

A New Tool for Food Industrial Plant Simulation and IoT Control

Filippo Catalano^{a,*}, Alessandro Leone^b, Biagio Bianchi^b, Antonia Tamborrino^b

^a Department of Biosciences and Territory, University of Molise, C.da Fonte Lappone – Pesche (IS), ITALY

^b Department of Agricultural and Environmental Science, University of Bari Aldo Moro, Via Amendola, 156/a – 70126 Bari, ITALY

f.catalano1@studenti.unimol.it; fil.cat@hotmail.com

The general objective of this research is to create a Simulation Software allowing the Optimization and Control of an IoT system for food industry applications.

An industrial plant is today considered a complex system as it is often composed of many types of machines characterized by a strong temporal variation of the productions.

The three missing factors to accomplish what above are:

- Simulation;
- Implementation of communicating objects also called IoT (Internet of Things) systems.
- Intelligent control;

In this paper, we show the tools developed and implemented in the Simulation software used to model a physical industrial process by mean of a user-friendly graphic interface allowing interactively defining different plant configurations, through the selection of basic graphic elements and their appropriate connection.

The software has been developed in Unity, a cross-platform engine used to create three-dimensional, two-dimensional, virtual reality, and augmented reality, as well as simulations.

The software consists of the following main sections:

- library;
- creation and management of the process scheme;
- simulation.

The "library" section contains the elements (blocks) used in the process scheme, to be completed with functions describing their specific behaviour. These elements represent physical components and logical connectors that allow connecting the different blocks generating an oriented graph.

The "creation and management of the process scheme" section is the graphical interface through which new configurations can be created. In particular, the connectors correspond to a flow of information from one block to another and contain the output variables determined by the function of the starting block. These values are used as input in the arrival block.

The "simulation" section allows simulating of previously designed configurations.

1. Introduction

Smart Manufacturing was certainly one of the precursors of the IoT world. IoT applications have been around for many years, long before we started talking about the Internet of Things. Today this sector is one of the most mature and combines issues related to automation with issues related to the world of robotics.

Smart Manufacturing also overlaps with the Industry 4.0 world (Lim, C.H., et al, 2020), with a real development policy to extend the introduction of digital in the world of industry towards Internet of Services (IoS) and Internet of People (IoP) (Yao, Xet al., 2017). A video camera is no longer alone in the condition of sending data and images, but is in the condition of doing so intelligently, depending, for example, on the images taken or the

temperature or brightness. In this way it can adapt its behaviour according to parameters of interest that can evolve over time (Long, C., 2017).

The packaging of food products can transfer important information on the quality of the product, on the way it was made and on all those who participated in the production and who are part of the food supply chain (Blackburn, J., and Scudder, G., 2009). Or the packaging of a drug can warn us if we are not taking it as established and can tell us how to remedy a possible forgetfulness.

The main areas of application of the Internet of Things (both for end consumers and for companies and manufacturing) are represented by those contexts in which there are "things" that can "speak" and generate new information.

The sectors most affected by IoT applications are the Smart Home, the Smart Building (Fucci, F., et al, 2016), the Smart City and the Smart Energy (Aluisio B. et al., 2018; Catalano F. et al., 2020), smart hazardous environments (Bragatto, P., et al., 2018) but also, and for a long time, Smart Manufacturing where smart industrial processes are considered.

2. Industrial processes modelling

An industrial process consists of a series of units integrated together in a rational way with the overall objective of converting certain raw materials into a finished product as economically as possible. During its operation, the plant must meet a series of conditions specified at the time of the project and imposed by the general situation (economic, social, environmental) in which the plant operates. During operation, an industrial plant is found to operate in dynamic conditions because of inputs that tend to displace it from stationary conditions. These are:

- Perturbations, unwanted, but inevitable in an industrial environment; for example: change of raw materials, aging or soiling of surfaces, fluctuations in pressure and temperature of the cooling or heating network, temperature variations between night and day;
- changes in operating conditions, desired and established by the operator or by the plant supervision system; for example: change type of product, optimization of operation (minimization of energy consumption, increase or decrease in production).

A schematic of the process variables is shown in Figure 1; the process can be seen as a system which, under the action of certain input variables (causes), generates certain output variables (effects).

