Using the mathematical model, we predict the time when the physical system will reach the threshold and send a signal to the physical system to reset the final temperature to some value that will keep the physical system within safe limits.

For this analysis, it is assumed that the temperature value of 45 °C is the maximum temperature capacity of the chamber after which the chamber or the objects inside the chamber may have a breakdown. We set the initial temperature at 35 °C from where the temperature rises to achieve 45 °C.

V. RESULT

A. Comparing Physical and Mathematical Models

Figure 8 and Figure 9 show the temperature profile when the hot and cold subsystems were turned on, respectively. The readings were taken using the thermal sensor that mapped out a 4 x 4 temperature region. The reference object was places in the middle of the cells 5, 6, 9 and 10. The two dimensional temperature profile with respect to time is shown in the figures. The mathematical model curve estimates the temperature profile at each following time step using the mathematical model (see Appendix I). The model takes the real time data at a time, t - 1, and calculates the data at time t. In such a manner, a predicted temperature profile is generated. It is observed that all the curves on the hot and the cold subsystem follow a similar trend. However, we see a significant temperature gap in both subsystems. For the hot subsystem, The thermal sensor curves and the mathematical model curve starts at 25 °C at $t = 0$, however, with time, the physical model fails to reach the desired temperature of 40 °C at a time ($t \approx 700$ secs) that the mathematical model predicted. These fluctuations can be explained in terms of the energy losses in the physical model. The mathematical model assumes that the physical model is an ideal system. However, in reality there will be heat losses through chamber walls, voltage fluctuation in the electronic components, and imperfections in the mechanical assembly. For the cold subsystem, all the curves have similar slopes. There is a ≈ 5 °C offset observed between the curves of the mathematical model and the physical model as the mathematical model reached at 40 °C during the hot subsystem's operation and the physical model only reached at ≈ 35 °C. Further, during the cold subsystem's operation there are two peltier chips trying to remove the heat from the system, therefore, the heat loss is mechanically facilitated.

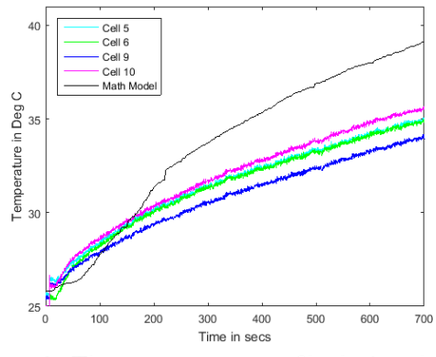

Figure 8: The temperature profile during the hot subsystem's operation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

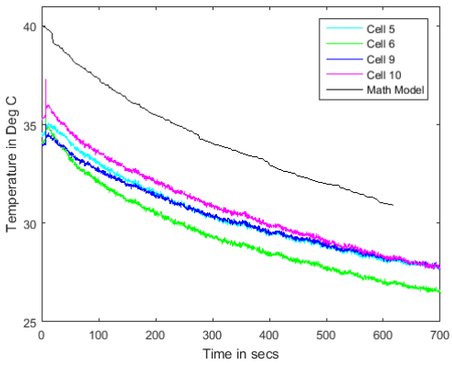

Figure 9: The temperature profile during cold subsystem's operation

B. Simulation Model

Table 1 shows the initial temperature of the peltier chip, reference object, and the air inside the chamber. The Figures 10, 11, 12 and 13 show the final temperature of the same components. It is observed that the final temperature of 40 °C (313.15 K) after the hot subsystem and the final temperature of 30 °C (303.15 K) after the cold subsystem is achieved. For each subsystem, all the three components come to a thermal equilibrium as the time progresses.

A steady state conjugate heat transfer solver was used to model the physical system. In reality, the system would be transient in nature due to factors like fluctuations in the input voltage and heat losses. However, due to limitations in the software program, a simplified ideal system was assumed. Moreover, the simulation model is only used as a visual aid to the mathematical system. All the predictions and data exchange is done using the mathematical system alone. However, the simulation model does hold significance as the mathematical model averages the temperature behavior of the entire chamber while the simulation model can help locate the exact temperature at any point inside the computational region. Figure 10 and Figure 12 show the cross-sectional view of the fluid region inside the chamber. Similarly, any cross section can be cut to visualize the temperature value at a particular location.

