

Creation of a digital twin for a commercial refrigeration facility based on the industry 4.0 platform Kiconex.

Author: Dorao Sánchez-Campos, Gonzalo

Supervisor: Montanary Morales, Antonio

Collaborating Entity: Kiconex

ABSTRACT This project consists of an application capable of managing Digital Twins of different refrigeration equipment. This virtual equipment will be used to simulate various situations and study the behavior of the machines without the need to use a real unit. The data obtained from the simulations will be used to optimize the installations and their operation. Kiconex is an Industry 4.0 platform that collects real-time data from installations that use refrigeration equipment. This data is stored in their database and can be accessed through their web application or via their API. However, all the stored information is past information, as the platform does not have any prediction or simulation capabilities. The objective of this project is to create digital twins of various commercial refrigeration equipment. Also, a tool with a graphical interface will be developed to manage these twins in a comfortable and intuitive manner. Although many different refrigeration units can be simulated, the scope has been focused on the equipment from the Más y Más supermarket in Los Villares, Jaén. The final product is an application with all the capabilities required by Kiconex. In case any new features or a change in the design is needed, it can be easily modified through Python programming using the PySide6 library.

INDEX TERMS IoT, Machine Learning, Digital Twin, Refrigeration, Simulation, Optimization.

I. INTRODUCTION

The objective of this project is the simulation of commercial refrigeration equipment. The Kiconex IoT platform has the capability to store historical data of numerous refrigeration equipment as well as other types of equipment. This data is stored in their database and can be accessed through their web application or via their API. However, all the stored information is past information, as the platform does not have any prediction or simulation capabilities.

Creating a digital twin of a facility with commercial refrigeration equipment will have very extensive coverage in a large number of facilities internationally.

The low development cost of these digital twins together with their reusability is a significant advantage when testing for optimization and design of facilities.

A. Motivation

The creation of a digital twin can address the lack of predictive capabilities mentioned above.

The digital twin will have a simulated controller identical to the real controller, only without the constraints of real-time operation. At any point in time, the digital twin can jump ahead a few minutes into the future, predicting the state of the analog

and digital inputs and calculating the corresponding state of their outputs. In this way, critical situations can be predicted and prevented.

B. Objectives

The project has the following list of technological, economic, and quality objectives.

- 1) TO INCREASE KNOWLEDGE ABOUT THE BEST TECHNOLOGICAL SOLUTIONS FOR DIFFERENT TYPES OF INSTALLATIONS

Thanks to the simulation, it will be possible to test the equipment in a multitude of situations, obtaining large amounts of data that can be used to analyze the performance of different technologies under various conditions. In this way, it will be possible to identify the most efficient and economical technologies for the design of future commercial refrigeration installations.

- 2) TO IMPROVE SYSTEM DESIGN BY PREDICTING UNIQUE OPERATING CONDITIONS THAT GIVE RISE TO ANOMALOUS BEHAVIOR.

Simulation of equipment under unique conditions will allow optimization of the actual equipment configuration to minimize the risk of failure and increase energy efficiency.

In turn, this data can be used by manufacturers to uncover potential critical situations not previously considered and improve their equipment in the next design iteration.

3) TO IMPROVE COLD CHAIN SAFETY THROUGH SIMULATION OF MALFUNCTIONS.

The simulation will allow the identification and correction of possible vulnerabilities in the real system and guarantee the correct maintenance of the cold chain at all times.

C. Alignment with UN Sustainable Development Goals

This project, like all those dealing with the improvement of the industry, is aligned with several Sustainable Development Goals (SDGs), as greater efficiency in industrial processes brings numerous positive repercussions.

1) INDUSTRY, INNOVATION AND INFRASTRUCTURE

The aim of digital twins is to facilitate the development of industry. Tests can be performed on simulated models before implementing solutions in the real world. This results in robust infrastructures that have already been tested in a multitude of borderline cases.

2) RESPONSIBLE CONSUMPTION AND PRODUCTION.

The use of digital twins makes it possible to study under which conditions to work in order to reduce energy consumption during cooling. Robust simulation rules out the need to oversize equipment, allowing for less costly and more efficient operation.

3) CLIMATE ACTION.

One of Kiconex's current goals is to develop systems that contribute to reducing the use of highly polluting refrigerants. Thanks to the virtualization of refrigeration systems, it will be possible to test Waterloop systems virtually before introducing them in new installations or replacing equipment in current installations. These systems distribute the cold in the installation through a loop of low-temperature glycol water, which is non-polluting.

II. PROJECT DEFINITION

The objective of this project is to create digital twins of various commercial refrigeration equipment. Also, a tool with a graphical interface will be developed to manage these twins in a comfortable and intuitive manner.

Although many different refrigeration units can be simulated, the scope has been focused on the equipment from the Más y Más supermarket in Los Villares, Jaén.

This installation consists of three different types of refrigeration units.

- XR60CX
- XW270K
- XM670K

A. Justification

These digital twins will be used to improve the planning of refrigeration installations, reducing the number of equipment required, their cooling capacity and consumption, thus impacting initial investment costs, operating costs, and maintenance costs.