Among the input variables it is possible to distinguish manipulable variables (which are available to the operator to make corrections in the process), from disturbances (which depend on the external environment and are beyond the control of the operator). While manipulable variables are also easily measurable, only some disturbances can be measured; the majority is instead to be considered unknown and unmeasurable.

Two categories can be essentially distinguished among the process output variables: performance variables and intermediate variables. The first are the controlled variables in a broad sense, generally not all directly measurable, linked to the objectives of the plant optimization; the second are the physical variables, which can be measured in some way, indicating the operation of the system and therefore the current performance.

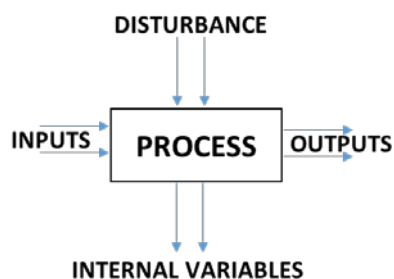


Figure 1: scheme of process variables

To give more details about how a real system can be implemented an example is shown below. In particular, the studied process is a refrigeration plant at service of a Cold Process (CP) where a first fluid product (P1) is refrigerated. As well known a Refrigeration Plant (RP) consists of two main sections: a cold section and a warm section of the thermodynamic cycle. Cold section is aimed to extract heat from the CP both by direct expansion or by mean of an intermediary fluid, e.g. glycol in case of freezing plants or water for refrigeration temperatures above 0 °C. this last option was chosen here. The warm section is aimed to dissipate both the heat coming from the CS and the heat used to rise refrigerant temperature (work-equivalent in vapour compression cycles or waste heat used in an absorption refrigeration cycle). This heat can be dissipated directly into the external

environment through a fan coil unit or recovered into a secondary working fluid mainly water to be used in downstream Hot Process (HP) heating another fluid product (P2). Again, last case was chosen here as water is supposed always below 100°C and the warm water is stored in an insulated warm Water Tank (WT). Moreover, an auxiliary Boiler (B) was added between WT and HP to rise water temperature to desired HP temperature. In Figure 1 an outline of the chosen plant is shown with the main input – output variables, including pipes and pumps.

