Ref. Ares(2019)6684733 - 29/10/2019

COORDINATED PRODUCTION
FOR BETTER RESOURCE EFFICIENCY

D4.4 – Report on an integrated framework for batch control and scheduling

Universidad de Valladolid
CSIC, CERTH, FRINSA, ASM

October 2019
www.spire2030.eu/copro

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723575

Project Details

PROJECT TITLE	Improved energy and resource efficiency by better coordination of production in the process industries
PROJECT ACRONYM	**COPRO**
GRANT AGREEMENT NO	**723575**
INSTRUMENT	**RESEARCH AND INNOVATION ACTION**
CALL	**H2020-SPIRE-02-2016**
STARTING DATE OF PROJECT	**NOVEMBER, 1ST 2016**
PROJECT DURATION	**42 MONTHS**
PROJECT COORDINATOR (ORGANISATION)	**PROF. SEBASTIAN ENGELL (TUDO)**

THE COPRO PROJECT

The goal of CoPro is to develop and to demonstrate methods and tools for process monitoring and optimal dynamic planning, scheduling and control of plants, industrial sites and clusters under dynamic market conditions. CoPro pays special attention to the role of operators and managers in plant-wide control solutions and to the deployment of advanced solutions in industrial sites with a heterogeneous IT environment. As the effort required for the development and maintenance of accurate plant models is the bottleneck for the development and long-term operation of advanced control and scheduling solutions, CoPro will develop methods for efficient modelling and for model quality monitoring and model adaption.

The CoPro Consortium

Participant No	Participant organisation name	Country	Organisation
1 (Coordinator)	Kechiche Universität Dortmund (TUDO)	DE	HES
2	INEOS Manufacturing Deutschland GmbH (INEOS)	DE	IND
3	Covestro Deutschland AG (COV)	DE	IND
4	Procter & Gamble Services Company NV (P&G)	BE	IND
5	Lenzing Aktiengesellschaft (LENZING)	AU	IND
6	Frinsa del Noroeste S.A. (Frinsa)	ES	IND
7	Universidad de Valladolid (UVA)	ES	HES
8	École Polytechnique Féderale de Lausanne (EPFL)	CH	HES
9	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis (CERTH)	GR	RES
10	IIM-CSIC (CSIC)	ES	RES
11	LeiKon GmbH (LEIKON)	DE	SME
12	Process Systems Enterprise LTD (PSE)	UK	SME
13	Divis Intelligent Solutions GmbH (divis)	DE	SME
14	Argent & Waugh Ltd. (Sabisu)	UK	SME
15	ASM Soft S.L (ASM)	ES	SME
16	ORSOFT GmbH (ORS)	DE	SME
17	Inno TSD (inno)	FR	SME

Document details

Deliverable Type	**Report**
Deliverable No	**4.4**
Deliverable Title	**Report on an integrated framework for batch control and scheduling**
Name of Lead Partner for this Deliverable	**Universidad de Valladolid**
List of Authors - name(s) and organisation(s)	**Carlos G. Palacín (UVa)** **Cesar de Prada (UVa)** **Carlos Vilas (CSIC)** **Daniel Adrián (ASM)** **Agustin Vilas (Frinsa)** **Chrysovalantou Ziogou (CERTH)** **Michael Georgiadis (CERTH)**
Version	**2.4**
Contractual Delivery Date	**30 October 2019**
Actual Delivery Date	**29 October 2019**

Dissemination level		
PU	Public	**X**
CO	Confidential, only for members of the consortium (including the Commission)	

Abstract

This document illustrates the need to integrate scheduling with control layer to provide extra flexibility to the decision-making process in line with Industry 4.0 paradigm.
Several approaches are presented, where control parameters of a batch process are integrated into the optimisation of the section scheduling, in order to minimize the use of resources and production times. These results are applied to an industrial case study, the FRINSA use case.

Revision History

The following table describes the main changes done in the document since it was created.

Revision	Date	Description	Author (Organisation)
V0.0	10/09/19	First version	Carlos G. Palacín (UVa)
V0.1	15/09/19	Added pseudocode of the final algorithm	Carlos G. Palacín (UVa)
V1.0	18/09/19	Outline modified	Carlos G. Palacín (UVa)
V1.1	22/09/19	Introduction upgraded	Cesar de Prada (UVa)
V1.2	23/09/19	Upgraded state of the art	Cesar de Prada (UVa)
V1.3	24/09/19	Upgraded model explanation	Carlos Vilas (CSIC)
V1.4	25/09/19	Reviewe by TUDO	Simon Wenzel (TUDO)
V2.0	01/10/19	Contribution from CERTH	Chrysovalantou Ziogou (CERTH)
V2.1	02/10/19	Updating of the document	Carlos G. Palacín (UVa)
V2.2	03/10/19	Review by TUDO	Simon Wenzel (TUDO)
V2.3	04/10/19	Corrected version	Georgios Georgiadis (CERTH)
V2.4	04/10/19	Updating of the document	Carlos G. Palacín (UVa)

Disclaimer

Table of contents

List of Figures

1 Executive summary

Besides coordinating the operation of different plants, the Copro project aims at integrating different decision layers of the control pyramid, in order to improve decisions taken by production management. Instead of making isolated decisions that only consider the interchange of static information with other layers, the integration of different decision layers provides additional degrees of freedom to each layer which can improve the global performance.

Within the CoPro project, Task 4.3 considers the integration of scheduling and optimal batch operation to develop a hierarchical control and optimisation scheme. The scheme is applied to the FRINSA use case where it integrates the optimal operation of batch operations into the production scheduling.

This deliverable analysis the challenge of such an integration and describes different approaches to combine control and scheduling, some of which have been tested for the use case of FRINSA. Special attentions is paid to the real-time and computational aspects of the implementation.

The task was coordinated by UVa and FRINSA, ASM, CSIC and CERTH contributed as partners.

2 Introduction

The typical control architecture that is implemented in most industrial plants comprises several hierarchical levels, where each level receives work orders from the layers above and operates with the information provided from below.

Starting from the bottom, the instrumentation and control layer allows to get information from the process and it provides a stable operation of the different assets. Better control is obtained if dynamic interactions among variables and operational constraints are taken into account explicitly. This functionality is provided by the advanced control and supervision layer with model predictive control (MPC), which is the key technology at this level.

Above, the MES (Manufacturing Execution Systems) layer provides a set of functionalities among which determining the best operating process conditions can be considered as the most relevant one. In continuous processes, this task is accomplished by RTO (Real Time Optimisation) systems that compute the operating points of the process that, satisfying the current process constraints, provide the optimal value of a cost function defined by the user, typically related to an economic objective. The computed optimal operating point is sent as set-points to the MPCs of the layer below. Most of the times, a RTO is based on static first principles models of the process and it is executed at a low sampling rate, while MPC controllers use dynamic linear models that were obtained by identification from step tests of the process. MPCs run at a higher frequency compared to RTOs. This creates inconsistences, both between the models and the way disturbances are handled. This calls for a better integration of these two layers. Several proposals have been made in the literature giving birth to the so called economic MPC, which integrates MPC and RTO using dynamic process models and economic targets, optimizing the degrees of freedom of the plant w.r.t. process performance rather than following pre-defined targets. See [1], [2] for more details about this approach.

In processes where batch or semi-batch units play an important role, the MES layer uses production scheduling to determine where and when each product must be processed. Potentially, a cost function is optimised respecting process and operational constraints and production targets. Scheduling is closely related to production organisation and has a big impact on the performance of the installation and results obtained. Because of this coupling, formulating and solving scheduling problems requires information from, and at the same time impacts, many other aspects and activities, e.g. production planning, assets maintenance, state of the process and actual capacities… If the decisions related to all these elements are taken in an isolated fashion, then this requires to fix some of the important variables, which leads to an operation with less degrees of freedom compared to the situation in the decisions are integrated.

Many proposals have been made to integrate scheduling with other elements with varying degrees of success. Integration of planning and scheduling seems natural, but the greater level of detail of scheduling and the large time horizons of planning poses serious dimensionality problems. Integration of scheduling and maintenance such that the changing state of health of the assets is considered when generating a schedule, has been proposed in the literature and appears to be a promising line of work that can bring clear benefits. See [3], [4] for a couple of applications in heat exchanger networks and evaporation plants.

In practice, scheduling is performed off-line at low frequency and the results are passed to the control layer with little feedback from the process, which is partly due to the lack of digitalisation in many plants. Only when significant changes or disturbances are present, manual adaptation or re-scheduling is applied, which may generate suboptimal production plans. In a similar way, since the scheduling uses models with limited granularity that not always represent all important information required for control, the generated control targets may not be feasible.

The integration of batch control and scheduling appears as natural as the integration of RTO and MPC, as it has similarities; however, it has its specific challenges. An integration makes more sense if scheduling is used in real-time using current information from the plant. Proposing future optimal actions are the adapted to cope with model uncertainties or changes that may take place in the process or production targets. Sequential control or MPC of the batch units operates within the same general framework, controlling the operation of those products assigned for processing to each batch unit and using the shared resources available according to the scheduling. Notice that depending on this availability, as well as to the control targets and the way of operation, the batch control / MPC may change critical parameters for the solution of the scheduling, such as duration of the batch cycle or consumption of energy or other resource. An integration of batch control and scheduling provides more flexibility and feasibility to the control recipes, improves common information and increases the space of feasible decisions of the scheduling. In contrast to this, standard structures impose rigid boundaries to each problem.

The integrated problem manages the main discrete and continuous degrees of freedom of the system. Nevertheless, the implementation of the solution in the process requires a layer of sequential control that uses DCS or PLC hardware. Also, the integration of advanced process control and scheduling requires a certain degree of digitalisation such that the information needed is available in a reliable and automated way. This is in line with the current trends favoured by Industry 4.0 programs which in a first step push for improving data collection and treatment as a base for exploiting the available information and for an implementing of better automated decision-making systems.

3 Integration of scheduling and optimal control of batch processes

Typical scheduling problems are formulated as mix-integer optimisation problems that use models of simple structure, but which involve a large number of equations and variables, many of which are binary. Sampled values of the process variables are used to describe the process dynamics that covers large horizons. In contrary, MPC of batch process typically uses smaller sets of DAEs to model the process, which involves fewer binary variables, but a significant number of process units may be in operation.

3.1 Monolithic approaches

There are several ways in which the integration can be formulated. Ideally, monolithic or full integration of the two problems in a single one offers clear advantages as it manages all available information and makes complete use of the degrees of freedom of the problem. Two main points of this approach are worth discussing: the first one is the modelling approach; the second one is related to the computation of the solution.

Regarding modelling, one can see two alternatives for the model: Incorporating the (discretised) continuous equations of the control problem into the scheduling framework or, vice versa, including discrete elements from the scheduling into the control problem.

The first alternative normally uses Resource Task Networks (RTN) to formulate the scheduling problem [5]. They model the availability of materials and process units as resources that are consumed or generated by the scheduling tasks. Time is divided into slots where tasks should take place, and the scheduling assigns tasks over time while ensuring feasibility of the resource levels. A typical resource balance equation is given as:

$$
\begin{aligned}
R_{r,t} = R_{r,t-1} &+ \sum_{i\in\mathcal{I}} \sum_{\theta=0}^{\tau_{i,m}} \mu_{i,m,r,\theta} \cdot w_{i,m,t,\theta} \\
&+ \sum_{i\in\mathcal{I}} \sum_{\theta=0}^{\tau_{i,m}} \nu_{i,m,n,r,\theta} \cdot \xi_{i,m,n,t,\theta} + \Pi_{r,t}
\end{aligned} \tag{1}
$$

Where R is the resource level, w and ξ are the past values of the states of discrete and continuous tasks, τ is the task length, Π are external transfer events and the other symbols represent parameters. The subscripts are: r, for each resource; t, for time slot; i, for tasks; and θ, for delays.

