UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TIAGO POSSATO

CONTROLANDO UMIDADE RELATIVA EM CÂMARAS FRIAS: MELHORIAS NO SISTEMA DE MEDIÇÃO E UMA ESTRATÉGIA DE CONTROLE BASEADA EM SISTEMAS DE EVENTOS DISCRETOS

PATO BRANCO

2024

TIAGO POSSATO

CONTROLANDO UMIDADE RELATIVA EM CÂMARAS FRIAS: MELHORIAS NO SISTEMA DE MEDIÇÃO E UMA ESTRATÉGIA DE CONTROLE BASEADA EM SISTEMAS DE EVENTOS DISCRETOS

Controlling Relative Humidity in Cold Chambers: Improvements in the Measurement System and a Control Strategy Based on Discrete Event Systems

Dissertação apresentada como requisito para obtenção do título de Mestre em Engenharia Elétrica do Programa de Pós-graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Jean Patrick da Costa

Coorientador: Prof. Dr. Marcelo Teixeira

PATO BRANCO
2024

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Pato Branco

TIAGO POSSATO

CONTROLANDO UMIDADE RELATIVA EM CÂMARAS FRIAS: MELHORIAS NO SISTEMA DE MEDIÇÃO E UMA ESTRATÉGIA DE CONTROLE BASEADA EM SISTEMAS DE EVENTOS DISCRETOS

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas E Processamento De Energia.

Data de aprovação: 13 de Dezembro de 2023

Dr. Jean Patric Da Costa, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Cesar Rafael Claure Torrico, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Marcelo Teixeira, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Richardson Ribeiro, Doutorado - Universidade Federal do Paraná (Ufpr)

À toda minha família.

ACKNOWLEDGEMENTS

À minha família, pela paciência e compreensão nos momentos em que minha dedicação à este trabalho demandou mais tempo e energia. Sua compreensão foi crucial para o sucesso alcançado. Estendo meus agradecimentos ao meus orientadores pelos valorosos conselhos e ajuda.

Ao Instituto Federal Catarinense (IFC), pela oportunidade de estar cursando este Mestrado. À UTFPR e a todos os técnicos administrativos e professores pelo apoio e ensinamentos.

Agradecimento especial ao Dr. Luiz Carlos Argenta, da Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina (EPAGRI), Carlos Roberto Rodrigues Gois, do INSTITUTO DE DESENVOLVIMENTO RURAL DO PARANÁ (IAPAR-EMATER) e Tiago Nunes, da Unigranjas Comercial Agrícola e Pecuária Ltda, pelo apoio nos experimentos com sensores. Agradeço também ao Mateus Hanser de Andrade e ao Mateus Zanini pelo apoio e parceria.

Este estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001, do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela Fundação Araucária (FA), pela Financiadora de Estudos e Projetos (FINEP) e pelo Instituto Federal Catarinense (IFC).

RESUMO

POSSATO, Tiago. Controlando Umidade Relativa em Câmaras Frias: Melhorias no Sistema de Medição e uma Estratégia de Controle Baseada em Sistemas de Eventos Discretos. 2024. 65 f. Dissertação. Programa de Pós-Graduação em Engenharia Elétrica. Pato Branco. 2024.

O cultivo da maçã é uma atividade agroindustrial que depende essencialmente da existência de câmaras frias para armazenar as frutas após a colheita, sob temperaturas em torno de 1°C e umidade relativa (UR) entre 92 e 95%rh. O controle de ambientes como esses é desafiador devido ao tamanho da câmara, suas entradas e zonas distintas e, especialmente, devido ao efeito que a temperatura e a UR impõem uma sobre a outra quando controladas em conjunto. Normalmente, espera-se que os ciclos de refrigeração associados a essas condições causem condensação no sensor de UR, afetando o monitoramento e o controle da UR. Este trabalho propõe uma estratégia para melhorar o controle da UR em câmaras frias. Primeiro, apresenta um novo Sistema de Medição (MS) que tolera altas taxas de UR, eventos de condensação e baixa ventilação. O sistema encapsula os dispositivos de medição em uma estrutura fechada que é controlada automaticamente para detectar a saturação e responder a ela adequadamente. Em seguida, com base no mapa de eventos observável a partir do MS proposto, surge um modelo formal orientado a eventos como forma de representar logicamente o sistema real, bem como suas regras para a coordenação esperada sob controle. Esse modelo é processado usando a Teoria de Controle Supervisório (TCS) de Sistemas de Eventos Discretos (SEDs) e leva a uma estratégia de controle que capta efetivamente a relação entre refrigeração e UR. O controlador é então convertido em código C usando uma ferramenta de geração automatizada, cujo desenvolvimento também faz parte das contribuições deste documento. São realizados experimentos para mostrar que o MS proposto reduz o tempo de resposta do sensor sob baixa ventilação e melhora a precisão da detecção de variações na UR. Além disso, a estratégia de controle, em conjunto com o MS, permite que o controlador antecipe as variações de UR, garantindo um controle mais preciso do processo e reduzindo o erro quadrático médio (RMSE) e o desvio padrão.

Palavras-chave: umidade relativa; medição de umidade; alta umidade; controle de sed; câmara fria.

ABSTRACT

POSSATO, Tiago. Controlling Relative Humidity in Cold Chambers: Improvements in the Measurement System and a Control Strategy Based on Discrete Event Systems. 2024. 65 f. Dissertação. Programa de Pós-Graduação em Engenharia Elétrica. Pato Branco. 2024.

Apple cultivation is an agroindustrial activity that depends essentially on the existence of cold chambers to store fruits after harvest, under temperatures around 1°C and Relative Humidity (RH) between 92 and 95%rh. Controlling environments like these is challenging because of the size of the chamber, its entrances, and distinct zones, and especially because of the effect that temperature and RH impose on each other when orchestrated in conjunction. Typically, the refrigeration cycles associated with these conditions are expected to cause condensation in the RH sensor, affecting the monitoring and control of RH. This work proposes a strategy to improve RH control in cold chambers. It first introduces a novel Measurement System (MS) that tolerates high RH rates, condensation events, and low ventilation. The system encapsulates the measurement devices in a closed structure that is automatically controlled to detect saturation and respond to it accordingly. Then, based on the event map observable from the proposed MS, a formal event-driven model emerges as a way to represent the real system logically, as well as its rules for expected coordination under control. This model is processed using the Supervisory Control Theory (SCT) of Discrete Event Systems (DESs) and leads to a control strategy that effectively captures the relationship between refrigeration and relative humidity. The controller is then converted into C code using an automated generation tool, whose development is also part of the contributions of this paper. Experiments are conducted to show that the proposed MS reduces the sensor response time under low ventilation and improves the accuracy of detecting variations in RH. Furthermore, the control strategy, in conjunction with MS, enables the controller to anticipate variations of RH, ensuring more precise process control and reduced Root Mean Squared Error (RMSE) and standard deviation.

Keywords: relative humidity; humidity measurement; high humidity; des control; cold chamber.

LIST OF FIGURES

CONTENTS

1 INTRODUCTION

The southern region of Brazil is responsible for almost all the apple production in the country, which in 2022 generated a revenue of more than US$388 million (IBGE, 2022). Apple producers sort and store apples in cold chambers after harvest. This practice prolongs their shelf life, delaying senescence and the ripening process (ANESE *et al.*, 2016; AKDEMIR; BAL, 2020; ARGENTA *et al.*, 2021). When properly handled, some apple cultivars can be stored for up to 10 months (ANESE; FRONZA, 2015). There are techniques to maintain fruit quality during storage, such as applying preservatives, controlling atmospheric gases in the chamber (ARGENTA *et al.*, 2022), and controlling the temperature and Relative Humidity (RH) of the air (ZHU *et al.*, 2023). However, post-harvest losses can still reach the order of 28% of total production (AKDEMIR; BAL, 2020; ARGENTA *et al.*, 2021), justifying attempts to optimize the storage process.

Temperature is a key factor in keeping the quality of stored apples (ANESE; FRONZA, 2015; LUFU; AMBAW; OPARA, 2020; SCHUDEL *et al.*, 2023). However, the temperature must be considered together with the RH level, which is one of the leading apple conservation strategies to reduce losses and increase storage time (LUFU; AMBAW; OPARA, 2020; SCHUDEL *et al.*, 2023). The appropriate RH for apple conservation ranges from 92 to 95%rh (WEBER *et al.*, 2012; ANESE; FRONZA, 2015; BRACKMANN *et al.*, 2016; LINKE *et al.*, 2023). RH outside ideal levels can cause wilting, yellowing, and weight loss and stimulate the appearance of rot, cracking, and other physiological disorders (LUFU; AMBAW; OPARA, 2020; ARGENTA *et al.*, 2021; ZHU *et al.*, 2023; SCHUDEL *et al.*, 2023).

Water loss occurs by the difference between the water vapor pressures on the inner surface of the apple and those in the surrounding air (AKDEMIR; BAL, 2020; LUFU; AMBAW; OPARA, 2020). Reducing temperature and controlling RH are the main ways to reduce this difference (AKDEMIR; BAL, 2020; LUFU; AMBAW; OPARA, 2020; BISHNOI; AHARWAL, 2022). The RH should be kept between 92 and 95%rh, depending on the cultivar stored (WEBER *et al.*, 2012; BRACKMANN *et al.*, 2016; AKDEMIR; BAL, 2020; LINKE *et al.*, 2023). If RH is not sufficient, the product can lose water, reducing mass and causing deterioration, nutritional loss, and internal degradation (WEBER *et al.*, 2012; AKDEMIR; BAL, 2020; LUFU; AMBAW; OPARA, 2020; ZHU *et al.*, 2023; SCHUDEL *et al.*, 2023). Furthermore, it cannot remain above the maximum values because it provides excellent conditions for the growth of microorganisms and causes cracking and flesh breakdown, among other diseases (WEBER *et al.*, 2012; LUFU; AMBAW; OPARA, 2020; LINKE *et al.*, 2023; ZHU *et al.*, 2023; SCHUDEL *et al.*, 2023). The effects of apple storage in low and high humidities can be seen in Figure 1.

Under ideal apple storage conditions, surfaces, including RH sensors, are expected to experience water condensation (dew formation) (LINKE *et al.*, 2021; PRAEGER *et al.*, 2021b; LINKE *et al.*, 2023). The sensor can become saturated, resulting in measurement errors and failures in the RH control system (HMSO, 1996; HU *et al.*, 2020; ZANELLA; SILVA; AL-

Figure 1 – Apples stored under bad RH conditions. In A, apples wilted and yellowed due to low RH. In B and C, cracking, rotting, and microorganism spots are caused by high RH.

Source: Adapted from Anese e Fronza (2015).

BINI, 2022). Another challenge is low ventilation in the chamber (PRAEGER *et al.*, 2021a; BISHNOI; AHARWAL, 2022), which can reduce the sensor response time. To address these problems, an initial version of a RH *MS* was proposed (POSSATO; TEIXEIRA; COSTA, 2023), using a commercial sensor integrated within a plastic housing and a fan system. The aim of the fan was to ensure constant, uninterrupted ventilation, hopefully improving response time. At the same time, the firmware was programmed to read the sensor and detect saturation episodes, thus activating the internal heater to evaporate the condensed water.

The proposed *MS* can be integrated with commercial RH controllers for practical purposes, which generally operate in the on-off mode with hysteresis, observing only the RH value to activate the humidifier. On-off control is required because humidifiers work with a pump and sprinkler nozzles that operate with a fixed water pressure. However, the number of signals observed in the plant can be increased by observing more details about the operation of the refrigeration system since it is the key to decrease RH inside the cold chamber (SHEN; WANG, 2020; ZHU *et al.*, 2023). By having available a more detailed plant, its interaction with RH can be more efficient, which is represented here by the proposal of a novel control strategy that subsumes and extends substantially the approach in Possato, Teixeira e Costa (2023). In this new strategy, the signals observable in the plant are seen as events part of a Discrete Event System (DES) model.

DESs is a class of systems that evolve based on the asynchronous occurrence of events at irregular time intervals (CASSANDRAS; LAFORTUNE, 2021), and the usual way to design them is by using diagram-based models, such as Finite State Machines (FSMs). An FSM that designs a DES model can be further processed using Supervisory Control Theory (SCT) (RAMADGE; WONHAM, 1989b) to result in a controller that holds some properties of interest, such as least restrictiveness, recognizes the controllability of events and does not ever block. In the literature, FSMs has been used, for example, to make control decisions in agro-industrial (LORENCENA *et al.*, 2020; SCHMIDT *et al.*, 2021) and factory automation (ZIELINSKI *et al.*, 2021).

When applied on industrial-scale DESs, FSMs may face limitations in the design, synthesis, simulation, test, and implementation steps. The Supremica tool (MALIK *et al.*, 2017; AKESSON *et al.*, 2023) is a straightforward alternative to address most of these limitations, but it does not reach implementation. This requires additional tools to convert FSM models into code. To address this, the DEScMaker tool Possato *et al.* (2024) was developed as an important complement to this work. The DEScMaker receives as input an FSM model outputting from Supremica and converts it into implementable C code.

In summary, this work proposes ways to improve RH control in cold chambers and adds the following main contributions to the literature: the previous RH *MS* is extended, improved, and tested with calibrated equipment; the new version of the plant is modeled using FSM, and the rules that guide the interaction among the plant components are also modeled and integrated with the plant; the synthesis procedure is conducted and this leads to a DES-based strategy to control RH in cold chambers; the tool for automated code generation in Possato *et al.* (2024) is tested as part of the implementation phase of the control system; the controller is integrated into a real environment, and its performance is observed and discussed.

This dissertation is presented in four main sections: The RH *MS* is presented in Chapter 2, including the theoretical background, a review of the literature on RH measurement, the experiments carried out and the results obtained. The Chapter 3 contains a discussion of the control methods applicable in the context of this work, the theory of DES control and supervisory control, and the modeling of the proposed control. The tool developed to generate code from a FSM is shown in the Chapter 4. The Chapter 5 brings the complete implementation of the proposal, with the integration of the *MS* code and the control strategy obtained with the code generation tool. Finally, conclusions and suggestions for future work are given in Chapter 6.

2 PROPOSAL OF A RH MS

This Chapter explains the development and validation of a RH *MS* for cold chambers that can handle storage conditions inside cold chambers, avoiding sensor saturation due to high RH and temperature variations. First, the theoretical basis and literature review on RH measurement is presented. This is followed by the development of the *MS* with two complementary strategies to improve RH measurement in this environment. The first strategy is maintaining constant airflow through the sensor by installing it in a plastic housing with a fan. The second strategy focuses on identifying and recovering from saturation. The assembly procedures and the method for determining saturation are described. Finally, the methods for experimental validation of the proposal and the results found are presented.

2.1 Relative Humidity of the Air

The apple and other fresh fruits and vegetables are stored in cold rooms, where refrigerated air regulates their temperature. Air is a mixture of gases. However, atmospheric air always contains moisture, essentially superheated vapor at low temperatures and partial pressure (SINGH; HELDMAN, 2014). The mass of water vapor in a specific volume of dry air is expressed by the *absolute humidity*. Its unit of measurement in the International System of Units (SI) is grams of water per cubic meter of air (g/m^3) (HMSO, 1996).

However, the capacity of air to absorb water is not fixed and depends mainly on temperature. Generally, the hotter the air, the more water vapor it can hold. For example, air saturation at 0°C is reached with an absolute humidity of $4.84g/m^3$, while at 5°C, this value increases to $6.80g/m^3$. Considering a sample of air with constant absolute humidity, the temperature at which this sample becomes saturated is called the *dew point*. This is when dew, or water condensation, forms. Its unit of measurement is the degree Celsius (°C) (HMSO, 1996). The *Relative Humidity (RH)* of air is the magnitude that measures the absolute humidity relative to the maximum amount of saturated air at the same temperature. Therefore, the RH of the air express, as a percentage (%rh), how saturated a sample of air is at the current temperature (HMSO, 1996).

The condensation of water vapor on the coil is the primary source of air humidity reduction. The difference between the air and the coil surface temperature causes dew formation. The dew freezes in the coil (SHEN; WANG, 2020) and does not return to the environment. The defrost process melts the ice and drains it out. The humidification systems then reload the lost air moisture (MOHAMMED; ALQAHTANI; EL-SHAFIE, 2021). However, the post-harvest moisture loss in apples is irreversible (BISHNOI; AHARWAL, 2022). Therefore, the humidifier must identify the reduction in RH and act immediately to prevent the difference between the water vapor pressures. Modern equipment uses RH sensors for this purpose. This sensor is an essential part of the system because it provides the information for control, which activates the

actuator to insert steam or water droplets into the environment (MOHAMMED; ALQAHTANI; EL-SHAFIE, 2021).

There are several ways to measure RH, including hygrometers with electronic sensors. These sensors, in turn, are manufactured with different materials and techniques, which will be presented in the next section.

2.2 Measurement of Relative Humidity

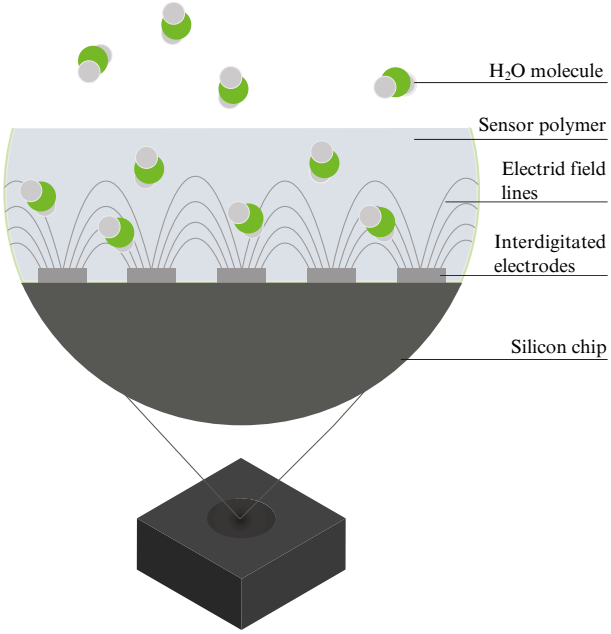
Hygrometers are used to measure RH. These devices can use mechanical sensors (HMSO, 1996), dry and wet bulb sensors (HMSO, 1996), chilled mirror sensors (HMSO, 1996), or electronic sensors (FARAHANI; WAGIRAN; HAMIDON, 2014; BLANK; EKSPERIANDOVA; BELIKOV, 2016; KUZUBASOGLU, 2022; MEMON *et al.*, 2022). Electronic sensors are made of organic or inorganic materials, which allow measurement of the response to changes in RH through an electronic circuit (KUZUBASOGLU, 2022; BLANK; EKSPERIANDOVA; BELIKOV, 2016).

RH sensors used in a humidifier are electronic devices developed based on different working principles and various hygroscopic sensing materials (FARAHANI; WAGIRAN; HAMIDON, 2014). The technology of sensing materials is such as lithium chloride electrolyte, ceramic, graphene, and polymer (FARAHANI; WAGIRAN; HAMIDON, 2014; BLANK; EKSPERIANDOVA; BELIKOV, 2016; KUZUBASOGLU, 2022; BAI *et al.*, 2022; LIU *et al.*, 2023; GINJA *et al.*, 2023). The transduction types are such as resistive, capacitive, reflectance, frequency, color change, and molecular rectifiers (FARAHANI; WAGIRAN; HAMIDON, 2014; BLANK; EKSPERIANDOVA; BELIKOV, 2016; MEMON *et al.*, 2022; KUZUBASOGLU, 2022; LIU *et al.*, 2023; SULLIVAN *et al.*, 2023; GINJA *et al.*, 2023). Finally, the sensing mechanism is such an electronic, protonic, electrolytic, and Grotthus principle (FARAHANI; WAGIRAN; HAMIDON, 2014; BLANK; EKSPERIANDOVA; BELIKOV, 2016; KUZUBASOGLU, 2022; LIU *et al.*, 2023; GINJA *et al.*, 2023).

All electronic sensors use changes in sensitive elements' physical and electrical properties. Commercial sensors typically comprise a material with a porous structure that can adsorb and desorb water vapor molecules, which undergo physical and electrical changes proportional to RH (FARAHANI; WAGIRAN; HAMIDON, 2014; GINJA *et al.*, 2023). Figure 2 shows the cross-section of a commercial polymer sensor, indicating the electrodes for detecting changes in the electric field caused by water molecules (SENSIRION, 2022).

How the humidifier controller reacts to variations in RH can follow several strategies, as presented in the literature review. Among the mentioned strategies, using DES is an alternative, so its concepts will be presented next.