Thanks to the simulation capabilities, it will be possible to estimate the optimal equipment configuration and equipment parameters to reduce consumption and equipment wear while meeting cooling targets.

In addition, it will be possible to study the impact of anomalies or failures and to make a prevention or action plan according to the simulation results.

Digital twins are gaining popularity in recent years thanks to the ability to experiment with a copy of the real equipment without having to incur additional costs for each test or put the facility, the products, or people at risk.

B. Objectives

The logic of 3 virtual controllers will be programmed and their physical response will be simulated. All this will be integrated into an intuitive graphical interface where the real and simulated values can be viewed and exported to Excel for further analysis. The user interaction with these digital twins will be through a simple and understandable graphical interface.

III. STATE OF THE ART

Kiconex is a remote monitoring and control system for various types of installations and/or equipment: air conditioning and industrial refrigeration, photovoltaic energy, electricity consumption meters, lighting installations, DHW, etc.

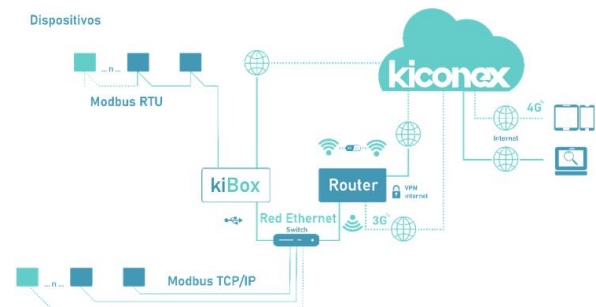


FIGURE 1. Kiconex is based on data storage in the cloud and is composed of a hardware part that captures data and sends it to the cloud via VPN (KiBox) and the control and supervision platform itself (myKiconex). Source: Kiconex.

The data is displayed on the myKiconex platform, from which it is possible to manage and monitor the installation remotely. The data can be represented through graphs and can even be downloaded for further study.

However, Kiconex does not have simulation capabilities. Despite its data storage and remote-control capabilities, it lacks the predictive capabilities needed to anticipate possible undesirable situations.

The refrigeration equipment used in the installations monitored by Kiconex is varied and there is not one digital twin valid for all of them. To ensure that the integration with the platform is

correct, a virtual model of each refrigeration equipment must be made individually.

IV. THE INSTALLATION

The installation to be simulated is a supermarket of the Más y Más chain in Los Villares, Jaén. The scheme of the installation is as follows:



FIGURE 1. The installation has 6 refrigeration units, 2 freezers, 4 refrigerated display cabinets and four storage chambers (3 refrigerated and 1 freezer). Source: Kiconex.

The installation has two INTARCON Sigilus air coolers, which provide the water used by the entire Waterloop system. At the time this paper was published, the installation of a third air cooler was underway to meet the cooling requirements.

V. THE TOOL

The tool will consist of a graphical interface designed with the Python library PySide6.

It will have the following capabilities:

- Connection to the Kiconex API
 - Credential validation.
 - Obtaining equipment data.
 - Obtaining historical data.
- Data analysis
- Physical model fitting
- Simulation
- Validation
 - Of the virtual controller (comparing real and simulated output vs. real input)
 - Of the physical model (comparing real and simulated behavior over the same period of time)
- Manual parameter tuning
 - To test different controller configurations
 - To simulate situations that cannot be tested on real equipment.
- Export real and simulated data to csv format for further analysis.

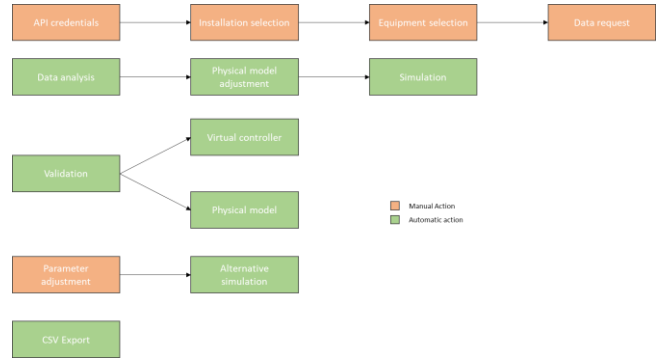


FIGURE 2. The tool has features that are triggered automatically and others that need the user's interaction.

A. User Interface (UI)

The user interface is the element that allows communication between the user, the API, and the digital twins. All visualization and simulation aspects of the application can be controlled from this interface.

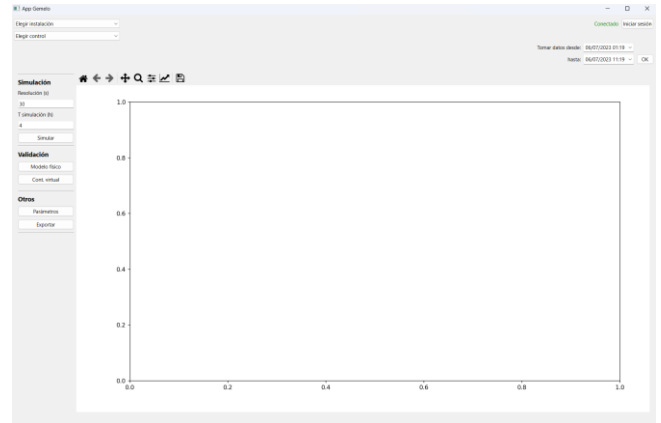


FIGURE 3. The user interface has the data collection actions in the upper part, the simulation actions in the left part and the data visualization occupies the rest of the screen.

