Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science in Mechanical Engineering

POLITECNICO
MILANO 1863

On-line algorithms for energy efficient control of machine tools: a step towards practical implementation

Supervisor
Prof. Andrea Matta

Co-Supervisor
PhD Nicla Frigerio

Candidate
Claudio Francesco Angelo Cornaggia
898418

Academic Year 2018/2019

Acknowledgements

Foremost, I would like to thank Prof. Andrea Matta and Nicla Frigerio for their continuous support, their positive attitude towards me and the great enthusiasm shown in valuing my ideas and efforts. It has been an honor and a pleasure to share this research experience with your team. It is due to people and professionals like you that our university globally outstands.

I also express my sincere gratitude to my unique family. You have always shown your belief in me and feeling you proud of my career gives me great satisfaction. I will never forget your sacrifices to raise me.

Further, a special thought goes to my girlfriend Ginevra. Thanks for being by my side in these years, everything about you makes me feel better. Your willingness is of great inspiration and your sweetness is rare.

And not least, I should also mention all my friends, from the childhood ones to those met at the university. Thank you for seeing and supporting me. Style is when we stand together.

Abstract

Over the last decade, the benefits of adopting sustainable solutions in manufacturing have gained an increasing global recognition. Specifically, the control of machine states is one of the most promising measure to pursue energy efficient operation. Based on service request, state control aims at minimizing the energy consumed during idle periods by properly switching off/on the machine tool. The selection of the optimal control is not trivial, since manufacturing systems are subject to several source of randomness, difficult to model a priori. However, most of the policies from literature assume the stochastic processes involved in the control problem to be known. Their practical implementation therefore requires a burdensome experimental campaign to fit the random variable distributions.

This work provides an on-line time-based policy able to effectively control the machine state, while learning from a real time collection of part arrivals. Considering that a startup transitory is required to resume operational readiness, an efficient solving algorithm is developed to exactly identify trade-off solutions, which improve machine sustainability while keeping its throughput high. Moreover, since the policy is optimized based on expectations, a measure is presented to mitigate the risk of deteriorating machine performance over the single service request. Finally, a change point detection method is introduced to handle a non-stationary arrival process. In this way, the control can be autonomously adapted to the dynamic behavior of the upstream portion of the system, thus overcoming a critical barrier for practical implementation. The benefits of the proposed algorithms are assessed by means of realistic numerical cases.

Besides the scientific results, this policy can be applied to a wide range of production cases, being tunable according to the user needs, adaptive to the changes of a real manufacturing environment, lightweight and highly autonomous.

Sommario

Nell'ultimo decennio, i benefici derivanti dall'adozione di soluzioni sostenibili nel settore manifatturiero sono stati progressivamente riconosciuti a livello globale. In particolare, una delle soluzioni più promettenti per ridurre il consumo energetico di una macchina utensile è il controllo dei suoi stati. Spegnere e riaccendere opportunamente la macchina sulla base delle richieste di servizio permette infatti di minimizzare l'energia assorbita durante i periodi di inattività. La selezione di un controllo ottimale non è semplice, in quanto i sistemi produttivi sono caratterizzati da diverse fonti di causalità, difficili da modellare a priori. Tuttavia, la maggior parte delle politiche in letteratura considera noti i processi stocastici coinvolti nel problema di controllo. La loro implementazione pratica richiede quindi un'onerosa campagna sperimentale per riconoscere le distribuzioni delle variabili casuali.

Questo lavoro fornisce una politica di controllo online e basata su variabili temporali, in grado di gestire lo stato della macchina imparando dai dati relativi all'arrivo delle parti, acquisiti in tempo reale dal sistema. Dal momento che, una volta spenta la macchina, è necessario un transitorio di accensione per ristabilirne l'operatività, un efficiente algoritmo risolutivo è sviluppato per identificare esattamente delle soluzioni di compromesso, che permettano il miglioramento della sostenibilità senza danneggiare la produttività. Inoltre, dato che la politica è ottimizzata sulla base di valori attesi, una misura è presentata per mitigare il rischio di deteriorare le prestazioni della macchina nella singola richiesta di servizio. Infine, una metodologia di individuazione dei punti di cambio è introdotta per gestire una successione di arrivi non stazionaria. In questo modo, il controllo può essere adattato autonomamente al comportamento dinamico del processo produttivo a monte, superando così un ostacolo critico per l'implementazione pratica. I benefici forniti dagli algoritmi proposti sono valutati per mezzo di realistici casi numerici.

Oltre ai risultati scientifici, questa politica può essere applicata ad una vasta gamma di casi produttivi, in quanto è calibrabile sulle esigenze dell'utilizzatore, adattiva ai cambiamenti di un reale ambiente manifatturiero, leggera dal punto di vista implementativo ed altamente autonoma.