To incorporate the dynamic control problem into this framework, the time slots of the scheduling are divided into a small number of finite elements and within each finite element the model equations of the controller are discretised using orthogonal collocation. Figure 1 represents this problem that results in a large set of algebraic equations, which are incorporated to the scheduling problem. The integrated scheduling and control problem tries to find the values of the continuous and discrete variables that minimises an economic cost function while satisfying the models and constraints of the scheduling and of the control. This is normally a mixed-integer non-linear programming problem (MINLP) that must be solved at every sampling time.

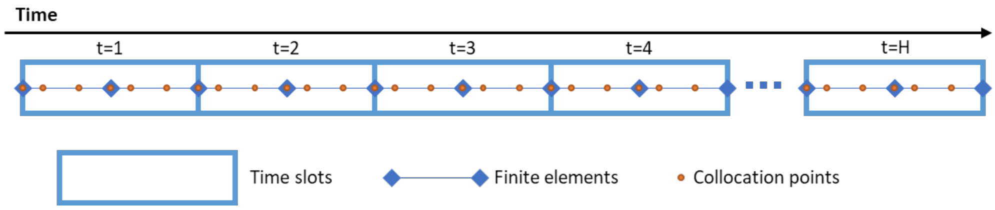


Figure 1: Time representation of the integrated scheduling and control formulation

The second point that was mentioned above concerns the question how to solve the resulting MINLP problem efficiently in a short time. Unfortunately, even using decomposition methods, the integrated scheduling/control problems can be solved only when dealing with small size problems. Because of that, examples in the literature are limited to problems with a small number of discrete variables such as single unit sequencing, e.g., grade changes in polymerisation reactors, or similar, [6].

The second modelling alternative, includes binary variables into the framework of control (MPC) as a way of representing the discrete decisions of scheduling problems. The associated dynamic optimisation problem corresponds to what is known as hybrid MPC, which in fact covers a wider set of problems. There are many theoretical results and applications of hybrid control [7], but not so many for the specific case of scheduling. The optimisation problem to solve is a dynamic mixed-integer programming (MIP) problem:

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{y}} J(\boldsymbol{x},\boldsymbol{u},\boldsymbol{y}) \\ &\boldsymbol{h}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u},\boldsymbol{y}) = \boldsymbol{0} \\ &\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{y}) \leq \boldsymbol{0} \\ &\boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{Z}^m \end{aligned} \tag{2}$$

This problem can be converted into a static MIP problem by discretisation, which leads to similar computational problems as the above-mentioned ones. Some alternative formulations avoid the use of integer variables by using sequential optimisation as in Figure 2, where the optimiser calls a dynamic simulation of the model each time it requires the values of the cost function J or of the constraints $\boldsymbol{g}$, [8]. The simulation contains all batch elements and the sequencing only requires to determine their starting times, which are continuous variables. Thus, the optimisation problem is reduced to a non-linear programming (NLP) one. Nevertheless, the sequential approach limits the performance of the solution such that only problems of limited size can be solved efficiently.

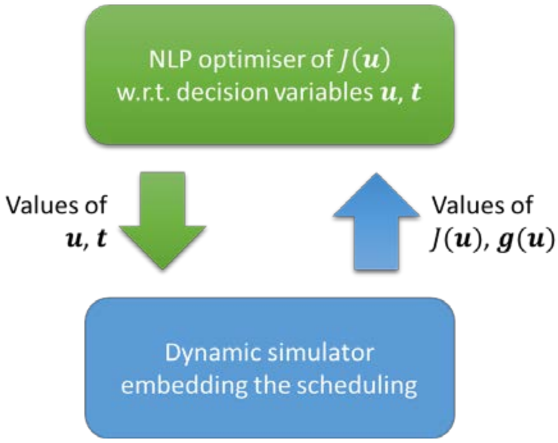


Figure 2: Sequential optimisation with the batch or discrete elements embedded in the simulation

3.2 Cooperative solutions

In order to deal with more realistic industrial problems, collaborative schemes with different modules that interchange information and share aims are required [9]. There, many possible schemes that range from hierarchical to more loose or cooperative structures.

In the hierarchical approach, as shown in Figure 3, the scheduling typically acts as the coordinator, which receives information about the actual durations of the tasks and their consumption of resources. The results of the coordinator level for the allocation and sequencing decisions are transferred to the control layer where optimal trajectories are computed and implemented. The control layer operates each batch process and has certain flexibility in order to adapt to the process conditions and to implement local optimisations. Frequent re-scheduling allows to take advantage of the updated information provided by the control layer.

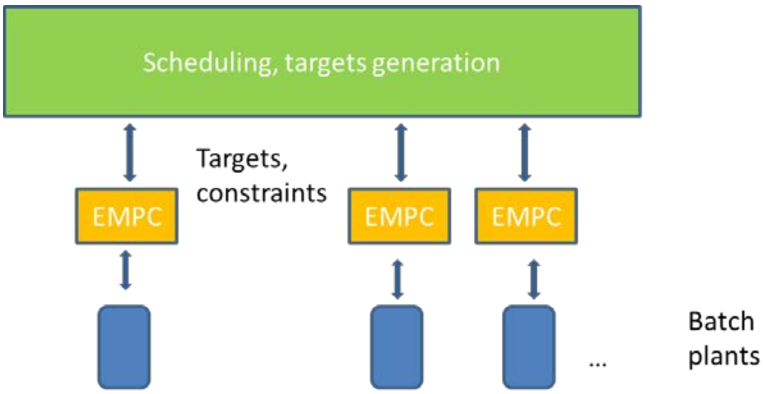


Figure 3: Hierarchical approach for integration of scheduling and batch control

An example application of these ideas can be seen in [10], where the joint operation of scheduling and MPC is performed via interchanging predictions and feasible ranges of operation between the individual layers.

At this point, it is worth to mention that it is difficult to propose detailed schemes for these problems because of the wide variety of scheduling problems found in practice. Thus, it is necessary to focus on specific families. Common difficulties as, for instance, the way of computing efficient and robust numerical solutions remain for all of them.

In addition, the advance in the implementation of digitalisation and open information systems, following the Industry 4.0 directives, is paving the way for the implementation of coordinated solutions. These advances offer architectures closer to the right-hand side of the well-known automation pyramid in Figure 4 than to the classical one on the left.

In the next section these integration procedures are applied to a particular industrial case. In a tuna canning plant (Frinsa use case), there is a discrete flow of product through continuous processing lines, that has to be submitted to a batch process in the middle of the production scheme. Sometimes, when a batch process is in the setup, it becomes the bottle neck of the production. This interruption in the continuous production can lead to decreasing the production rate of the whole system if the sections do not work at the same pace.

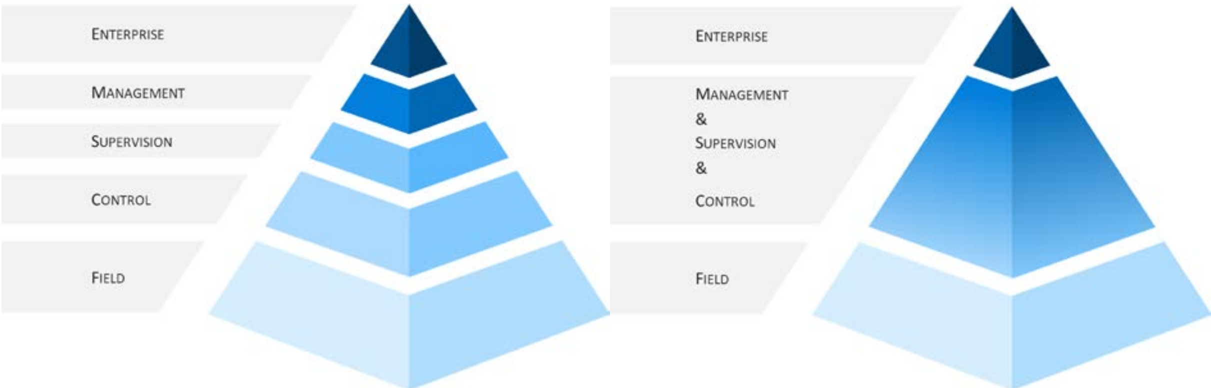


Figure 4: Automation pyramid comparison between the classical with well-defined levels and one with levels blurred[1]

[1] Pyramid diagrams adapted from an original by © Copyright Showeet.com / CC BY-NC-SA

4 Application to the FRINSA use case – Part I

In the tuna canning plant, a multiproduct thermal treatment is carried out in several parallel similar autoclaves. All the cans must go through this treatment known as sterilisation. The cans are introduced in industrial metallic carts that are gathered in groups up to the capacity of the autoclaves. The scheduler has to calculate the size of the groups, allocate them in the appropriate autoclave, and then compute the start instant of every sterilisation process. The type of cans/products inside the carts gathered determines the length of the sterilisation process relative to that group and the temperature set point. In addition, the scheduler must consider hard time constraints related to every cart, v.gr. every can has a maximum waiting time since it is released until it starts the sterilisation procedure.

4.1 Optimised control of the autoclaves

The thermal treatment guarantees that the temperature in the coolest spot of the can in the least accessible place is higher than a certain temperature for a predefined duration. This relation temperature/time is computed to assure that the bacteria lethality exceeds a fixed threshold. The evolution of temperature and other properties depends, not only on time, but on the spatial position. The models must be formulated in terms of partial differential equations (PDE). Each sterilisation unit consists of a plate heat exchanger, the autoclave and the cans. The whole sterilisation unit is modelled, from the heat transmission in the plates to the temperature distribution inside the cans.

$$\begin{gathered} m_w c_{p,w} \frac{\partial T_{w,j}}{\partial t} = -v_w m_w c_{p,w} \frac{\partial T_{w,j}}{\partial z} + U \cdot A_p \left(T_{s,j} - 2T_{w,j} + T_{w,j+1} \right) \\ \frac{\partial T}{\partial t} = \alpha \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) + \frac{\partial^2 T}{\partial z^2} \right] \end{gathered} \tag{3}$$

For instance, these PDEs show the temperature evolution in the plates and the temperature in the product. There is a more detailed explanation on how to obtain accurate models in [11].

In the heat equation of the plate the parameters m_w and v_w are the mass and velocity of water respectively. Whereas $c_{p,w}$ is the specific heat of water, U is the heat transfer coefficient and A_p the contact area between plates. Last $T_{w,j}$ and $T_{s,j}$ are the water and steam temperatures in the plate j respectively. In the retort temperature evolution model, r and z are spatial dimensions and α is the thermal diffusivity. See [12] for a whole model explanation. Some model simulation results can be seen in Figure 5, where the evolution of the lethality (red line) is obtained from the sterilisation temperature evolution (blue line).