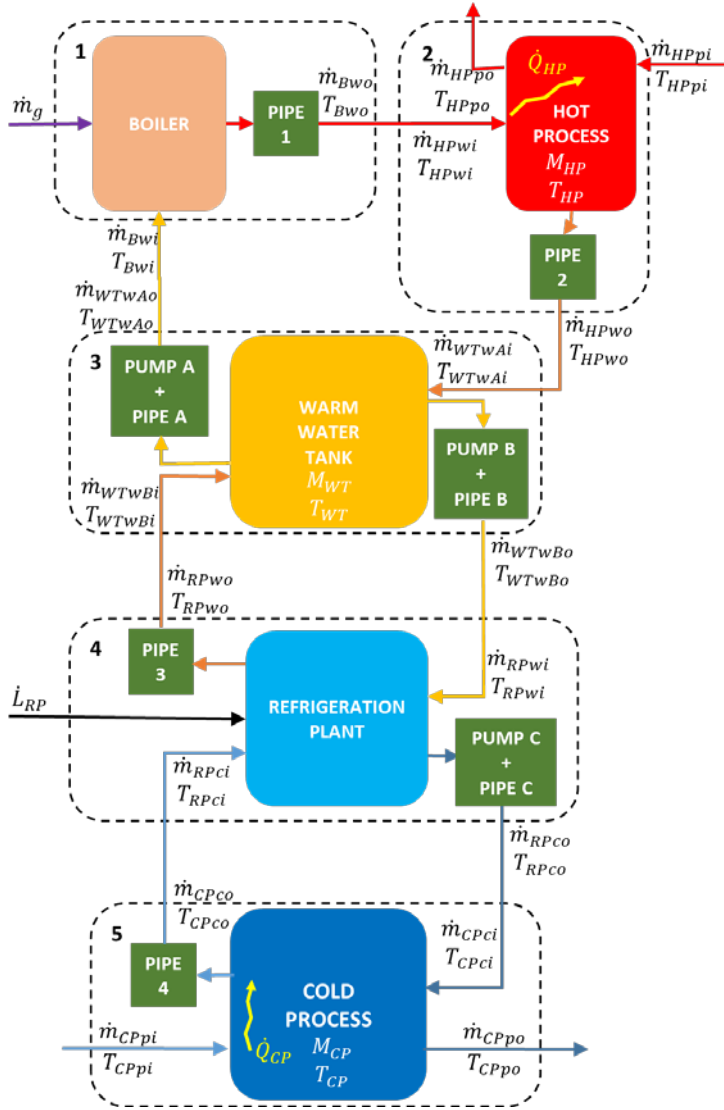


Figure 1. Outline of the simulated plant

The mathematical models of the different subsystems are not shown here for brevity, but they can be easily found in literature as they represent standard processes used in any food industry. These equations are in general nonlinear Differential Algebraic Equations (DAEs). A condition generally hypothesized is the no-delay condition: pumps' response time is small enough compared with controlled parameters variation and the work to operate them is neglected. In this case we can suppose that the flow rate does not vary within each circuit:

$$\dot{m}_{WTwBo} = \dot{m}_{Bwi} = \dot{m}_{Bwo} = \dot{m}_{HPwi} = \dot{m}_{HPwo} = \dot{m}_{WTBwi} = \dot{m}_{PPA}$$

$$\dot{m}_{WTwAo} = \dot{m}_{RPwi} = \dot{m}_{RPwo} = \dot{m}_{WTwAi} = \dot{m}_{PPB}$$

$$\dot{m}_{RPco} = \dot{m}_{CPci} = \dot{m}_{CPco} = \dot{m}_{RPci} = \dot{m}_{PPC}$$

Finally, in a full-scale application some of or all the models can be substituted with IoT components that give in real time the values of the corresponding output variables. This way, coupling two copies of the same system, one corresponding to all IoT components, the other to a fully simulated one, it is possible to evaluate system

future trend against disturbances variation to determine in real time, through an external controller, optimal (or suboptimal) values for the controlled external inputs.

3. Simulation framework

3.1 General aspects

Based on the study of the state of the art previously carried out (Perone et al., 2017; Tamborrino et al., 2019; Catalano et al., 2020), we wanted to direct all the research activities towards the creation of a Simulation Framework that allows the Optimization and Control of a generic industrial plant from an IoT perspective. As said an industrial plant is today to be considered a complex system as it is often not only composed of numerous types of machines but is also characterized by a strong temporal variation of production. This complexity, therefore, is reflected in the design of the plants serving the industry and increases even more if the design must consider the behaviour of the plants during operation due to the high variability of the parameters affecting the dynamics of the system.

In fact, the designer has no to design tools that allow him to simulate a multiplicity of plant configurations such as to be able to choose the optimal one from time to time. All the solutions currently available allow at most to build mechanical models, or CFD models, or even multiphysics models, but a system vision has not yet been implemented, except in applications mainly for educational purposes and, by their nature, very limited. Furthermore, the current plant design methodologies do not consider dynamic and flexible solutions, except in some cases in some applications for buildings. This aspect is essentially due to the lack of framework or software that is sufficiently general purpose for the simulation, optimization and intelligent control of a set of "objects" which, by communicating with each other, can make the system always work in an optimal way.

The two missing factors to accomplish the above are:

- general purpose simulation,
- intelligent control.

Therefore, the research was focused in particular on the creation of a Framework complete with user-friendly interface that allows to simulate, design and control in an optimal way (or as we will see sub-optimal) any system industrial configured as an IoT system whose components can be described through appropriate physical-mathematical models. In particular, in this paper is described the simulation tool, which is the hardest to implement.

In industrial plants, the essential elements constituting the different processes can be identified and schematized in terms of block diagrams. Each block represents in a pictorial way the cause - effect relationship between input and output variables. By introducing the mathematical model of the element into the block, a quantitative evaluation of the evolution of the system over time is possible. In particular, there were developed the following subjects.