Table 1: Initial temperature values of different components

Component	Temperature (K)	subsystem
Peltier chip	330.15	hot
Reference object	298.15	hot
Air	298.15	hot
Peltier chip	280.15	cold
Reference object	313.15	cold
Air	313.15	cold

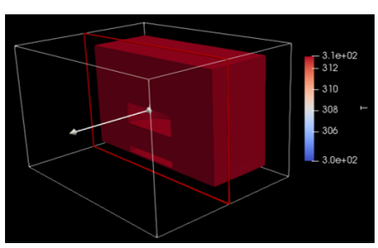

Figure 10: Final air temperature after the hot subsystem's operation

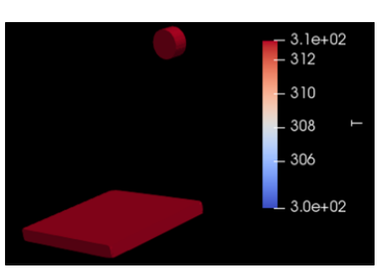

Figure 11: Final temperature of the peltier chip and the reference object after hot subsystem's operation

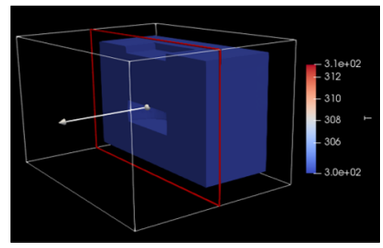

Figure 12: Final air temperature after cold subsystem's operation

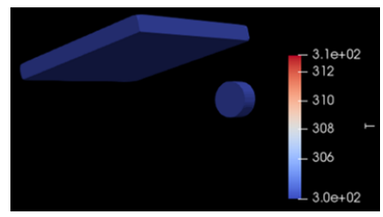

Figure 13: Final temperature of the peltier chip and the reference object after cold subsystem's operation

C. What-If Analysis

Figure 14 show the results from the what-if analysis. The physical model was initiated to go from 35 to 45 °C. However, 45 °C is assumed to be the catastrophic point. By extrapolating the mathematical model, it is predicted that the physical system will reach the catastrophic point at **663.10 secs**. Here, 40 °C is set as the decision point or the safety mark at which the system should redirect itself to another final temperature. The mathematical model predicted that the physical system will reach the 40 °C mark at **385.90 secs**. Therefore, before the physical system could reach the decision point, a signal is sent to the physical system to redirect back to 35 °C in order to maintain safety.

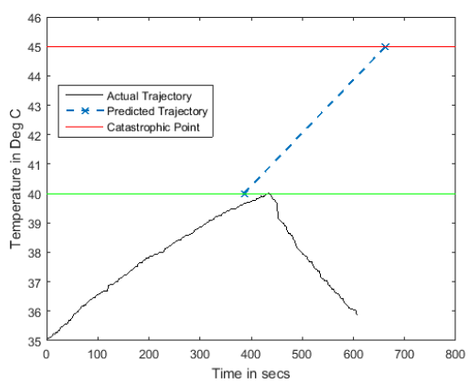

Figure 14: Actual temperature trajectory and predicted trajectory

VI. RELATED WORKS

The works in literature mostly describe DT applications and propose multi layered frameworks to seamlessly connect between the physical and its digital copy. In [10] the authors describe the application of DT for planning projects in the production system of the digital factory. The solution uses commercial licensed simulation software. The works in [11] describe the benefits of using the DT for an ultra-precision machining part. While there is detail on the machine vision software integration with commercial hardware, there is no mention about the construction of the DT from scratch. The work in [12] distinguishes Digital Thread, Digital Shadow and the DT. The concept of "reflection" is brought out for the twin's information health and the capability to perform its function with uncertainity and margin for errors. The authors in [13] build a power-grid based IEEE-39 bus DT for microgrid security. This intelligent data driven model can learn and is a self healing system. The work in [14] describe a production line digital twin where their metric the "overall equipment efficiency" is calculated from the sensor data collected. Here a data driven model is used. The work in [15] makes an attempt to bring out the framework for the DT engine. A three