A multi-objective problem is proposed, where the manipulated variables are the temperature set point and the duration of the sterilisation. The objectives are: to minimise the heating resource consumption (steam); to maximise the quality, which is measured as colour loss; and minimising process duration. The lethality constraints have to be fulfilled. If we denote as J_1 steam consumption and as J_2 colour loss, we can obtain the Pareto front by solving the optimisation problem (4) for several values of κ, where the inputs pt and T_{sp} are the process time and the temperature set point respectively, κ is a fixed value to constrain the objective function J_2, i.e. $\kappa \in \left[\underline{\kappa}, \overline{\kappa}\right]$, where $\underline{\kappa}$ is the minimum possible loss colour and $\overline{\kappa}$ is the maximum one, and $\boldsymbol{x}$ are the states of the model.

$$\begin{gathered}\min_{pt,T_{sp}} J_1 \\ s.t. J_2 \geq \kappa \\ \boldsymbol{h}(\dot{\boldsymbol{x}}, \boldsymbol{x}, pt, T_{sp}) = \boldsymbol{0} \\ \boldsymbol{g}(\boldsymbol{x}, pt, T_{sp}) \leq \boldsymbol{0}\end{gathered} \tag{4}$$

By default, one Pareto optimal combination temperature/time has been chosen for every type of product. This set of durations of the different processes is passed to the scheduler to determine the order of the sterilisations.

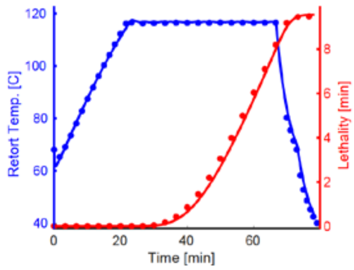


Figure 5: Comparison between experimental data (marks) obtained at a tuna canning site and model simulation results (continuous lines)[2]

4.2 Optimised scheduling of the sterilisers

The optimisation of the scheduling does not consider that the duration depends on the temperature, and therefore it is fixed during the computation. Usually, one scheduler will solve the mixed-integer lineal programming (MILP) optimisation problem (5), where I is the set of sterilisation processes, and st_i, and pt_i are the starting and processing times of every sterilisation process $i \in I$, MK represent the makespan, which is the last ending time of the sterilisations, and the vector $\boldsymbol{y}$ contains all Boolean and continuous variables used to model the cart gathering, the allocation in the autoclaves, and the ordering of the sterilisations in the optimisation problem. The parameters A and b model

$$\begin{gathered}\min_{\boldsymbol{y}} MK \\ s.t. MK \geq st_i + pt_i \quad \forall i \in I \\ A \cdot \boldsymbol{y} \leq b \\ \boldsymbol{y} \geq 0\end{gathered} \tag{5}$$

those constraints as linear constraints.

Here, we have defined the problem using continuous slots. Defining the time base as continuous instead of discrete, allows to improve the accuracy of the solution. Despite this, the resolution time increases, although it is kept in an acceptable level [13].

The scheduler gathers the carts in groups, where the type of carts (meaning type of product inside the cart) must not be different, unless their sterilisation procedures are similar, one Boolean variable determines if one cart is included in one group. Then, the groups are assigned to an autoclave, which

[2] Source: [12] / CC BY-NC

is also represented by a Boolean variable. In addition, to ease the operators work, carts cannot travel far away from their releasing lines. As the carts have a maximum waiting time, the sterilisation process of any group must start before all the time limits of the carts included have past. The scheduler then has to compute the starting point (st) of every sterilisation procedure. Their duration (pt) is dictated by the type of carts introduced. Precedence constraints are added to model the sterilisation order in every autoclave [14].

4.3 Including control into the scheduling

To ensure convexity of the problem and a reduced computational time, the optimisation problem is modelled again as a MILP problem. First, the Pareto front is calculated offline (see Figure 6) by solving the multi-objective where the axes are scanned iteratively. This provides a hyperplane that relates all the Pareto optimal solutions to their respective combination of temperature set point and duration of the procedure.

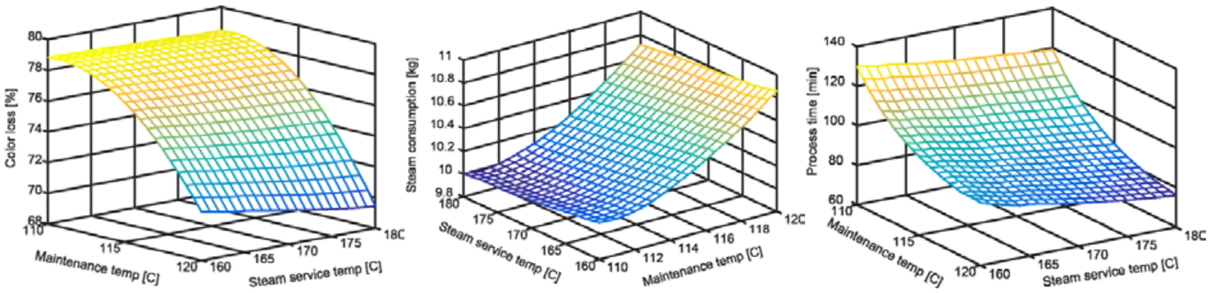


Figure 6: Representations of the Pareto fronts

The lethality constraint sets a minimum value for sanitary reasons. This value can be exceeded, at the expense of penalizing the objective functions. This possible increase in the lethality value creates a feasible region over the curve that relates the manipulated variables. As this relation is convex, the feasible region created is also convex, see Figure 7.

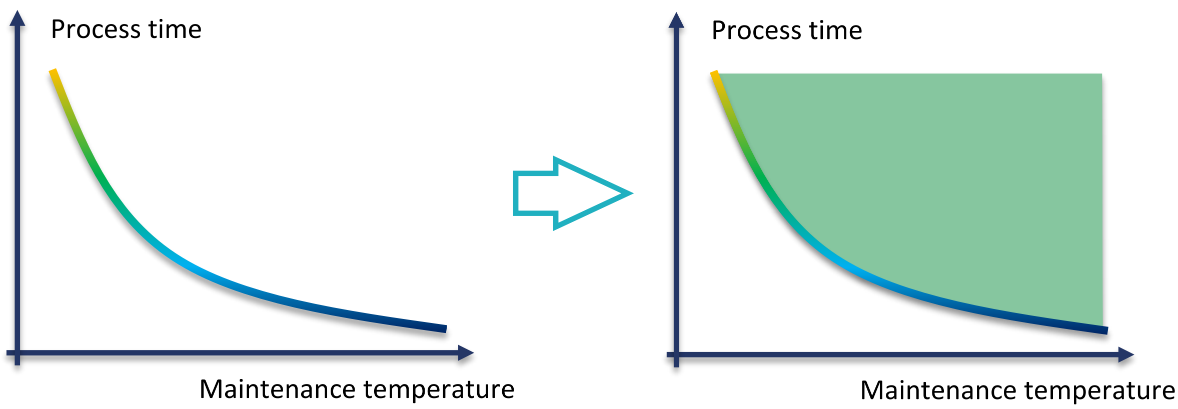


Figure 7: Extrapolation of the relation maintenance temperature set point vs. length of the sterilisation (left) and the representation of the feasible region (right)

Therefore, this control decision can be included easily in the MILP scheduling problem by linearizing the region using as many straight lines as necessary to approximate the curve sufficiently well (Figure 8), i.e. add the linear constraints shown in (6).

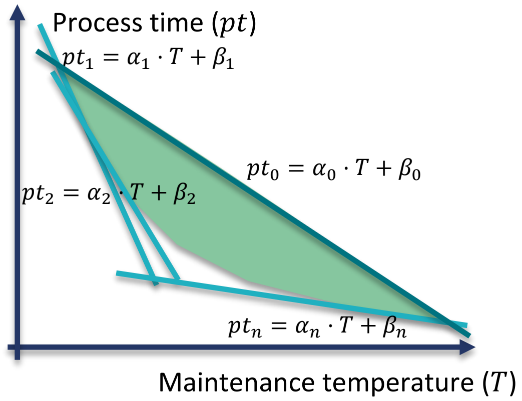


Figure 8: Linearization example of the feasible region

$$
\begin{aligned}
pt &\geq \alpha_i \cdot T + \beta_i \quad \forall i \in [1, n] \\
pt &\leq \alpha_0 \cdot T + \beta_0
\end{aligned} \tag{6}
$$

4.4 On considering shared resources

When the parallel pieces of equipment share a common resource, it is distributed among all according to the current demand of each one. Usually, the parallel pieces of equipment do not start at the same time and the demands depends on the product they are processing. Consequently, their respective resource consumption profiles differ. Without incorporating the real consumption profiles to the scheduling, the available limit must be overestimated, which results into a waste of resources. On the other hand, if this is not taken into account, it could produce infeasible scenarios, where the supply does not match the demand. This will lead to a forced stopping of the previous lines in order to be able to handle all incoming production [15].

In the Frinsa use case, the autoclaves share steam from the boilers as the heating resource. The sterilisation process does not maintain a constant steam demand, but it varies during the process. The sterilisation process can be divided into three different phases: the heating phase, the plateau phase, and the cooling phase. During the first phase the temperature inside the autoclave is increased up to the temperature set point, where the sterilisation process itself begins. Then, the plateau or maintaining phase begins, where the temperature has to be kept above the set point for a precomputed duration to achieve the required lethality. Finally, during the cooling phase the cans temperature is reduced to certain value. See the Figure 9 for more information.

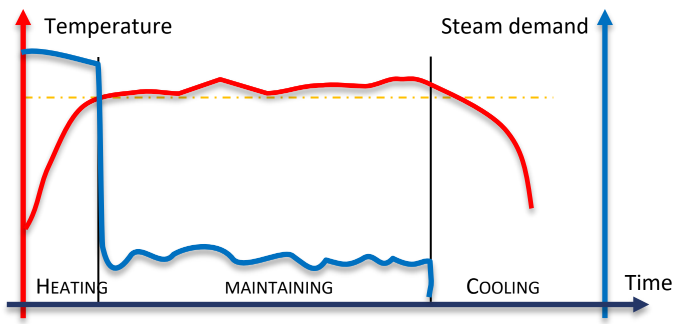


Figure 9: Representation of one sterilisation process phases

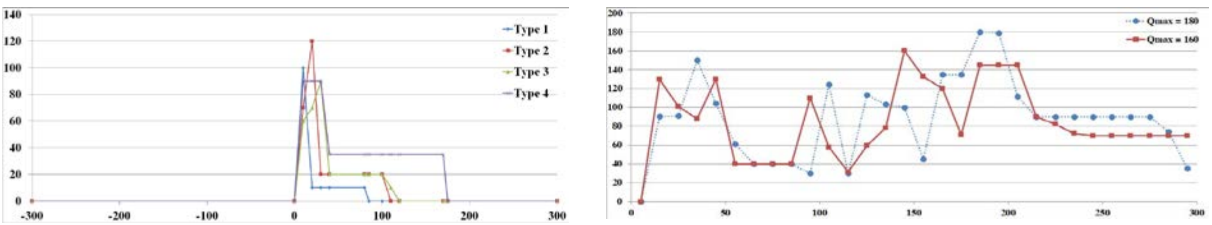
Whereas the steam demand along the plateau phase does not affect the system significantly, during the heating phase of the autoclaves the high demand lowers the pressure of steam in the resource network. This decrease can produce an increase in the time required to achieve the temperature set point. In this case, the predefined schedule will no longer be suitable, and it will have to be tailored to the new scenario.

Hence, the consumption profile must be included into the scheduling model. Here, we present two different approaches that were implemented in the sterilisation section. The first one sets the maximum supply capacity of the system so that the scheduler will prevent the autoclaves to demand more than it is available. The second one considers that the available capacity is never sufficiently large and it deals with the rise in the sterilisation duration, where this delay is modelled. Nevertheless, both options need to include some models of the performance of the autoclave.