Figure 2 – Cross-section of a commercial sensor. The water molecules accumulate on the surface of the porous polymer structure and interfere with the electric field lines that change the material's capacitance. The RH is determined by combining the capacitance and temperature measurements.



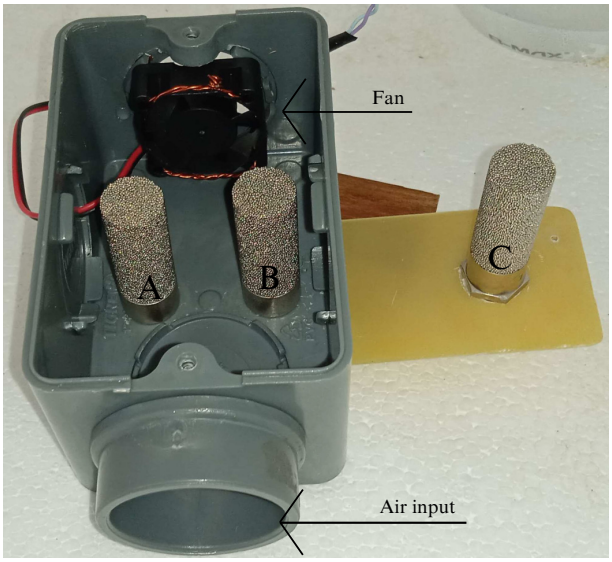
Source: Adapted from (SENSIRION, 2022)..

2.3 Assembly of the Measurement System

The system comprises a set of a RH sensor, a fan, and a plastic housing. The sensor was installed in the plastic housing with entrances on two opposite sides to ensure adequate ventilation. The fan was installed on one side to provide the minimum airflow velocity through the sensor. The fan removes the air from the housing, preventing self-heating from interfering with the measurements. The airspeed is constant and was adjusted close to 1m/s using an anemometer (Instrutherm AD-250, São Paulo, Brazil). Figure 3 demonstrates the assembly of the prototype. The sensor used (indicated as **A**) is a model SHT31-DIS-F with a measurement range of 0 to 100%rh, a tolerance of $\pm$2%rh over the entire range, and a time constant of 8s. It also has a membrane with a thickness of 100μm to prevent pollution and water from hitting the sensor directly and an internal heater to eliminate condensed water. The figure also shows the sensors **B** and **C**, presented in Subsection 2.5.

The sensors are connected to an ESP32 microcontroller (Espressif Systems, Shanghai, China). The measurement strategy is implemented as a dedicated task, and the firmware is developed with the FreeRTOS operating system. The acquisition frequency is 3Hz, and it calculates a moving average filter of order three every 1s to decrease measurement noise. The resulting components form the proposed *Measurement System (MS)*, including the sensor, fan, plastic housing, microcontroller, and firmware that implements the measurement strategy.

Figure 3 – Prototype of the system without the lower plastic case cover. Also visible are the three sensors and the mounted fan. The arrows indicate the air input and the fan.



Source: Author (2023).

2.3.1 Improvements in the RH MS

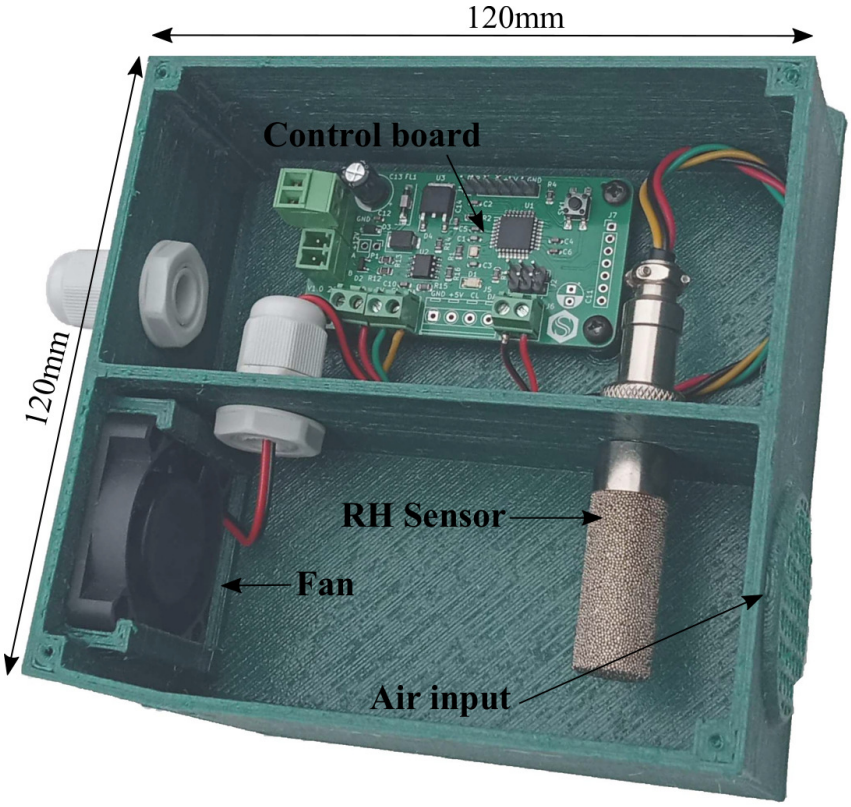
The upgraded *MS* comprises a set of a RH sensor, model SHT31-DIS-F (Sensirion AG, Staefa, Switzerland), a control board, a fan, and a plastic housing. The sensor has an internal heater activated by a command. The plastic housing has entrances on two opposite sides to ensure ventilation. The fan is installed on one side to provide the minimum airflow through the sensor. The complete *MS* is shown in Figure 4. The electronic board includes a power supply, an RS485 communication interface, a microcontroller, and connection terminals. The RH and temperature values can be read through the serial interface using the MODBUS RTU protocol [1].

2.4 Strategy for Detection and Recovery from Saturation

The firmware to read and monitor the RH sensor is embedded in the electronic board. The operation is summarized in Figure 5. The acquisition frequency is 3Hz, and a moving average filter of order three is calculated every 1s to decrease measurement noise. The standard deviation (σ) is calculated, and the sensor is saturated when $\sigma < \varepsilon$, where ε is a value for comparison. With the sensor saturated, the recovery procedure is started. The current (initial) RH is stored in v_i, and the controller sends a command to activate the heater. When the current RH value v decreases by δ, the heater is turned off and an interval α is awaited to stabilize the sensor readings. Finally, the system returns to normal.

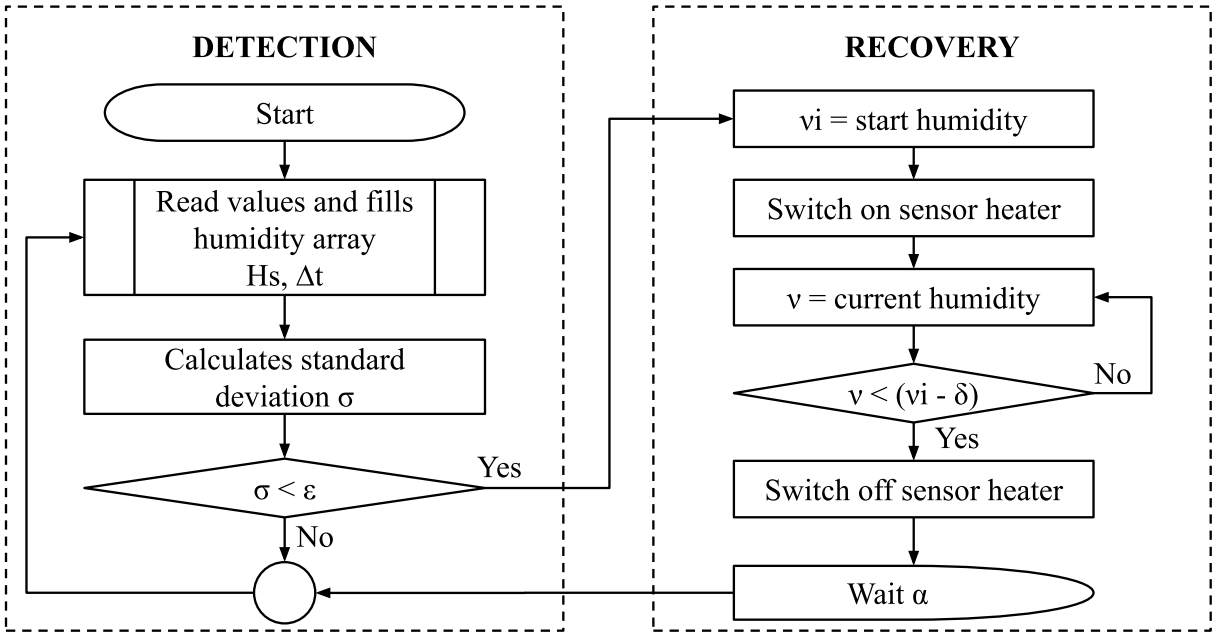
[1] https://modbus.com/specs.php

Figure 4 – Final version of the *MS*. Plastic housing without cover, sensor, control board, and fan.



Source: Author (2023).

Figure 5 – Flowchart of the method for reading RH, detecting and recovering saturation.



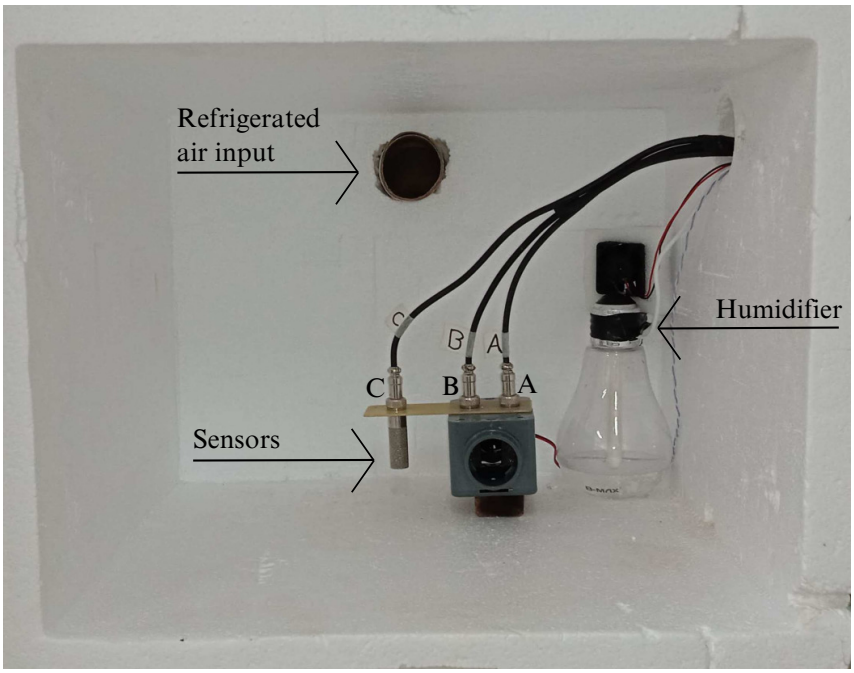
Source: Author (2023).

2.5 Validation

As shown in Figure 3, two additional sensors were used to validate the proposal. The sensor **B** is installed with the sensor of the system (**A**) and is used as a reference to compare the results. It is installed under ventilation without using the internal heater. This setup allows comparison between the data collected on sensors **A**, **B**, and **C**. The sensor **C** is installed outside the housing to measure RH under the same conditions as used in commercial cold chambers, exposed to the natural ventilation variations of the chamber.

The proposed method was validated by exposing the sensors to RH above 90%rh. The prototype was installed in a cold test chamber with a temperature between 0.5 and 5°C. The RH of the air was controlled using an ultrasonic air humidifier. Figure 6 shows the test setup.

Figure 6 – Setup assembled inside the test chamber. Sensors mounted below the refrigerated air input to replicate real-world systems. The humidifier has a fan to mix the humid air evenly throughout the chamber.



Source: Author (2023).

Three experiments were carried out in order to validate the *MS* prototype, compare the sensors, simulate humidity condensation events and apply the proposed system. Experiment 1 compares the three sensors under the same controlled conditions. Experiment 2 considers the setup of the *MS*, integrating the three sensors from the previous experiment. Finally, Experiment 3 aims to investigate the impact of the method and the proposed *MS* on the control of RH. Data were collected every 2 seconds for all experiments. The analysis calculates the difference between the values and the Pearson correlation coefficient (r).

In turn, the improved version of the *MS* was installed in a cold room under commercial apple storage conditions. A commercial RH and temperature meter, model HMT337 (Vaisala, Vantaa, Finland), was used as a reference (**standard**) for comparison. The standard was cali-

brated in a specialized laboratory (ISO/IEC 17025), and the maximum measurement uncertainty found was ±1.1%rh. The acquisition interval of the standard is 1 minute, which is sufficient to monitor apples during their months-long storage period. A sensor (sensor B) of the same model used in *MS* was installed next to the standard for comparison.

The experiments for the improved *MS* were carried out by installing the sensors under the same conditions (Figure 7) to verify their operation and the correlation of the values. The proposed *MS* was then installed (Figure 8), and the data was collected and analyzed. Sensor saturation tests were also carried out by opening the chamber door to cause a rapid change in temperature (CARNEIRO; GASPAR; SILVA, 2017), causing condensation on the sensors. The following sections present the results and discuss the investigations.

Figure 7 – Setup of the sensor verification. On the right is a photo of the test chamber, installed in a cold chamber. On the left is a detail of the sensor installation.

Source: Author (2023).

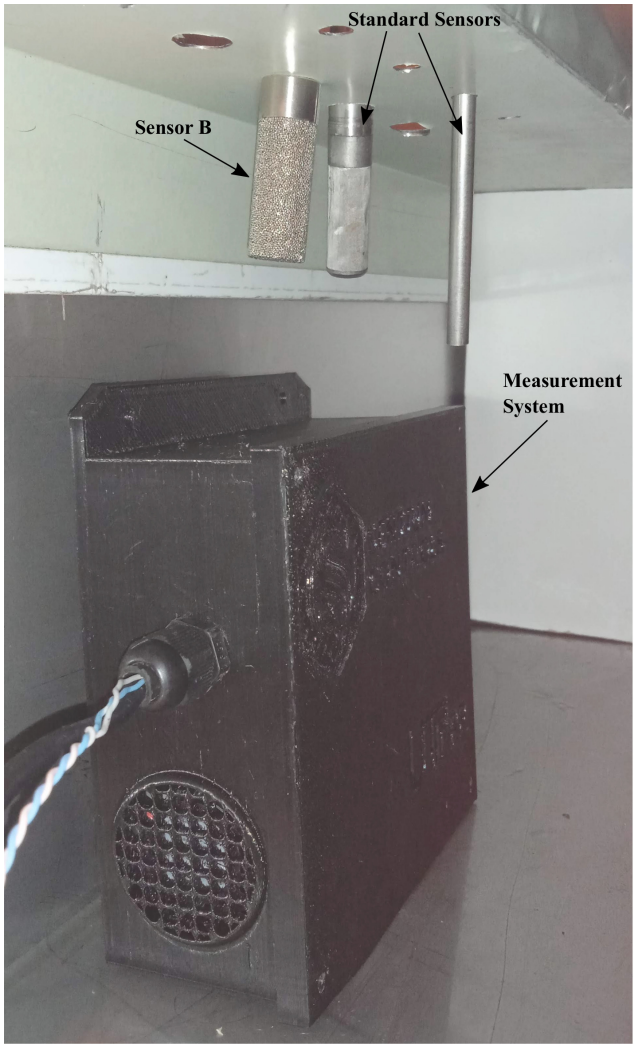
2.6 MS Results and Discussion

2.6.1 Experiment 1: Checking the sensors

This experiment checks the three sensors before mounting the *MS*. They were installed under the same temperature (2.4°C±0.5), RH (94.8%rh ±0.81), and ventilation conditions (1m/s). The data presented in Figure 9 are relative to an interval of 30min (n=900). The RH does not remain stable due to temperature control. However, this variation is positive for the study as it is possible to analyze the response time of the sensors.

The difference in sensor **A** with respect to sensor **B** is 0.65%rh ±0.49. Sensor **C** shows a difference of 0.91%rh ±0.18 compared to sensor **B**. This variation in the difference between the sensors shows different response times. In the range shown, the sensor **A** reaches the maximum

Figure 8 – Setup of the *MS* verification.



Source: Author (2023).

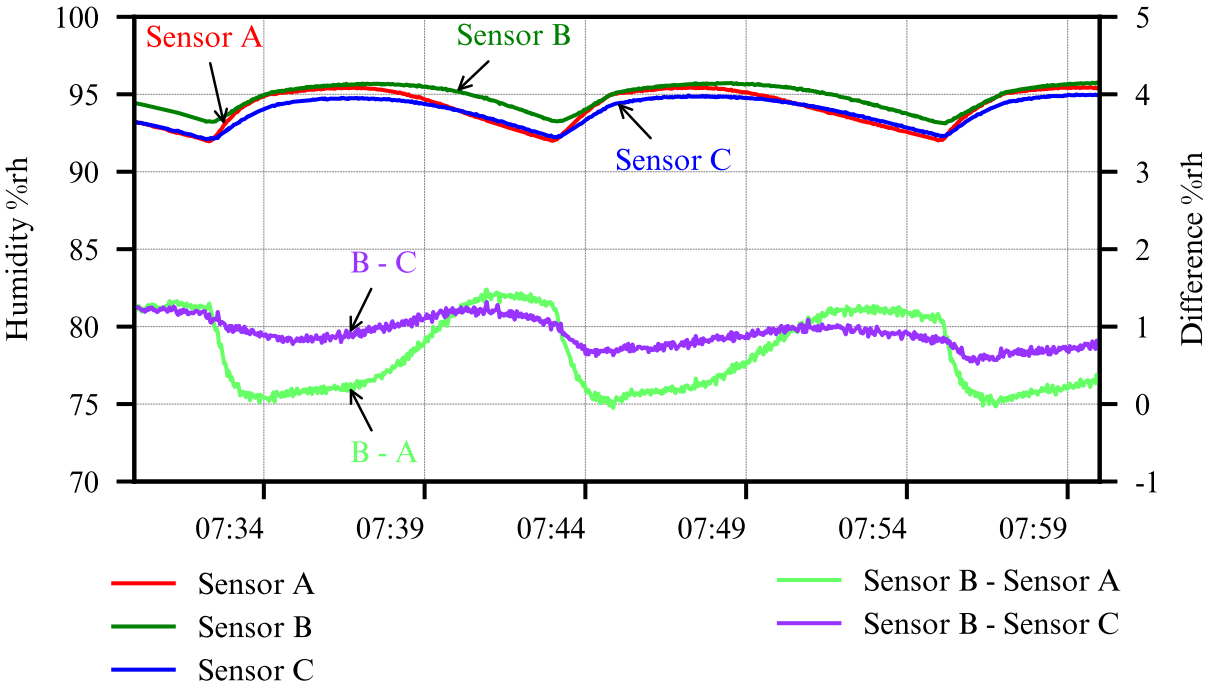
Table 1 – Pearson's correlation of RH between the sensors in experiments 01 and 02

	Experiment 01		Experiment 02	
	Sensor B	**Sensor C**	**Sensor B**	**Sensor C**
Sensor A	0.93111 ±0.00873	0.97094 ±0.00376	0.95585 ±0.00566	0.60264 ±0.04168
Sensor B	-	0.97980 ±0.00262	-	0.70627 ±0.03282

Source: Author (2023).

value up to 20s before the sensor **C**. However, although the curves do not overlap exactly, this difference is within the tolerance of ±2%rh declared by the manufacturer. Pearson's correlation coefficients (r), presented in Table 1, confirm the proportional relationship between sensor readings. The coefficients between readings show values close to 1, with confidence intervals smaller than ±0.01, for a confidence level of 95% and a p-value close to zero.

Figure 9 – Relative humidity and difference between sensors B and A and B and C, under same environment conditions.



Source: Author (2023).