Contents

List of Figures

List of Tables

Nomenclature

Variables

$\Delta E/\Delta e$ Difference in machine energy consumption with respect to the AON policy (random variable/occurrence)

$\hat{g}(\boldsymbol{\tau}|h)$ Expected value of the energy consumed in a cycle by the machine obtained with $\hat{f}^{norm}_{X|H}(x|h)$

A/a Machine interarrival time (random variable/occurrence)

D Paired difference in machine energy consumption

D_h Difference between $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ and $\boldsymbol{\tau}_{\boldsymbol{impl}}$ in terms of objective function $\hat{g}(\boldsymbol{\tau}|h)$

E/e Machine energy consumption in a cycle (random variable/occurrence)

e_i^{path} NPE consumed through path i

H/h Kernel smoothing parameter or bandwidth (random variable/occurrence)

Q/q Waiting time in a cycle (random variable/occurrence)

q_i^{path} Waiting time a part has to experience through path i

s Machine energy state

X/x Machine starvation time (random variable/occurrence)

Functions

$\check{g}_k(\boldsymbol{\tau})$ Sample-based objective function

$\hat{\theta}_k(\boldsymbol{\tau})$ Expected throughput obtained with $\hat{f}_k(x)$

$\hat{f}^{norm}(x,h)$ Starvation times joint distribution function

$\hat{f}_{X|H}^{norm}(x|h)$ Kernel density estimator truncated and normalized over the positive domain

$\hat{f}_k(x)$ Estimated starvation times distribution

$\hat{f}_{X|H,-i}(x|h)$ Leave-one-out estimator of $\hat{f}_{X|H}(x|h)$

$\hat{f}_{X|H}(x|h)$ Kernel density estimator

$\hat{g}_k(\boldsymbol{\tau})$ Expected value of the energy consumed in a cycle by the machine obtained with $\hat{f}_k(x)$

$\phi'(x, \boldsymbol{\tau})$ Energy output function derivative with respect to x

$\Phi(\boldsymbol{\tau})$ Expected NPE consumed in a cycle

$\phi(x, \boldsymbol{\tau})$ Energy output function

$\theta(\boldsymbol{\tau})$ Expected machine throughput

$\tilde{L}(h)$ Cross validated pseudo-likelihood function

$\tilde{L}^{scaled}(h)$ Scaled cross validated pseudo-likelihood function

$C(n)$ Implementation cost function

$f_A(a)$ Machine interarrival time probability density function

$f_E(e|\boldsymbol{\tau})$ Probability density function of E

$f_H(h)$ KDE bandwidth probability density function

$f_Q(q|\boldsymbol{\tau})$ Continuous part of the distribution function of Q

$F_X(x)$ Machine starvation time cumulative distribution function

$f_X(x)$ Machine starvation time probability density function

$F_{\Delta E}(\Delta e|\boldsymbol{\tau})$ Cumulative distribution function of ΔE

$f_{\Delta E}(\Delta e|\boldsymbol{\tau})$ Continuous part of the distribution function of ΔE

$F_Q(q|\boldsymbol{\tau})$ Cumulative distribution function of Q

$g(\boldsymbol{\tau})$ Expected value of the energy consumed in a cycle by the machine

$H(\boldsymbol{\tau})$ Expected part waiting time in a cycle

$HR(x)$ Hazard rate of X

$I(\cdot)$ Indicator function

$ISE(h)$ Integrated square error

$K(\cdot)$ Kernel function

$L(h)$ Pseudo-likelihood function

$MISE(h)$ Mean integrated square error

$N(x-m,v)$ Normal distribution with mean m and standard deviation v

Parameters

α Significance level for BC_a confidence interval construction

α_h Bandwidth implementation policy parameter (probability)

α_{CPD} Smoothness controller for the change point score

Δe_{target} Target difference in machine energy consumption with respect to the AON policy

Δn Observation interval between two algorithm iterations

δ_e Target probability (energy risk constraint)

δ_q Target probability (waiting time risk constraint)

ϵ Maximum admissible throughput reduction with respect to the AON policy

η_{CPD} Threshold for the change point score

ι Constant of the negative exponential function

B Number of bootstrap resamples

C_0 Implementation cost

h^* Optimal bandwidth

h_k^* Optimal bandwidth at iteration k

H_{max} Upper bound for the expected waiting time

k_{CPD} Size of the sliding window used to generate a single $\boldsymbol{X}(i)$

n^* Observation interval required by the Implementation cost function to reach approximately the 5% of its initial value

n_{CPD} Number of retrospective subsequences in each set $\boldsymbol{\mathcal{X}}$

q_{target} Target waiting time in a cycle

ST Safety threshold

t_a Mean starvation time

t_p Machine processing time

t_{su} Machine startup duration

w_q Part holding power

w_s Machine power consumption in state s

Sets

$\check{\boldsymbol{\tau}}_{\boldsymbol{k}}$ Estimated control parameters with $\check{g}_k(\boldsymbol{\tau})$

$\check{\boldsymbol{x}}(\boldsymbol{\tau}) = \{\check{x}_1(\boldsymbol{\tau}), \ldots, \check{x}_n(\boldsymbol{\tau})\}$ Machine starvation times obtained with the recursive function

$\hat{\boldsymbol{\tau}}_{\boldsymbol{k}} = \{\hat{\tau}_{off,k}, \hat{\tau}_{on,k}\}$ Estimated control parameters