4.4.1 Maximum available supply

Computing the aggregate of a continuous quantity at every time using a continuous time base is a complex task. Unlike using a discrete time basis, the evolution of a process cannot be directly known. To overcome this problem, we have introduced different time bases. The continuous time base to solve the scheduling; and a discrete time base, to compute the aggregation of the steam consumption of all the sterilisation processes.

Each product has a different sterilisation program that varies in its span and temperature set point. Therefore, steam consumption profiles also differ. To include the steam consumption profiles in the scheduling problem, first they are linearized. As the system uses a continuous time base, the linearization of the profiles is done by approximating piecewise functions. However, the order of the sterilisation procedures is unknown, hence the profile followed by each procedure is also unknown. To be able to handle all possibilities, the set of time instants chosen to linearize every consumption profile coincide. Therefore, using the convex-hull method [16] and interpolation, one can estimate the actual consumption for every sterilisation process at every time instant. See Figure 10(a) for more information [17].



(a) (b)

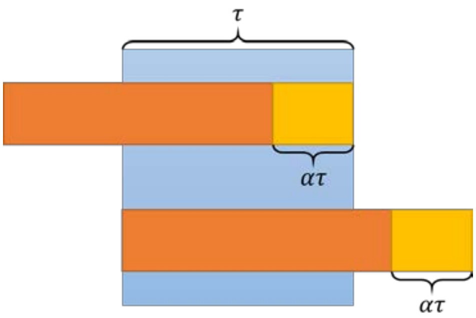
Figure 10: Example of piecewise functions to approximate steam consumption (a); Example of the overall steam consumption of the whole section (b)

The scheduler includes these models and maintains the overall consumption under a limit in every sample of the discrete time base, see Figure 10(b).

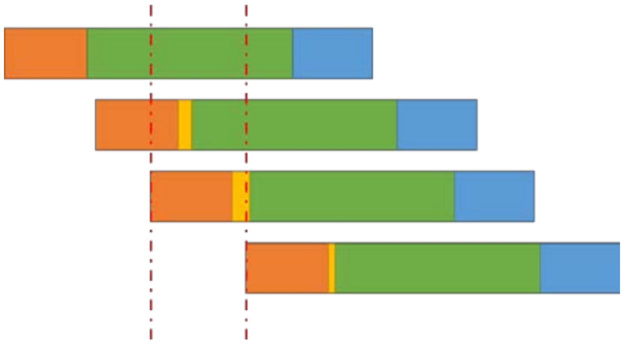
4.4.2 Altering the duration of the processes

In this version, instead of estimating the real steam consumption, we consider the variations in the duration of the sterilisation processes due to a high resource demand in the system. If two high demanding processes run at the same time, both see their respective duration extended by a time proportional to the coincidence span, see Figure 11(a).

In the sterilisation section, this high demanding process corresponds to the heating phase of the sterilisation. Even though during the plateau phase there is also steam consumption, the capacity of the system can handle this without problem as shown in Figure 11(b).



(a)



(b)

Figure 11: Representation of the effect of the coincidence of high demanding procedures (a); Representation of the effect in the sterilisation procedures (b)

In order to cope with the variation of the process length caused by the optimizer, first the process time is divided into two variables (7). One variable indicates the heating phase duration (h_i), the other variable indicates the sum of the plateau and cooling phases (mc_i), as they are constant with respect to their relative type of can. Then, the heating phase is computed as the minimum possible heating phase duration (φ) plus the interference suffered from the rest of heating phases. The interference is approximated as the aggregation of all the interferences suffered in pairs ($\iota_{i,i'}: i, i' \in I, i \neq i'$).

$$
\begin{aligned}
pt_i &= h_i + mc_i \quad \forall i \in I \\
h_i &= \varphi + \sum_{i' \in I : i' \neq i} \iota_{i,i'}
\end{aligned}
\tag{7}
$$

As a result, the scheduler will be able to handle the synchronisation of the heating phases to reduce the interference when possible, and to prevent infeasible situations where one sterilisation process is set to start but the previous one has not yet ended.

In the Figure 12, one example solution for a scheduling is shown. On the vertical axis the different autoclaves are represented and on the horizontal axis the time is indicated. The sterilisation processes are divided into two: the heating phase and the phase of the plateau and cooling. The different colours represent the different types of products that have to be sterilised, therefore the length of the second phase is equal for all the procedures with the same colour. Meanwhile, the heating phase depends on the coincidence with other heating phases, therefore it changes for every sterilisation procedure [18], [19].

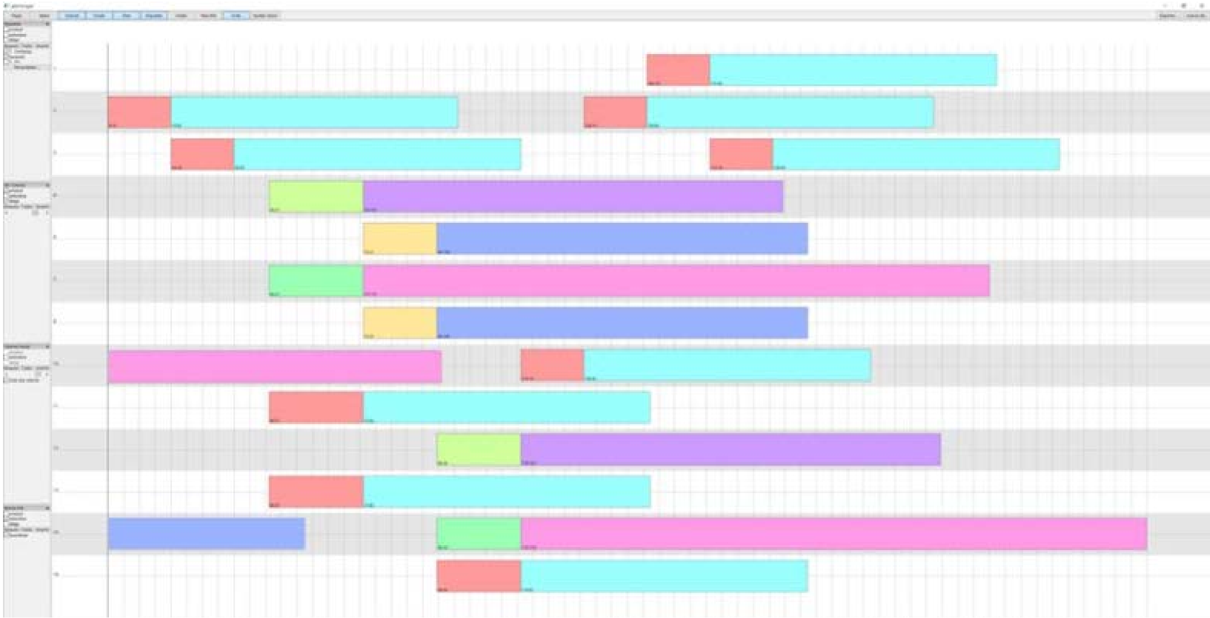


Figure 12: Example of a scheduling solution with the high demand phase interference integrated

4.5 Real-time implementation

Analogous to the control of the procedure, that reacts to changes in the system in real-time, the scheduler that integrates those controls must adapt itself. Hence, we propose to apply a rolling horizon approach, which enables a real-time execution of the tool. Rather than obtaining a solution for a whole week that cannot include the control parameters, the real-time optimisation will be able to propose optimal thermal profiles that focus just on a small period. Only the near future arrivals of carts are included as inputs to the system. At the beginning of every execution the system reads the current state, adapts and optimises again based on the current situation.

To improve the performance of this real-time control and schedule optimisation, it has an interface to the MES system. It can be executed automatically by the system and suggests allocation and control policies to the operators. The suggestions incorporate the allocation of the carts and the control parameters to set. If the operator does not accept those suggestions, the scheduler must reorganise the system.

To do so, we have defined a main procedure that reads the database, compares the current status to the predicted one, readjust the inputs of the optimiser and then runs it again. This procedure is explained in the pseudocode in Algorithm 1. First, it computes the time that has passed since the last iteration, then it captures the actions that occurred during this period and it makes a new prediction based on these past actions. Subsequently, it removes the wrong predictions and suggestions that were not followed and modifies the constraints of the optimisation problem. Afterwards, it executes the optimisation again.

As the prediction of carts is updated at each iteration with information from the MES system, this is be able to handle changes in the processing lines, as emergency stops or increases in the production, or failures of the autoclaves, and readjust the schedule.

Algorithm 1: Pseudocode for the real-time scheduling tool

```
BEGIN
    previous ← now
    now ← getTime()
    RealCartArrivals ← readDatabase(previous, now)
    PredCartArrivals ← PredCartArrivals ∪ predictArrivals(previous, now,
predictionHorizon)
    cleanCarts(RealCartArrivals, PredCartArrivals, now)
    adjustConstraints(model, PredCartArrivals, RealCartArrivals, now)
    optimize(model)
END
```

In the pseudocode shown in Algorithm 2, it is shown how the system handles an upgrade of the system to the current status. First, it modifies the model to include the real status of the system, i.e., the real arrival times of the carts instead of the predicted times. Then, it adds new constraints in case of a not predicted arrivals to the system.

Algorithm 2: Pseudocode for the adjustContraints procedure

```
adjustConstraints()
    input:
    • model, optimisation model in OPL to pass to the optimizer
    • PredCartArrivals, matrix that has all the data respective to the carts
      predicted by the system using the overall planification from the ERP system,
      the first carts have been corrected to the real arrival of carts, in order to
      update the model to the current state of the section
    • RealCartArrivals, matrix that has all the data respective to the carts that
      have arrived but were not contemplated by the prediction and must be included
      in the current state
    • now, time instant where the software tool started to execute, to know up to
      when trust the prediction of carts
BEGIN
    i ← 0
    while PredCartArrivals[i].arrivalTime ≤ now
        alterConstraints(model, PredCartArrivals[i])
        i ← i + 1
    end while
    for j ← 0 to size(RealCartArrivals.arrivalTime)
        appendCarts(model, RealCartArrivals[j])
    end for
END
```

Finally, in Algorithm 3 the procedure to compare the predicted arrivals to the real arrivals is shown. The optimisation model has to constrain the start point of the sterilisation procedures with respect to the real release times of the carts from the sealing lines.

This procedure checks the list of predicted carts and compares it with the list of real arrival times. The carts that have been predicted but have not arrived are removed from the list. The carts that have arrived but have not been predicted are included in the next iteration of the optimiser.

Algorithm 3: Pseudocode for the cleanCarts procedure

```
cleanCarts()
   input:
      • RealCartArrivals, matrix that has all the data respective to the carts that the
        system has freed from the sealing lines to the sterilisation section
      • PredCartArrivals, matrix that has all the data respective to the carts
        predicted by the system using the overall planification from the ERP system
      • now, time instant where the software tool started to execute, to know up to
        when trust the prediction of carts
BEGIN
    I ← 0
    cartArrived ← new vectorIntegers()
    cartChecked ← new vectorIntegers()
    while PredCartArrivals[i].arrivalTime ≤ now
        for j ← 0 to size(RealCartArrivals.arrivalTime)
            if j ∉ cartChecked then
                if compareCarts(PredCartArrivals[i], RealCartArrivals[j])
                    append(cartArrived, i)
                    append(cartChecked, j)
                    j ← size(RealCartArrivals.arrivalTime)
                end if
            end if
        end for
        I ← I + 1
    end while
    for I ← (I - 1) to 0
        if I ∉ cartArrived then
            remove(PredCartArrivals, i)
        end if
    end for
    for j ∈ cartChecked
        remove(RealCartArrivals, j)
    end for
END
```

4.6 Summary and Conclusion

To adapt to the direction proposed by Industry 4.0, where flexibility becomes a key point, the automation methodology has to upgrade from the automation levels standard to a more adaptable one. The process lines must be variable aiming to on-demand production, where the intermediate stocks tend to disappear. Following this approach, the scheduling must be combined with control strategies, to incorporate the effects that they have one in the other.