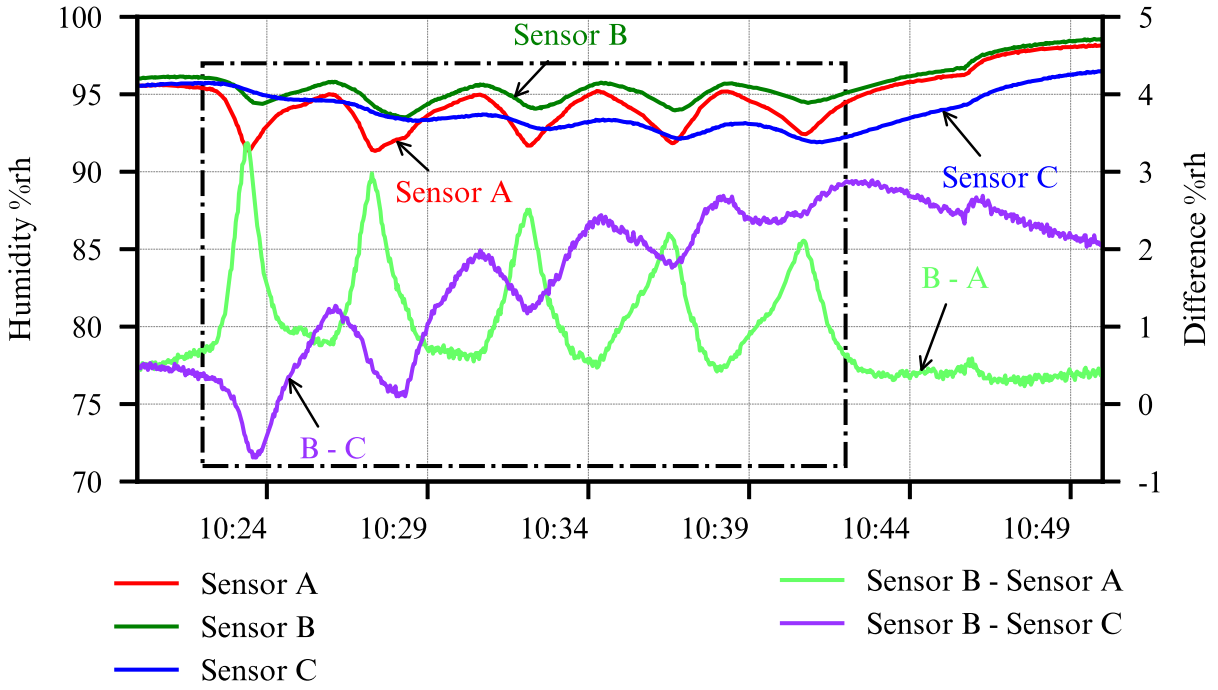
2.6.2 Experiment 2: Prototype of the measurement system

To perform this experiment, the *MS* was mounted as in Figure 3. Sensors **A** and **B** are under airflow of 1m/s. The average temperature is 2.4°C±1.3. Data from 30min (n=900) were used to allow comparison with Experiment 1. Figure 10 shows the readings of the two sensors and the differences with respect to the sensor **B**. The difference between the curves can be seen in the area highlighted by the dashed rectangle. Under ventilation, the **A** and **B** sensors show more variation than the sensor **C**. The difference between sensor **A** and sensor **B** is 0.99%rh ±0.66. The sensor **C** has a difference of 1.68%rh ±0.95.

Although the difference between the **A** and **B** sensors has increased, the correlation coefficient (Table 1) between the readings is still very similar to that of experiment 1. The coefficient between **A** and **B** remains stable compared to the first experiment. Sensor **C** values decrease, indicating a reduction in signal correlation. This difference reinforces the visual confirmation in Figure 10 and Figure 12 and shows, statistically, that the sensor **C** presents different results from the sensors **A** and **B**.

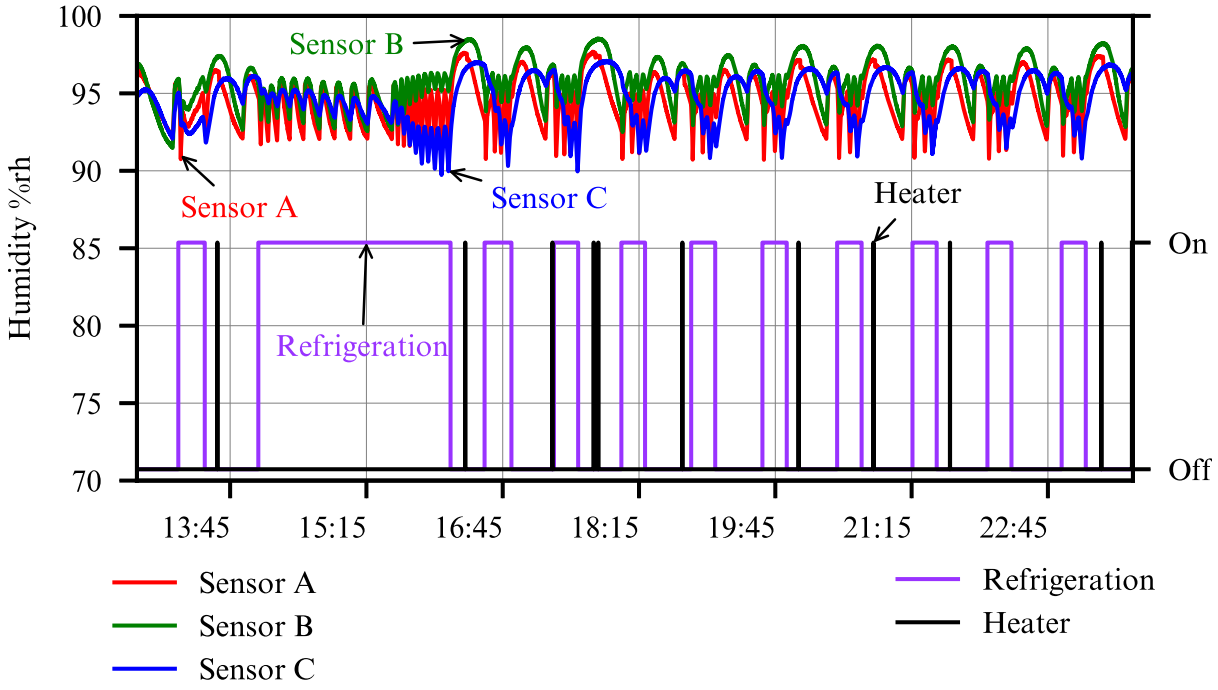
In this same experiment, the performance of the saturation detection and recovery method was also tested. Figure 11 shows data from 11 hours of records. The black line represents the state of the sensor heater, which is triggered every time the sensor saturates. It is essential to mention that all saturation events are detected when the cooling system is off, reinforcing our justification for the need for ventilation.

Figure 10 – Data collected from the three sensors in 30min. Sensors A and B are under ventilation. Sensor C is subject to the normal condition of the chamber.



Source: Author (2023).

Figure 11 – System operation over 11 hours with the indication of sensor heating and refrigeration operation.

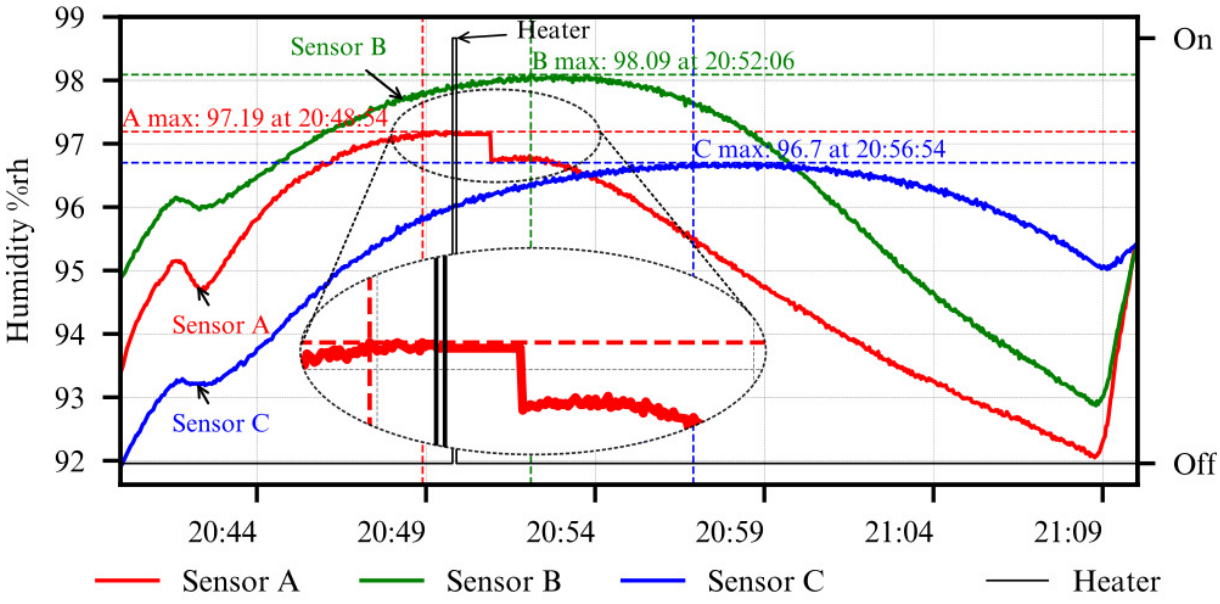


Source: Author (2023).

Figure 12 highlights a 30min interval of the same data from Figure 11. This image shows the maximum RH reached by each sensor and the instant of this event. Note that the difference

of up to **8min** between the peak values measured by the **A** and **C** sensors is more significant than the 20s found in Experiment 1 (Figure 9). The step in the line of the sensor **A**, after the heater is turned off, shows the moment when the *MS* updates the readings again. This step is generally between 0.3%rh and 0.4%rh for $\alpha = 60$s and $\delta = 0.5$%rh. This satisfactorily covers the relationship between the stability of the readings and the interval in which data are not updated to the rest of the system. The step emphasized in Figure 12 occurs consistently whenever the system heats the sensor.

Figure 12 – Details of the operation of the *MS*. Please note the scale of the humidity axis, which is reduced compared to the other graphs. The maximum value of the three sensors is pointed, including the time when it occurred.



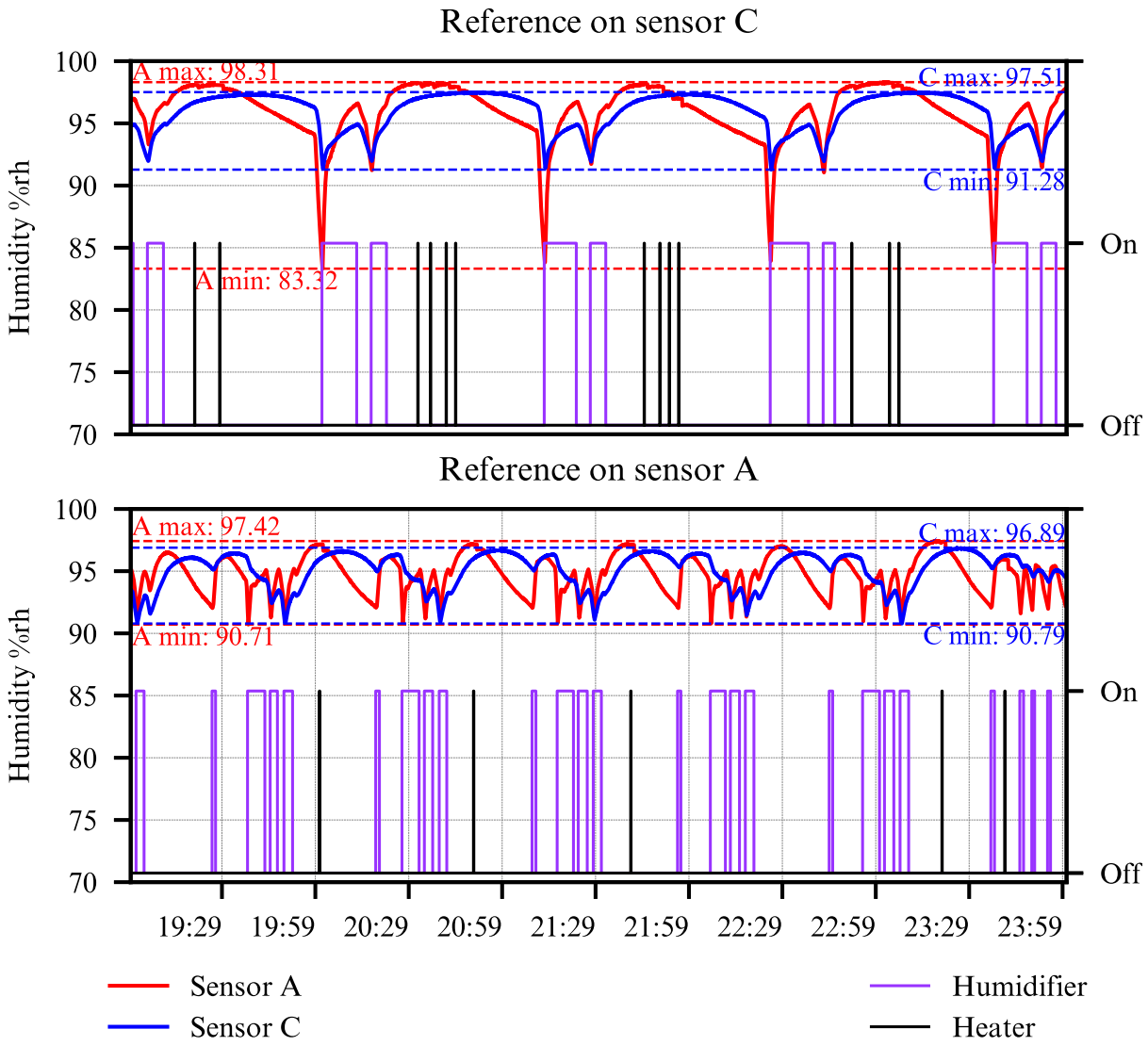
Source: Author (2023).

2.6.3 Experiment 3: Effects of the measurement system on RH control

The last experiment aims to analyze the impact of the proposed method on the control of RH. A humidifier was controlled using the on/off method and configured to turn on with 92%rh and off with 95%rh.

In the first step, the controller used RH provided by the sensor **C**. The sensor **A** was used as a reference in the second step. Figure 13 shows the collected data. When using the sensor **C** as a reference, the readings were between 91.28 and 97.51%rh in the sensor **C** and 83.32 and 98.31%rh in the sensor **A**. The humidifier was activated 10 times for 68min, and the sensor **A** was considered saturated 13 times within this interval. When the control was based on the sensor **A**, the values were between 90.71 and 97.42%rh for the sensor **A** and between 90.79 and 96.89%rh for the sensor **C**. The humidifier was activated 25 times for 66min and the sensor **A** was saturated five times.

Figure 13 – Comparison of the control system, using sensor A and sensor C as a reference. The dashed lines indicate the minimum and maximum values recorded by the two sensors over 5 hours.



Source: Author (2023).

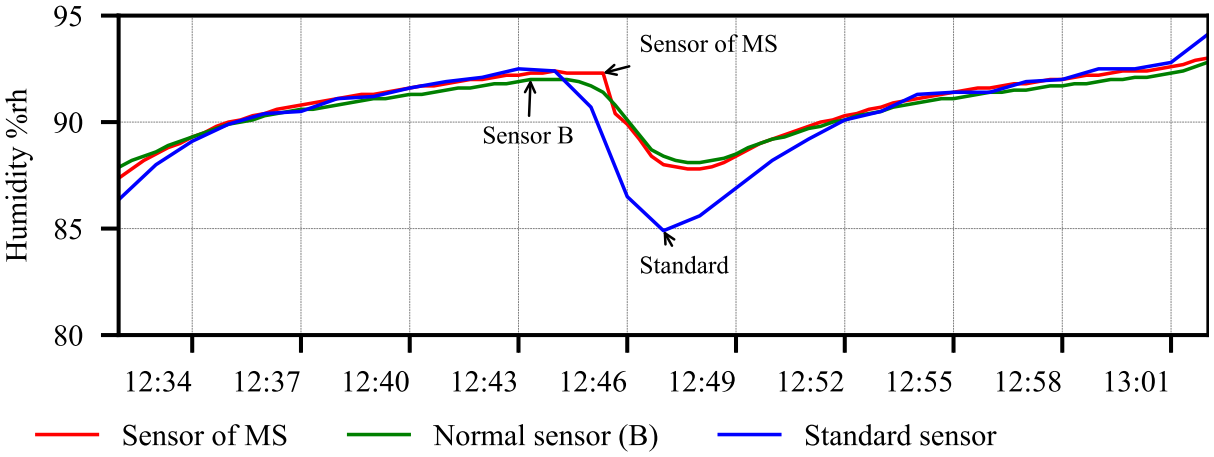
The results obtained in experiment 3 show the applicability of the proposed. When using the sensor without ventilation as a reference, the actual humidity inside the chamber, measured by the ventilated sensor, is approximately 7%rh lower at some moments. In addition, it exceeds by almost 1%rh the maximum value measured by the sensor **C**. In both cases, this may decrease the storage period and the quality of the fruits.

2.6.4 Improved MS Experiments

The sensor verification data for the improved *MS* is shown in Figure 14. Pearson's correlation of the *MS* sensor and Sensor B is 1.00 ±0.003 (95%). The correlation of the *MS* sensor

and standard is 0.94 ±0.046 (95%) The average and correlation between the values were very close, indicating that the sensors behave similarly when installed under the same conditions.

Figure 14 – Thirty minutes of data with the sensors installed under the same conditions. The RH average read for proposed *MS* is 90.8%rh, for sensor B is 90.6%rh, and for the standard is 90.3%rh.

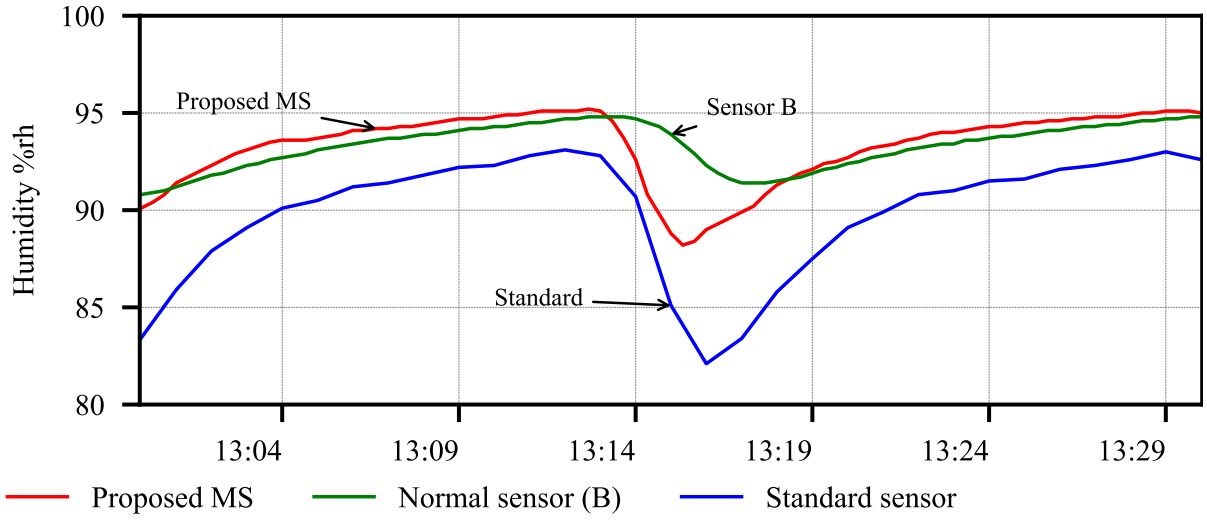


Source: Author (2023).

The Figure 15 shows data from 30 minutes of readings from the sensors installed as Figure 8. Although the average RH of the two sensors is the same as that of the standard, the proposed *MS* closely follows the humidity readings of the standard sensor. Pearson's correlation supports this observation since the value between the *MS* readings and the standard is 0.97 ±0.025 (95%). Meanwhile, the correlation between sensor B and the standard is 0.82 ±0.142 (95%).

Figure 15 –]

The RH average read for proposed *MS* is 93.3%rh, for sensor B is 93.3%rh, and for standard is 89.9%rh.