$\boldsymbol{\tau}_{\boldsymbol{b}}^*$ Constrained minimum of $g(\boldsymbol{\tau})$ along the throughput boundary

$\boldsymbol{\tau}^* = \{\tau_{off}^*, \tau_{on}^*\}$ Off-line optimal control parameters

$\boldsymbol{\tau} = \{\tau_{off}, \tau_{on}\}$ Switching policy control parameters

$\boldsymbol{\mathcal{X}}(i)$ Set of n_{CPD} retrospective subsequence samples $\boldsymbol{X}(i)$ starting from observation x_i

$\boldsymbol{a} = \{a_1, \ldots, a_n\}$ Collected interarrival times data

$\boldsymbol{D_0}$ Set of non-zero paired difference in machine energy consumption

$\boldsymbol{D}$ Set of paired difference in machine energy consumption

$\boldsymbol{X}(i)$ Subsequence of observations starting from x_i with length k_{CPD}

$\boldsymbol{x} = \{x_1, \ldots, x_n\}$ Collected starvation times data

$\boldsymbol{\tau_{impl}} = \{\tau_{off,impl}, \tau_{on,impl}\}$ Implemented control parameters

$\mathcal{D}_{con}$ Set of candidates solutions for the constrained optimization problem

$\mathcal{D}_{unc}$ Set of candidates solutions for the unconstrained optimization problem

ΔE_b Set of points which satisfy the energy risk constraint at equality

$\mathcal{T}_{off}$ Set of the local minima of $g(\{\tau_{off}, \cdot\})$

$\mathcal{T}_{on}$ Set of the local minima of $g(\{\cdot, \tau_{on}\})$

Q_b Set of points which satisfy the waiting time risk constraint at equality

TH_b Throughput boundary

Other Symbols

$\bar{C}_i$ Average probability of detecting the ith change point

$\bar{F}_a$ Average number of false alarms

$\check{q}_i(\boldsymbol{\tau})$ Generic waiting time in a cycle obtained with the recursive function

$\check{x}_i(\boldsymbol{\tau})$ Generic starvation time obtained with the recursive function

$\Delta\theta(\boldsymbol{\tau})$ Expected throughput reduction percentage with respect to the AON condition

Δe_i Generic observation of variable ΔE

Δe_{max} Maximum possible occurrence of ΔE

Δe_{min} Minimum possible occurrence of ΔE

$\Delta g(\boldsymbol{\tau})$ Expected energy saving percentage with respect to the AON condition

γ Order of magnitude for the maximum of $\tilde{L}(h)$

$\hat{\theta}_{AON,k}$ Maximum achievable machine throughput (AON policy) with $\hat{f}_k(x)$

Λ_i Manifestation of path i over the feasible ones

μ Paired difference mean

π_e Proportion of observations in which Δe_{target} is exceeded

π_q Proportion of observations in which q_{target} is exceeded

τ_{off}^{LB}	Lower bound for the switch off parameter (theorem 4.3)
θ_{AON}	Machine throughput with AON policy
$\tilde{\tau}_{off,i}$	Local minimum of $g(\{\tau_{off}, \cdot\})$
$\tilde{\tau}_{on,j}$	Local minimum of $g(\{\cdot, \tau_{on}\})$
ϖ	Limit sojourn duration in the standby state (theorem 4.2)
ξ	Minimum sojourn duration in the standby state
g_{AON}	Expected value of the energy consumed in a cycle by the machine with AON policy
k	Algorithm iteration number
n	Observation number and sample size of vector $\boldsymbol{x}$
q_i	Generic waiting time observation
x_i	Generic starvation time observation

Chapter 1

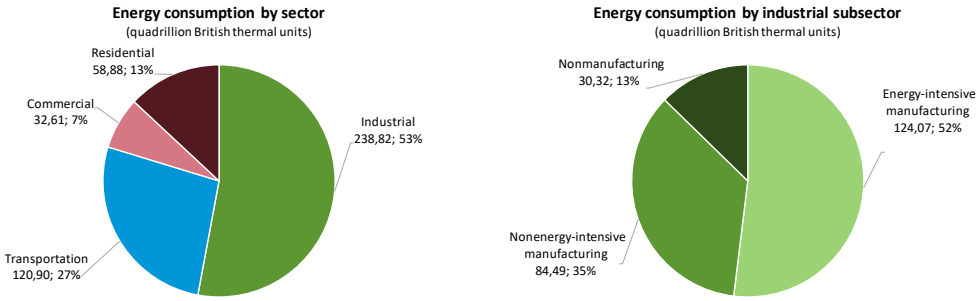
Introduction

In the last ten years, alongside the classical manufacturing system research topics concerning productivity improvement and quality enhancement, *energy efficiency* has become a significant field of interest. The need for systems with *"eco-green" functionalities* is driven both by the latest political regulations, aimed at containing the environmental impact of industrial processes, and by the competitive advantage achievable through cost reduction. Nevertheless, some critical issues limit the widespread implementation of sustainable solutions in manufacturing industries. New models and methods must therefore be developed to overcome these barriers.