Here, different approaches have been presented to deal with the scheduling of batch processes incorporating the optimal control of the equipment. In addition, the problem of scheduling with shared resources is also managed. All the implementations paths have been designed to obtain real-time optimization tools to apply in the plant. These tools can adapt to the current status of the system and offer decision support to the operators.

The optimization algorithms have been shown applied to a particular use case in a tuna canning plant, however all the methods presented can be easily extended to others industries.

5 Application to the FRINSA use case – Part II

Thermal food sterilisation process needs to optimally handle a number of products with different features and quality requirements. The optimisation in this case means that for a given number of products and plant equipment, it is necessary to find the optimum way to sterilise the products within minimum plant operation time, cost and energy consumption. The plant equipment refers to the number and capacity of the retorts as well as the available resources. *Figure 13* shows an overview of the steriliser unit with the parallel heat exchanger (PPHE) into consideration.

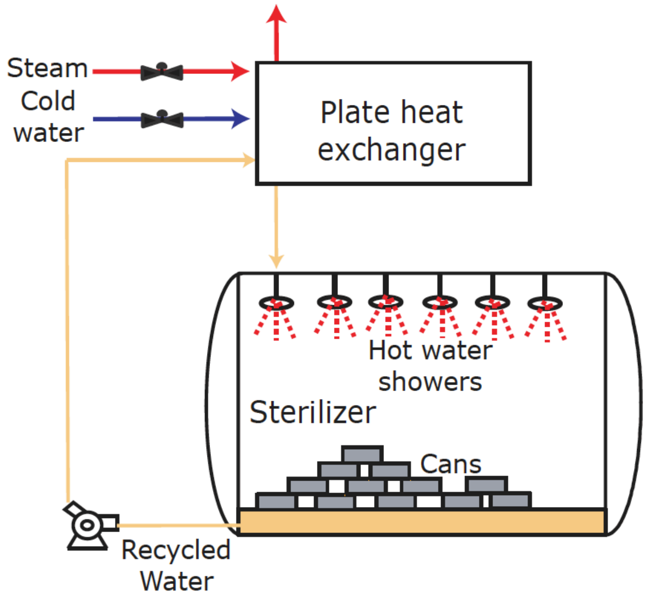


Figure 13: Steriliser unit with parallel heat exchanger[3]

The operation cycle begins by the insertion of preheated water (≈ 60°C) into the steriliser and recycled to the PHE. The water temperature in the retort increases until it reaches the operation point (T_{retort} =110-130°C). Subsequently the temperature is kept for a given amount of time and the steam valve in the PHE is closed. After the lethality specifications are met, cold water is introduced to cool down the cans. The duration of the 2-stage cooling is 13mins. Initially the inlet water temperature is set at 65°C while the retort temperature is above 75°C and then the inlet water temperature is set at 20°C for the remaining cooling time.

This work involves the development of a model-based optimisation and control framework for the network of sterilisers based on FRINSA case study. A reduced order model (ROM) was developed with low computational complexity, which was verified against experimental data from FRINSA. Subsequently an NMPC framework was developed that relies on the ROM and a Nonlinear Programming Problem (NLP) problem formulation. Finally, the NMPC interacts with an RTO framework that considers economic and energy consumption aspects. The goals of the coordinated RTO and NMPC framework are:

- the optimum operation of multiple sterilisers using common steam resources
- the online determination of the start of cooling phase (lethality target)

[3] Source: [12] / CC BY-NC

- the usage priorities to improve the overall utilisation of the sterilisers
- the consideration of energy consumption

5.1 Control oriented reduced order model (ROM) of the sterilissation process

The reduced order model (ROM) that was developed consists of three parts and was used at the coordinated optimisation and control framework. It was necessary to capture the generic trend and behaviour of the involved subsystems at the system level and to derive a model which is suitable for control.

5.1.1 Can temperature governing equations

The governing equations for the can temperature (T_{can}) evolution profiles are:

$$\frac{dT_{can}}{dt} = Bi_2(T_{w,R} - (c_o + c_1 + c_2)(1+q/2)) - Bi_1(c_o(1+q/2) - T_{w,R}) + 2MBi_R(T_{w,R} - (1+q)(c_o + c_1/2 + c_2/3)) \quad (8)$$

where $q = \frac{Bi_R(T_{w,R}/(c_o + c_1/2 + c_2/3) - 1}{2 + Bi_R}$ (9)

$$Bi_R = \frac{h_R R}{k}, Bi_1 = \frac{h_{Z1} L}{k}, Bi_2 = \frac{h_{Z2} L}{k}, M = \frac{L^2}{R^2}$$

where $T_{w,R}$ is the retort water temperature, k is the thermal conductivity, L and R are the length and the radius, $h_{(R,Z1,Z2)}$ is the convective heat transfer coefficient right, top and bottom of the can. In each time step c_o, c_1 and c_2 have to be found from the solution of the following 3x3 system of nonlinear algebraic equations:

$$T_{can} = (c_o + c_1/2 + c_2/3)(1+q/2) \quad (10a)$$

$$c_1(1+q/2) = Bi_1(c_o(1+q/2) - T_A) \quad (10b)$$

$$(c_1 + 2c_2)(1+q/2) = Bi_2(T_A - (c_o + c_1 + c_2)(1+q/2)) \quad (10c)$$

Integrating a single ordinary differential equation in conjunction with a system of three algebraic equations the evolution of complete two-dimensional temperature profile in the can is derived.

5.1.2 Parallel plate heat exchanger model (PPHE)

Let us assume a parallel plate heat exchanger (PPHE) with mass flow rate of water M_w and mass flow rate of steam M_s. A steady state problem for the counter current parallel heat exchanger is considered. The equations can be integrated in closed form and scaled back to the whole heat exchanger to give:

$$Q = M_s c_{ps}(T_{s,in} - T_{s,out}) \quad (11)$$

$$Q = M_w c_{pw}(T_{w,out} - T_{w,in}) \quad (12)$$

$$Q = UA(T_{w,out} - T_{s,in} - (T_{w,in} - T_{s,out})) / \ln((T_{w,out} - T_{s,in}) / (T_{w,in} - T_{s,out})) \quad (13)$$

where *Q* is the rate of heat transferred in the PPHE. The heat transfer coefficient and the (one side) surface area of the plate is *U* and *A* respectively. Q can be eliminated between the above equations to leave with a system of two equations with two unknowns (the outlet temperatures). The system of equations (11)-(13) is highly non-linear and can be replaced by the linear system:

$$T_{w,out} = \frac{T_{w,in} + p2\left(\left(1-p1\right)T_{s,in} - T_{w,in}\right)}{1-p1\,p2} \tag{14}$$

$$T_{s,out} = T_{w,in} + p1\left(T_{s,in} - T_{w,out}\right) \tag{15}$$

$$p1 = \exp\left(UA\left(\frac{1}{M_w c_{pw}} - \frac{1}{M_s c_{ps}}\right)\right) \tag{16}$$

$$p2 = \frac{M_s c_{ps}}{M_w c_{pw}} \tag{17}$$

For a given inlet temperatures to PHE the outlet temperatures are computed by (14) and (15) and then Q by equation (11).

5.1.3 Steriliser model

The heat balance in the steriliser is the leading equation of the model. The water temperature $T_{w,R}$ is dominated by the equation:

$$M_{wR} c_{pw} \frac{dT_{w,R}}{dt} = Q_{PHE} - Q_{can} - Q_{env} \tag{18}$$

$$Q_{PHE} = M_s c_{ps}(T_{s,\mathrm{in}} - T_{s,out}) \tag{18a}$$

$$Q_{can} = n_{can} M_{can} cp_{can} \frac{dT_{can}}{dt} \tag{18b}$$

$$Q_{env} = h_c A_c (T_{env} - T_{wR}) \tag{18c}$$

where M_{wR} is the water mass in steriliser, Q_{can} is the heat adsorbed by the cans and Q_{env} is a simple loss term where h_c, A_c are the corresponding heat transfer coefficient and the metal cover area of the retort respectively and T_{env} is the environment temperature. The total model consists of the main equation (18) which must be solved simultaneously to the can heating submodel (one ODE and three algebraic equations) and to the PPHE model (two algebraic equations). Everything else is post processing (e.g. lethality, colour) or control actions (e.g. steam valve) of the core model.

The model is verified against experimental data from FRINSA. *Figure 14* shows the simulated retort temperature (solid line) compared to the experimental data (dots).

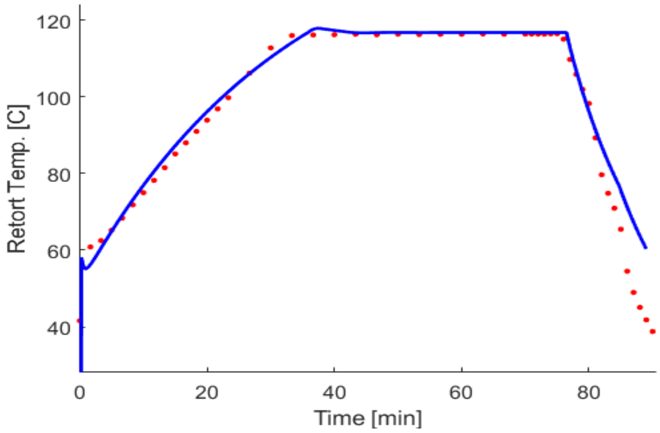


Figure 14: Evolution of the retort temperature

5.2 Problem formulation of the RTO and NMPC

An NMPC framework was developed that considers both the temperature and the lethality set-points in order to coordinate the operations of the steriliser. Preliminary results were obtained for the concurrent operation of the sterilisers using the specifications for real cans used by FRINSA.

5.2.1 Nonlinear Model Predictive Control (NMPC) Framework

The objective was to reach a predefined temperature and a lethality limit under hard constraints when multiple sterilisers are at the heat-up stage concurrently. The control framework considers the steam availability limitations and its effect to the operation of the sterilisers. Also, different start-up time profiles and various lethality set-points due to product variation were considered. Furthermore, the cooling stage initiates based on the lethality set-point instead of a fixed time duration from the beginning of the heat-up process. NMPC computes online a finite-time constrained optimisation problem over a prediction horizon (T_p), using the current state of the process as the initial state. The optimisation yields an optimal control sequence ($u_k..u_{k+Nc}$) over a control horizon (T_c), which is partitioned into N_c intervals and only the first control action (u_k) for the current time is applied to the system. We consider the following formulation of the NMPC problem:

$$\min J = \sum_{j=1}^{Np} (\hat{y}_{k+j} - y_{sp,k+j})^T QR(\hat{y}_{k+j} - y_{sp,k+j}) + \sum_{l=0}^{Nc-1} \Delta u_{k+l}^T R1 \Delta u_{k+l} \tag{19}$$

$$\text{s.t.:} \quad \dot{x} = f_d(x,u), \quad y = g(x,u) \tag{20a}$$

$$e_k = (y_{pred} - y_{meas})_k \tag{20b}$$

$$\hat{y}_{k+j} = y_{pred,k+j} + e_k \tag{20c}$$

where *u,y,x* are the manipulated, the controlled and the state variables, $y_{pred}, y_{meas}, y_{sp}$ are the predicted, the measured variables and the desired set-points and *QR, R1* are the output tracking and the input move weighing matrices. The minimisation of functional *J* (eq. 19) is also subject to constraints of *u, x* and *y*. The controlled variables are the retort temperature ($T_{w,R}$) and the lethality (F_0) and the manipulated variable is the steam flow. A centralised approach (*Figure 15*) is applied for the control of the sterilisation process that uses the nonlinear model described by eq (8)-(18).