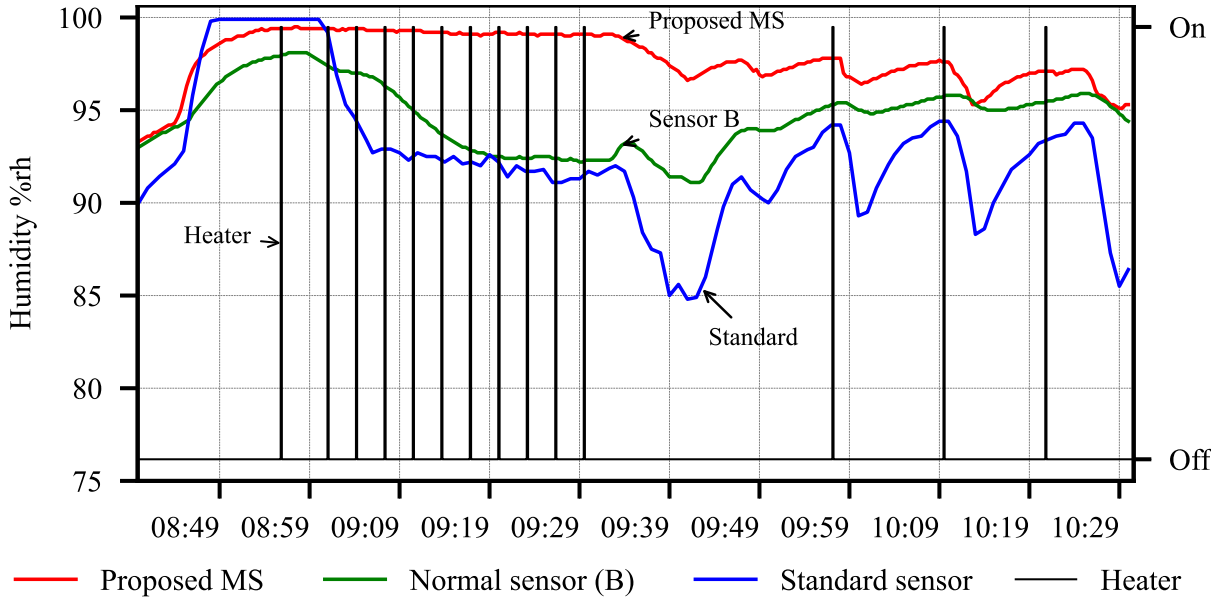


Source: Author (2023).

There is a difference of 3.4%rh between the RH measured by the sensors and the standard. This offset was significantly smaller (≈0.5%rh) when the sensors were checked (Figure 14). This error may be associated with the hysteresis of the sensors and the standard and requires further investigation. However, if the difference persists, it can be compensated for by adjusting the sensor, as shown in Montoya *et al.* (2020).

The data from the saturation experiment are shown in Figure 16. The method of detecting saturation can be seen to work, activating the heater of the *MS* sensor continuously until the RH begins to vary again. Although the RH read by the *MS* is closer to the standard (>99%rh), there is a significant delay before the readings are recovered. This may be associated with the sensor's heating process and requires further investigation. In Figure 17, there is a graph of another test in which the standard remains in saturation for a longer time, but *MS* does not even reach RH >99%rh.

Figure 16 – Saturation event

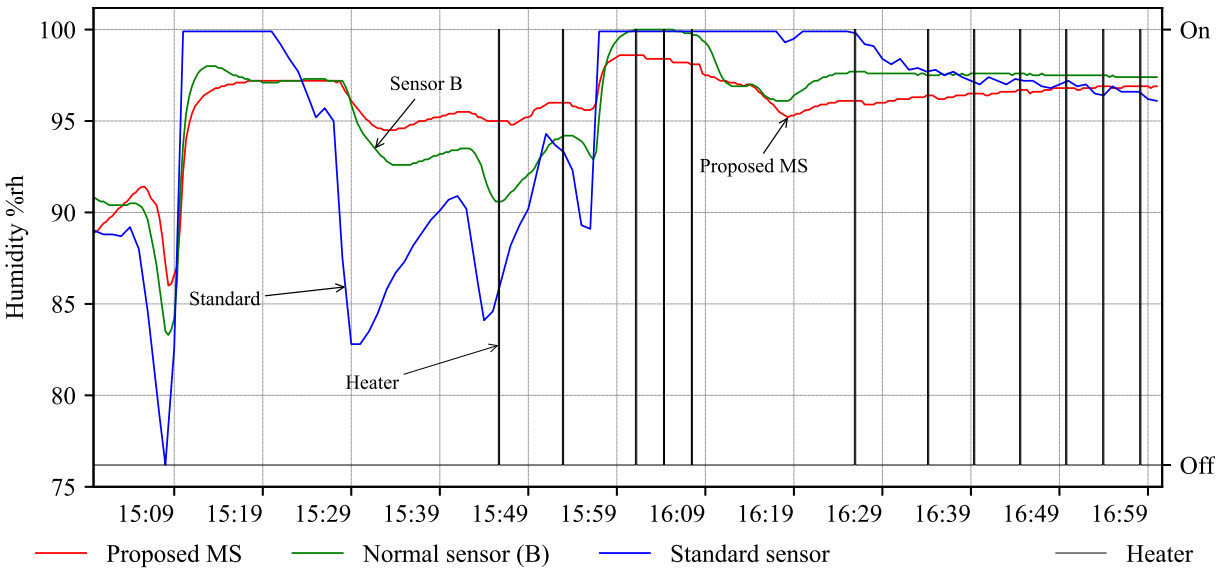


Source: Author (2023).

The experiments showed that the proposed *MS* could detect the variation in RH more quickly than the sensor without ventilation installed next to the standard sensor. On the other hand, when exposed to saturation events, its performance is even lower than that of the standard. It should be taken into account that the standard has a permanent heating system, which prevents the temperature of the RH sensor from falling below the dew point temperature. As a result, condensation occurs rarely in the standard. It should be noted, however, that the most up-to-date sensor model from the same manufacturer as the standard costs US$2,096.00 (+taxes)[2]. However, the estimated cost of the proposed *MS* is US$100.00, so it can be considered low cost.

[2] https://store.vaisala.com/en/products/HMP7, accessed on October 18, 2023.

Figure 17 – Further saturation tests on the sensors.



Source: Author (2023).

Continuous airflow, in conjunction with saturation detection and recovery, improves the reliability of the *MS*. It is possible to detect variations in RH in a shorter time and ensure control operation even under extreme RH conditions. Applying the *MS* to the humidity controller improves the results obtained, keeping the humidity closer to the programmed values. However, regardless of the sensor used, assuming it can withstand the extreme conditions of RH, it is still possible to improve the control results, as will be shown in the next Chapter.

3 RH CONTROL BASED ON DES

Humidifiers are used to maintain RH levels inside the cold chamber (LUFU; AMBAW; OPARA, 2020). These devices have actuators that insert droplets or water vapor into the environment to increase the RH. In the cold chamber, condensation of water vapor on the coil is the primary source of RH reduction. The difference between the air temperature and the surface temperature of the coil causes dew formation. The dew freezes on the coil (SHEN; WANG, 2020; ZHU *et al.*, 2023) and does not return to the environment. The defrosting process melts the ice and drains the water out of the chamber. Since post-harvest humidity loss in apples is irreversible (BISHNOI; AHARWAL, 2022), the humidifier must identify the reduction in RH and act immediately to avoid the problem.

A cold chamber is a large piece of equipment built according to the particular needs of each producer. They can have capacities of up to hundreds of tons of product. The height, width, and length change depending on the available space. In addition, the refrigeration system is dimensioned by storage capacity and the amount of product processed daily during the harvest. As for refrigeration control, despite the well-known advantages of variable compressor control, such as energy and maintenance savings, it is expected to find new installations with on/off control, with installation cost being the main determining factor when choosing.

The air humidifier, when present, operates in on/off mode. In commercial chambers, it is necessary to use sprinkler nozzles to generate the mist since ultrasonic humidifiers such as the one from Mohammed, Alqahtani e El-Shafie (2021) do not have sufficient capacity for this purpose. The nozzles work at a constant flow rate, which is why humidifiers use this drive method.

The moment the humidifier is switched on and off can be determined via a timer or based on the signal from a RH sensor. Control using a timer requires intensive research into the structure of the chamber and the product stored (DELELE *et al.*, 2009) or trial and error based on the experience of the equipment operator. Closed-loop control, using the RH sensor as a reference, appears to be the best alternative, as long as a reliable measurement structure is used, such as the one presented in Chapter 2. Therefore, this chapter reviews the literature that evaluates humidifier control methods in cold storage for fresh produce, the theory about DES and supervisory control, and the proposed control strategy.

3.1 Literature review

Classical controllers such as Proportional-Integral-Derivative (PID) have shown strong dynamic performance in controlled environments, including greenhouses and closed environments (HEIDARI; KHODADADI, 2017; SU; YU; ZENG, 2020; WANG; ZHU, 2020; MATAMOROS *et al.*, 2020; CUI *et al.*, 2020). Although these approaches, using traditional PID or self-tuning methods, offer promising results in simulations based on experimental prototypes,

they can suffer from relative errors. To address this, more advanced strategies like Model-based Predictive Control (MPC) have been used to preemptively estimate the future state of the system, as observed in climate chambers, greenhouses, and Heating, Ventilating, and Air Conditioning (HVAC) systems in studies like (DOSTáL; FERKL, 2014; ITO; TABEI, 2021; CHEN; MATTSON; YOU, 2022; ENDO *et al.*, 2022; BAHRAMNIA *et al.*, 2019). However, MPC relies on parameters that may be poorly known and affect the control design. Even when parameters are identified, they may still have uncertainties, leading to issues such as parametric sensitivity.

To mitigate parametric dependencies, fuzzy strategies (KAEWWISET; YODKHAD, 2017; TUTUNCU; OZCAN, 2019), neural networks (FUADY *et al.*, 2017), and reinforcement learning (SUN *et al.*, 2022) can be found in the literature. While these approaches yield promising results, their complexity may render them impractical.

In this sense, approaching the problem as DES can be an alternative. Exposing the system through FSMs allows comprehensive plant observation, considering sporadic events that are more difficult to embed in other approaches. For example, in Lorencena *et al.* (2020) and Schmidt *et al.* (2021), the control of temperature and RH in poultry houses was approached from the point of view of DES. These articles present a hybrid solution (CASSANDRAS; LAFORTUNE, 2021), where the evolution of the system is tracked through discrete events, and the temperature control is time-oriented, with temperature parameters adjusted by an already trained neural network (SCHMIDT *et al.*, 2021). The neural network has multiple inputs and calculates the temperature parameters and (discrete) operating levels of the exhaust fans to keep the process under control.

Although existing work has explored DES-based approaches to controlling temperature and RH, it is crucial to develop specific strategies customized to the unique requirements of cold chambers considering the coupling between RH and the refrigeration system. This approach can potentially improve the control and performance of humidifiers in these environments. Taking into account, a control strategy based on DES was developed for practical application, without parametric dependence, to minimize RH errors and control the process. The following section presents the theory of DES and their control.

3.2 Foundations of DES

DESs are systems that evolve according to the unexpected occurrence of events at irregular, possibly unknown, intervals (CASSANDRAS; LAFORTUNE, 2021). They are opposed to the system whose evolution depends continuously on time. For example, a discrete event could be pushing a button, a component arrival, a disturbance or set-point change in a control system, etc.

Observing and designing a system as DES helps to understand it and find ways to improve it. In general, DES models are created using the notion of *strings* formed with *events* taken

from an alphabet Σ. Σ^* denotes the set of all finite strings formed with events in Σ, including the *empty string* ε. Any subset $L \subseteq \Sigma^*$ is called a *language*.

FSMs are acceptable mechanisms to represent languages. An FSM is a 5-tuple $G = (\Sigma,Q,q^\circ,Q^m,\rightarrow)$, where Σ is the set of events, Q is the set of states, $\rightarrow \subseteq Q \times \Sigma \times Q$ is the transition relation, q° is the initial state, and $Q^m \subseteq Q$ is the subset of marked states that represent tasks completed by the DES modeled by G.

It is usual for FSMs to be exposed graphically. For two any states $q_1,q_2 \in Q$, $q_1 \xrightarrow{\sigma} q_2$ denotes a transition from state q_1 to q_2 with event $\sigma \in \Sigma$. The same notation can be generalized to strings $s \in \Sigma^*$, that is, $q_1 \xrightarrow{s} q_2$ means that q_2 is reached from q_1 with s, while $G \xrightarrow{s} q$ means that a string s is possible in G, from its initial state.

The language generated by G is denoted by $L(G) \subseteq \Sigma^*$ and defined as $L(G) = \{s \in \Sigma^* \mid G \xrightarrow{s} q \in Q\}$. It represents the set of all the strings of events that can occur in G. Differently, $L^m(G) \subseteq L(G)$ is defined as $L^m(G) = \{s \in \Sigma^* \mid G \xrightarrow{s} q \in Q^m\}$ and denotes the language marked by G, i.e. those strings that lead DES to complete tasks.

As a DES is usually formed by a set $J = \{1, \cdots ,m\}$ of components, it is convenient to design each component as an FSM G_j, for $j \in J$, and combine them by *synchronous composition* (CASSANDRAS; LAFORTUNE, 2021). Given two FSMs, $G_1 = (\Sigma_1,Q_1,q_1^\circ,Q_1^m,\rightarrow_1)$ and $G_2 = (\Sigma_2,Q_2,q_2^\circ,Q_2^m,\rightarrow_2)$, their synchronous composition, denoted $G_1 \| G_2$, is an operation that merges events that are enabled by both G_1 and G_2; events enabled by only one FSM and unknown by the other are interleaved in any order; finally, events enabled by one FSM and known, but not enabled, by the other, are disabled upon composition. This is actually the main mechanism for imposing restrictions on a DES by modeling. A state is considered marked after composition if the corresponding combination of states is also marked in the input FSMs.

3.2.1 DES Control

A DES often includes $n \geq 1$ components that are expected to work together. Each component can be modeled by an FSM G^i, and the so-called open-loop behavior is then given by the composition $G = G^1 \| \ldots \| G^n$, which is called the *plant*. Together, FSMs G^i are expected to interact with each other, and it is the role of a controller to define how and when they do. An approach that allows DES controllers to be automatically computed from a set of FSMs is SCT (RAMADGE; WONHAM, 1989a), as long as control specifications are provided.

Specifications are rules that restrict actions in G, such as imposing priority on one component over another. Analogously to G, a specification E^j can also be modularly exposed so that $E = E^1 \| \ldots \| E^m$, for $m \geq 1$, is the general specification model. It can be joined to G to form the *desired behavior* under control, i.e., a sub-behavior of the open-loop denoted $K = G \parallel E$. Observe that, therefore, E must only restrict G without creating any new behavior.

From K, a controller can be calculated (or synthesized). This operation further considers the controllable nature of the events, which is formally expressed by partitioning the set of

events of G into $\Sigma = \Sigma_{\mathrm{c}} \dot{\cup} \Sigma_{\mathrm{u}}$. Events in Σ_{c} can be inhibited in G, while those in Σ_{u} cannot be prevented directly by the controller.

Then K is said to be controllable with respect to G if, for all $s \in \Sigma_G^*$ and all $\mu \in \Sigma_{\mathrm{u}}$, whenever $K \xrightarrow{s} x_K$ and $G \xrightarrow{s} x_G \xrightarrow{\mu} x_G{}'$, there exists $x_K{}' \in Q_K$, such that $K \xrightarrow{s} x_K \xrightarrow{\mu} x_K{}'$.

That is, K is controllable if it allows the uncontrollable events whenever they are also allowed by the plant. When the model K is not controllable, it can be reduced iteratively to its largest controllable (supremal) submodel. This computation is called *synthesis* in SCT, and $\sup\mathcal{C}(K,G)$ is the usual notation of the resulting model. In practice, $\sup\mathcal{C}(K,G)$ represents the most permissive behavior possible to be imposed by control on G while complying with controllability and specification. If $\sup\mathcal{C}(K,G)$ is also nonblocking (CASSANDRAS; LAFORTUNE, 2021), it is said to be *optimal*.

Implementing the FSM that models $\sup\mathcal{C}(K,G)$ has advantages compared to empirical programming because it automates manual tasks by involving modular, usually simple, design tasks, which tends to increase development and maintenance speed, especially when the control logic changes frequently.

3.2.2 Modular synthesis

For many applications, it is reasonable to design the plant model of a DES as a set of independent components to be coordinated by the controller according to multiple specifications, also constructed modularly. When these multiple plants and specifications are synchronized, and the result is a single controller, the solution is said to be *monolithic*.

In monolithic synthesis, the *state-space explosion* problem may become a barrier for SCT to be adopted in the industry. Modularisation (QUEIROZ; CURY, 2005; MOHAJERANI; MALIK; FABIAN, 2017; ROSA; TEIXEIRA; MALIK, 2018) is an alternative to simplify synthesis by exploiting subsets of a few components, hopefully reducing the computational effort necessary to obtain the controller.

The tool presented in this paper supports the generation of code for both monolithic and *Local Modular Controllers* ($\mathbf{LMC\text{-}P}$) (QUEIROZ; CURY, 2005). In $\mathbf{LMC\text{-}P}$, each specification leads to a local controller, which is calculated using only plants affected by the events of the specification.

Definition 1. *Let E^i, for $i \in I = \{1, \cdots, m\}$, be specifications modeled with events in Σ_{E^i}; and G^j, for $j \in J = \{1, \cdots, n\}$, be asynchronous plants designed with events in Σ_{G^j}. For $i = 1, \cdots, m$, the local plants G_i^{loc}, associated with each specification E^i, are $\mathrm{G}_i^{loc} = \|_{j \in J_i} G^j$, such that $J_i = \{j \in J \mid \Sigma_{G^j} \cap \Sigma_{E^i} \neq \emptyset\}$.*

In words, each local plant is associated only with the plants restricted by E^i. Upon synthesis, the $\mathbf{LMC\text{-}P}$ creates a set of local controllers S_i^{loc} that are individually nonblocking. However, the conjunction of controllers can be blocking, even when individually nonblocking.

In this case, the controllers are said to be conflicting. In (QUEIROZ; CURY, 2005) and older results, some conditions guarantee the non-conflict of local controllers. When the non-conflicting condition is achieved, the joint action of local controllers over the plant is optimal. That is, if the set $\{\mathrm{S}_i^{loc}, i = 1, \cdots, m\}$ is non-conflicting, then $S = \|_{i=1}^{m} \mathrm{S}_i^{loc}$, for S modeling $\sup\mathcal{C}(K,G)$.

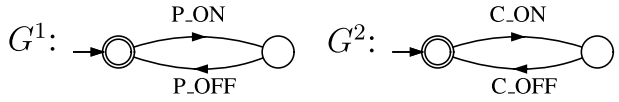
In the next section, the RH system and its control structure are seen as a DES, and modeled with FSMs, as presented here. The RH controller is synthesized in Section 3.4, and the resulting FSM is converted into code with the tool showed in Chapter 4 and applied in Chapter 5.

3.3 Modeling the RH control as a DES

To model the RH control system, a traditional humidifier found in commercial cold chambers was considered, consisting of a water pump, a fan (optional), and a RH sensor. The fan is activated in parallel with the pump, so it was not included in the models in this work.

Thus, Figure 18 presents the humidifier (G^1) and the refrigeration system (G^2) models. These models are represented by two states that indicate on or off. In this work, the refrigeration is not controlled, so the G^2 model only captures its operating state, and no control specifications are provided.

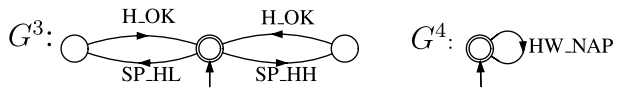
Figure 18 – Models of the humidifier and refrigeration system. G^1 represents the water pump. G^2 represents the compressor.



Source: Author (2023).

Since RH is an analog signal, the RH sensor was discretized into three states to model it. The states indicate whether RH is within or outside the set limits. The behavior of the RH sensor is modeled in G^3 and shown in Figure 19. Two adjustable parameters are considered to trigger sensor events. The maximum (**SPh**) and minimum (**SPl**) set-points. These parameters describe the acceptable limits for RH and are set by the humidifier operator. The event SP_HH is triggered whenever $RH > SPh$. The event SP_HL means that RH is less than the programmed minimum set-point $RH < SPl$. The event H_OK, in turn, is observed when RH is within limits ($SPl <= RH <= SPh$).

Figure 19 – Model G^3 for the RH sensor with the events SP_HH, H_OK, and SP_HL. Model G^4 for the event HW_NAP



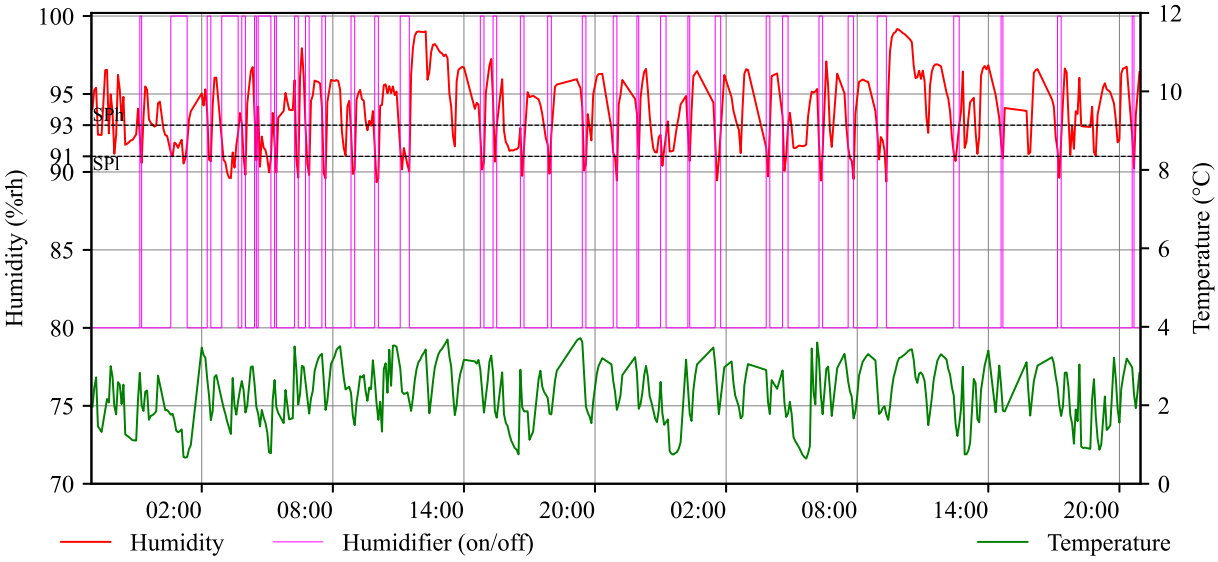
Source: Author (2023).