1.1 Background

According to the U.S. Energy Information Administration [1], the amount of energy absorbed by the industrial sector in 2018 accounts for more than 50% of the world energy consumption (cf. figure 1.1). In particular, energy-intensive manufacturing (food, pulp and paper, basic chemicals, refining, iron and steel, non-ferrous metals and non-metallic minerals) is the largest component in the sector, being responsible for the 52% of the total industrial absorptions. This subsector is followed by nonenergy-intensive manufacturing (metal-based durables, other chemicals and other manufacturing) and non-manufacturing (agriculture, mining and construction) with a share of 35% and 13%, respectively.

Looking to the future, the long-term projections provided in the *International Energy Outlook 2019* [1] show that the gross output from industrial activities (i.e. a measure of sales across industrial sectors) is supposed to double between 2018 and 2050, resulting in an increase of industrial energy consumption (cf. figure 1.2). Nevertheless, the industrial gross output



(a) Worldwide energy demand. (b) Industrial sector energy demand.

Figure 1.1: Energy consumption data from the *International Energy Outlook 2019*.

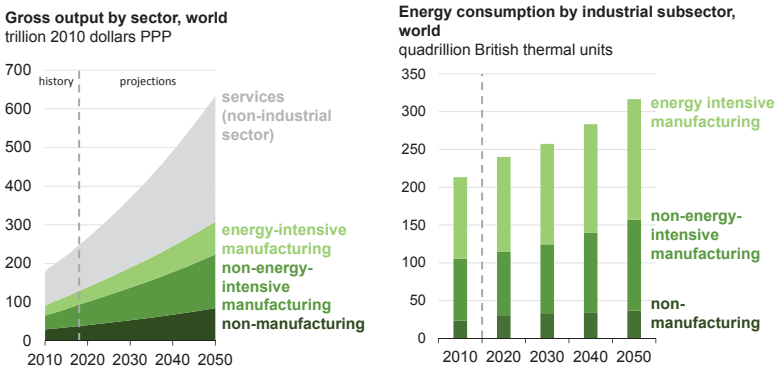


Figure 1.2: Gross output and industrial energy consumption (2010-2050) [1].

growth outpaces the energy consumption increase, due to the higher share of nonenergy-intensive manufacturing and the enhancements concerning industrial *energy efficiency*.

These improvements are especially achieved in those countries that are part of the Organisation for Economic Cooperation and Development (OECD). In fact, while the difference between the industrial gross output growths in OECD and non-OECD countries is supposed to progressively diminish, the latter are clearly characterized by a stronger increase of industrial energy consumption (cf. figure 1.3). In detail, at the end of the analyzed period, China remains the world's largest single industrial energy consumer, while India's consumptions nearly triple (cf. figure 1.4). This mainly results from the continuous shift of energy-intensive manufacturing towards non-OECD Asia.

Finally, the price of electrical energy is progressively increasing [2], becoming a significant factor for the competitiveness of a company.

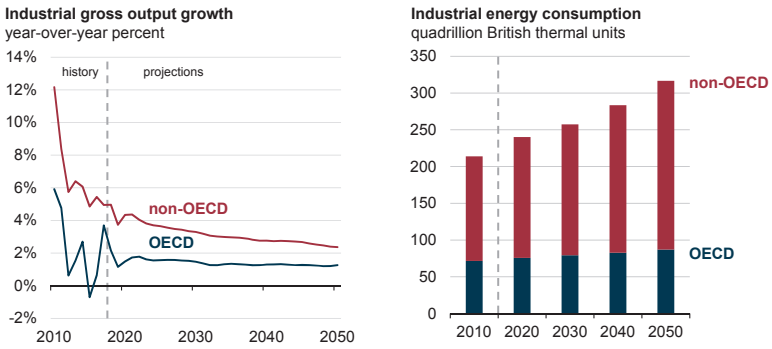


Figure 1.3: Gross output growth and industrial energy consumption in non-OECD and OECD countries (2010-2050) [1].

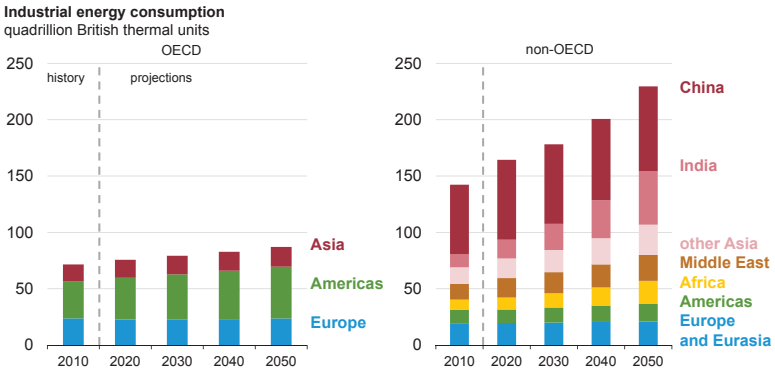


Figure 1.4: Industrial energy consumption by country (2010-2050) [1].