- The tools implemented in the simulation software used to model a process and/or an industrial plant by mean of an intuitive graphic interface that allows to interactively define different plant configurations, through the selection of basic graphic elements and their appropriate connection.
- The methods used for simulation and intelligent management of the system.
- The graphic interface software is composed of the following main sections:
- Library: it contains the elements (blocks) to be included in the process diagram, complete with calls to matlab functions that describe the physical-mathematical behaviour, which are used and called by the simulation algorithm.
- Creation and management of the process scheme: it is the fulcrum of the framework. The graphic interface allows to create new configurations or manage configurations that have already been designed and validated: in particular, the configurations will be composed of the elements present in the library or new elements to be created and, if required, stored as new elements in the library. The elements present in the library are mainly representative blocks of physical components (machinery) and logical connectors which, when properly used, allow to connect the different blocks, thus generating an oriented graph. Each side of the graph will correspond to a flow of information / matter / energy / etc. from one block to another and will contain the output variables of the starting block function whose values will be used as inputs for the arrival block. Appropriate information integrity checks have been implemented.
- Simulation: it allows to simulate the operation of the configured configuration. The simulation algorithm recalls the functions of the individual blocks making up the system in an order that can be random, defined by the designer based on his experience or managed by the system through the possible search for the "optimal" calling order (blocks or nodes of the graph representing the plant), which allows to reduce, searching the smallest (or close to it) feedback arc set of the graph, the number of implicit operations in the solution method.

Each section can be partially modified by the user of the software in order to allow customization of the projects to be created and managed as well as the search for an "optimal" configuration based on criteria that the designer will define from time to time.

A "New project" can thus be defined in the software, ie a new configuration of elements that together make up a process, whose elements are the individual blocks of the process to be simulated.

3.2 Software

The development of the node editor made use of an already existing framework: https://github.com/General/Node_Editor_Framework to which changes have been made to make it compatible with the specifications indicated above. In particular, the *CustomFunctionNode* node has been introduced, through the appropriate modifications of the Node class already created in the starting framework regarding the *AutoLayout* parameter set to *True* in order to have a node that adapts to the content vertically. The Node class allows you to inherit two functions as follow.

- *NodeGUI*: it contains all the visual and rendering part of the node with all the instructions that allow you to view the parameters of the single node.
- *Calculate*: which contains the logical part of the calculation of the node function (which in our case will not be used as it is delegated to appropriate Matlab scripts).

Subsequently, a new class called Parameter was created which allows you to have custom attributes for each parameter. Finally, again within the same class, the following two new variables were added.

- *Parameters*: a list of parameters with related attributes
- *FilePath*: a string that contains the path of the Matlab script containing the Input-Output relations (model) relative to the node.

The nodes thus defined can in any case be stored in the Library to be subsequently reused simply by changing the parameter values. The code section contains:

- the parameters (Work Parameters) added in the node (e.g. physical constants or other parameters) other than Input and Output),
- the connection points between one node and another.

At the end you will see an image like the one shown in Figure 2.