The last relevant event observed in the plant is the interval between starts. It symbolizes a minimum interval between pump starts to ensure the electric motor ad-

heres to its duty type. The model G^4, in Figure 19, shows this component. Therefore, the controllability of events is defined as $\Sigma_c = \{P\_ON, P\_OFF\}$, while $\Sigma_u = \{C\_ON, C\_OFF, SP\_HH, SP\_HL, H\_OK, HW\_NAP\}$.

With plant models, the control specifications are modeled. The control strategy is based on theoretical and experimental analysis. Conceptually, in a closed environment, when the temperature decreases, RH increases. However, with activated refrigeration, water condenses in the evaporator coil (SHEN; WANG, 2020; ZHU *et al.*, 2023), freezing and causing a drop in RH. This behavior was also observed experimentally by installing sensors in two commercial fruit storage cold chambers. The investigation shows that cooling causes a drop in RH, as can be seen in Figure 20 and Figure 21, which confirms the theoretical analysis. It was also observed that RH increases without a humidifier with the refrigeration off.

Figure 20 – Data collected in a 759m^3 commercial cold chamber, with on-off control. The controller was set to be turned on at 91%rh and off at 93%rh. The average RH is 94.0%rh ±2.22%rh. The Root Mean Squared Error (RMSE) for values above the programmed value is 2.19%rh, and for values below, it is 0.29%rh. The RH remained 65.22% of the time above the SPh limit and 10.43% below SPl.

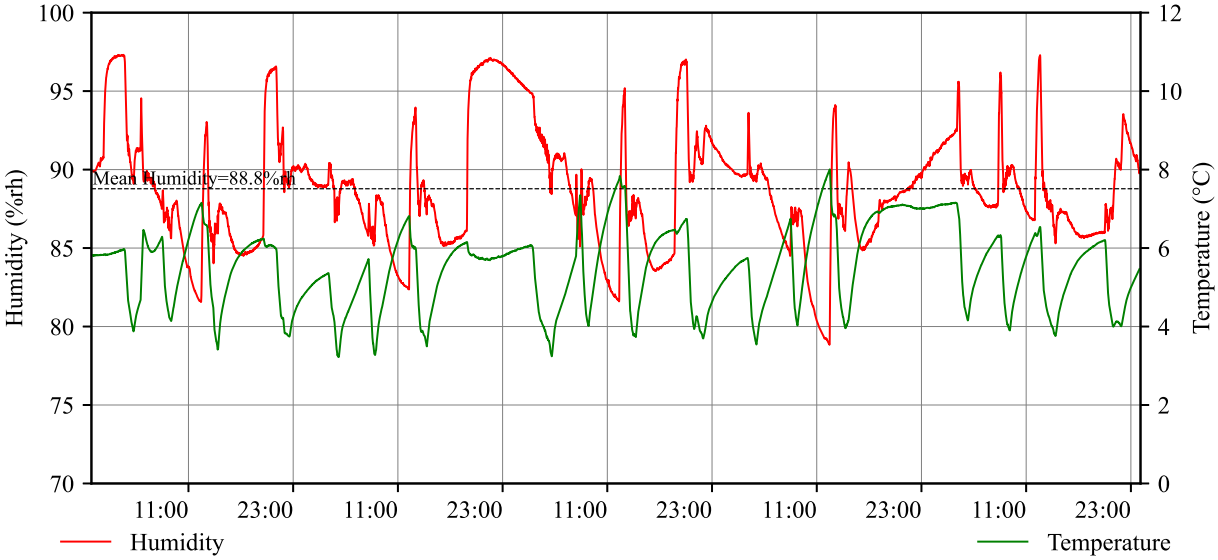


Source: Author (2023).

Therefore, the specifications for the humidifier operation consider the interaction between refrigeration and RH. It is expected that the plant G^3 is always in the acceptable state, representing that RH is within the programmed limits. To achieve this, the P_ON event is disabled until the SP_HL event is observed. At the same time, if the event C_ON occurs, P_ON is enabled with SP_HL and H_OK. This behavior allows the humidifier to take action on the plant before $RH < SPl$. The specification E^1 is shown in Figure 22.

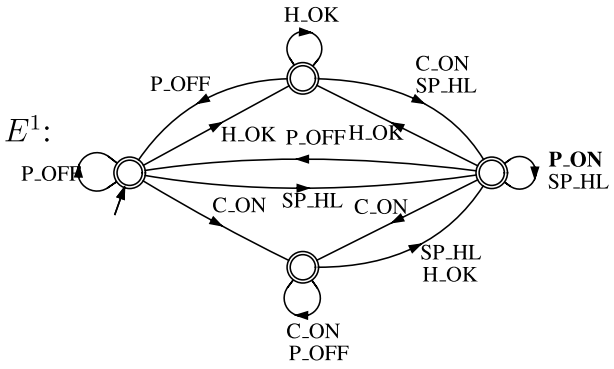
The coordination of the event P_OFF is carried out by the specification E^2, shown in the Figure 23. The humidifier will be turned off for two moments. With the refrigeration off,

Figure 21 – Data collected in a 2429m^3 commercial cold chamber without automatic RH control. In this case, the humidifier is activated together with the refrigeration. The average RH is 88.8%rh ±3.79%rh. It was impossible to calculate error statistics concerning the programmed values since the chamber does not have a RH controller. However, the mean RH resulted in 88.8%rh, with a standard deviation of ±3.79%rh.



Source: Author (2023).

Figure 22 – Specification E^1, that disables the event P_ON.



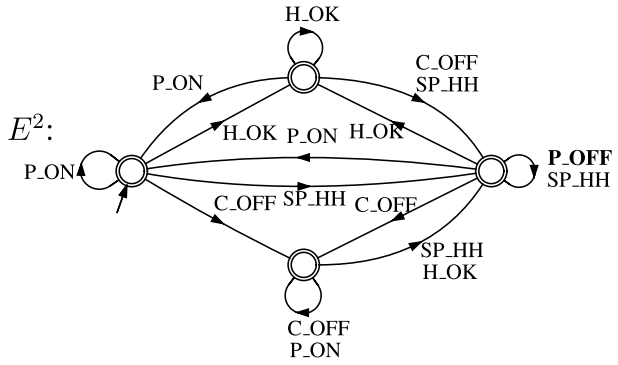
Source: Author (2023).

the event P_OFF is enabled when events SP_HH and H_OK are observed. However, the event P_OFF is enabled only with SP_HH if refrigeration is enabled.

The specifications E^1 and E^2 work together, adding preventive action capability to humidifier control. The equipment prevents RH from falling below the programmed minimum levels. It is also turned off before RH crosses the maximum value.

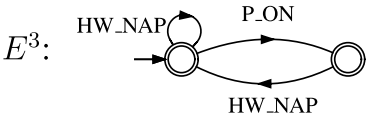
The setting that limits the minimum interval between the humidifier activations is added to the control structure by specification E^3, in Figure 24. Whenever an event P_ON occurs, a timer triggers the event HW_NAP at the end of a time t. Observing this interval is essential to ensure the electric motor's life and the entire system's performance over time. The constant motor starting can lead to overheating and the actuation of thermal protection devices, which disables the equipment until the problem is fixed.

Figure 23 – Specification E^2, that disables the event P_OFF.



Source: Author (2023).

Figure 24 – Specification E^3. Disables the start of the humidifier pump until the interval between starts is over.



Source: Author (2023).

3.4 Synthesis the Controller

The synchronous composition $G = G^1 \parallel G^2 \parallel G^3 \parallel G^4$, is performed from the plant models, resulting in a FSM with 12 states and 40 transitions. The composition of the specifications, $E = E^1 \parallel E^2 \parallel E^3$, results in a FSM with 27 states and 174 transitions. The model E is composed with the plant G to obtain the model K. $K = G \parallel E$, which has 193 states and 707 transitions. In this work, K is controllable. Thus, after the synthesis, the supervisor S, representing $\sup\mathcal{C}(K,G)$, has maintained the same number of states and transitions as K.

Modeling, synchronization, and synthesis operations were performed in the Supremica tool (AKESSON *et al.*, 2023; MALIK *et al.*, 2017). Despite allowing these operations, the free version of Supremica does not offer an option for practical implementation of the supervisor obtained, such as exporting code. Therefore, a tool was developed to analyze the Supremica file and generate code from it. This tool was called DEScMaker (POSSATO *et al.*, 2024; POSSATO, 2023a), and will be presented in the next Chapter.

4 DESCMAKER: AN AUTOMATIC CODE GENERATOR FROM FSMS

When a system is modeled as an FSM, the purpose is to synthesize sequences of events to be allowed under control. It is assumed that events occur spontaneously in the system (also known as *plant*), and they have to be externally coordinated by a properly programmed dedicated hardware (such as *microcontroller* or *Programmable Logic Controller* (PLC)). A formal approach that automates the synthesis of controllers from FSMs is the *SCT* (RAMADGE; WONHAM, 1989b), which defines a mature mathematical method for extracting controllers from higher-level models (FOKKINK *et al.*, 2022).

Despite its practical relevance and formal background, the SCT is limited in converting synthesized controllers into implementable code. This depends on code generation tools for different synthesis architectures and modeling resources. There are some tools for code generation, such as DESTool (MOOR; SCHMIDT; PERK, 2008), libFAUDES (MOOR; SCHMIDT; PERK, 2008), DESLAB (CLAVIJO; BASILIO; CARVALHO, 2012), CIF (BEEK *et al.*, 2014), Nadzoru (PINHEIRO *et al.*, 2015), UltraDES (ALVES; MARTINS; PENA, 2017). However, they work only in combination with the specific modeling tool that implements it, which in turn does not usually gather all the resources needed to guide the entire controller project, from conception through modeling, simulation, synthesis, model-checking, code generation, implementation, and monitoring. Furthermore, existing approaches usually focus on the implementation of PLC (QUEIROZ; CURY, 2002; VIEIRA; CURY; QUEIROZ, 2006; LJUNGKRANTZ *et al.*, 2007; BASILE; CHIACCHIO, 2007; HASDEMIR; KURTULAN; GOREN, 2008; LEAL; CRUZ; HOUNSELL, 2009; SILVA; QUEIROZ, 2010; UZAM, 2012; GOBE *et al.*, 2016; VIEIRA *et al.*, 2017; MCCARTHY *et al.*, 2022), and do not include outputs for higher-level data science-oriented languages, such as Python.

A tool that gathers many functionalities for the entire development of a controller is *Supremica* (AKESSON *et al.*, 2023). This tool provides an intuitive modeling environment and a complete set of synthesis algorithms either for monolithic, modular, incremental, or variable-based synthesis. Supremica also provides a modular view of how models are related so that they can be separated accordingly. Model-checking resources and simulation functions are also available to allow an empirical check of the correctness of models and constraints (MALIK *et al.*, 2017; RENIERS; MORTEL-FRONCZAK, 2018; FOKKINK *et al.*, 2022). Supremica also provides some output formats, such as XML, but (at least in its free release) does not export models in implementable code, such as C, Python, or other hardware-compatible forms for PLC or microcontrollers, limiting its usefulness.

This chapter presents the DEScMaker tool (POSSATO *et al.*, 2024; POSSATO, 2023a), which is designed to generate C code from correct-by-construction controllers designed, synthesized, and checked in Supremica. From the Supremica's output controller, DEScMaker reads, interprets, and organizes this file and constructs the equivalent infrastructure in C, which can be directly transferred to compatible hardware. The three-dimensional implementation architecture

of Queiroz e Cury (2002) is adopted for the conversion of FSM into C code, where supervisors, event handling, and physical integration are addressed in a distinct way. The code resulting from DEScMaker can be compiled and tested on a computer prior to deployment, serving as a tool for simulation, validation, and previsualization of the effect of the joint solution.

4.1 The DEScMaker implementation

The DEScMaker (POSSATO, 2023a) is a tool programmed in Python that receives a Supremica file as input and exports a directory with a project in C. The input file must contain at least one synthesized supervisor, which implies that it also supports modular projects of control. The directory with the generated code is organized to allow the quick implementation of the project and the easy replacement of the supervisor's files in case of any modeling change. The directory and file structure generated by DEScMaker can be found in Figure 25.

Figure 25 – Directory tree of generated code.

```
.
├── CMakeLists.txt
├── lib
│   ├── CMakeLists.txt
│   └── supervisor: Library for manipulating supervisor structures.
│       ├── CMakeLists.txt
│       ├── sup_debug.h: Debug commands.
│       ├── supervisor.c: Library functions.
│       └── supervisor.h: Definitions of the structures and function headers.
├── run.sh: Script for compiling and running the code in Linux.
└── src
    ├── main.c: Main file with examples of using the generated code.
    ├── event_handler
    │   ├── event_handler.c: Functions of the event handler
    │   ├── event_handler.h
    │   ├── events.c: Event creation.
    │   ├── events.h: Event definitions.
    └── supervisors: Folder with the generated supervisors.
        ├── supervisor_list.c: Make and linking the list of supervisors
        ├── supervisor_list.h: List of supervisors
        ├── sup[s].c: Instantiation and linking supervisor structures.
        └── sup[s].h: Supervisor definitions.
```

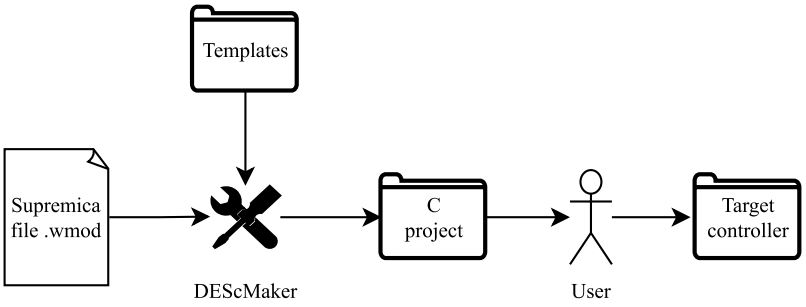
Source: Author (2023).

In this case, the input file has only one supervisor, named "sup", which results in the files "sup.c" and "sup.h" under "src/supervisors/". If the file has two or more supervisors, as in the modular approach, all code files and headers are placed in the folder "supervisors".

Figure 26 provides a visual representation of the flow around code generation. DEScMaker must be called from the command line, indicating the path of the Supremica file to be processed and the name of the output file. DEScMaker then scans the file, identifies the supervisors, and builds the output project. The Supremica's file is structured in XML format, although it is saved with the extension '.wmod'. This file contains elements and attributes to represent events, plants, and supervisors. Within each supervisor, there are other nested elements to rep-

resent events and transitions. After identifying the supervisors, events, and transitions, DEScMaker uses the class Template from Python strings to make string substitutions in the templates and save the code. The generated project can be compiled and executed. Depending on the target platform, moving the files to a specific structure may be necessary. The user can automate this task by creating a custom script.

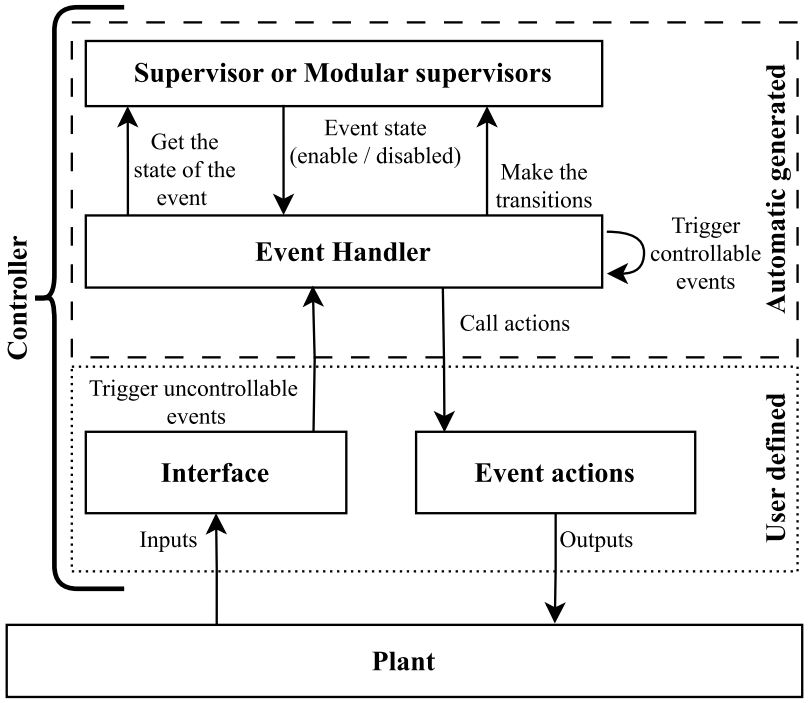
Figure 26 – Usage flow of DEScMaker



Source: Author (2023).

The proposed structure for the generated code is based on the hierarchy presented by Queiroz e Cury (2002), which is adapted and shown in Figure 27. At the top is the *automatically generated* code, highlighting the *supervisors* and the *event handler*, which will be discussed later. In the center is the *user defined* area, with functions specific to the target platform. In the user area, the uncontrollable events are transmitted to the event handler, and the actions associated with the events, usually controllable events, are defined.

Figure 27 – Proposed structure for the operation of the generated code.



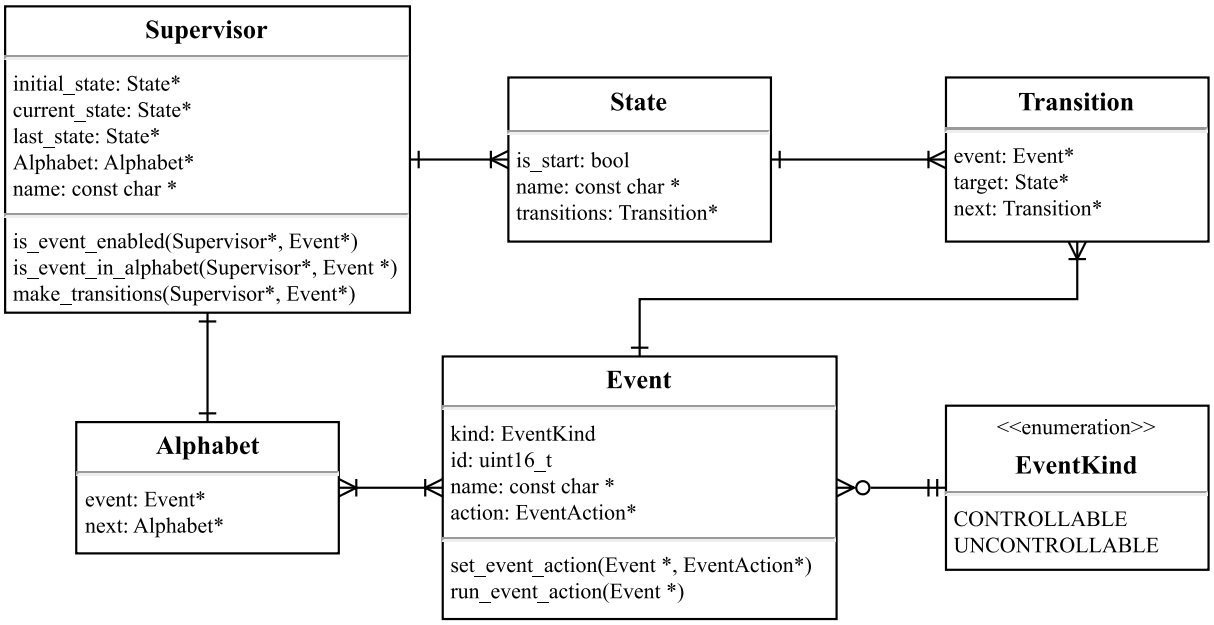
Source: Author (2023).