layer concept is introduced and a basic implementation is attempted. There is no mention of real time data and action based on the "what-if" analysis. The work in [16] proposes a DT framework for health and well-being in smart cities. The idea here is to integrate X73 complaint personal health devices into the framework. The data analytics is limited to data driven models. The work in [17] describes the architecture of a digital twin laboratory. A multi layered model is proposed but no system building is shown. The work in [18] illustrate with an example the extension of a smart city monitoring system into a primitive DT.

In summary none of the works show construction of a DT from scratch for a given application using open source software. Our work advances the state of art by performing a "what-if" analysis on the data and performs an actuation on the physical system. Thus, our framework supports bidirectional, real time data based fully functional DTs.

VII. DISCUSSION AND CONCLUSION

Results from our DT framework facilitates the goal of this paper to establish feasibility of the DT technology using free and open-source software programs. Moreover, an application scenario for a portable table-top temperature-controlled chamber was demonstrated by having the mathematical model in the digital system predict the temperature profile of the physical model and also prevent a potentially catastrophic event by redirecting the physical system to safety.

However, in the current state, our system is unable to give more accurate results because of certain system limitations. Some of the limitations are as follows:

a) The OpenFOAM solver only takes in one input and computes one output before it is ready to take another input. We require the solver to take in multiple values and compute output in real time. Therefore, the PDE solver built into OpenFOAM is not efficient enough to reduce the overall computation cost.

b) In the simulation model, we can only compute one layer of heat transfer at a time. There is no control loop built into the SimFlow software that would allow us to run the hot subsystem and cold subsystem simultaneously. A separate setup is required to execute the same.

c) The UI Path Studio only works on windows based system while the Physical model was designed on a Linux based system. This increases the overall computation cost of the system as certain additional steps are required to establish prompt communication between the digital and physical systems.

In order to address these issues and make the system more efficient, a multi-physics model using open-source program like OpenModelica [19] could be constructed that would more accurately model the electrical and mechanical systems. Further, it could operate and automate the entire digital system in order to reduce the high computation cost. Additionally, a physical model with better thermal management can be designed to prevent significant energy losses.

VIII. REFERENCES

[1] https://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications/

[2] https://www.mouser.in/ProductDetail/Sensirion/SHT31-ARP-B25kS?qs=y6ZabgHbY

[3] https://www.openplcproject.com/

[4] https://www.eclipse.org/ditto/

[5] https://www.freecadweb.org/

[6] https://sim-flow.com/

[7] https://openfoam.org/

[8] https://www.paraview.org/

[9] https://www.uipath.com/product/studio

[10] F. Biesinger, D. Meike, B. Krass, and M. Weyrich, "A Case Study for a Digital Twin of Body-in-White Production Systems General Concept for Automated Updating of Planning Projects in the Digital Factory," 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2018.

[11] R. Zhao, D. Yan, Q. Liu, J. Leng, J. Wan, X. Chen, and X. Zhang, "Digital Twin-Driven Cyber-Physical System for Autonomously Controlling of Micro Punching System," IEEE Access, vol. 7, pp. 9459–9469, 2019.

[12] M. Borth, J. Verriet, and G. Muller, "Digital Twin Strategies for SoS 4 Challenges and 4 Architecture Setups for Digital Twins of SoS," 2019 14th Annual Conference System of Systems Engineering (SoSE), May 2019.

[13] W. Danilczyk, Y. Sun, and H. He, "ANGEL: An Intelligent Digital Twin Framework for Microgrid Security," 2019 North American Power Symposium (NAPS), Oct. 2019.