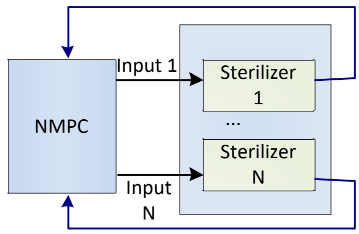


Figure 15: Centralised NMPC architecture

5.2.2 Tuning of the NMPC parameters

The response of the NMPC controller depends on the model and its parameters. The validity of the model was verified at the previous sections. The values for the necessary parameters were selected after a thorough analysis by running various operating scenarios. We were interested on determining the appropriate sampling time and the prediction horizon duration at the minimum computational cost and the maximum accuracy for the results. A model mismatch was introduced at the optimisation of the NMPC controller (ideal and process model) in order to be able to study the response of the NMPC controller under uncertainties and deviations imposed by the process measurements against the mathematical model.

Initially a scenario where only one steriliser operates was simulated with different prediction horizon length (T_p=1-5). At *Figure 16* the NMPC scenario results are shown. The scenario involved the operation of one steriliser for one batch of cans. The scenario duration was 1.5hr. We can observe that the simulation time increases along with the prediction horizon and decreases as the sampling interval of the model increases. In order to select the appropriate parameter values the accuracy of the model against the experimental data is explored.

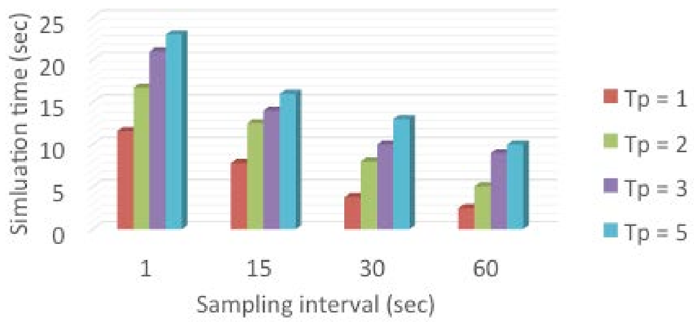


Figure 16: Sampling interval and optimisation time for one steriliser

As far as the accuracy is concerned, the modification of the sampling interval does not affect the model accuracy. On the other hand, accuracy is improved for a prediction horizon of Tp=2, compared to Tp=1 and the overshoot after the heat-up stage is minimized. The selected NMPC parameters for the rest of the analysis Tp=2, ts=60sec.

5.2.3 Operation of multiple sterilisers concurrently

After the tuning of the parameters of the NMPC, a set of scenarios where multiple sterilisers operate concurrently was explored. The sole purpose of this study was to evaluate the flexibility of the NMPC under various conditions and different operation requirements. An indicative list of the executed scenarios is:

- Scenario C1. Simultaneous operation at different lethality set-points
  - A) apply a common maximum steam flow constraint for each steriliser
  - B) apply different maximum steam flow constraints
- Scenario C2. Operation with different start-up time
  - A) common lethality set-point
  - B) different lethality set-point
- Scenario C3. Operation with different start-up time and lethality set-point
- Scenario C4. Simultaneous operation of sterilisers with different priorities

At each scenario the behaviour of 1 to 16 sterilisers is studied along with the respective computational requirements. We assume that each steriliser processes the same amount and type of cans. *Figure 17* presents the temperature evolution of 5 sterilisers that start to heat-up at the same time and have the same steam flow upper bound but each one has a different lethality set-point.

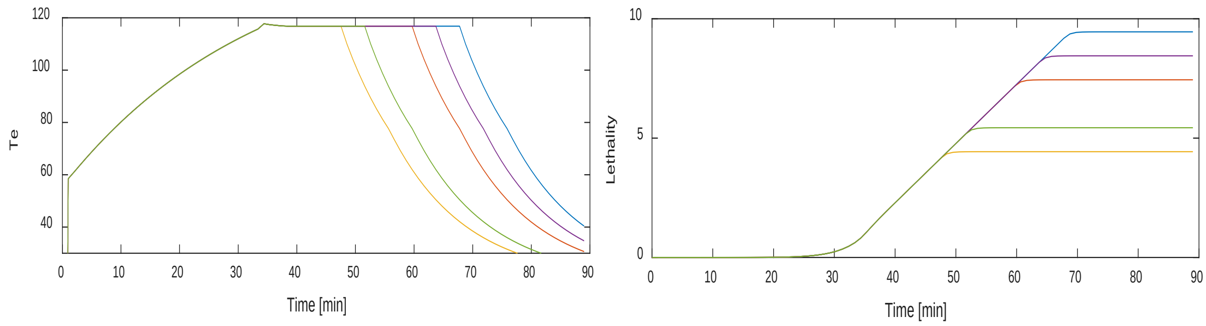


Figure 17: Concurrent operation of 5 sterilisers a) temperature evolution b) lethality evolution

Figure 17 shows that each steriliser reached at the same time the temperature set-point and that the overall operation is affected by the lethality set-point. The results of another scenario are presented at *Figure 18*.

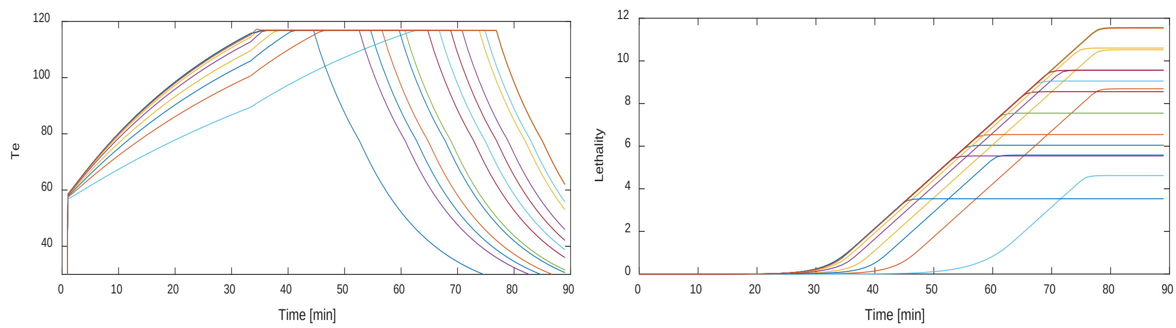


Figure 18: Concurrent operation of 16 sterilisers with different steam flow upper bound a) temperature evolution b) lethality evolution

The heat-up duration is affected by the steam flow's upper bound. As the upper bound decreases, the heat-up time increases. Also, since each steriliser has different lethality set-point the retention time varies.

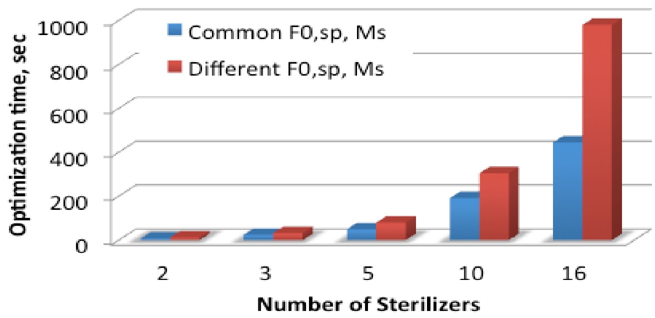


Figure 19: Computational requirements under common or different set-points for lethality and steam flow

Figure 19 presents the computational requirements when different lethality set-point and maximum steam flow rate is applied at each steriliser (Scenario C1). It can be seen that the computational requirements increase significantly when different set-points are applied to each steriliser. Another case is presented at *Figure 20*. We can see how the sterilisers operate when different start up time is applied and each one has its own lethality set-point.

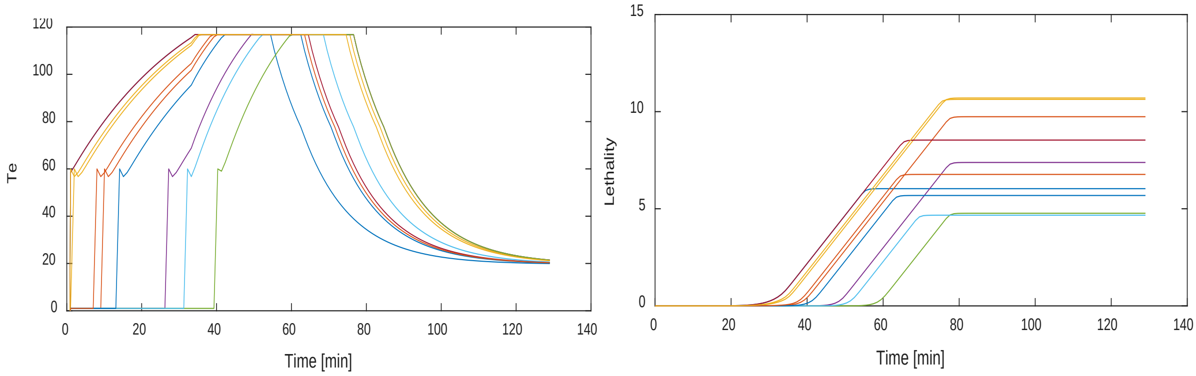


Figure 20: Operation of 16 sterilisers that start to heat-up at different time a) temperature evolution b) lethality evolution

In all scenarios the lethality requirement is satisfied and the overall operation time is modified according to the start-up time or the bounds of the steam flow rate. As a result, we can conclude that the centralised NMPC can operate under a wide range of conditions.

5.2.4 Development of a distributed NMPC architecture

The next part of our study involves the response analysis of a distributed NMPC architecture. For this purpose, a distributed NMPC framework was developed and applied to the same can sterilisation process. The models of the can, the PPHE and the steriliser (eq. (8)-(18)) were used by each controller with the same assumptions. The main difference from the previous formulation is that the upper bounds of the steam flow rate are modified and depend on the number of concurrent sterilisers that are on heat-up stage.

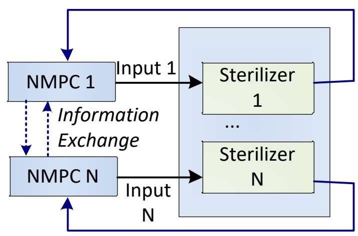


Figure 21: Information flow of the distributed NMPC architecture

A nonlinear NLP problem is solved at each iteration while the NMPC parameters of the individual controllers remain the same as in the centralised architecture (Tc=1, Tp=2, ts=60sec). In order to explore the response of the distributed NMPC architecture a scenario where 3 sterilisers start to operated concurrently is presented. At each steriliser two subsequent batches of cans are entered.

Figure 22 presents the retort temperature evolution while *Figure 23* presents the duration of the batches at each steriliser. The steam flow rate is adjusted according to the number of sterilisers that are concurrently at heat-up stage. Thus, the heat-up time is affected accordingly when multiple sterilisers are activated. For example, the slope of the 2nd (yellow) steriliser changes when the 1st steriliser reaches the heat maintenance stage and when the 3rd steriliser enters the heat-up stage.