4.1.1 The Supervisor Structure

Supervisors are represented through linked lists, with structures representing events, states, and transitions. These structures are linked to allow access and navigation through the automaton. A library enables operations to be performed on the supervisors, with functions to add a callback function to each event and check if an event is enabled, among other functionalities. Each event can have an associated callback function that is executed whenever the event is executed. The generated code does not perform dynamic memory allocation. Thus, it is safer to implement on microcontrollers with limited memory.

Logically, a supervisor is represented by a structure that points to an initial state, the current state, the last state, and an alphabet. The alphabet is a list of all the events of the supervisor. A state has a list of transitions. A transition has the target state and an associated event. The event has a type (controllable or uncontrollable) and an associated action callback. Figure 28 illustrates these relationships.

Figure 28 – Crow's foot entity relationship diagram for the structures of a supervisor.



Source: Author (2023).

The structure representing a state has a variable, to indicate whether it is the initial, and a pointer to a list of transitions. It is from this list of transitions that the program searches for enabled events. The list of transitions points to the first transition of the state. The next transitions are associated with the transition structure itself. In addition, the transition has a pointer to the event that triggers it and to the target state.

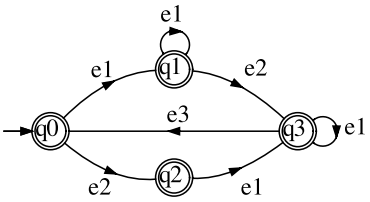
An event, in turn, is represented by a structure with an event type (controllable or uncontrollable), id, name, and a pointer to an action function. This function is associated with the event in the user area at the start of the program. It is important to note that the events are

global and not linked directly to a supervisor. This is why there is only one event structure, even using the modular approach. Next section shows some snippets of the generated code, with the creation, filling, and linking of the structures that represent the FSM.

4.2 Code generation example

Consider the FSM in Figure 29, corresponding to a supervisor with four states, seven transitions, and three different events. The Supremica file with this automaton was submitted to the DEScMaker, which created the code structure and saved it in "src/supervisors/sup.c".

Figure 29 – Simple example of FSM for code generation.



Source: Author (2023).

Variables of type Alphabet are declared to create the list of events observed by the supervisor. This creates a variable with reserved memory space for an Alphabet structure, but its fields have no values assigned yet. Next, the alphabet variables are initialized and linked. This creates a list where it is possible to navigate from the first to the last event in the alphabet. In this example, the last element of the 'sup_e3_evt2' variable is NULL because there are no more elements in the list. This pattern also repeats when creating the other lists. The Source Code 1 shows the code for declaring, initializing, and linking the alphabet.

Source Code 1 – Alphabet creation and initialization

```
1  #include "../event_handler/events.h"
2  #include "sup.h"
3  // Declares the Alphabet variables
4  const Alphabet sup_e1_evt0;
5  const Alphabet sup_e2_evt1;
6  const Alphabet sup_e3_evt2;
7  // Initializes and links Alphabet variables.
8  const Alphabet sup_e1_evt0 = {&e1, &sup_e2_evt1};
9  const Alphabet sup_e2_evt1 = {&e2, &sup_e3_evt2};
10 const Alphabet sup_e3_evt2 = {&e3, NULL};
```

Source: Author (2023).

Then the variables of type State, for representing the FSM states, are declared as in Source Code 2.

Once the alphabet and states are defined, the transition lists can be created and linked. Source Code 3 exemplifies the transitions of the states $q0$ and $q1$.

The states are initialized in Source Code 4, now filled with all the necessary information, especially the pointer to the corresponding list of transitions.

Source Code 2 – States creation

```
1 const State sup_q0;
2 const State sup_q1;
3 ...
```

Source: Author (2023).

Source Code 3 – Creating and linking transitions

```
1  //Transitions for state q0
2  const Transition sup_q0_t0;
3  const Transition sup_q0_t1;
4  const Transition sup_q0_t0 = {&e1, &sup_q1, &sup_q0_t1};
5  const Transition sup_q0_t1 = {&e2, &sup_q2, NULL};
6  //Transitions for state q1
7  const Transition sup_q1_t0;
8  const Transition sup_q1_t1;
9  const Transition sup_q1_t0 = {&e1, &sup_q1, &sup_q1_t1};
10 const Transition sup_q1_t1 = {&e2, &sup_q3, NULL};
11 ...
```

Source: Author (2023).

Source Code 4 – Filling in the states

```
1 const State sup_q0 = {true,  SUP_DEBUG_STR("q0"), &sup_q0_t0};
2 const State sup_q1 = {false, SUP_DEBUG_STR("q1"), &sup_q1_t0};
3 ...
```

Source: Author (2023).

Finally, Source Code 5 shows the structure of the supervisor, which will compose the list of supervisors (in the file "src/supervisors/supervisor_list.c") used by the event handler to query the event status and perform transitions. The variable of type 'Supervisor' is declared and initialized with pointers to the initial state, the current state, the last state (NULL because it has not yet been executed), the first item in the event list, and the supervisor's name.

Source Code 5 – Creating and linking the supervisor

```
1 Supervisor sup = {&sup_q0, &sup_q0, NULL,  &sup_e1_evt0, "sup"};
```

Source: Author (2023).

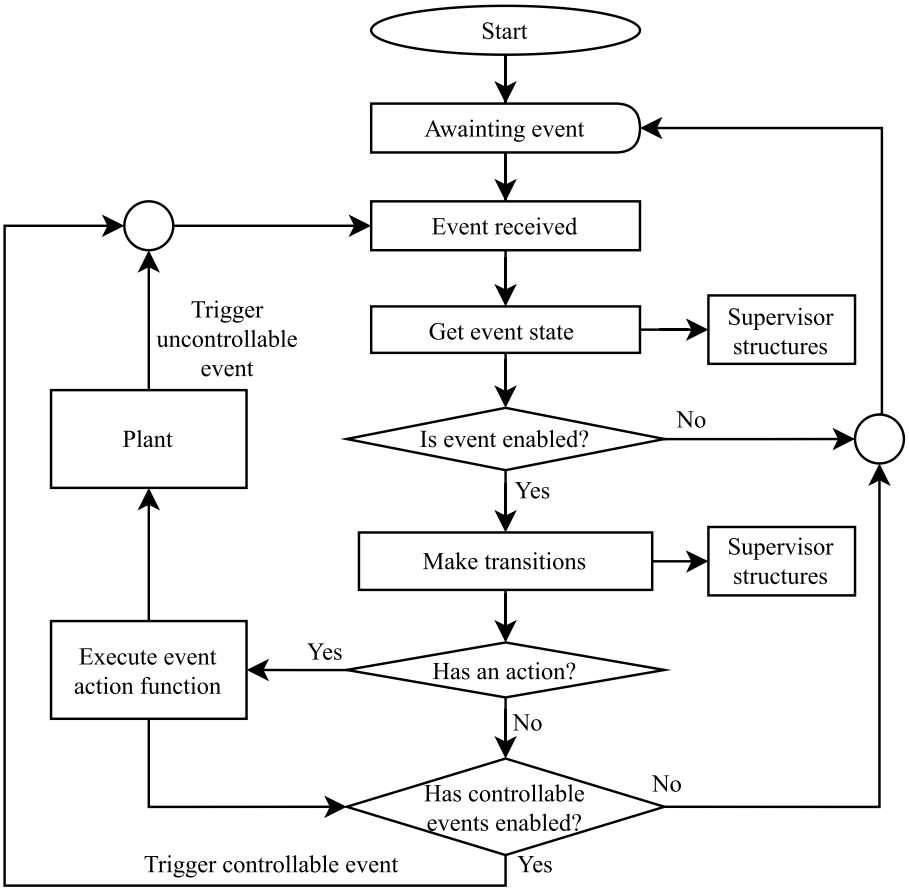
The event handler is proposed to check whether an event is enabled and make the transitions in the automata accordingly.

4.3 The Event Handler

The event handler manages the events of the controller. For the correct operation of the proposed code, every event must be submitted to the event handler through the "trigger_event" function. The event handling checks whether the event is enabled by supervisors. If enabled, it is executed. Executing an event means making the transitions in the FSM, calling the function associated with the event action, and managing the controllable events enabled in the

plant. Figure 30 presents a flow chart to explain the operation of the event handler, including its interaction with plant and supervisors.

Figure 30 – Flowchart for the event handler operation.



Source: Author (2023).

The modular control approach is incorporated into the project in the event handler. It is implemented through the list of supervisors, in "src/supervisors/supervisor_list.c". This allows the event handler to check that the event is enabled on all supervisors before executing it. If an event is not enabled in a supervisor, then it is not enabled in the plant, which materializes the idea of event disabling for the conjunction of modular controllers.

4.4 The User Area

The user area includes the functions for interfacing with the physical plant, defined specifically for each target platform. In the *Interface* block, the process signals are read, converted into uncontrollable events, and transmitted to the event handler. In the *Event actions* block, functions are created with the specific action that should be executed whenever an enabled event is handled. These functions must be associated with the events. Usually, each controllable event has an associated action, such as turning on or off a digital output.

A complete case study in which inputs and outputs are read and controlled remotely, using the Modbus TCP protocol, can be found in Appendix A and in Possato *et al.* (2024), Possato (2023c).

4.5 Considerations about DEScMaker

The DEScMaker can generate code for monolithic and LMC-P controllers. Compared to Assmann *et al.* (2021), DEScMaker has a more straightforward structure and is more intuitive to use. It is also easier to integrate with existing projects as it does not rely on third-party libraries. The design of the project also facilitates the implementation of hybrid control by allowing the user to control the execution of the control algorithms in the user area.

The event handler is a good way to interconnect the supervisor and the physical plant. At the same time, it acts as an interface between these two layers, which allows replacement, for example, the implementation structure of the supervisors without the need for user code refactoring. In the case of a decentralized architecture, with the supervisors' state machines deployed on different hardware, the event handler is the agent that communicates with these external devices.

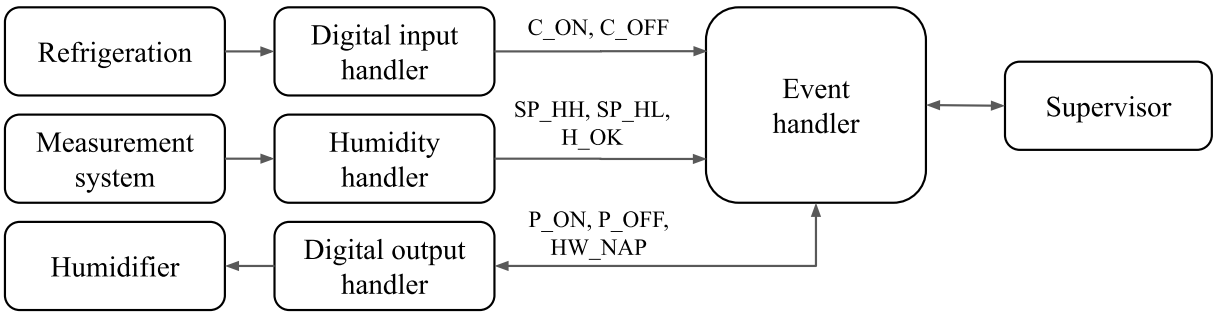
The tool that generates code in Python (POSSATO, 2023b; POSSATO, 2023e) has the structure of operation and use similar to DEScMaker. The Python implementation is easier to handle, test, and debug since the code compilation step is suppressed. Furthermore, it is data science-oriented and can straightforwardly gather cognitive skills for SCT.

The next Chapter demonstrates the practical application of DEScMaker in controlling RH in cold chambers.

5 INTEGRATION OF MS AND CONTROL STRATEGY

The obtained controller that implements the control strategy proposed in Chapter 3 was converted into C code with DEScMaker, presented in Chapter 4. The resulting code was loaded onto a control board developed for this work. The RH *MS*, proposed in Chapter 2, was connected to the same hardware. The Figure 31 presents a diagram showing the main components of the application and their association with the events. The refrigeration state is captured by digital input and the C_ON and C_OFF events are passed to the event handler. A dedicated task reads the RH values and emits the SP_HL, H_OK, and SP_HH events.

Figure 31 – Diagram of integration between the main components of the application.



Source: Author (2023).

The humidifier is controlled by events P_ON and P_OFF, which have associated actions to turn a digital output of the controller on and off when called by the event handler, which is created by DEScMaker. Finally, the timer that triggers the HW_NAP event is started by the same function associated with the P_ON event. An on-off controller was also implemented to obtain reference results for comparison. The temperature was kept between 0.5 and 2°C in all experiments. The configuration of the RH set-points was $SPl = 92\%rh$ and $SPh = 95\%rh$.

The main metric used to evaluate the results is the Root Mean Squared Error (RMSE). It quantitatively measures the difference between the programmed and actual values. It is calculated by the square root of the mean square error. It is useful because it penalizes significant errors more than minor errors. Two RMSE calculations are performed. One for values above the maximum set point (RMSE_H) and one below the minimum set point (RMSE_L). The following equations show the RMSE calculations, where n is the number of samples and h_i is the value of the RH sample.

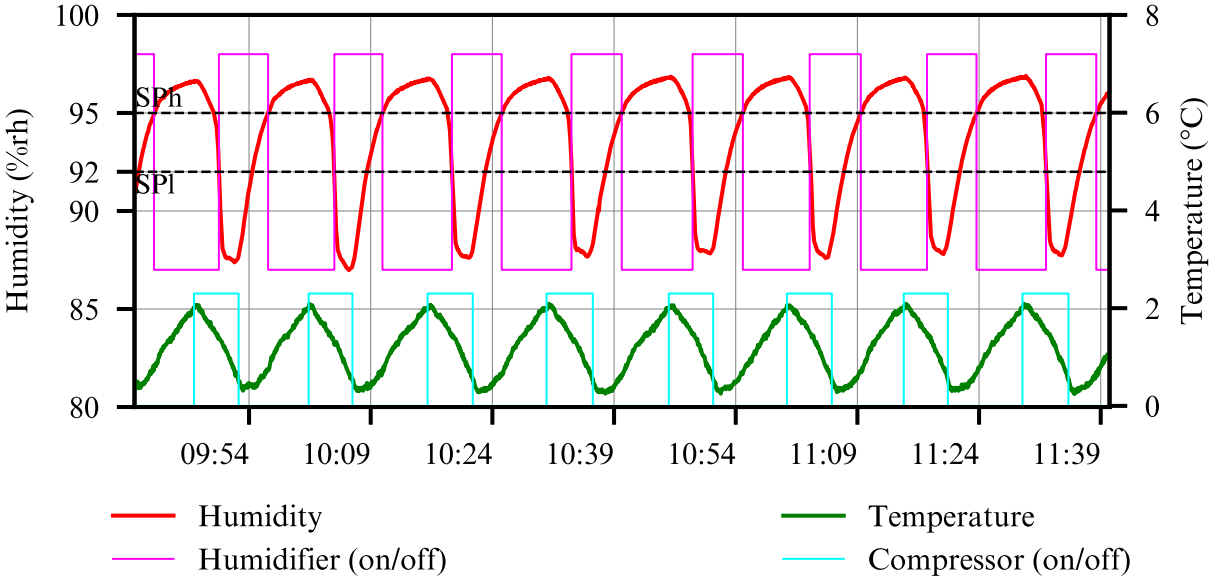
$$RMSE\_H = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(max(0, h_i - SPh)^2)} \tag{1}$$

$$RMSE\_L = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(max(0, SPl - h_i)^2)} \tag{2}$$

5.1 Results and Discussions

The data presented in Figure 32 show the humidifier under on-off control. In all cycles, RH is lower than SPl and higher than SPh. This is because the controller acts on the humidifier only when the set points are reached. Therefore, the system is influenced by the measurement delay, the time required to homogenize RH in the room, and the coupling of RH with the temperature and the refrigeration system.

Figure 32 – Two hours of humidification and refrigeration under the control of the on-off method. The humidifier and RH data are on the left axis. On the right axis, temperature and compressor data. The mean RH value is 93.7%rh ±3.23.



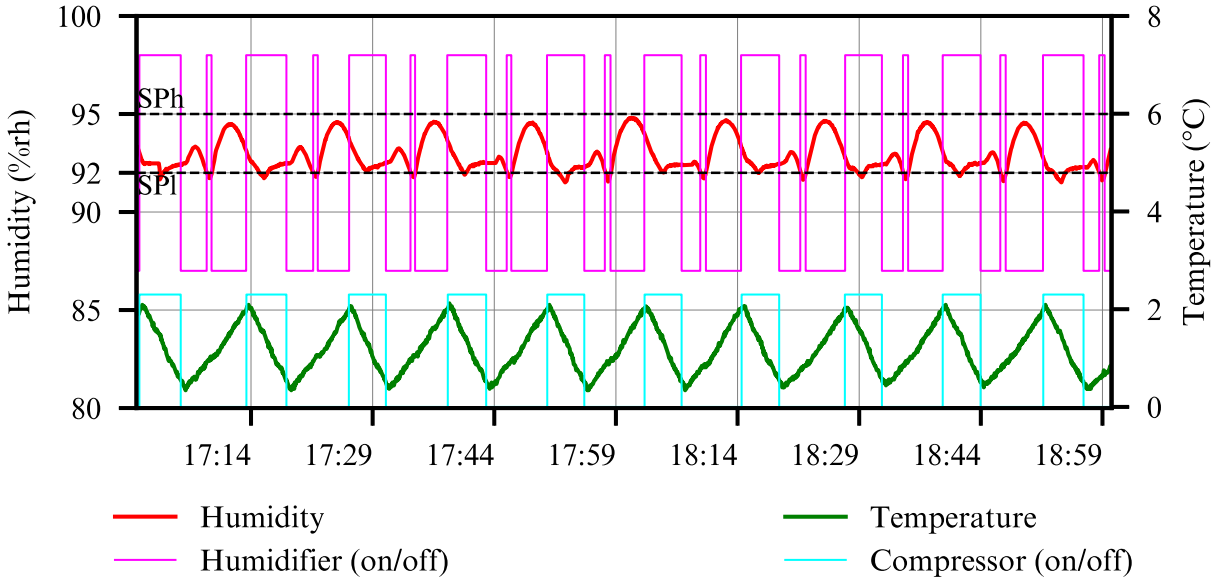
Source: Author (2023).

Figure 33 shows the data collected with the system under control as proposed. The humidifier starts with the compressor when RH is less than SPh. The humidifier is turned off with the compressor when the humidity exceeds SPl. In this way, the system anticipates changes in RH imposed by refrigeration.

Table 2 presents the statistical analysis of the collected data. A gain in almost all indicators is observed, highlighting the efficiency of the proposed method. There is a decrease in the standard deviation, which brings RH within the limits. Reduced RMSE indicates fewer measurement peaks, which are closer to the reference values. It is also important to note the decrease of the out-of-limit indicator (time over SPh and below SPl). This shows that controlling the RH with the proposed method exposes the stored product to shorter periods under non-standard conditions.

The data obtained can be compared with measurements taken in a commercial chamber. In the data in Figure 32, the RMSE for values above the programmed value is 2.19%rh, and for values below it is 0.29%rh. The RH remained 65.22% of the time above the SPh limit and

Figure 33 – Two hours of humidification under control as proposed. The humidifier and RH data are on the left axis. On the right axis, temperature and compressor data. The mean RH value is 92.9%rh ±0.87.



Source: Author (2023).

Table 2 – Statistics of the data presented

	On-off	**Proposed method**	Gain (%)
Average (%rh)	93.7	92.9	-
Standard deviation (%rh)	±3.23	±0.87	73.0
RMSE_H (%rh)	0.94	0.0	100.0
RMSE_L (%rh)	1.77	0.07	96.0
Time over SPh (%)	54.0	0.0	100.0
Time under SPl (%)	27.4	10.2	62.8
Humidifier activation count	9	20	-122.2
Humidification time (%)	42.3	43.8	-3.5

Source: Author (2023).

10.43% below the SPl. Since in the experiments carried out, RH remained 100% of the time below the SPh limit, we can assume that the proposed method can improve the RH indicators in that chamber.

The increased number of times the humidifier has been turned on does not represent a problem. The specification E^3 in Figure 24 prevents the humidifier from actuating at an unsafe frequency. In works such as that of Ito e Tabei (2021), the constant actuation of the humidifier is observed. To implement this, equipment that accepts control at a relatively high frequency is necessary, which could be unfeasible, especially for small and medium producers.