1) API CONNECTION

There are several ways to establish the connection to the API (Application Programming Interface). This API allows the user to interact with the Kiconex back-end and with the digital twins running in the local environment.

Automatic connection

To use the application, you must have access to the Kiconex API to obtain the data from the controllers and their historical records. To do this, a connection token is required. At startup, the application looks for a file called token.txt in the folder from which it is executed. If the file exists, a test query is made to the API with this token and, if valid, the connection is established, and the token is saved. At that moment, the user will appear as connected in the upper right corner.

Login

After clicking the Login button, a username and password can be entered. By clicking on Login, these credentials are sent to the API to obtain a token.

The credentials are validated and, in case of being a valid user and password, this token will be used for the connection and saved in token.txt. Otherwise, it will give credentials error and will prompt the user to try again.

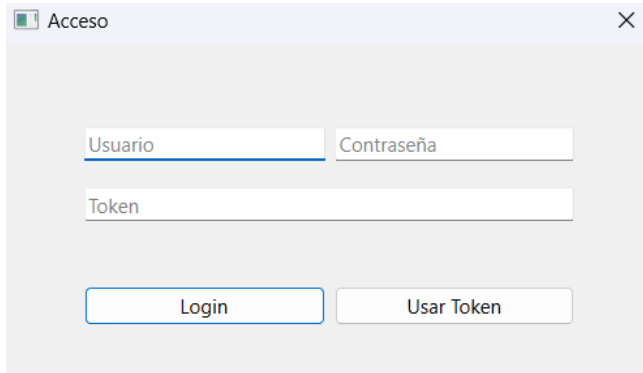


FIGURE 4. The login dialog has an option for user-password login as well as a direct token connection.

Direct token

If the user has a token saved somewhere other than token.txt, they can use this token from the same window. Instead of entering username and password, they must enter the token and click on Use Token. This token will be validated and saved or discarded following the same method as in the previous case.

2) CONTROL SELECTION

Once the connection to the API has been established, an installation and a control must be selected to obtain its data. This is done in the drop-down menu at the top left.

The first thing to do is to select an installation. In this case, only the Más y Más los Villares facility has been implemented, but the application has the ability to expand the list of facilities.

Once the installation to be simulated has been selected, the application asks the API for data about this installation. It will receive information on all the controllers of the installation and will show them in the drop-down list of *Choose Controller*.

Once the controllers have been obtained, the controller type will appear next to its descriptive name in the drop-down list.

This indicates that the application has successfully received the data from the API and the virtual controllers of each of the units are being initialized.

To finish the update, the application asks the API for the parameters of each controller. This operation is performed in parallel with threading, so that several requests can be made at the same time to improve the response time of the query. With threading, several threads are created in the processor that can be executed in parallel. In this way, you can request data and wait for the server's response without blocking the execution of other requests or the main code.

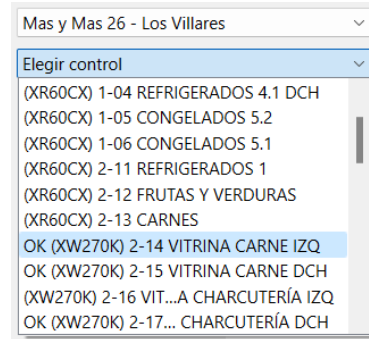


FIGURE 5. Once the controller is fully initialized, an OK will be displayed at the beginning of its name.

3) DATA COLLECTION

By default, 10h of data is taken from the current time, but this is configurable thanks to the date and time dropdowns. Once OK is pressed, the API request is made.

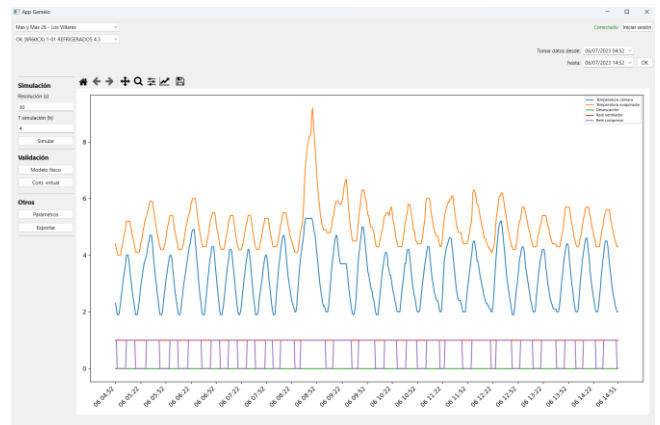


FIGURE 6. The data coming from the API is represented in the interface with a legend.