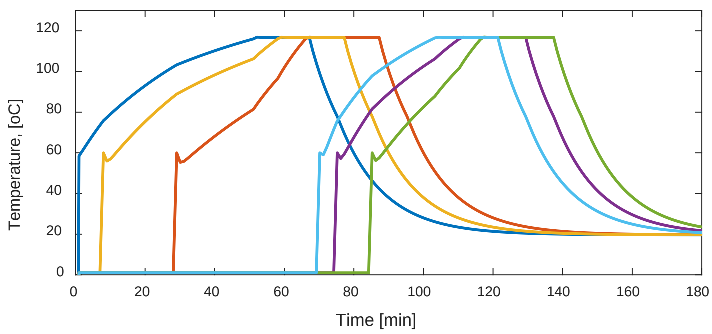


Figure 22: Operation of 3 sterilisers that start to heat-up concurrently

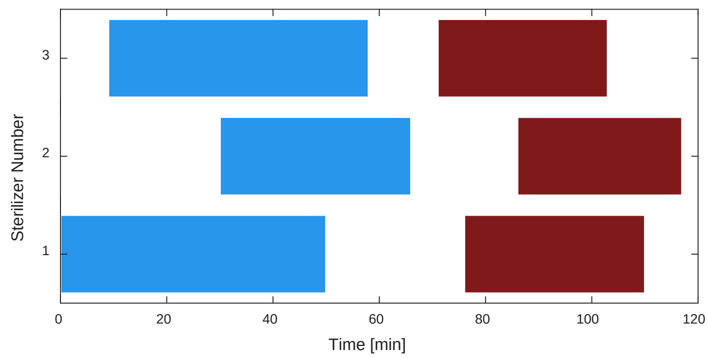


Figure 23: Batch duration

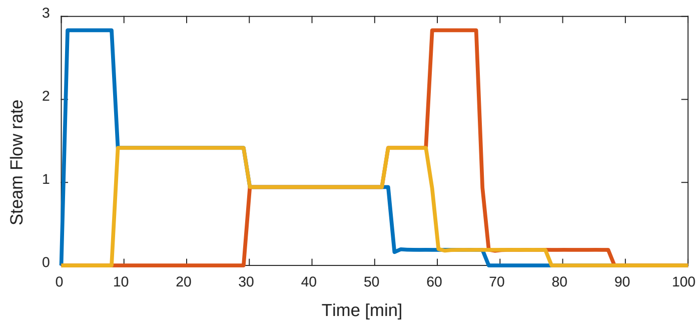


Figure 24: Steam distribution between 3 sterilisers during heat-up stage

Finally, *Figure 24* shows the distribution of steam between the 3 sterilisers. Initially the 1st steriliser operates. The 2nd steriliser start to heat-up at t=8min and the steam is shared equally between the 2 sterilisers. Finally, the 3rd steriliser is activated at t=30min and the shared steam is further reduced between the 3 sterilisers. At t=50min the 1st steriliser reaches the temperature target and the steam request is reduced. After an hour only the 3rd steriliser is still at heat-up stage and is allowed to use the maximum steam flow rate from the network. Overall it can be seen that the steam is adjusted according to the operating stage of the sterilisers.

Another scenario that was applied, involves the concurrent operation of multiple sterilisers (1 to 16) for one batch of cans. The goal of this scenario was to evaluate the response of the NMPC architectures.

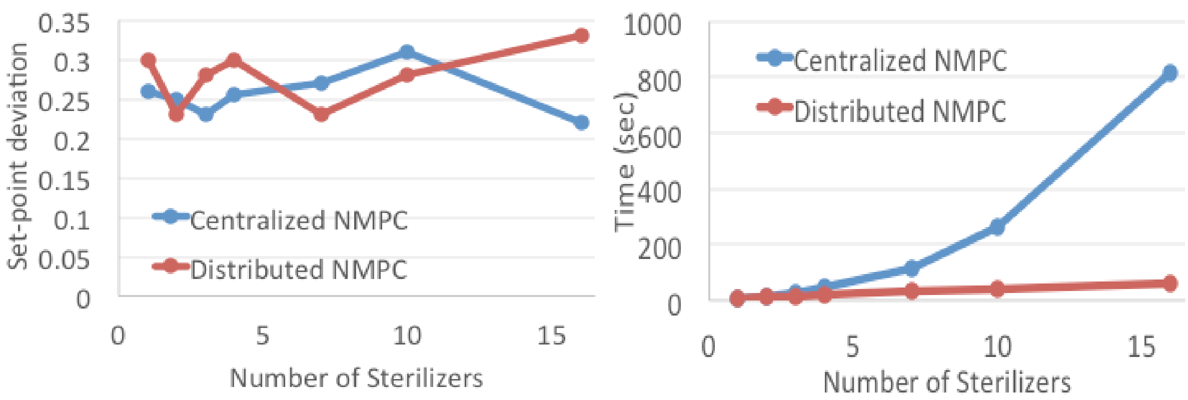


Figure 25: Comparison between centralised and distributed NMPC a) Lethality set-point deviation, b) Computational requirements

Figure 25 shows the lethality set-point deviation and the computational requirements. It is observed that both type of controllers, centralised and distributed NMPC, are able to satisfy the requirements regardless of the number of the steriliser. The main difference is at the significant decrease of the optimisation time in case the distributed NMPC is deployed.

5.2.5 Real-time Optimisation (RTO) framework

The RTO application optimises the process operating conditions and updates the local NMPCs. Dynamic RTO does not require steady-state conditions to be applied and the formulation of the DRTO problem is similar to the approach used to solve NMPC. The main difference is the inclusion of economic aspects in the objective function. Thus, the objective of the RTO framework is to improve the economic performance of the network of sterilisers that use common resources, in terms of increasing the usage time of the sterilisers and reducing the overall energy consumption.

5.2.6 Problem formulation

RTO incorporates different aspects related to the steriliser's operation in order to compute operational decisions that optimizes the operation efficiency and maximizes economy in terms of operating cost (consumed energy and operation time of each steriliser). The goals of the coordinated RTO and NMPC framework are to:

- Maximise production profit
- Minimise operation cost
- Minimise energy consumption

The RTO considers the cost of materials (canned food), values of products, and costs of production as functions of operating conditions. An objective function is specified in terms of these quantities over a predetermined period of time and it can be expressed as:

$$\max J = w_{Bcan}\sum_{i} pr_{can,j} B_{can,i} - w_{proc}\sum_{j} OC_{can,j} t_{proc,j} - w_{egy}\sum_{k} pr_{egy} E_{proc,k} \tag{21}$$

where pr_{can} and pr_{egy} are the costs related to the can that are being processed and the cost of energy, OC_{can} is the operating cost related to the specific product, t_{proc} is the processing time of the specific batch and E_{proc} is the energy consumption related to the specific batch. Also $w_{[Bcan,proc,egy]}$ are the weights for each term. Both the operating and economic terms include constraints on operating conditions, as process variables must be within certain limits due to steam availability (valve ranges (0% to 100% open). *Figure 26* shows the hierarchical structure of the RTO and the NMPC. At the higher layer, RTO is performed to compute the optimal operating conditions with respect to the performance index representing an economic criterion. At the lower layer the NMPC framework that was discussed earlier is applied and guarantees that the target values transmitted from the higher layer are attained.

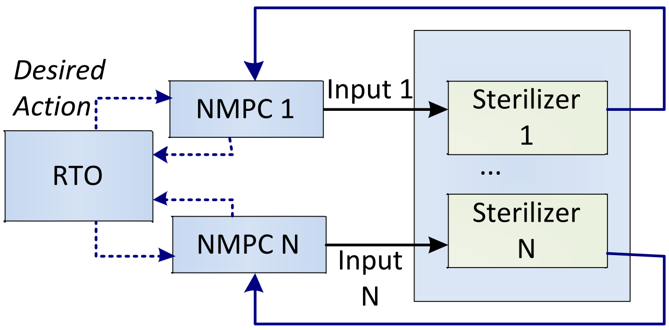


Figure 26: Hierarchical structure of the RTO and the NMPC

In this work the RTO and the NMPC formulation use the same nonlinear dynamic model of the process as described at Section 5.22. The adopted model needs to be periodically updated by means of some kind of reconciliation procedure to deal with changing operating conditions due to disturbances. The NMPC framework updates the RTO about the current state of the sterilisation process when RTO is invoked.

5.2.7 Evaluation of steriliser's operation using coordinated RTO and NMPC

In order to explore the response and behaviour of the coordinated RTO with the NMPC an indicative scenario is presented. The RTO runs every 20mins while the NMPC has a sampling time of 1min. A minimum steam flow rate is available for the sterilisers when they are initially activated and we assume that the total amount of available steam can be distributed to the sterilisers that are in heat-up stage. The execution of the RTO yields the values of each steriliser's priority, which is subsequently translated into available amount of steam flow.

The scenario involves the operation of 5 sterilisers that are initially activated within a 30min window time frame. Overall the scenario shows the operation of the sterilisers for one work shift (8hrs). During this time period 33 batches are accommodated. The batches are not identical to each other, which means that they have different number of cans and type of cans to process. The interval between two consecutive batches is randomly selected and ranges from 5mins to 15mins which is a typical time for the steriliser to empty and the new batch to be loaded for processing. We assume that there is always a cart waiting to enter the steriliser and there are no periods of inactivity of the sterilisers.

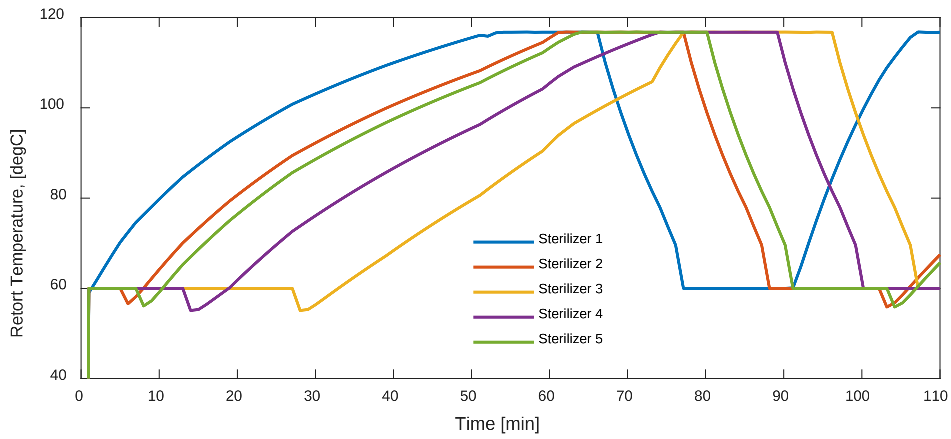


Figure 27: Operation of 5 sterilisers that have an overlap during their heat-up stage (1 batch)

Figure 27 show the retort temperature evolution for the first batch of each of the sterilisers. The slope of the temperature is affected every time the RTO is executed and new sterilisers are activated. The coordinated RTO gives priority to the sterilisers that are closer to the end of the heat-up stage compared to the sterilisers that were recently activated.