In Delele *et al.* (2009) and Guo *et al.* (2020), studies are carried out on the humidification process in cold chambers. However, the focus is on optimizing the process as a whole using Computational Fluid Dynamics (CFD). In Delele *et al.* (2009), the humidifier is turned on using

on-off by time. This provided an average RH of 96.3%rh, but the standard deviation and other statistics are not presented. The results achieved can be applied together with the proposals in this work to further improve the cold chamber humidification process.

6 CONCLUSION

This study highlights the importance of managing RH in cold chambers to preserve apples. The challenges of RH measurement and the impact of low ventilation and water condensation on the sensors due to low temperature, high RH, and temperature fluctuations were detailed. In addition, methods and research in the RH control in cold chambers were evaluated. After comparison, a strategy based on Discrete Event System (DES) was adopted, primarily due to its lack of parametric dependency.

The proposed *MS* showed results similar to the standard, and the proposed techniques confirmed their effectiveness in improving the correlation between the two systems. It also adds reliability to the data since the method of detecting and recovering from saturation keeps the sensor working at all times, independently of adverse conditions. Saturation detection and recovery need further investigation to improve the results. It is possible that, with adjustments to the parameters used to move forward in the stages of the process, the goals will be achieved more successfully. Considering that the cost of building the proposed system is around US$100.00, the proposal is attractive for application in small and medium-sized apple producers.

As for the control strategy, the process was modeled based on the operation of the humidifier and the cooling system. The models incorporated the dependence between temperature and RH and the interference of refrigeration on air RH. Mathematical operations achieved a FSM for process control. A tool DEScMaker was developed to generate code from the automaton, which was deployed to control a test chamber. Finally, data were collected and analyzed.

The humidifier control strategy successfully achieved its objectives. The refrigeration characteristics were effectively incorporated into the RH control. These results demonstrate the possibility of using FSM for more sophisticated RH control methods than simple on-off control, with the advantage of not depending on specific parameters.

Combining the enhanced *MS* and the DES-based control strategy demonstrated a reliable alternative for RH control in cold chambers. Mounting the RH sensor with the fan is a simple and cost-effective solution. It can be done independently of changing the sensor readout programming and applies to any RH sensor. The integrated operation of the proposed *MS* and control was effectively maintained RH within the desired limits, adding more reliability to the process.

For practical application, DEScMaker converts a supervisor modeled with FSM into C code, as well as providing all the necessary structures for implementing the controller compatible with SCT. DEScMaker handles ordinary FSM but can be extended to support other types of FSM.

Future work could include adding trigger levels to the humidifier to control the reactivity of actions, allowing for more precise adjustments. The control strategy could also be validated at an industrial level, with the objective of further developing and better integrating the *MS* and the control proposal. In DEScMaker, due to the execution nature of the event handler, interleav-

ing is not yet addressed in this work and is recommended for future work. The DEScMaker, as presented, is only designed to deal with ordinary automata. Other concepts, such as timed automata, FSM delay, or statecharts, are not yet part of the scope of this work and may be added in the future. Finally, the use of tables instead of linked lists to represent transitions will be investigated and compared.

BIBLIOGRAPHY

AKDEMIR, S.; BAL, E. Quality changes in apple in evaporative cooling store. **Erwerbs-Obstbau**, v. 62, p. 61–67, 3 2020. ISSN 0014-0309. Disponível em: https://doi.org/10.1007/s10341-019-00458-w. Acesso em: 28 dez. 2023.

AKESSON, K. *et al.* **Supremica**. [*S.l.*], 2023. Disponível em: http://www.supremica.org/. Acesso em: 28 dez. 2023.

ALVES, L. V.; MARTINS, L. R.; PENA, P. N. UltraDES - A Library for Modeling, Analysis and Control of DES. **IFAC-PapersOnLine**, v. 50, p. 5831–5836, 7 2017. Disponível em: https://doi.org/10.1016/j.ifacol.2017.08.540. Acesso em: 28 dez. 2023.

ANESE, R. O. *et al.* Mass loss by low relative humidity increases gas diffusion rates in apple flesh and allows the use of high CO_2 partial pressures during ultralow O_2 storage. **Scientia Horticulturae**, v. 198, p. 414–423, 1 2016. ISSN 03044238. Disponível em: https://doi.org/10.1016/j.scienta.2015.12.015. Acesso em: 28 dez. 2023.

ANESE, R. O.; FRONZA, D. **Fisiologia Pós-Colheita em Fruticultura**. UFSM: Colégio Politécnico, 2015. ISBN 9788563573896. Disponível em: https://bit.ly/3Q2FgI2. Acesso em: 28 dez. 2023.

ARGENTA, L. C. *et al.* Maintenance of 'luiza' apple fruit quality as affected by postharvest practices. **Revista Brasileira de Fruticultura**, v. 44, 2022. Disponível em: https://doi.org/10.1590/0100-29452022905. Acesso em: 28 dez. 2023.

ARGENTA, L. C. *et al.* Characterization and quantification of postharvest losses of apple fruit stored under commercial conditions. **HortScience**, v. 56, p. 608–616, 5 2021. ISSN 0018-5345. Disponível em: https://doi.org/10.21273/HORTSCI15771-21. Acesso em: 28 dez. 2023.

ASSMANN, J. V. *et al.* Distributed embedded platform for controllers following the sct. *In*: **Int. Conf. on Electrical, Communication, and Computer Engineering**. IEEE, 2021. p. 1–6. Disponível em: https://doi.org/10.1109/ICECCE52056.2021.9514113. Acesso em: 28 dez. 2023.

BAHRAMNIA, P. *et al.* Modeling and controlling of temperature and humidity in building heating, ventilating, and air conditioning system using model predictive control. **Energies**, v. 12, p. 4805, 12 2019. ISSN 1996-1073. Disponível em: https://doi.org/10.3390/en12244805. Acesso em: 28 dez. 2023.

BAI, L. *et al.* High-performance humidity sensor based on plasma modified graphene. **Applied Physics Express**, v. 15, p. 106501, 10 2022. ISSN 1882-0778. Disponível em: https://doi.org/10.35848/1882-0786/ac8d48. Acesso em: 28 dez. 2023.

BASILE, F.; CHIACCHIO, P. On the implementation of supervised control of discrete event systems. **IEEE Trans. on Control Systems Technology**, v. 15, p. 725–739, 7 2007. Disponível em: https://doi.org/10.1109/TCST.2006.890281. Acesso em: 28 dez. 2023.

BEEK, D. A. van *et al.* Cif 3: Model-based engineering of supervisory controllers. *In*: ____. **Tools and Algorithms for the Construction and Analysis of Systems**. Springer, 2014. cap. Tool Demonstrations, p. 575–580. Disponível em: https://doi.org/10.1007/978-3-642-54862-8_48. Acesso em: 28 dez. 2023.

BISHNOI, R.; AHARWAL, K. R. Experimental and theoretical analysis of mass transfer in a refrigerated food storage. **Heat and Mass Transfer/Waerme- und Stoffuebertragung**, Springer Science and Business Media Deutschland GmbH, p. 1–11, 4 2022. ISSN 14321181. Disponível em: https://doi.org/10.1007/S00231-022-03217-Y. Acesso em: 28 dez. 2023.

BLANK, T.; EKSPERIANDOVA, L.; BELIKOV, K. Recent trends of ceramic humidity sensors development: A review. **Sensors and Actuators B: Chemical**, v. 228, p. 416–442, 6 2016. ISSN 09254005. Disponível em: https://doi.org/10.1016/j.snb.2016.01.015. Acesso em: 28 dez. 2023.

BRACKMANN, A. *et al.* Relative humidity and its interaction with the wounds on decay incidence and ripening of 'galaxy' apples during cold storage. **Bioscience Journal**, p. 857–862, 2016. ISSN 19813163. Disponível em: https://doi.org/10.14393/BJ-v32n4a2016-29189. Acesso em: 28 dez. 2023.

CARNEIRO, R.; GASPAR, P.; SILVA, P. 3D and transient numerical modeling of door opening and closing processes and its influence on thermal performance of cold rooms. **Applied Thermal Engineering**, v. 113, p. 585–600, 2017. ISSN 1359-4311. Disponível em: https://doi.org/10.1016/j.applthermaleng.2016.11.046. Acesso em: 28 dez. 2023.

CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. 3. ed. Springer International Publishing, 2021. ISBN 978-3-030-72272-2. Disponível em: https://doi.org/10.1007/978-3-030-72274-6. Acesso em: 28 dez. 2023.

CHEN, W.-H.; MATTSON, N. S.; YOU, F. Intelligent control and energy optimization in controlled environment agriculture via nonlinear model predictive control of semi-closed greenhouse. **Applied Energy**, v. 320, p. 119334, 8 2022. ISSN 03062619. Disponível em: https://doi.org/10.1016/j.apenergy.2022.119334. Acesso em: 28 dez. 2023.

CLAVIJO, L. B.; BASILIO, J. C.; CARVALHO, L. K. DESLAB: A scientific computing program for analysis and synthesis of discrete-event systems. **IFAC Proceedings Volumes**, v. 45, p. 349–355, 2012. Disponível em: https://doi.org/10.3182/20121003-3-MX-4033.00056. Acesso em: 28 dez. 2023.

CUI, S. *et al.* Study on decoupling control system of temperature and humidity in intelligent plant factory. *In*: **2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)**. IEEE, 2020. v. 9, p. 2160–2163. ISBN 978-1-7281-5244-8. ISSN 2693-2865. Disponível em: https://doi.org/10.1109/ITAIC49862.2020.9339036. Acesso em: 28 dez. 2023.

DELELE, M. *et al.* Optimization of the humidification of cold stores by pressurized water atomizers based on a multiscale cfd model. **Journal of Food Engineering**, v. 91, p. 228–239, 3 2009. ISSN 02608774. Disponível em: https://doi.org/10.1016/j.jfoodeng.2008.08.027. Acesso em: 28 dez. 2023.

DOSTáL, J.; FERKL, L. Model predictive control of climatic chamber with on-off actuators. *In*: **IFAC Proceedings Volumes**. [*s.n.*], 2014. v. 47, p. 4423–4428. ISSN 14746670. Disponível em: https://doi.org/10.3182/20140824-6-ZA-1003.01571. Acesso em: 28 dez. 2023.

ENDO, A. *et al.* Fuzzy and neural prediction intervals for robust control of a greenhouse. *In*: **2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. IEEE, 2022. p. 1–8. ISBN 978-1-6654-6710-0. Disponível em: https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882701. Acesso em: 28 dez. 2023.

FARAHANI, H.; WAGIRAN, R.; HAMIDON, M. Humidity sensors principle, mechanism, and fabrication technologies: A comprehensive review. **Sensors**, v. 14, p. 7881–7939, 4 2014. ISSN 1424-8220. Disponível em: https://doi.org/10.3390/s140507881. Acesso em: 28 dez. 2023.

FOKKINK, W. *et al.* Supervisor synthesis: Bridging theory and practice. **Computer**, v. 55, p. 48–54, 10 2022. Disponível em: https://doi.org/10.1109/MC.2021.3134934. Acesso em: 28 dez. 2023.

FUADY, G. M. *et al.* Extreme learning machine and back propagation neural network comparison for temperature and humidity control of oyster mushroom based on microcontroller. *In*: **2017 International Symposium on Electronics and Smart Devices (ISESD)**. IEEE, 2017. p. 46–50. ISBN 978-1-5386-2778-5. Disponível em: https://doi.org/10.1109/ISESD.2017.8253303. Acesso em: 28 dez. 2023.

GINJA, G. A. *et al.* A humidity sensor based on bacterial nanocellulose membrane (bnc). **IEEE Sensors Journal**, v. 23, p. 3485–3492, 2 2023. ISSN 1530-437X. Disponível em: https://doi.org/10.1109/JSEN.2023.3234222. Acesso em: 28 dez. 2023.

GOBE, F. *et al.* Synthesis tool for automation controller supervision. *In*: **International Workshop on Discrete Event Systems**. IEEE, 2016. p. 424–431. Disponível em: https://doi.org/10.1109/WODES.2016.7497883. Acesso em: 28 dez. 2023.

GUO, J. *et al.* Characteristic analysis of humidity control in a fresh-keeping container using cfd model. **Computers and Electronics in Agriculture**, v. 179, p. 105816, 12 2020. ISSN 01681699. Disponível em: https://doi.org/10.1016/j.compag.2020.105816. Acesso em: 28 dez. 2023.

HASDEMIR, I. T.; KURTULAN, S.; GOREN, L. An implementation methodology for supervisory control theory. **Int. Journal of Advanced Manufacturing Technology**, v. 36, p. 373–385, 3 2008. Disponível em: https://doi.org/10.1007/s00170-006-0843-5. Acesso em: 28 dez. 2023.

HEIDARI, M.; KHODADADI, H. Climate control of an agricultural greenhouse by using fuzzy logic self-tuning pid approach. *In*: **2017 23rd International Conference on Automation and Computing (ICAC)**. IEEE, 2017. p. 1–6. ISBN 978-0-7017-0260-1. Disponível em: https://doi.org/10.23919/IConAC.2017.8082074. Acesso em: 28 dez. 2023.

HMSO. **A Guide to the Measurement of Humidity**. [*S.l.*], 1996. Disponível em: https://bit.ly/3ROxhQe. Acesso em: 28 dez. 2023.

HU, C. *et al.* Design method of the heating system for controllable heating humidity sensor. *In*: **2020 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)**. IEEE, 2020. p. 62–68. Disponível em: https://doi.org/10.1109/ICSMD50554.2020.9261685. Acesso em: 28 dez. 2023.

IBGE. **Produção de Maçã**. 2022. Disponível em: https://bit.ly/3RLGb15. Acesso em: 28 dez. 2023.

ITO, K.; TABEI, T. Model predictive temperature and humidity control of greenhouse with ventilation. *In*: **Procedia Computer Science**. [*s.n.*], 2021. v. 192, p. 212–221. ISSN 18770509. Disponível em: https://doi.org/10.1016/j.procs.2021.08.022. Acesso em: 28 dez. 2023.

KAEWWISET, T.; YODKHAD, P. Automatic temperature and humidity control system by using fuzzy logic algorithm for mushroom nursery. *In*: **2017 International Conference**

on Digital Arts, Media and Technology (ICDAMT). IEEE, 2017. p. 396–399. ISBN 978-1-5090-5210-3. Disponível em: https://doi.org/10.1109/ICDAMT.2017.7905000. Acesso em: 28 dez. 2023.

KUZUBASOGLU, B. A. Recent studies on the humidity sensor: A mini review. **ACS Applied Electronic Materials**, 9 2022. ISSN 2637-6113. Disponível em: https://doi.org/10.1021/acsaelm.2c00721. Acesso em: 28 dez. 2023.

LEAL, A. B.; CRUZ, D. L. L. da; HOUNSELL, M. da S. Supervisory control implementation into programmable logic controllers. *In*: **Int. Conf. on Emerging Technologies and Factory Automation**. IEEE, 2009. p. 1–7. Disponível em: https://doi.org/10.1109/ETFA.2009.5347090. Acesso em: 28 dez. 2023.

LINKE, M. *et al.* Water vapour condensation on the surface of bulky fruit: Some basics and a simple measurement method. **Journal of Food Engineering**, v. 307, p. 110661, 10 2021. ISSN 02608774. Disponível em: https://doi.org/10.1016/j.jfoodeng.2021.110661. Acesso em: 28 dez. 2023.

LINKE, M. *et al.* Measurement of water vapor condensation on apple surfaces during controlled atmosphere storage. **Sensors**, v. 23, p. 1739, 2 2023. ISSN 1424-8220. Disponível em: https://doi.org/10.3390/s23031739. Acesso em: 28 dez. 2023.

LIU, E. *et al.* An overview of flexible sensors: Development, application, and challenges. **Sensors**, v. 23, p. 817, 1 2023. ISSN 1424-8220. Disponível em: https://doi.org/10.3390/s23020817. Acesso em: 28 dez. 2023.

LJUNGKRANTZ, O. *et al.* Implementing a control system framework for automatic generation of manufacturing cell controllers. *In*: **Proceedings 2007 IEEE International Conference on Robotics and Automation**. IEEE, 2007. p. 674–679. Disponível em: https://doi.org/10.1109/ROBOT.2007.363064. Acesso em: 28 dez. 2023.

LORENCENA, M. C. *et al.* A framework for modelling, control and supervision of poultry farming. **International Journal of Production Research**, v. 58, p. 3164–3179, 5 2020. ISSN 0020-7543. Disponível em: https://doi.org/10.1080/00207543.2019.1630768. Acesso em: 28 dez. 2023.

LUFU, R.; AMBAW, A.; OPARA, U. L. Water loss of fresh fruit: Influencing pre-harvest, harvest and postharvest factors. **Scientia Horticulturae**, v. 272, p. 109519, 10 2020. ISSN 03044238. Disponível em: https://doi.org/10.1016/j.scienta.2020.109519. Acesso em: 28 dez. 2023.

MALIK, R. *et al.* Supremica–an efficient tool for large-scale discrete event systems. **IFAC-PapersOnLine**, v. 50, p. 5794–5799, 7 2017. Disponível em: https://doi.org/10.1016/j.ifacol.2017.08.427. Acesso em: 28 dez. 2023.

MATAMOROS, M. *et al.* Temperature and humidity pid controller for a bioprinter atmospheric enclosure system. **Micromachines**, v. 11, p. 999, 11 2020. ISSN 2072-666X. Disponível em: https://doi.org/10.3390/mi11110999. Acesso em: 28 dez. 2023.

MCCARTHY, D. *et al.* A model-based approach to automated validation and generation of plc code for manufacturing equipment in regulated environments. **Applied Sciences**, v. 12, p. 7506, 7 2022. Disponível em: https://doi.org/10.3390/app12157506. Acesso em: 28 dez. 2023.

MEMON, M. M. *et al.* Surface acoustic wave humidity sensor based on hydrophobic polymer film. **Journal of Electronic Materials**, v. 51, p. 5627–5634, 10 2022. ISSN 0361-5235. Disponível em: https://doi.org/10.1007/s11664-022-09791-5. Acesso em: 28 dez. 2023.

MOHAJERANI, S.; MALIK, R.; FABIAN, M. Compositional synthesis of supervisors in the form of state machines and state maps. **Automatica**, v. 76, p. 277 – 281, 2017. Disponível em: https://doi.org/10.1016/j.automatica.2016.10.012. Acesso em: 28 dez. 2023.

MOHAMMED, M.; ALQAHTANI, N.; EL-SHAFIE, H. Development and evaluation of an ultrasonic humidifier to control humidity in a cold storage room for postharvest quality management of dates. **Foods**, v. 10, p. 949, 4 2021. ISSN 2304-8158. Disponível em: https://doi.org/10.3390/foods10050949. Acesso em: 28 dez. 2023.

MONTOYA, A. *et al.* Design and implementation of a low-cost sensor network to monitor environmental and agronomic variables in a plant factory. **Computers and Electronics in Agriculture**, v. 178, p. 105758, 11 2020. ISSN 01681699. Disponível em: https://doi.org/10.1016/j.compag.2020.105758. Acesso em: 28 dez. 2023.

MOOR, T.; SCHMIDT, K.; PERK, S. libFAUDES - An open source C++ library for discrete event systems. *In*: **2008 9th International Workshop on Discrete Event Systems**. IEEE, 2008. p. 125–130. ISBN 978-1-4244-2592-1. Disponível em: https://doi.org/10.1109/WODES.2008.4605933. Acesso em: 28 dez. 2023.

PINHEIRO, L. P. *et al.* Nadzoru: A software tool for supervisory control of DES. **IFAC-PapersOnLine**, v. 48, p. 182–187, 2015. Disponível em: https://doi.org/10.1016/j.ifacol.2015.06.491. Acesso em: 28 dez. 2023.

POSSATO, T. **Automated code generator from Supremica to C**. 2023. Disponível em: https://bit.ly/3DTsCEu. Acesso em: 28 dez. 2023.

POSSATO, T. **Automated code generator from Supremica to Python**. 2023. Disponível em: https://bit.ly/3saIY99. Acesso em: 28 dez. 2023.

POSSATO, T. **DEScMaker case study**. 2023. Disponível em: https://bit.ly/47w3zoA. Acesso em: 28 dez. 2023.

POSSATO, T. **DEScMaker case study video**. 2023. Disponível em: https://bit.ly/3KFqELY. Acesso em: 28 dez. 2023.

POSSATO, T. **DESPythonMaker case study**. 2023. Disponível em: https://bit.ly/3OD0haF. Acesso em: 28 dez. 2023.

POSSATO, T.; TEIXEIRA, M.; COSTA, J. P. D. A relative humidity measurement system tolerant to condensation events applied to apple storage. *In*: **2023 7th International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)**. [*s.n.*], 2023. p. 1–6. Disponível em: https://doi.org/10.1109/INSCIT59673.2023.10258509. Acesso em: 28 dez. 2023.