1.2 Energy Efficiency in Manufacturing

In the manufacturing field, *energy efficiency* is defined as the relationship between the results achieved and the resources used, where the energy input is the only resource considered [3]. Going into detail, *energy efficiency* can be addressed at different hierarchical levels of the automation pyramid [4]:

- Global supply chain
- Facility / Factory
- Production line
- Machine tool

In the followings, the machine tool level is analyzed, which accounts for approximately 50% of the total amount of electricity utilized in manufacturing [5]. The above-mentioned absorptions are however significantly greater

than the theoretical requirements for chip removal since, neglecting losses and process inefficiencies, most of the energy is employed for creating stable process conditions and for peripheral functions. In detail, this surplus must be attributed to the machine auxiliary units (e.g. chiller, hydraulic, pneumatic, cutting coolant and tool change units) which, without being directly involved in the chip removal process, support the spindle and axes units operation. When the machine tool is visiting a not-operative state (idle or failure), some auxiliary units may still require a significant amount of power, resulting in non value adding absorptions.

In light of these considerations, the energy consumption of a machine tool can be broken down into two main constituents:

- **Non-Processing Energy (NPE)** – It indicates whenever energy is used during not-operative states, i.e. consumed without adding value to the part. It generally results from integrating a constant power coefficient overtime.

- **Processing Energy (PE)** – It accounts for the energy consumed by the machine tool while it is working on parts, commonly characterized by:

  - A *constant* energy, which derives from the *base load* power demanded for the processing time.
  - A *variable* energy, which depends on the cutting parameters and the toolpath strategy.

Different technical and organizational strategies are focused on achieving *energy efficiency* in the manufacturing industries. They either consider the whole machine or its single components:

- **Machine Design** – It helps to minimize the power demands of components, reducing energy losses (e.g. machine lightening, friction reduction and improvement of energy transformations). Many design enhancements have been proposed and investigated, such as the use of more efficient motor drives, handling, clamping, hydraulic and pneumatic systems.

- **Reuse and Recovery Systems** – They include the implementation of any measure to prevent energy from being released as a loss and rather be used in other forms of desired work. Most energy recovery systems exchange thermal or kinetic energy, for example the Kinetic Energy Recovery System (KERS).

- **Process Parameters Optimization** – It is focused on identifying the process parameters (e.g. spindle rotating speed or cutting speed) which allow to minimize the PE. Problems in this category are clearly multi-objective, since the trade-off between cycle time and power demand has to be explicitly addressed.

- **Machine State Control** – It aims to reduce the NPE by properly switching off some auxiliary units (together or independently) when production is not required. Since service is interrupted when the machine is visiting a low power consumption state, the machine tool undergoes a startup transitory to resume operational readiness as soon as one or a pre-determined number of parts need to be processed. Nevertheless, the time required to restore service may significantly affect the throughput. The strategies covered by this field can be further classified according to an increasing detail level in the production planning and control (PP&C) hierarchy. In particular:

  - **Energy-Efficient Production Scheduling (EES)**, which refers to scheduling the production activities with the objective to improve *energy efficiency*. In detail, production plans are defined in order to allocate jobs to a specific machine minimizing the number of unproductive periods. By gathering together the time intervals in which the machine is idle, energy saving potential arises: it is therefore possible to switch the machine tool into a low power consumption state respecting constraints on tardiness or total makespan.
  - **Energy-Efficient State Control (EEC)**, which refers to managing machine state transitions in order to reduce or minimize its NPE consumptions. It focuses on the control level and provides policies to be applied during production progress, without a deterministic knowledge of the next part arrival. Since machine vacations in a low power consumption state are not planned at priori, the management of the trade-off between energy saving and throughput reduction is more challenging.

1.3 Off-Line and On-Line Problems

Manufacturing systems are commonly characterized by the presence of stochastic processes (e.g. arrivals, failures, startup durations) and noisy information. Nevertheless, production decisions must be taken even in the

presence of uncertainty over the underlying sources of randomness, models and objective functions. On the basis of information availability (i.e. knowledge level of random variable distributions), manufacturing problems can be therefore classified in two categories [6, 7]:

- **Off-line problems** – The stochastic processes involved in the system are assumed to be known when addressing the problem, therefore optimization is performed once for all and the final decision is implemented on the field. In reality, an a priori experimental campaign is required to collect the large data sets needed to fit the random variable distributions.
- **On-line problems** – The stochastic processes involved in the system are assumed to be unknown during problem resolution, therefore learning and optimization are executed simultaneously and iteratively. In particular, the real time data collected from the system are progressively employed to fit the random variable distributions. Even though this category is a more faithful representation of real industrial cases, the issue of estimates uncertainty has to be managed, especially when few information are available.

1.4 Research Focus

This work is aimed at improving and extending the available models for *energy efficiency* in manufacturing. In detail, the focus is on the energy-efficient state control (EEC) of resources while they are not working on parts (i.e. the part flow is interrupted). Specifically, machine tools executing machining operations are considered, since they are one of the most complex and energy demanding production equipment.