4) SIMULATION

When requesting the data from the API, the physical model fitting is executed in parallel. Thanks to this parallel action, you can run the simulation as soon as you have the actual data represented on the screen.

The simulation section has several options. The first is the resolution of the simulation in seconds. The real data has been taken with a period of 60s, but the simulation can be run with shorter steps to ensure fidelity or longer steps to speed up the simulation and reduce the computational load on low performance devices.

The second parameter that can be adjusted in the simulation is the simulated time. It indicates how many hours in the future are to be simulated. There is no limit to this value, but visualization will be difficult if more than 12h are simulated. However, thanks to the ability of the application to export these graphs to Excel, further analysis of extensive simulations can be performed.

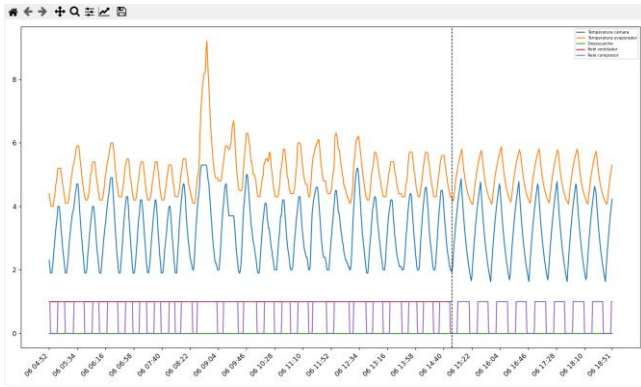


FIGURE 7. The simulated data is shown following the real data separated by a dashed vertical line. This shows that there is continuity between the real and simulated equipment.

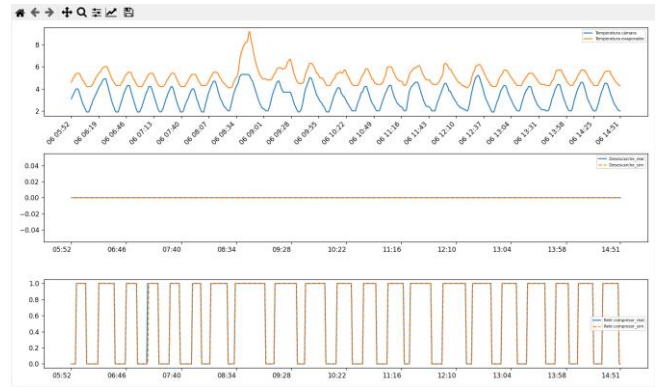


FIGURE 8. The real outputs are represented as a solid blue line, while the simulated ones are a dashed orange line. The match between the two is almost perfect.

A simulation with 180s steps is noticeably faster, but results in a step response and is less true to reality. Although data is recorded every minute in the Kiconex database, the real controller measures at a much higher frequency and its response takes no more than a few seconds. A long duration step in the simulation influences the delay that the virtual controller has and will therefore cause steeper peaks and a worse response.

In addition, the native Matplotlib toolbar can be used to zoom in on the curves, move the graph as a whole or even activate or deactivate curves according to the user's needs.

The simulation will start from the moment the data requested from the API ends. This can end at the current time or at any point in the past for which data is available.

5) VALIDATION

The simulation result can be a good indicator of whether the physical model and the virtual controller are working correctly. However, comparing past data with simulated future data is not a guarantee of the correct functioning of the digital twin. This is why two additional functions have been implemented to check the veracity of the results.

Virtual Controller

To check that the virtual controller has been configured correctly, the validation is performed by clicking on the Virtual Controller button. At the start of the validation, the data previously obtained from the API will be divided in two. The first part will have a duration equal to one complete defrost cycle (or 1h if no defrost is activated). This data is used to obtain the status of the real controller at that moment. The rest of the data is used for comparison with the simulation being performed.

The real input data is iterated, obtaining the output of the virtual controller and comparing it with the output of the real controller at that moment. This results in a graph where the user can see the actual inputs and one more graph for each controller output. In most cases, these will be a defrost and a compressor.

Physical Model

Once the virtual controller has been validated, the whole physical model can be validated by a full simulation. The state of the virtual controller is updated in the same way as before.

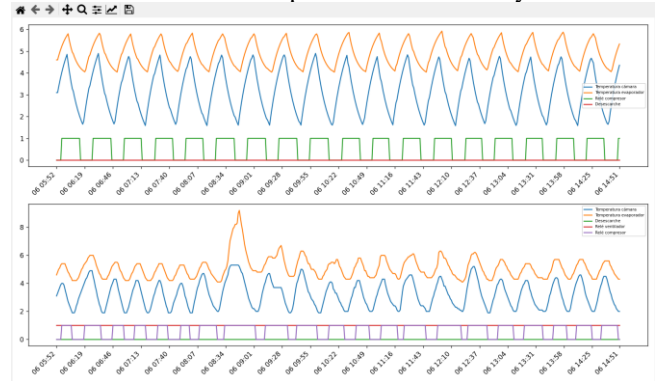


FIGURE 9. The simulated data is compared with the real data. Apart from one anomaly caused by opening the door of the refrigerator, the two graphs have similar shape and values.