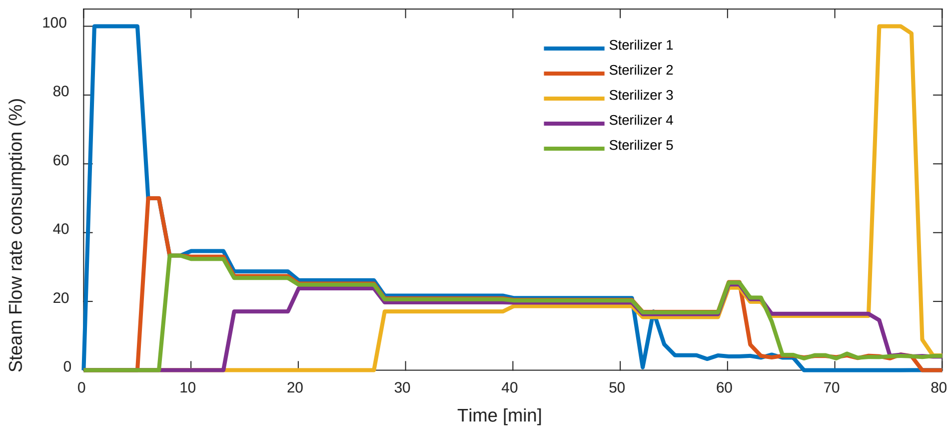


Figure 28: Steam distribution between the sterilisers (1 batch)

Figure 28 presents the distribution of the steam flow among the sterilisers during the first batch of their operation. It is observed that the steam flow is adjusted appropriately based on the outcome of the real-time optimisation.

An overview of the temperature evolution is presented at *Figure 29*. It is observed that they have a significant overlap during their heat-up stage. *Figure 30* shows how the steam is distributed among the sterilisers during the 8-hour shift. Overall it is observed that there are only short periods of time where only one steriliser is activated and can have the maximum amount of steam during its heat-up stage. Finally, at *Figure 31* we can see how that batches are being processed during the 8-hour shift.

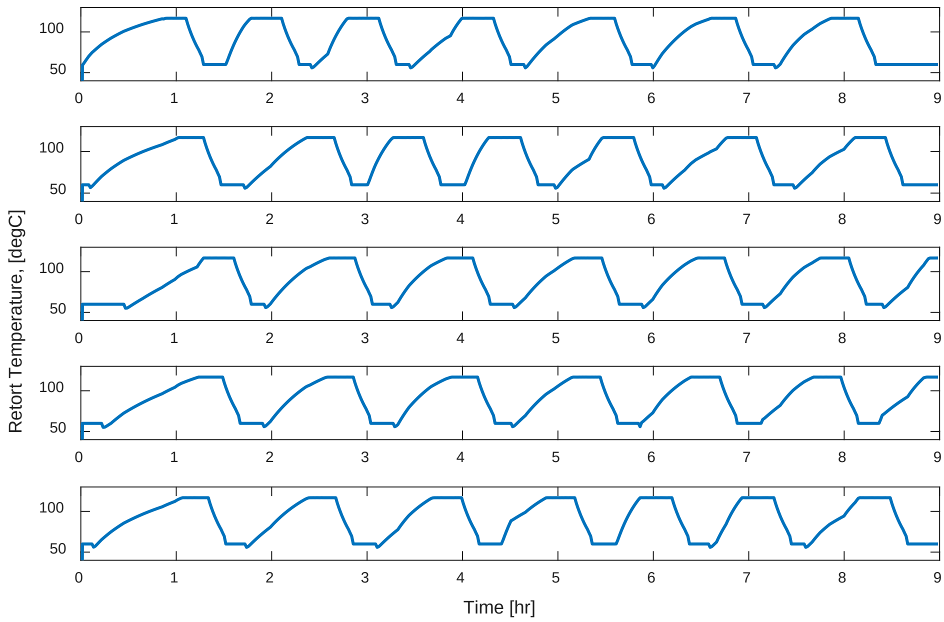


Figure 29: Operation of 5 sterilisers (during an 8-hour shift)

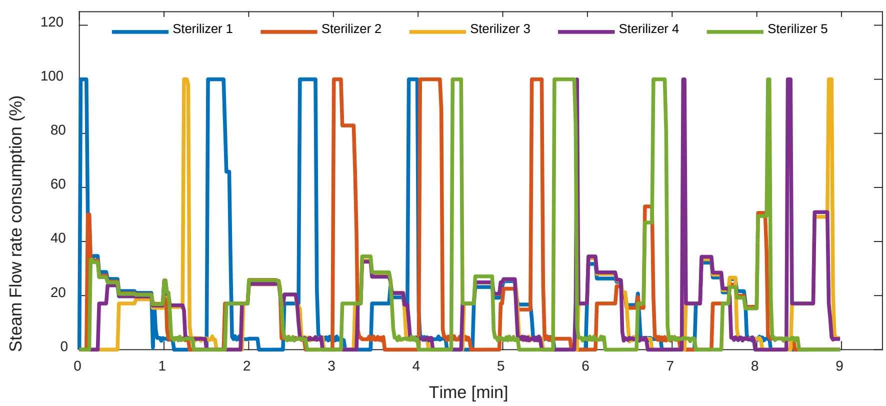


Figure 30: Steam distribution between the sterilisers (during an 8-hour shift)

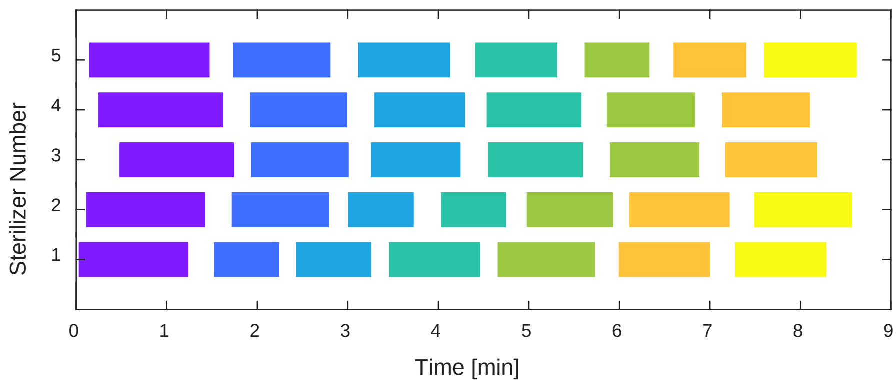


Figure 31: Gantt chart of the batches (during an 8-hour shift)

Overall the coordinated RTO and NMPC framework is able to handle the variability between the batches and can successfully perform the sterilisation of the cans by utilizing the optimum amount of steam under strict constraints related to the quality of the canned food (lethality and colour). The computational requirements of the entire simulation are within an acceptable range (3-5mins).

5.3 Summary and conclusion

Here, the optimal control of the thermal food sterilisation process have been considered. In particular, the optimal way to sterilise a given number of products to be processed utilising the plant resources, e.g. number and size of retorts, steam availability etc. is found.

Initially, a reduced order model (ROM) is developed that captures the behaviour of all related subsystems in just 2 ODEs and five algebraic equations, which has been verified against experimental data from the FRINSA plant. As a result, a compact model of the complex sterilisation process that is suitable for control has been generated. Based on that model, a coordinated RTO and NMPC framework has been developed for the consideration of this problem, whose objectives are the maximisation of production profit, minimisation of operation cost and minimisation of energy consumption. The role of the NMPC is to reach the temperature and lethality limits imposed by the RTO under the given resource considerations. A centralised and a distributed NMPC architecture has been developed, both being able to operate under a wide range of conditions, however the distributed NMPC architecture proved to be computationally superior. The RTO operates in a higher level and is responsible for updating the local NMPCs and optimizing the process operating conditions.

The proposed framework has been successfully evaluated through numerous use cases using real data provided by FRINSA. Overall, optimal solutions were generated that can significantly reduce the amount of steam used and improve the plant productivity within acceptable CPU times.

References

[1] S. Engell, “Feedback control for optimal process operation,” *J. Process Control*, vol. 17, no. 3, pp. 203–219, Mar. 2007.
[2] C. de Prada *et al.*, “Integration of RTO and MPC in the Hydrogen Network of a Petrol Refinery,” *Processes*, vol. 5, no. 1, p. 3, Mar. 2017.
[3] F. Lozano Santamaria and S. Macchietto, “Integration of Optimal Cleaning Scheduling and Control of Heat Exchanger Networks Undergoing Fouling: Model and Formulation,” *Ind. Eng. Chem. Res.*, vol. 57, no. 38, pp. 12842–12860, Sep. 2018.
[4] C. G. Palacín, J. L. Pitarch, C. Jasch, C. A. Méndez, and C. de Prada, “Robust integrated production-maintenance scheduling for an evaporation network,” *Comput. Chem. Eng.*, vol. 110, pp. 140–151, Feb. 2018.
[5] C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski, and M. Fahl, “State-of-the-art review of optimization methods for short-term scheduling of batch processes,” *Comput. Chem. Eng.*, vol. 30, no. 6, pp. 913–946, May 2006.
[6] L. T. Biegler, “Advanced optimization strategies for integrated dynamic process operations,” *Comput. Chem. Eng.*, vol. 114, pp. 3–13, Jun. 2018.
[7] J. Lunze and F. Lamnabhi-Lagarrigue, Eds., *Handbook of hybrid systems control: theory, tools, applications*. Cambridge, UK ; New York: Cambridge University Press, 2009.
[8] C. de Prada, M. Rodriguez, and D. Sarabia, “On-Line Scheduling and Control of a Mixed Continuous−Batch Plant,” *Ind. Eng. Chem. Res.*, vol. 50, no. 9, pp. 5041–5049, May 2011.
[9] S. Engell and I. Harjunkoski, “Optimal operation: Scheduling, advanced control and their integration,” *Comput. Chem. Eng.*, vol. 47, pp. 121–133, Dec. 2012.
[10] R. Mazaeda, S. P. Cristea, and C. de Prada, “Hierarchically coordinated economic MPC plantwide control of mixed continuous-batch units in process industries with application to a beet sugar plant,” *Optim. Control Appl. Methods*, vol. 0, no. 0.
[11] M. Nauta, “D1.1 Efficient construction of efficient models,” p. 66.
[12] C. Vilas Fernández and A. A. Alonso, “Real time optimization of the sterilization process in a canning industry,” 2018.
[13] C. A. Floudas and X. Lin, “Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review,” *Comput. Chem. Eng.*, vol. 28, no. 11, pp. 2109–2129, Oct. 2004.
[14] C. G. Palacín, C. de Prada, and J. L. Pitarch, “Operator support in optimal load-allocation among equivalent equipment,” 2018. [Online]. Available: http://hdl.handle.net/10662/8643.
[15] C. de Prada, R. Mazaeda, and S. P. Cristea, “Receding horizon scheduling of processes with shared resources,” *Comput. Chem. Eng.*, vol. 125, pp. 1–12, Jun. 2019.
[16] E. Balas, *Disjunctive Programming*. Cham: Springer International Publishing, 2018.
[17] C. G. Palacín and C. de Prada, “Optimal Coordination Of Batch Processes with Shared Resources,” *IFAC-Pap.*, vol. 52, no. 1, pp. 826–831, Jan. 2019.
[18] C. G. Palacín, C. A. Méndez, and C. de Prada, “Slots Startup Synchronization with Shared Resources Dependency,” *Chem. Eng. Trans.*, vol. 74, pp. 1321–1326, May 2019.
[19] C. G. Palacín, P. Riquelme, and C. de Prada, “Scheduling óptimo de procesos Batch de duración interdependiente,” presented at the XL Jornadas de Automática, 2019, pp. 560–567.