The *scientific relevance* of this work results from addressing the control problem in an innovative on-line framework: solutions are provided in real time, while collecting data from the shopfloor. Moreover, particular attention is devoted to the management of the trade-off between energy saving and throughput reduction. Also, since the control is based on a data-driven estimate, the risk of incurring in not advantageous solutions is included in problem formulation. Switching towards *industrial relevance*, the proposed EEC policy can be implemented on a wide range of machining centers with *"eco-green" functionalities*, being flexible and highly autonomous.

To guarantee a better understanding, the detailed thesis contribution as well as the outline of the thesis structure are given in section 2.7.

Chapter 2

Literature Review on Energy Efficient State Control

The scope of this chapter is to provide a detailed literature review on the energy efficient state control (EEC) topic. In particular, more than 50 papers are collected and classified according to several criteria, which correspond to the axes of table 2.1. This classification is aimed at providing the knowledge base over which the thesis contribution will be grafted.

This chapter is structured in seven sections, which extensively illustrate the main classification criteria. After a brief introduction, section 2.2 addresses the machine energy states contemplated in literature. In section 2.3, the characteristics of the two different levels of analysis (i.e. single machine and production system) are described, focusing especially on their sources of stochasticity. In section 2.4, the peculiar model types used to abstract the controlled systems are detailed. In section 2.5, the control methods aimed at achieving energy efficiency are introduced. In section 2.6, the scope of the collected papers is analyzed (i.e. optimization or performance evaluation). Finally, section 2.7 closes the review illustrating how the thesis contribution fits in the current state of the art and outlying the structure of the work.

2.1 Introduction

Under the consideration of massive energy consumption of machine tools, EEC of machine states addresses the problem of energy efficiency at the lowest level of the production planning and control hierarchy. In particular, EEC literature provides policies to efficiently reduce the non-processing energy (NPE) consumed by machine tools during production progress, without knowing the occurrence of next part arrival. The work proposed by Mouzon et al. [8] can be

Table 2.1: EEC papers classification.