6) PARAMETER TUNING

Selecting Parameters will open a tab with all the parameters of the virtual controller.

These parameters can be modified to change the simulation conditions. The Reset button returns the parameters to their original values obtained from the API.

7) EXPORTING DATA

The user can export the current plot to Excel. After clicking the Export button, a dialog will open asking for the name of the file and the path where it will be saved.

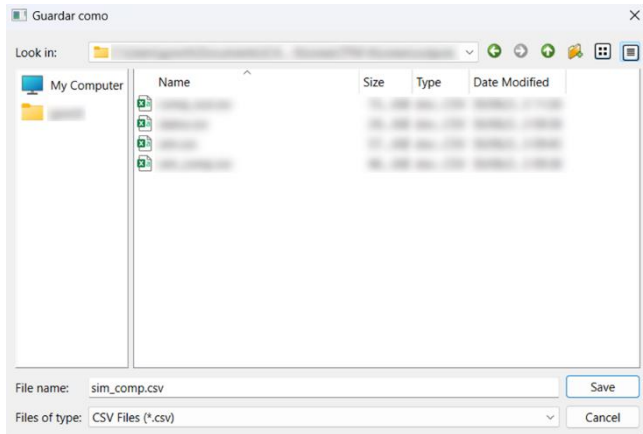


FIGURE 10. The file will be saved as a .csv file and can be analyzed later in other programs such as Excel.

Depending on the chosen graph, the following data can be saved:

- The actual data obtained from the API.
- The actual data together with the simulated section.
- The validation of the virtual controller. Actual vs. simulated data.
- The validation of the physical model. Real vs. simulated data.

B. Virtual controllers

All the controllers that have been virtualized have common attributes and functions. For this reason, a base class called Controller has been programmed, which will have the basic methods and fields of all virtual controllers.

In addition, there are some methods that exist in all controllers, but whose operation is different depending on the type of controller. These methods are overwritten in each subclass.

1) XR60CX

The XR60CX controller is used for supermarket refrigerated cabinets. There are 9 in total. They are suitable for both refrigerated and frozen, never dropping below -20°C .

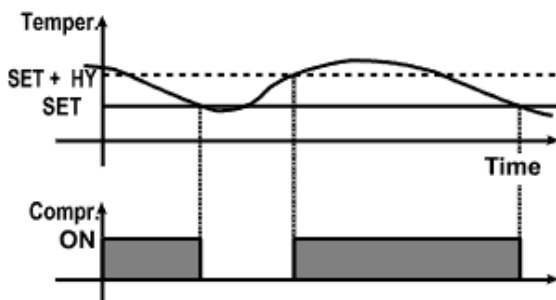


FIGURE 11. When the measured temperature exceeds the set point (SET) plus a differential called hysteresis (HY), the compressor is activated. The compressor remains on until the measured temperature returns below SET or until a defrost starts. Source: Dixell XR60CX manual.

Inputs:

- Probe temperature 1 (Chamber probe)
- Probe temperature 2 (Evaporator probe)

- Probe temperature 3 (Optional. Condenser probe)
- Probe temperature 4 (Optional. Not used in any equipment)

Outputs:

- Compressor relay
- Defrost relay (By electric resistance or hot gas)
- Fan relay

Parameters:

The actual controller has dozens of parameters. For the digital twin, only the most relevant parameters have been selected:

- SET. (Set point). Temperature set point.
- HY. (Differential). Intervention differential for the set point.
- idF (Time between defrosts). The controller waits "idF" hours from the start of one defrost until the start of the next one.
- MdF (Maximum duration first defrost). Maximum time in minutes from the start of a defrost to the end of a defrost.
- dFP (Probe selection for defrost). Probe selection for parameter "dtE". Controls end of defrost by maximum temperature.
- dtE (Defrost end temperature). If the measurement of the probe chosen in "dFP" exceeds this value, defrost is aborted.
- odS (Delay of the outputs at start-up). At least "odS" minutes must elapse before the outputs can be activated from the moment the unit is switched on.
- dPo (Perform defrost at start-up). Boolean. Starts controller operation with a defrost or waits "idF" hours.
- AC (Anti-short-cycle delay). Interval in minutes between compressor stop and next compressor start.
- rtr (Percentage of P1-P2 for regulation). Allows to define the regulation according to the percentage of the temperatures measured by the first and the second probe according to the formula $(rtr(P1-P2)/100 + P2)$.
- P3P. (Presence of third probe). Boolean.

Memory

The XR60CX class has a memory object that stores the state of the controller. It stores the last values of the inputs, the last values of the outputs, the time when each output was last modified (in order to know when to stop a defrost, when to start the next defrost or when a cooling cycle is considered short), the current time and the time when the equipment was started.

2) XW270K

The XW270K controller is used for the refrigerated display cabinets. There are 4 in total. Its normal temperature does not drop below 0°C .