[14] C. Assawaarayakul, W. Srisawat, S. D. N. Ayuthaya, and S. Wattanasirichaigoon, "Integrate Digital Twin to Exist Production System for Industry 4.0," 2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), 2019.

[15] Z. Zhang, J. Lu, L. Xia, S. Wang, H. Zhang, and R. Zhao, "Digital twin system design for dual-manipulator cooperation unit," 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Jun. 2020.

[16] F. Laamarti, H. F. Badawi, Y. Ding, F. Arafsha, B. Hafidh, and A. E. Saddik, "An ISO/IEEE 11073 Standardized Digital Twin Framework for Health and Well-Being in Smart Cities," IEEE Access, vol. 8, pp. 105950–105961, 2020.

[17] M. Li, Y. Ma, Z. Yin, and C. Wang, "Structural Design of Digital Twin Laboratory Model Based on Instruments Sharing Platform," 2020 Chinese Control And Decision Conference (CCDC), Aug. 2020.

[18] F. Villanueva, O. Acena, J. Dorado, R. Cantarero, J. F. Bermejo, and A. Rubio, "On building support of digital twin concept for smart spaces," 2020 IEEE International Conference on Human-Machine Systems (ICHMS), Sep. 2020.

[19] https://www.openmodelica.org/

IX. APPENDIX I: MATHEMATICAL MODEL

The mathematical model is developed for the hot and cold subsystems as follows:

A. Hot Subsystem

For the hot subsystem, the heat is first transferred from the chip to the sink base through conduction, then from the sink base to the fins of the sink through conduction, then from the fins' tips to the air inside the chamber through convection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

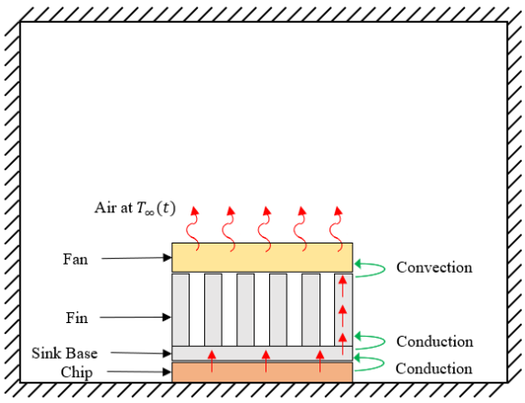

Figure 15: Hot subsystem heat transfer overview

We begin by evaluating the heat transfer from the chip to the heat sink base:

Using the Specific Heat formula, the amount of heat transfer from the chip to the sink is given by:

$$Q = \ m_{sink} \times C_{p,sink} \ \times T_{sink}$$

Here, m_{sink} is the mass of the sink, $C_{p,sink}$ is the specific heat of the sink, and T_{sink} is the temperature of the sink. Therefore, the rate of heat transfer can be given as:

$$dQ = \ m_{sink} \times C_{p,sink} \ \times dT_{sink}$$

$$\frac{dQ}{dt} = m_{sink} \times C_{p,sink} \times \frac{dT_{sink}}{dt} \tag{1}$$

Also, the rate of heat transfer as per the Fourier's Law is given by:

$$\frac{dQ}{dt} = (\frac{k_{sink} \times A_{contact}}{L_{sink}}) \times (T_{chip} - T_{sink}) \tag{2}$$

Here, k_{sink} is the thermal conductivity of the sink, $A_{contact}$ is the Area of surface contact between the chip and the sink base, L_{sink} is the thickness of the sink base, T_{chip} is the temperature of the chip, and T_{sink} is the temperature of the sink.