Reference				Can et al., 2019 [4]	Huang et al., 2019 [9]	Li et al., 2019 [51]	Marzano et al., 2019 [43]	Squeo et al., 2019 [27]	Wang et al., 2019a [40]	Wang et al., 2019b [58]	Zhang and Jiang, 2019 [57]	Zou et al., 2019 [10]	Duque et al., 2018 [11]	Fei et al., 2018 [12]	Huang et al., 2018 [59]	Li et al., 2018 [48]	Maccio and Down, 2018a [30]	Maccio and Down, 2018b [31]	Renna, 2018 [54]	Wang et al., 2018a [55]	Wang et al., 2018b [28]	Frigerio et al., 2017a [53]	Frigerio et al., 2017b [37]	Wang et al., 2017 [13]	Su et al., 2017 [14]	Zou et al., 2017 [15]	Brundage et al., 2016 [16]	Cronrath et al., 2016 [62]	Frigerio and Matta, 2016 [36]	Frigerio et al., 2016 [53]	Jia et al., 2016 [41]	Maccio and Down, 2016 [32]	Su et al., 2016a [17]	Su et al., 2016b [18]	Wang et al., 2016 [56]	Wei and Wang, 2016 [61]	Zou et al., 2016 [19]	Frigerio and Matta, 2015a [50]	Frigerio and Matta, 2015b [44]	Frigerio et al., 2015 [35]	Jia et al., 2015 [42]	Maccio and Down, 2015 [33]	Sun et al., 2015 [20]	Brundage et al., 2014a [21]	Brundage et al., 2014b [22]	Frigerio and Matta, 2014 [38]	Chang et al., 2013 [47]	Frigerio and Matta, 2013 [45]	Guo et al., 2013 [39]	Li and Sun, 2013 [29]	Maccio and Down, 2013 [34]	Sun and Li, 2013 [23]	Niu et al., 2012 [24]	Prabhu et al., 2012 [46]	Prabhu and Taisch., 2012 [25]	Wang et al., 2012 [60]	Chen et al., 2011 [52]	Li et al., 2011 [26]	Mouzon et al., 2007 [8]
	Machine Energy States	Standby (Out-of-Service)	One	x	x	x	x		x	x		x	x	x	x		x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x		x	x	x	x
	Machine Energy States	Standby (Out-of-Service)	Several					x			x					x					x														x	x														x					x				
	Machine Energy States	Idle (On-Service)		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x			x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x
	Machine Energy States	Startup					x	x			x		x	x			x	x	x	x	x	x	x				x	x	x	x	x	x			x			x	x	x	x	x	x	x		x		x	x		x	x	x		x	x	x	x	x
	Machine Energy States	Working		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	Machine Energy States	Closedown																		x								x			x				x						x								x				x		x	x		x	x
	Machine Energy States	Breakdown (Failed)		x	x	x			x	x		x	x	x	x	x				x	x			x		x	x				x				x		x				x			x	x		x			x		x					x		
Level of Analysis	Single Machine			x			x	x					x										x						x							x			x	x							x		x	x		x		x		x		x		x	x
Level of Analysis	Production System				x	x			x	x	x	x		x	x	x	x	x	x	x	x	x		x	x	x	x	x		x	x	x	x	x	x		x	x			x	x	x	x	x		x			x		x		x	x		x		
Level of Analysis	Part Flow		Discrete			x			x	x	x			x	x	x			x	x	x	x		x	x			x		x	x		x	x	x			x			x		x							x		x			x		x		
Level of Analysis	Part Flow		Continuous		x							x														x	x										x							x	x		x												
Level of Analysis	Line Balancing		Balanced													x			x			x			x					x	x		x	x				x			x									x		x					x		
Level of Analysis	Line Balancing		Unbalanced		x	x			x	x		x		x	x				x	x	x	x		x		x	x			x					x		x						x	x	x		x								x				
Level of Analysis	Number of Controlled Machines/Stations		One						x	x				x		x					x			x							x				x		x	x			x																x		
Level of Analysis	Number of Controlled Machines/Stations		Some			x			x	x		x		x	x		x	x		x	x	x		x							x	x					x	x						x	x							x							
Level of Analysis	Number of Controlled Machines/Stations		All		x						x								x						x	x	x	x		x			x	x				x				x	x				x			x				x	x				
	Buffer Capacity		Finite		x	x			x	x	x	x	x	x	x	x			x	x	x	x		x	x	x	x		x	x	x		x	x	x	x	x	x		x	x		x	x	x		x			x		x					x		
	Buffer Capacity		Infinite														x	x					x									x										x							x		x		x	x					
	Stochastic Processes	Reliability Model	Bernoulli			x										x			x			x								x	x							x			x																x		
	Stochastic Processes	Reliability Model	MTBF and MTTR: Exponential						x	x		x	x	x	x											x	x																	x	x		x					x							
	Stochastic Processes	Reliability Model	MTBF and MTTR: Geometric																	x	x			x											x		x																						
	Stochastic Processes	Reliability Model	MTBF: Weibull, MTTR: Exponential		x																																													x									
	Stochastic Processes	Processing Time															x	x					x						x			x				x				x		x							x		x		x	x	x	x			
	Stochastic Processes	Interarrivals					x	x					x		x		x	x	x				x					x	x			x				x		x	x	x		x				x		x	x		x		x	x	x	x			x
	Stochastic Processes	Startup duration															x	x					x						x			x								x		x				x			x		x								
	Stochastic Processes	Closedown duration																																															x				x						
	Model Type		Analytical		x	x	x	x		x		x			x	x	x	x		x			x		x	x	x		x		3	x	x	x	x	x	x		x	x	x	x		x	x	x	x	x	x	x	x	x	x	x			x		x
	Model Type		Simulation						x		x	x	x	x					x		x	x		x				x	x	x								x					x												x	x		x	
Control Method	Control Policies & Rules			x			x	x	x				x	x			x	x	x		x	x	x	x	x				x	x	x	x	x	x		x		x	x	x	x	x				x		x	x		x		x	x	x	x	x	x	x
Control Method	Control Rule		Time	x			x	x																					x									x	x							x		x	x		x			x	x	x		x	x
Control Method	Control Rule		Buffer Level						x				x	x			x	x	x		x	x	x	x					x	x	x	x				x		x		x	x	x							x		x		x				x		
Control Method	Control Rule		Other						x				x						x						x								x	x																							x		
Control Method	Buffer Level Information		Upstream						x				x	x			x	x			x	x	x	x					x	x	x	x						x		x	x	x							x		x		x				x		
Control Method	Buffer Level Information		Downstream						x					x					x		x			x						x	x					x					x																		
Control Method	Real Time Decision Making		Energy Opportunity Window		x					x		x			x	x				x						x	x	x							x		x							x	x		x					x							
Control Method	Real Time Decision Making		Other			x					x																																x							x									
Scope	Optimization				x	x	x	x				x										x	x		x	x	x		x	x			x	x		x		x	x	x						x		x	x	x	x						x		x
Scope	Objective Function	Cost/Payoff	Energy		x	x						x											x			x	x									x				x											x								x
Scope	Objective Function	Cost/Payoff	Completion Time																																																									x
Scope	Objective Function	Cost/Payoff	Status Switching		x							x														x																																	
Scope	Objective Function	Cost/Payoff	Production Loss / Reward		x	x						x											x			x	x									x				x																			
Scope	Objective Function	Cost/Payoff	Maintenance		x																																																						
Scope	Objective Function	Cost/Payoff	Control (Fixed)			x																																																					
Scope	Objective Function	Cost/Payoff	Resilience									x																																															
Scope	Objective Function	Cost/Payoff	Collaboration Request																							x																																	
Scope	Objective Function	Cost/Payoff	Inventory Holding																				x													x				x																			
Scope	Objective Function	Cost/Payoff	Part Rejection																				x																	x																			
Scope	Objective Function	Cost/Payoff	Waiting time																																															x								
Scope	Objective Function	Energy Consumed Per Part / Cycle					x	x														x			x				x	x			x	x				x	x							x		x	x								x		
Scope	Objective Function	Energy Consumption to Final Production Horizon																																																x									
Scope	Constraints on KPI		Throughput				x	x														x			x		x		x	x			x	x				x	x											x							x		
Scope	Constraints on KPI		Workforce																														x																										
Scope	Constraints on KPI		Number of Switch-Off Commands																																				x																				
Scope	Constraints on KPI		Mean (Expected) Waiting Time																																													x										
Scope	Optimization Methodology	Analytical	Exact (Closed-Form)																						x								x						x									x	x		x								
Scope	Optimization Methodology	Analytical	Numerical				x	x																	x				x				x	x					x							x		x	x										x
Scope	Optimization Methodology	Heuristic			x	x						x										x				x	x		x	x								x																					x
Scope	Optimization Methodology	Dynamic Programming																					x													x				x										x									
Scope	Optimization Occurence		Once (Off-Line)					x														x	x		x				x	x			x	x		x		x	x	x						x		x	x		x						x		x
Scope	Optimization Occurence		Repeated (Real Time)		x	x	x					x														x	x																							x									
Scope	Performance Evaluation			x			x		x	x	x		x	x	x	x	x	x	x	x	x			x				x	x		x	x			x		x				x	x	x	x	x		x		x		x	x	x	x	x	x	x	x	x