POSSATO, T. *et al.* Automated Code Generation for DES Controllers Modeled as Finite State Machines. *In*: BARBOSA, H.; ZOHAR, Y. (Ed.). **Formal Methods: Foundations and Applications**. Cham: Springer Nature Switzerland, 2024. p. 113–130. ISBN 978-3-031-49342-3. Disponível em: https://doi.org/10.1007/978-3-031-49342-3_7. Acesso em: 28 dez. 2023.

PRAEGER, U. *et al.* Influence of room layout on airflow distribution in an industrial fruit store. **International Journal of Refrigeration**, Elsevier, v. 131, p. 714–722, 11 2021. ISSN

0140-7007. Disponível em: https://doi.org/10.1016/J.IJREFRIG.2021.06.016. Acesso em: 28 dez. 2023.

PRAEGER, U. *et al.* Water condensation on fruit surfaces during apple storage. **Acta Horticulturae**, p. 591–596, 11 2021. ISSN 0567-7572. Disponível em: https://doi.org/10.17660/ActaHortic.2021.1327.78. Acesso em: 28 dez. 2023.

QUEIROZ, M.; CURY, J. Synthesis and implementation of local modular supervisory control for a manufacturing cell. *In*: **Sixth International Workshop on Discrete Event Systems, 2002. Proceedings.** IFAC, 2002. p. 377–382. Disponível em: https://doi.org/10.1109/WODES.2002.1167714. Acesso em: 28 dez. 2023.

QUEIROZ, M.; CURY, J. Modular multitasking supervisory control of composite discrete-event systems. **16th IFAC World Congress**, Prague, Czech Republic, 2005. Disponível em: https://doi.org/10.3182/20050703-6-CZ-1902.00300. Acesso em: 28 dez. 2023.

RAMADGE, P.; WONHAM, W. The control of discrete event systems. **Proceedings of the IEEE**, v. 77, p. 81–98, 1989. ISSN 00189219. Disponível em: https://doi.org/10.1109/5.21072. Acesso em: 28 dez. 2023.

RAMADGE, P. J.; WONHAM, W. M. The control of discrete event systems. **Proceedings of the IEEE**, IEEE, v. 77, n. 1, p. 81–98, 1989. Disponível em: https://doi.org/10.1109/5.21072. Acesso em: 28 dez. 2023.

RENIERS, M.; MORTEL-FRONCZAK, J. van de. An engineering perspective on model-based design of supervisors. **IFAC-PapersOnLine**, v. 51, p. 257–264, 2018. Disponível em: https://doi.org/10.1016/j.ifacol.2018.06.310. Acesso em: 28 dez. 2023.

ROSA, M.; TEIXEIRA, M.; MALIK, R. Exploiting approximations in supervisory control with distinguishers. *In*: **International Workshop on Discrete Event Systems**. Sorrento, Italy: [*s.n.*], 2018. p. 13–18. Disponível em: https://doi.org/10.1016/j.ifacol.2018.06.272. Acesso em: 28 dez. 2023.

SCHMIDT, L. *et al.* A combined solution for flexible control of poultry houses. **International Journal of Computer Applications in Technology**, v. 67, p. 232, 2021. ISSN 0952-8091. Disponível em: https://doi.org/10.1504/IJCAT.2021.121539. Acesso em: 28 dez. 2023.

SCHUDEL, S. *et al.* Solution roadmap to reduce food loss along your postharvest supply chain from farm to retail. **Food Packaging and Shelf Life**, v. 36, p. 101057, 4 2023. ISSN 22142894. Disponível em: https://doi.org/10.1016/j.fpsl.2023.101057. Acesso em: 28 dez. 2023.

SENSIRION. **Humidity sensors Experts in environmental sensing**. [*S.l.*], 2022. Disponível em: https://bit.ly/46u9QQA. Acesso em: 28 dez. 2023.

SHEN, Y.; WANG, S. Condensation frosting detection and characterization using a capacitance sensing approach. **International Journal of Heat and Mass Transfer**, Pergamon, v. 147, p. 118968, 2 2020. ISSN 00179310. Disponível em: https://doi.org/10.1016/j.ijheatmasstransfer.2019.118968. Acesso em: 28 dez. 2023.

SILVA, Y. G.; QUEIROZ, M. H. de. Formal synthesis, simulation and automatic code generation of supervisory control for a manufacturing cell. *In*: **Symposium Series in Mechatronics**. ABCM, 2010. p. 418–426. Disponível em: https://bit.ly/45l9lXK. Acesso em: 28 dez. 2023.

SINGH, R. P.; HELDMAN, D. R. Introduction to food engineering. *In*: ____. Elsevier, 2014. cap. Psychrometrics, p. 593–616. Disponível em: https://doi.org/10.1016/B978-0-12-398530-9.00009-7. Acesso em: 28 dez. 2023.

SU, Y.; YU, Q.; ZENG, L. Parameter self-tuning pid control for greenhouse climate control problem. **IEEE Access**, v. 8, p. 186157–186171, 2020. ISSN 2169-3536. Disponível em: https://doi.org/10.1109/ACCESS.2020.3030416. Acesso em: 28 dez. 2023.

SULLIVAN, R. P. *et al.* Humidity sensors based on molecular rectifiers. **Nanoscale**, v. 15, p. 171–176, 2023. ISSN 2040-3364. Disponível em: https://doi.org/10.1039/D2NR04498F. Acesso em: 28 dez. 2023.

SUN, Y. *et al.* Intelligent distributed temperature and humidity control mechanism for uniformity and precision in the indoor environment. **IEEE Internet of Things Journal**, v. 9, p. 19101–19115, 10 2022. ISSN 2327-4662. Disponível em: https://doi.org/10.1109/JIOT.2022.3163772. Acesso em: 28 dez. 2023.

TUTUNCU, K.; OZCAN, R. Embedded fuzzy logic control system for refrigerated display cabinets. **Arabian Journal for Science and Engineering**, v. 44, p. 9529–9543, 11 2019. ISSN 2193-567X. Disponível em: https://doi.org/10.1007/s13369-019-03892-w. Acesso em: 28 dez. 2023.

UZAM, M. A general technique for the plc-based implementation of rw supervisors with time delay functions. **The International Journal of Advanced Manufacturing Technology**, v. 62, p. 687–704, 9 2012. Disponível em: https://doi.org/10.1007/s00170-011-3817-1. Acesso em: 28 dez. 2023.

VIEIRA, A. D.; CURY, J. E. R.; QUEIROZ, M. H. de. A model for PLC implementation of supervisory control of DES. *In*: **Int. Conf. on Emerging Technologies and Factory Automation**. [*s.n.*], 2006. p. 225–232. ISBN 0-7803-9758-4. Disponível em: https://doi.org/10.1109/ETFA.2006.355436. Acesso em: 28 dez. 2023.

VIEIRA, A. D. *et al.* A method for PLC implementation of supervisory control of DES. **IEEE Transactions on Control Systems Technology**, v. 25, p. 175–191, 1 2017. Disponível em: https://doi.org/10.1109/TCST.2016.2544702. Acesso em: 28 dez. 2023.

WANG, L.; ZHU, Z. Research on temperature and humidity decoupling control of constant temperature and humidity test chamber. *In*: **IOP Conference Series: Materials Science and Engineering**. [*s.n.*], 2020. v. 711, p. 012104. ISSN 1757-8981. Disponível em: https://doi.org/10.1088/1757-899X/711/1/012104. Acesso em: 28 dez. 2023.

WEBER, B. *et al.* Relative humidity and its interaction with the storage temperature of 'gala' apples and mutants. **Ciência Rural**, v. 42, p. 2159–2165, 2012. ISSN 0103-8478. Disponível em: https://redalyc.org/articulo.oa?id=33124575001. Acesso em: 28 dez. 2023.

ZANELLA, A. R. A.; SILVA, E. da; ALBINI, L. C. P. CEIFA: A multi-level anomaly detector for smart farming. **Computers and Electronics in Agriculture**, v. 202, p. 107279, 2022. ISSN 01681699. Disponível em: https://doi.org/10.1016/j.compag.2022.107279. Acesso em: 28 dez. 2023.

ZHU, Z. *et al.* Method to delay frost formation under high relative humidity by choosing proper heat exchange temperature difference and air velocity. **Case Studies in Thermal Engineering**, v. 45, p. 102922, 5 2023. ISSN 2214157X. Disponível em: https://doi.org/10.1016/j.csite.2023.102922. Acesso em: 28 dez. 2023.

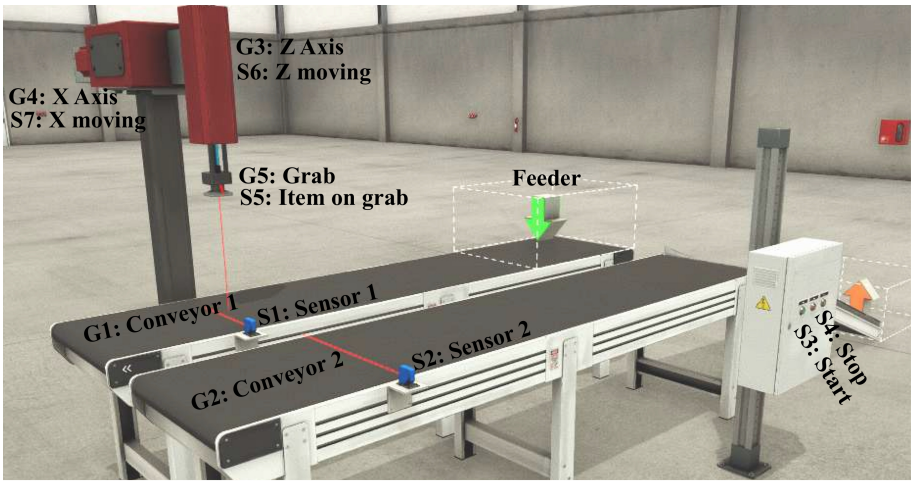
ZIELINSKI, K. M. *et al.* Flexible control of discrete event systems using environment simulation and reinforcement learning. **Applied Soft Computing**, v. 111, p. 107714, 11 2021. ISSN 15684946. Disponível em: https://doi.org/10.1016/j.asoc.2021.107714. Acesso em: 28 dez. 2023.

APPENDIX

APPENDIX A – DEScMaker Case Study

A case study was implemented to illustrate the proposed tool. The experimental plant, shown in Figure 34, consists of conveyors, sensors, a control panel, and a Two-Axis Pick & Place arm. The arm has sensors to indicate if the axes are moving and if there is a grabbed workpiece. The Factory I/O[1] simulator was used to represent the plant and implement the controller. This tool reproduces by software the same conditions as a real plant, in terms of devices, hardware, communication channels and protocols, such as Modbus TCP used here, and many other resources. Yet, it avoids having to maintain the real infrastructure. A variety of scenarios can be constructed using the Factory I/O tool within a short period of time and at low costs, so that this becomes a viable strategy for control engineering practice and test.

Figure 34 – Experimental plant for demonstration.



Source: Author (2023).

The operation consists of moving the workpieces from one conveyor to another. When the start (S_3) button is pressed, the conveyor 1 (G^1) turns on. When sensor 1 (S_1) detects an item, conveyor 01 is turned off. Then, the arm moves down the Z-axis (G^3). When the sensors grab (S_5) detects the item, it activates the grab (G^5), moves up the Z-axis and moves the item to the other conveyor via the X-axis (G^4). When it reaches the destination, it turns off conveyor 2 (G^2), moves the item down to sensor 2 (S_2), and releases it. When released, it turns on conveyor 2 and returns the arm. In parallel, as soon as the workpiece is grabbed and the Z-axis moves up, it turns on the conveyor 1. Sensors S_6 and S_7 indicate whether the Z and X axes are moving.

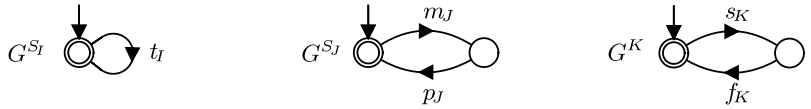
A.1 Plant modeling

Figure 35 shows the automata modeling each equipment of the system presented in Figure 34. Models G^{S_I}, for $I = 1, 2, 3, 4, 5$, represent the behavior of the respective sensor S_I. The models G^{S_J}, for $J = 6, 7$, represent the respective sensor S_J, which indicates, respectively, whether the Z and X axes are moving and paused. Finally, each G^K, for $K = 1, 2, 3, 4, 5$, models

[1] https://factoryio.com

of the respective actuator or motor K. Event t_I indicates that sensor S_I has been triggered; events m_J represents that the axis is moving, while p_J indicates that the axis is paused; events s_K represent the commands that activate actuators K and events f_K represent their switching off. It has been assumed that $\Sigma_{\mathrm{c}} = \{s_K, f_K\}$, while $\Sigma_{\mathrm{u}} = \{t_I, m_J, p_J\}$. The composition $G = G^{S_I} \parallel G^{S_J} \parallel G^K$ is an automaton with 128 states and 1536 transitions.

Figure 35 – Plant models.



Source: Author (2023).

A.2 Specification modeling

The following specifications (see Figure 36) are considered to coordinate the plant.

E^1**:** Coordinate the start of conveyor 1 according to the start button.

E^2**:** Turn off conveyor 1 every time an item arrives at sensor 1.

E^3**:** Disable the activation of the Z-axis whenever the X-axis is moving.

E^4**:** Coordinate the Z-axis drive to pick up and drop the workpiece on the conveyors.

E^5**:** Elevate the Z-axis as the grab picks up or releases a workpiece.

E^6**:** Move the X-axis forward as a workpiece is picked up and the Z-axis rises.

E^7**:** Move the X-axis back as a workpiece is released and the Z-axis rises.

E^8**:** Grab a workpiece whenever the grab sensor detects it.

E^9**:** Release a workpiece whenever sensor 2 detects it.

E^{10}**:** Turn on conveyor 2 whenever a workpiece is dropped on it.

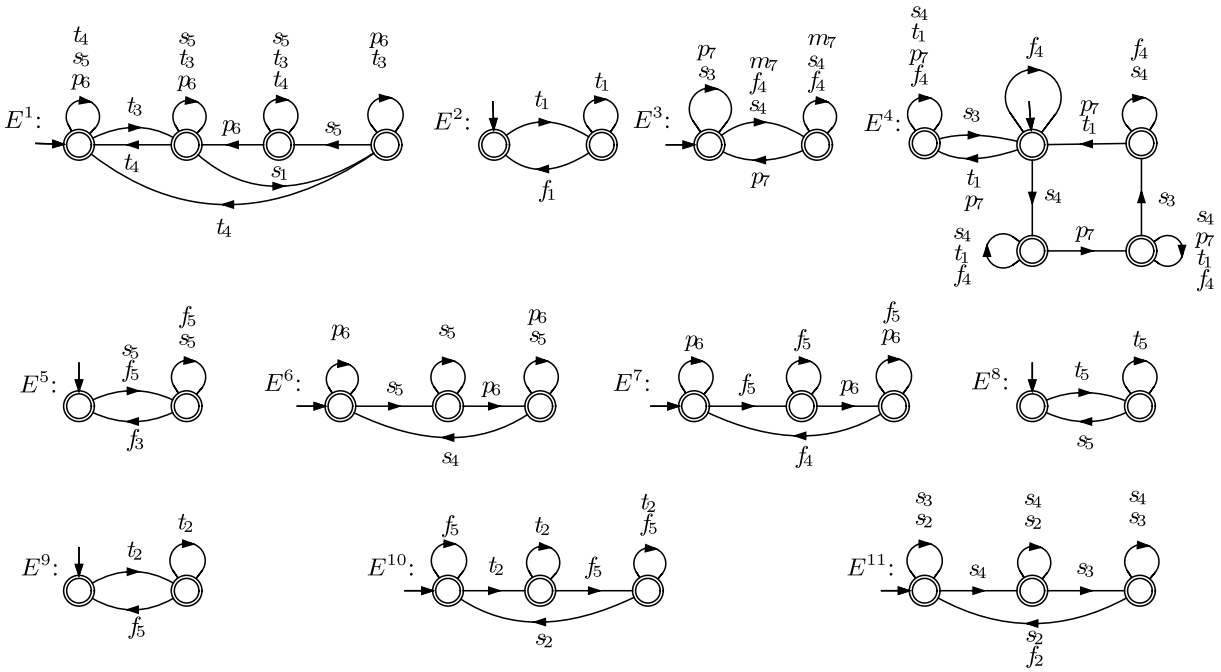
E^{11}**:** Turn off conveyor 2 whenever the Z-axis moves down to unload a workpiece.

The composition $E = \|_{i=1}^{11} E^i$ results in an automaton with 37344 states and 447784 transitions.

A.3 Synthesis

For the synthesis of the controller, the input model $K = G \| E$ resulted in 2507796 states and 22774582 transitions. This gives an idea of the computational complexity to compute the

Figure 36 – Specification models.



Source: Author (2023).

controller, as the synthesis algorithm iterates through the state space searching for controllability and non-blocking violations. Although this operation is polynomial on the state space of K, the model K itself grows exponentially with the size of the systems, which justifies **LMC-P** as a much simpler non-exponential strategy.

In this example, we first apply the monolithic synthesis over K. It turns out that, for the example, K reveals to be controllable and non-blocking, so the synthesized supervisor is the K model itself. Due to the high number of states and transitions, which would complicate the implementation, its maintainability, and make the hardware solution more expensive, **LMC-P** was adopted. This resulted in 11 local supervisors, where the worst-case modular synthesis explored 40 states, in comparison with 2507796 in the monolithic case. It turns out that the composition $\|_{i=1}^{11}\mathrm{S}^i$ returned 2507796 states, therefore the same as in the monolithic case, which means that the **LMC-P** solution is optimal, i.e. $\|_{i=1}^{11}\mathrm{S}^i = K$.

A.4 Conversion and control of the plant

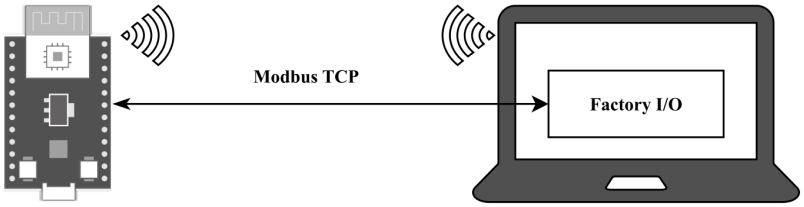
After synthesizing the supervisors, the Supremica file was uploaded to DEScMaker for code generation, which was embedded in a PlatformIO[2] project for the ESP32[3] platform. The routines for connecting to the Wi-Fi network, connecting to the Modbus TCP server (Factory I/O), and the functions for reading the inputs and controlling the outputs have been inserted

[2] https://platformio.org

[3] https://espressif.com/en/products/devkits

in the user section. The user section completes the three-level hierarchy according to Figure 27. Figure 37 demonstrates the architecture of the case study, showing the connection between the microcontroller and the simulator.

Figure 37 – Diagram illustrating the case study. The ESP32 and the computer are connected to the Wi-Fi network, allowing communication between the microcontroller and the simulator through the Modbus TCP protocol.



Source: Author (2023).

The average real time (n=10) to generate the files, measured with the Linux 'time' tool, was 373ms. The ESP32 application's memory usage consists of: The supervisor library, which uses 214 bytes of flash, and the event handler, which occupies 176 bytes of flash and is independent of the size of the FSM. The events occupy 344 bytes of flash and 344 bytes of RAM. The supervisors, in total, consume 9964 bytes of flash and 9964 bytes of RAM. Totaling 10698 bytes of flash and 10308 bytes of RAM. [4]

The complete case study code can be found in (POSSATO, 2023c), which also includes the Factory I/O simulation file and the file with the supervisor modeled in Supremica. In "application/src/main.cpp," it can be seen that there is no control logic, only the reading of the digital inputs, with the triggering of the respective uncontrollable events and the functions to control the outputs associated as "action function" to the controllable events. A video of the plant operation under control can be accessed at (POSSATO, 2023d).

This case study illustrates how DEScMaker converts a FSM into functional code. Additional steps were required for the practical application of the generated code in the proposed scenario, such as the connection via Modbus TCP to the simulator and the Wi-Fi connection. This additional effort is necessary since the code generated is generic and only deals with supervisors and event handling. If the actuators and sensors were connected directly to the microcontroller board, other adjustments would be required for implementation.

[4] The Inspect functionality of PlataformIO was used to measure the use of memory. Available at: https://docs.platformio.org/en/stable/home/index.html#project-inspect