Equating equation (1) and equation (2),

$$m_{sink} \times C_{p,sink} \times \frac{dT_{sink}}{dt} = (\frac{k_{sink} \times A_{contact}}{L_{sink}}) \times (T_{chip} - T_{sink})$$

Rearranging and integrating with respect to time, t,

$$\int \frac{dT_{sink}}{T_{chip} - T_{sink}} = \frac{k_{sink} \times A_{contact}}{m_{sink} \times C_{p,sink} \times L_{sink}} \times \int dt$$

$$-ln(T_{chip} - T_{sink}) = \frac{k_{sink} \times A_{contact} \times t}{m_{sink} \times C_{p,sink} \times L_{sink}} + C \tag{3}$$

Evaluating boundary condition to find the integration constant, C,

When t = 0, equation (3) becomes

$$-ln(T_{chip} - T_{sink,0}) = C$$

Substituting back into equation (3),

$$-ln(T_{chip} - T_{sink}) = \frac{k_{sink} \times A_{contact} \times t}{m_{sink} \times C_{p,sink} \times L_{sink}} - ln(T_{chip} - T_{sink,0})$$

Rearranging the equation, we get the **Temperature of the sink base as a function of time** as:

$$\boxed{T_{sink}(t) = \frac{T_{sink,0} - T_{chip}}{e^{a_1 \times t}} + T_{chip}} \tag{4}$$

where a_1 is given by:

$$a_1 = \frac{k_{sink} \times A_{contact}}{m_{sink} \times C_{p,sink} \times L_{sink}}$$

We now evaluate the heat transfer from the sink base to fins, and from fins to the air:

The Fin Heat Transfer Rate for fins of finite length with heat loss via convection from the tips of fins is given by:

$$\frac{dQ}{dt} = N \times M \times \frac{\sinh(m \times L_{fins}) + \frac{h_{air}}{m \times k_{sink}} \times \cosh(m \times L_{fins})}{\cosh(m \times L_{fins}) + \frac{h_{air}}{m \times k_{sink}} \times \sinh(m \times L_{fins})} \tag{5}$$

where,

$$N = Total\ Number\ of\ fins$$

$$M = \sqrt{h_{air} \times P \times k_{sink} \times A_{cs}} \times (T_{sink}(t) - T_\infty)$$

$$P = \frac{A_{sa}}{L_{fin}}$$

$$m = \sqrt{\frac{h_{air} \times P}{k_{sink} \times A_{cs}}}$$

Here, L_{fins} is the length of the fins, h_{air} is the convection coefficient of air, k_{sink} is the thermal conductivity of the sink, A_{cs} is the cross sectional area of a fin, T_∞ is the air temperature inside the chamber, $T_{sink}(t)$ is the temperature of the sink base with respect to time (from equation (4)), and A_{sa} is the surface area for a fin.

Using the Specific Heat formula, the amount of heat transfer from the fin to the air is given by:

$$Q = \ m_{air} \times C_{p,air} \ \times T_\infty$$

Here, m_{air} is the mass of the air inside the chamber, $C_{p,air}$ is the specific heat of the air, and T_∞ is the air temperature inside the chamber. Therefore, the rate of heat transfer can be given as:

$$dQ = \ m_{air} \times C_{p,air} \ \times dT_\infty$$

$$\frac{dQ}{dt} = m_{air} \times C_{p,air} \times \frac{dT_\infty}{dt} \tag{6}$$

Equating equation (5) and equation (6),

$$m_{air} \times C_{p,air} \times \frac{dT_\infty}{dt} = N \times M \times a_2$$

where,

$$a_2 = \frac{\sinh(m \times L_{fins}) + \frac{h_{air}}{m \times k_{sink}} \times \cosh(m \times L_{fins})}{\cosh(m \times L_{fins}) + \frac{h_{air}}{m \times k_{sink}} \times \sinh(m \times L_{fins})}$$

Rearranging and integrating with respect to time, t,

$$\int \frac{dT_\infty}{M} = \frac{N}{m_{air} \times C_{p,air}} \times a_2 \times \int dt$$

Let the constants in M be a_3,

$$a_3 = \sqrt{h_{air} \times P \times k_{sink} \times A_{cs}}$$

Therefore,

$$\int \frac{dT_\infty}{a_3 \times (T_{sink}(t) - T_\infty)} = \frac{N}{m_{air} \times C_{p,air}} \times a_2 \times \int dt$$

$$\frac{-ln(T_{sink}(t) - T_\infty)}{a_3} = \frac{N}{m_{air} \times C_{p,air}} \times a_2 \times t + C \tag{7}$$