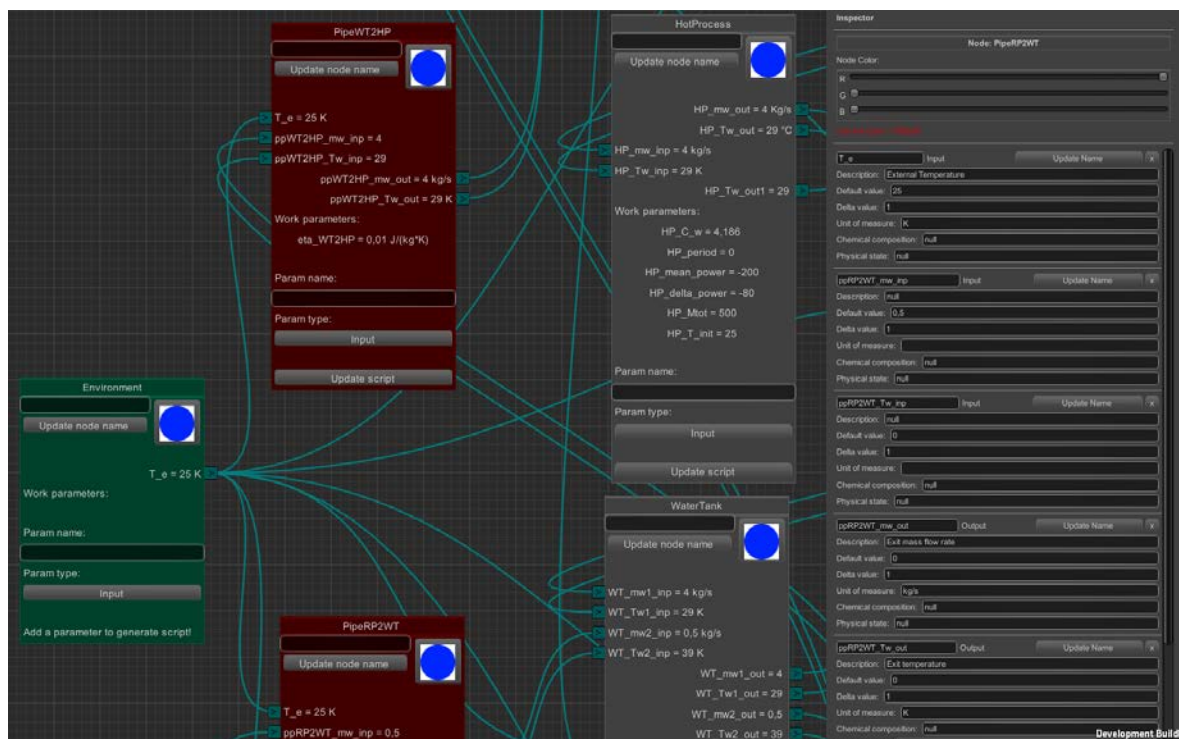


Figure 2: Detail of the plant scheme showing the external environment, the HP and WT processes and two pipes

As for the structure of a new project, this is located inside a specific folder in which there will be two text files (if not present they will be generated automatically) called Node.tmp and Graph.tmp: they represent the templates used respectively to define the nodes and structure of the system graph.

The placeholders of the Node.tmp template are marked with the symbol | before and after the placeholder name:

- |outputs| are the output node parameters;
- |FunctionName| corrisponde al nome del nodo;
- |inputs| matches the input parameters of the node;
- |VarAssignment| it corresponds to the list of assignments regarding the inputs and the WorkParameters.

The Matlab script is automatically prepared with all the information relating to input, output, work parameters taken from the Node.tmp Template, leaving the user the ability to write the code of his own model just below the last comment.

Once the MatLab script has been generated, the parameter file is generated for which two new classes have been created called SerializedCanvas and JsonNode to structure the .json file in the best possible way which will contain all the data exchanged between the Unity code and the Matlab code.

During Runtime, the Matlab code will read the data from the various nodes present within these two new files including the constants Δt (Time step) n_t (Number of time steps).

3.3 Test of the framework

In Table 1 the values of all the constants used in the simulation are shown, while the state variables have the same initial value: $T_{HP} = T_{WT} = T_{CP} = 25^\circ\text{C}$.

Table 1. Constants used in the simulation

Parameters	Values	Parameters	Values
\dot{m}_{PPA}	4 kg/s	c_w	4.187 kJ/kgK
\dot{m}_{PPB}	0.5 kg/s	h_{fg}	50,000 kJ/kg
\dot{m}_{PPC}	20 kg/s	η_B	0.98
M_{HP}	500 kg	COP	3
M_{WT}	1000 kg	T_E	25 °C
M_{CP}	500 kg	\dot{Q}_{HP}	-280 kW

In this case the explicit form of the analytical solution is:

$$x = \begin{cases} T_{HP} = 14.7930 \exp(-0.000796 * t) + 10.207 \\ T_{WT} = 6.4373 \exp(-0.011924 * t) - 10.640 \exp(-0.000086386 * t) + 29.200 \\ T_{CP} = -3.2210 \exp(-0.011924 * t) - 10.739 \exp(-0.000086386 * t) + 38.960 \end{cases}$$

whose final (stationary) value is:

$$x_\infty = -K^{-1}Pu = \begin{cases} T_{HP} = 10.207 \\ T_{WT} = 29.200 \\ T_{CP} = 38.960 \end{cases}$$

Simulation was carried out using a time step $\Delta t = 10 \text{ s}$, while the number of time steps $n_t = 8640$ for a total duration of $t_{fin} = 86400 \text{ s} = 24 \text{ h}$. The solution obtained with the numerical technique implemented in this framework is very close to the analytical one even just a little less close to it if compared with the fully implicit solution method. The absolute error is less than $0.15 \text{ }^\circ\text{C}$ (0.65 % of the analytical solution) for T_{HP} , less than $0.14 \text{ }^\circ\text{C}$ (0.45 % of the analytical solution) for T_{WT} , null for T_{CP} .