The operation of all controllers to be virtualized is similar. The peculiarity of the XW270K is that they have a heating function in order not to go below a minimum temperature. An SAA parameter is used, which uses a nominal temperature setpoint and by default uses the same value as SET. If the temperature drops below SAA - HY at any time, the heater is activated until the temperature is again higher than SAA. There are no other significant differences in operation compared to the XR60CX.

Inputs:

- Probe temperature 1 (Chamber probe)
- Probe temperature 2 (Evaporator probe)
- Probe temperature 3 (Optional. Condenser probe)

Outputs:

- Compressor relay 1/2
- Defrost relay 1/2
- Evaporator fan relay
- Condenser fan relay

This type of controller has two relays for defrost and compressor. This allows to have one or two units in parallel operation depending on the refrigeration demand. In this installation only one relay of each type is being used.

Parameters

The actual controller has dozens of parameters. For the digital twin, the relevant parameters have been selected:

- SET
- Hy
- idF
- MdF
- dtE
- odS
- dPo
- AC

The abbreviations and operation of all these parameters are identical to those of the XR60CX. The SAA parameter has not been included since it is not used for heating in any of the equipment in the installation.

Memory

Same structure as in XR60CX.

3) XM670K

The XM670K controller can be configured to operate in two ways. The first, which is the one to be simulated, is the way the previous two controllers operate. However, it can also control the operation of the air coolers that exchange heat from the Waterloop system to the outside.

In the case under study, the operation is similar to the previous controllers. It tries to maintain the temperature between SET and SET+HY.

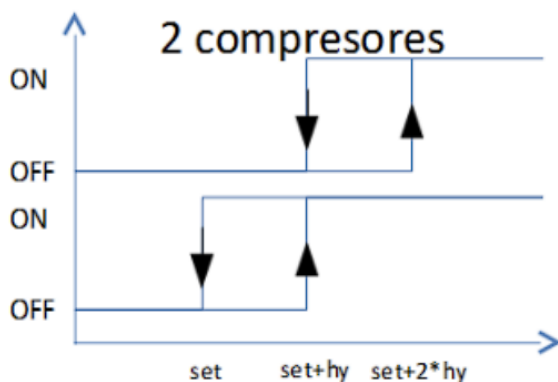


FIGURE 12. It has the possibility of activating several compressors in cascade for high demand situations. Source: Dixell XM670K manual.

It does not have heating.

Inputs:

- Probe temperature 1 (Chamber probe)
- Probe temperature 2 (Evaporator probe)
- Probe temperature 3 (Condenser probe)
- Waterloop temperature
- Water inlet
- Water outlet

Outputs:

- Compressor relay 1/2
- Defrost relay 1/2
- Evaporator fan relay
- Condenser fan relay

Parameters

The simulated operation is the same as the previous ones.

- SET
- Hy
- idF
- MdF
- dtE
- odS
- dPo
- AC

Memory

Same structure as in XR60CX.

C. Physical model

1) INTRODUCTION

The physical model of all the curves of all the equipment studied has a similar behavior. As they are simple thermal models, the response can be represented approximately as a first order transfer function. In this way, each curve would be determined by a final temperature and a time constant. In addition, a delay is added, since the response between the activation/deactivation of the compressor and the start of cooling/heating of the chambers is not immediate due to thermal inertia of the system. In conclusion, each variable to be simulated can be characterized with 3 parameters for cooling and 3 parameters for heating).

2) CURVE ANALYSIS

Each controller's cooling or heating curve can be characterized by three parameters: final temperature, time constant and delay. Each controller has a method that receives a Pandas dataframe with input and output data, which is analyzed to obtain its parameters. A dataframe is a data structure and is the basis of all analysis with the Pandas Python library. It is comparable to a csv file.

Edges

The first thing to do is to obtain the rising and falling edges of the compressor in order to divide the curves into cooling and heating cycles. These are the points where the compressor changes from 0 to 1 or vice versa and can be obtained easily with numpy's diff function, which calculates the difference between one dataframe row and the next. The dataframe

positions where this value is 1 or -1 are the rising and falling edges.

Limits

Using the data from a falling or rising edge to the next edge of the same type, a complete cooling and heating cycle is obtained.

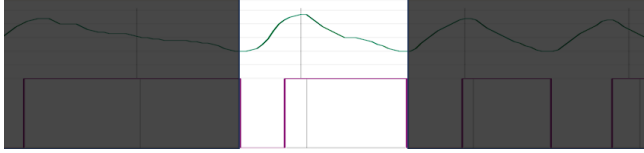


FIGURE 13. The data is taken as a whole cycle at a time. Source: Kiconex online platform.

In this example, a falling edge is used, so the first type of curve is heating. The heating curve is complete since it starts after the delayed falling edge and is known to end before the cooling curve. The cooling curve cannot be analyzed with these data, as it may continue for a few minutes after the next falling edge due to the delay caused by the inertia of the system.

In this way, we will study the heating curves from falling edge to falling edge and the cooling curves from rising edge to rising edge.