Evaluating boundary condition to find the integration constant, C,

When t = 0, equation (7) becomes:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

$$\frac{-ln(T_{sink}(t)-T_{\infty,0})}{a_3}=C$$

Substituting back into equation (7),

$$\frac{-ln(T_{sink}(t)-T_{\infty})}{a_3}=\frac{N}{m_{air}\times C_{p,air}}\times a_2\times t-\frac{ln(T_{sink}(t)-T_{\infty,0})}{a_3}$$

$$ln(\frac{T_{sink}(t)-T_{\infty,0}}{T_{sink}(t)-T_{\infty}})=\frac{a_3\times N}{m_{air}\times C_{p,air}}\times a_2\times t$$

Let,

$$\frac{a_3\times N}{m_{air}\times C_{p,air}}\times a_2=a_4$$

Rearranging the equation, the **Temperature of the Air inside the chamber as a function of time** is given as:

$$\boxed{T_{\infty}(t)=T_{sink}(t)-\frac{T_{sink}(t)-T_{\infty,0}}{e^{a_3\times t}}} \quad (8)$$

B. Cold Subsystem For the cold subsystem, the heat is first transferred from the chip to the aluminium plate through conduction, then from the aluminium plate to the air through convection

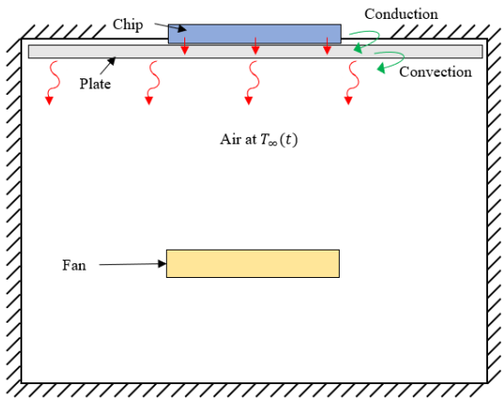

Figure 16: Cold subsystem heat transfer overview

We begin by evaluating the heat transfer from the chip to the aluminium plate:

Using the Specific Heat formula, the amount of heat transfer from the chip to the plate is given by:

$$Q=\ -m_{plate}\times C_{p,plate}\ \times T_{plate}$$

Here, m_{plate} is the mass of the plate, $C_{p,plate}$ is the specific heat of the plate, and T_{plate} is the temperature of the plate. Since the heat is absorbed, there is a negative sign. Therefore, the rate of heat transfer can be given as:

$$dQ=\ -m_{plate}\times C_{p,plate}\ \times dT_{plate}$$

$$\frac{dQ}{dt}=-m_{plate}\times C_{p,plate}\times\frac{dT_{plate}}{dt} \quad (9)$$

Also, the rate of heat transfer as per the Fourier's Law is given by:

$$\frac{dQ}{dt}=-(\frac{k_{plate}\times A_{contact}}{L_{plate}})\times(T_{chip}-T_{plate}) \quad (10)$$

Here, k_{plate} is the thermal conductivity of the plate, $A_{contact}$ is the Area of surface contact between the chip and the plate, L_{plate} is the thickness of the plate, T_{chip} is the temperature of the chip, and T_{plate} is the temperature of the plate.