4. Conclusions

The software thus created allows the simulation of even very complex systems and with execution times almost proportional to the number of nodes thanks to an algorithm specially developed for the choice of the optimal execution order of the scripts that will be the subject of a forthcoming publication in progress. Preparation. In any case, at present neither the control part (which in any case, albeit with some additional complications, can be implemented as a separate node) nor the optimization part (this can also be carried out where the number of configurations to study is controllable. Therefore, it is expected soon to implement the following functions:

- *Intelligent Management System: a supervisor who supervises all the operations described above.*
- *Configurator: a code that allows you to parameterize a series of configurations so that you can easily switch from one lay-out to another.*
- *Optimizer: a code that allows you to select the configuration that is optimal according to criteria that the designer must identify.*

The framework was tested simulating a food refrigeration plant with heat recovery in a hot process. Finally, we will proceed to the experimental validation by modeling e.g. plant prototypes comparing experimental and numerical results. This will allow to validate both the numerical solver already implemented and the other functions still to be implemented.

Acknowledgments

The authors have contributed to the same extent to the present study.

References

- Aluisio B., Dicorato M., Forte G., Litrico G., Trovato M. 2018. Integration of heat production and thermal comfort models in microgrid operation planning, *Sustainable Energy, Grids and Networks* 16, 37–54
- Ayr U., Tamborrino A., Catalano P., Bianchi B., Leone A. 2015. 3D computational fluid dynamics simulation and experimental validation for prediction of heat transfer in a new malaxer machine. *Journal of Food Engineering*, vol. 154, p. 30-38, ISSN: 0260-8774, DOI: 10.1016/j.jfoodeng.2014.12.022.
- Blackburn J. and Scudder G. 2009. Supply chain strategies for perishable products: The case of fresh produce. *Production and Operations Management*. 18 (2). 129-137.
- Bragatto P., Faramondi L., Failla F., Gnoni M.G. 2018. Potential and Limits of IoT for Hazardous Job in Process Industries, *Chemical Engineering Transactions* Vol. 67, 865-870.
- Catalano F., Perone C., Iannacci V., Leone A., Tamborrino A., Bianchi B. 2020. Energetic analysis and optimal design of a CHP plant in a frozen food processing factory through a dynamical simulation model. *Energy Conversion and Management*, 225 (2020), 113444. doi.org/10.1016/j.enconman.2020.113444.
- Fucci F., Perone C., La Fianza G., Brunetti L., Giametta F., Catalano P. 2016. Study of a prototype of an advanced mechanical ventilation system with heat recovery integrated by heat pump, *Energy and Buildings* 133, 111–121.
- Lim C.H., Loo V.L.K., Ngan S.L., How B.S., Ng P.Q.W., Lam H.L. 2020, Optimisation of Industry Revolution 4.0 Implementation Strategy for Palm Oil Industry in Cyber Security, *Chemical Engineering Transactions* Vol. 81, 253-258.
- Long C., Cao Y., Jiang T., and Zhang Q. 2017. Edge Computing Framework for Cooperative Video Processing in Multimedia IoT Systems, *IEEE Transactions on Multimedia*, Vol. No. 2017.
- Perone C., Catalano F., Tamborrino A., Giametta F., Bianchi B., Ayr U. 2017. Study and Analysis of a Cogeneration System with Microturbines in a Food Farming of Dry Pasta. *Chemical Engineering Transactions*, vol. 58, 2017, 499-504, doi: 10.3303/CET1758084.
- Tamborrino A., Perone C., Catalano F., Squeo G., Caponio F., Bianchi B. 2019. Modelling energy consumption and energy-saving in high-quality olive oil decanter centrifuge: Numerical study and experimental validation. *Energies* 2019, 12, 2592; doi:10.3390/en12132592.
- Yao X., Zhou J., Lin Y., Li Y., Yu H. and Liu Y. 2019. Smart manufacturing based on cyber-physical systems and beyond. *Journal of Intelligent Manufacturing*, 30:2805–2817.