From the extracted data, the following information can be extracted:

- End of heating curve (maximum temperature). Value and index are obtained. This is not the value to which the curve tends, but the last value recorded during heating.
- Start of heating curve (minimum temperature). Value and index are obtained. Only the data prior to the end of heating are taken into account to avoid confusion with the data of the following cooling curve.

To study a cooling curve, the same is done by inverting the role of maximum and minimum temperatures.

Once the start and end of the curve are obtained, each curve is stored from the start index to the end index, thus obtaining only the necessary information for each curve.

Curve shape

For each curve, the `curve_fit` function of the SciPy library is used to obtain the first order transfer function that best fits the actual curve data. The first order function has been defined such that:

$$T_t = T_{t-1} + (T_f - T_{t-1}) * (1 - e^{-\frac{1}{\tau}})$$

The function uses the actual initial value of the curve and the estimated parameters to obtain a simulated curve. The process is repeated until the estimated parameters are satisfactory, and the real and simulated curves are similar. The temperature at which the curve tends, and its time constant are obtained, which gives two of the three parameters necessary to have the curve fully characterized.

In addition, the R^2 value of `curve_fit` is obtained to know how good the curve fit has been. This value is between 1 (perfect fit) and 0 (no relationship between real and simulated).

This process is repeated for all cooling and heating curves in the dataframe and the data for all these curves are stored. Further analysis of this dataset will provide the global parameters to be used in the simulation.

Each curve has the following data:

- Compressor start index.
- Type (Cooling/Heating)
- Index and initial value of the curve
- Index and final value of the curve
- End temperature parameter
- Time constant parameter
- R^2

Final analysis

For all the curves analyzed, the delay is calculated as the difference between the initial index of the curve and the index at which the compressor started. Similarly, the duration of the curve is calculated as the difference between the final and initial rates.

For the final temperature, time constant, delay and duration, the population mean and standard deviation are calculated and those curves that are more than 2 standard deviations away from the mean are eliminated. The elimination of anomalies gives a result that is more similar to the normal operation of the equipment.

For R^2 values, only those curves with a value greater than 0.8 are taken. A value lower than this may mean that the curve fitting has not been correct or that the curve has been anomalous due to disturbances during the monitoring period.

Once the outliers have been eliminated, the mean values of final temperature, time constant and delay are taken. Outliers are those measurements that deviate excessively from the sample mean. They are taken as invalid and are ignored in the rest of the statistical analysis. With this, the curves are characterized and can be simulated.

3) SIMULATION

Simulation is the main task of the digital twin. Thanks to the physical model that has been adjusted and the virtual controller, the behavior of the equipment in the future can be predicted.

Memory update

The first step is to set the state of the virtual controller at the initial time of the simulation.

The data obtained from the API is used to copy the state of the real controller at a given time. To do this, it is necessary to know the state of the inputs and outputs at that time, the last time the defrost was activated (and the current duration of the defrost if it is activated) and the time of startup of the equipment. The current status of inputs and outputs is found in a single row of the dataframe data obtained from the API. To obtain the rest of the necessary information, the program iterates from the end of the dataframe to the beginning to determine the remaining parameters. To speed up this operation, only "idF" hours are analyzed, which is what is needed to see an entire defrost cycle. If "idF" = 0, 1 hour is analyzed. In this case, it is assumed that

the equipment has been started up long enough to be operating normally, so the default start-up date is taken as the epoch, i.e., January 1, 1970.

Step by step simulation

Once the controller configuration is correct, the simulation itself can proceed.

The `simulate()` method of the controller in question is used. This method receives the time interval to be simulated and the duration of the time step in the simulation (the time resolution). For each step, the measurements of the input variables and the current time are used. This information is passed to the virtual controller to provide the corresponding outputs. With the inputs and outputs, the physical model is used to simulate the response for as long as defined in the method.

Simulation steps are performed until the simulation length defined in the method is reached, after which the results are returned so that they can be represented in the user interface.

VI. CONCLUSIONS

The interface that was developed has all the required functionalities. The result is a complete application with predictive capabilities in which both real and simulated data can be observed and stored for later analysis.

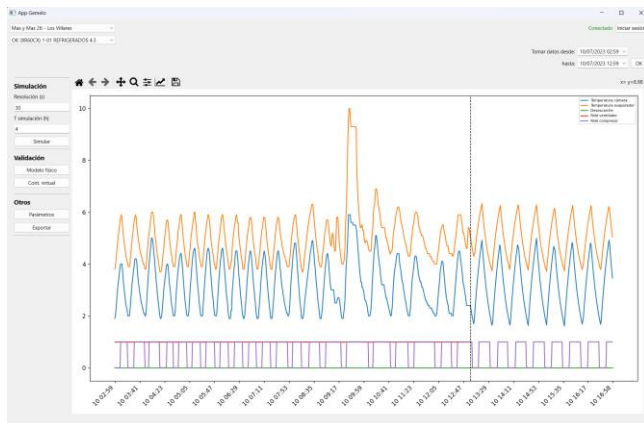


FIGURE 14. The application shows real data followed by simulated data.