Equating equation (9) and equation (10),

$$m_{plate}\times C_{p,plate}\times\frac{dT_{plate}}{dt}=(\frac{k_{plate}\times A_{contact}}{L_{plate}})\times(T_{chip}-T_{plate})$$

Rearranging and integrating with respect to time, t,

$$\int\frac{dT_{plate}}{T_{chip}-T_{plate}}=\frac{k_{plate}\times A_{contact}}{m_{plate}\times C_{p,plate}\times L_{plate}}\times\int dt$$

$$-ln(T_{chip}-T_{plate})=\frac{k_{plate}\times A_{contact}\times t}{m_{plate}\times C_{p,plate}\times L_{plate}}+C \quad (11)$$

Evaluating boundary condition to find the integration constant, C,

When t = 0, equation (11) becomes

$$-ln(T_{chip}-T_{plate,0})=C$$

Substituting back into equation (11),

$$-ln(T_{chip}-T_{plate})=\frac{k_{plate}\times A_{contact}\times t}{m_{plate}\times C_{p,plate}\times L_{plate}}-ln(T_{chip}-T_{plate,0})$$

Rearranging the equation gives the **Temperature of the plate as a function of time**,

$$\boxed{T_{plate}(t)=\frac{T_{plate,0}-T_{chip}}{e^{a_5\times t}}+T_{chip}} \quad (12)$$

where a_5 is given by,

$$a_5=\frac{k_{plate}\times A_{contact}}{m_{plate}\times C_{p,plate}\times L_{plate}}$$

We now evaluate the heat transfer from the plate to air:

Using the Specific Heat formula, the amount of heat transfer from the plate to the air is given by:

$$Q=\ -m_{air}\times C_{p,air}\ \times T_{\infty}$$

Here, m_{air} is the mass of the air inside the chamber, $C_{p,air}$ is the specific heat of the air, and T_{∞} is the temperature of the plate. Since the heat is absorbed, there is a negative sign. Therefore, the rate of heat transfer can be given as:

$$dQ=\ -m_{air}\times C_{p,air}\ \times dT_{\infty}$$

$$\frac{dQ}{dt}=-m_{air}\times C_{p,air}\times\frac{dT_{\infty}}{dt} \quad (13)$$

Also, the rate of heat transfer as per Newton's Law of Cooling is is given by:

$$\frac{dQ}{dt}=-h_{air}\times A_{sa}\times(T_{plate}(t)-T_{\infty}) \quad (14)$$

Here, h_{air} is the heat transfer coefficient of the air, A_{sa} is the surface area of the plate exposed to the air, $T_{plate}(t)$ is the temperature of the plate with respect to time (given by equation (12), and T_{∞} is the temperature of the air inside the chamber.

Equating equation (13) and equation (14),

$$-m_{air}\times C_{p,air}\times\frac{dT_{\infty}}{dt}=-h_{air}\times A_{sa}\times(T_{plate}(t)-T_{\infty})$$

Rearranging and integrating with respect to time, t,

$$\int\frac{dT_{\infty}}{(T_{plate}(t)-T_{\infty})}=\frac{h_{air}\times A_{sa}}{m_{air}\times C_{p,air}}\times\int dt$$

$$-ln(T_{plate}(t)-T_{\infty})=\frac{h_{air}\times A_{sa}\times t}{m_{air}\times C_{p,air}}+C \quad (15)$$

Evaluating boundary condition to find the integration constant, C,
When t = 0, equation (15) becomes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIC.2021.3051798, IEEE Internet Computing

$$-ln(T_{plate}(t) - T_{\infty,0}) = C$$

Substituting back into equation (15),

$$-ln(T_{plate}(t) - T_{\infty}) = \frac{h_{air} \times A_{sa} \times t}{m_{air} \times C_{p,air}} - ln(T_{plate}(t) - T_{\infty,0})$$

$$ln(\frac{T_{plate}(t) - T_{\infty,0}}{T_{plate}(t) - T_{\infty}}) = \frac{h_{air} \times A_{sa} \times t}{m_{air} \times C_{p,air}}$$

Let,

$$\frac{h_{air} \times A_{sa}}{m_{air} \times C_{p,air}} = a_6$$

Rearranging the equation gives the **Temperature of the Air inside the chamber as a function of time**,

$$\boxed{T_{\infty}(t) = T_{plate}(t) - \frac{T_{plate}(t) - T_{\infty,0}}{e^{a_6 \times t}}} \quad (16)$$