The logic of the three desired virtual controllers has been successfully programmed and their physical response has been simulated and validated. This physical model can still be fine-tuned, but it is robust enough to serve its purpose.

The interface has been designed in a simple and intuitive manner. It is divided into clear sections and has not been overloaded with information. It integrates well with the Kiconex ecosystem and allows for further analysis of real and simulated data thanks to its export to csv file feature. With this addition, Kiconex's platform will be more complete, adding predictive capabilities to their IoT tool.

The application will allow for convenient testing of equipment in various situations without the need to use real equipment. These tests will generate information that will be used to optimize the design of installations and the operation of the equipment already installed. In addition, it will serve to detect critical conditions and eliminate possible risks in the physical equipment.

The reduction of necessary equipment and the optimization of its use will make it possible to meet the SDGs mentioned above.

VII. FUTURE WORK

Although the basic functionality of the tool is complete, its design can be improved, and its capabilities expanded.

In the future, other installations with other types of controllers will be simulated. To do this, the digital twins of these controllers will have to be programmed. This task will not be too costly since a base class has been established for all controllers. The elements that differ from this base class for each controller also follow a similar pattern. The three controllers that have been implemented have code with a common structure. For most of the new controllers to be virtualized, it will be possible to follow the scheme of one of the already implemented controllers without significant changes.

At this time, only one installation has been considered. In the installation selection dropdown, the user can see three installations, but two of them are placeholders and are there only to demonstrate how the application would look like with several installations available. The only way to introduce new installations is to modify the code and add their name and UUID (Universal Unique Identifier). While this is a simple task, it would be convenient to create a way to add new installations from within the application itself. In addition, the information of these installations should be saved so that the next time the application is started, the added installations can be seen.

One of the functionalities that could not yet be introduced in the project due to time restrictions is the simulation of disturbances. The machines will not always be operating in a stable regime, so the simulation capacity must be extended to consider the following phenomena, among others:

- Sudden heating due to introduction of food, maintenance, or other causes.
- Temporary loss of power due to power failure.
- Failure of one or more probes.
- Compressor failure or defrosting.

The simulation is deterministic, while the real data is more stochastic. The behavior of the simulation could be closer to the real behavior if the following random elements were introduced:

- Noise in the measurements
- Noise in the physical response
- Randomness in the delay of the controller response.
- Introduction of small disturbances at random times.

Behavior closer to reality can help detect situations that would not occur in a deterministic environment. In addition, if the possibility of simulating equipment start-up from room temperature were added, a greater number of situations could be studied.

So far, the fit of the physical model of the equipment has been mostly satisfactory. This may not be the case in the future, so it is advisable to plan the future development of the application with this in mind.

At this point, the physical model is automatically adjusted, and the user does not have the ability to see its parameters nor to modify them. A section could be added to the application where these parameters can be viewed and, in case the automatic fitting was not quite correct, modified until the simulated response is comparable to the real response.

The introduction of all these improvements could provide a better user experience and make the application more robust over time.

In case any new features or a change in the design is needed, it can be easily modified through Python programming using the PySide6 library.

VIII. REFERENCES

- [1] Dixell. “CONTROLADORES PARA MOSTRADORES CANALIZADOS XM660K-XM670K -MANUAL PARA VERSIÓN SW 5.4d”. March 2022. <https://webapps.emerson.com/Dixell/Content/Pages/Manuales/XM400600/XM660K-XM670KR5.4/XM660K-XM670K-ES-r5.4.pdf>
- [2] INTARCON. “Regulación Electrónica XW270K”. October 2018. <https://www.intarcon.com/pdfs/ES/XW270K.pdf>
- [3] Dixell. “Controladores digitales con gestion deshielo y ventiladores XR60CX”. April 2015. <https://webapps.emerson.com/Dixell/Content/Pages/Manuales/XR-CX/XR60CX/XR60CX-SP.pdf>
- [4] Naciones Unidas. “Objetivos de desarrollo sostenible”. September 2015. <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [5] Ogunleke, Samuel. “Why Is Python Popular for Data Science?”. January 2022. <https://www.makeuseof.com/why-is-python-popular-for-data-science/>
- [6] Hampel, Clare. “The Growing Popularity of Digital Twin Technology”. January 2023. <https://globaledge.msu.edu/blog/post/57227/the-growing-popularity-of-digital-twin-t>
- [7] Kiconex. “Monitorización/Supervisión y Control de dispositivos Advanced Connectivity Solutions for Smart Installations (IoT) Industry 4.0”, Catálogo 2022. <https://www.intarcon.com/pdfs/ES/catalogo-kiconex-2022.pdf>
- [8] Intarcon. “EQUIPOS DE REFRIGERACIÓN GAMA COMERCIAL”, Tarifa 2023. https://www.intarcon.com/pdfs/ES/2023-tarifa-comercial-INTARCON-LQ.pdf?utm_source=analytics&utm_medium=contenidoabierto&utm_campaign=cp-web&utm_id=catalogos&utm_term=tarifa-comercial-2023