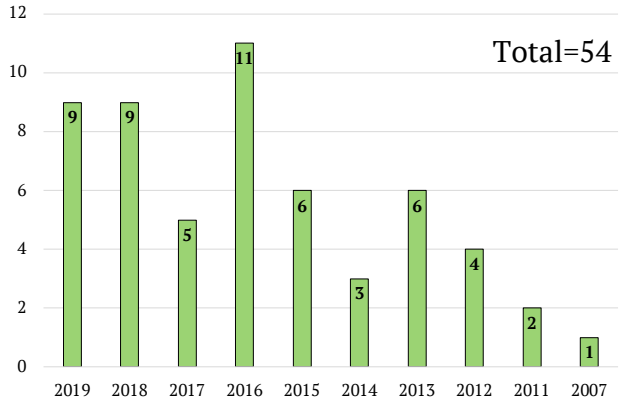


Figure 2.1: Number of EEC papers published for each year.

definitely considered the forerunner of this stream of research. In detail, they develop several dispatching rules to effectively decrease energy consumptions of especially underutilized manufacturing equipment. In addition, considering a non-stationary arrival process of parts, they also exploit artificial neural network algorithms to predict the next arrival in the system.

In more than a decade, this research field extended in terms of contributions, level of analysis, scope and modeling accuracy. Without going into the details of the different classification criteria, figure 2.1 shows the amount of works yearly published.

2.2 Machine Energy States

Under state control, a machine tool visits several states during a production run, which differ in terms of power request. Since average models are generally employed in EEC literature, the energy consumed by the machine in a given state results from the product between the sojourn time in that state and a constant power request. A comprehensive description of the considered machine states is herewith provided, even though some of them are not necessarily employed in each paper:

- **Standby (Out-of-service)** – Some machine modules are deactivated in order to reduce power consumptions. Since only emergency services are active, the machine cannot process a part, being in a "sleeping mode". Some papers [8–26] characterize the standby state with null power absorptions, assuming that the machine tool can be completely switched off.

By accurately establishing which machine modules should be deactivated, it is even possible to generate several standby states, each with different power consumptions and startup times. For example, Squeo et al. [27] model several sleeping states by individually controlling single machine components. On the contrary, Wang et al. [28] consider two standby states, i.e. light sleep and deep sleep, whose power consumptions respectively account for the 50% and 30% of the working power. Finally, Li and Su [29] introduce several energy hibernation states for each machine.

- **Idle (On-Service)** – The machine is ready to process a part with all its modules activated, therefore the power consumption is higher than the standby one. In this state, the machine is in blocking or starvation conditions.

- **Startup** – It accounts for the transitory the machine has to experience to resume its operational readiness. During this state, all the procedures required to make the machine suitable for processing are executed, so that quality and tolerance requirements can be met. The power consumption is generally greater compared to other not-productive states.

  In a real manufacturing environment the startup time is stochastic, since it is affected by the system state. Nevertheless, it is generally considered constant in EEC literature, even though some exceptions are present. For example Maccio and Down [30–34] and Frigerio et al. [35–37] model the startup time as an exponentially distributed random variable, in order refer to the queuing theory framework and derive analytical formulas. In addition, Frigerio and Matta [38] consider the startup time as explicitly dependent on the machine sojourn time in the standby state, defining several startup functions.

  Generally, the startup state is uninterruptible. In fact, once a machine enters this state, no additional transitions are allowed until its operational readiness is resumed. Nevertheless, Maccio and Down [33] consider systems in which a machine can be switched off while warming up.

- **Busy** – The machine is processing a part, requiring a greater amount of power with respect to the idle state. It also includes all the non-cutting states experienced while the part is loaded on the machine.

- **Closedown** – It accounts for the transitory the machine has to undergo to switch from the idle to the standby state. When considered, it is

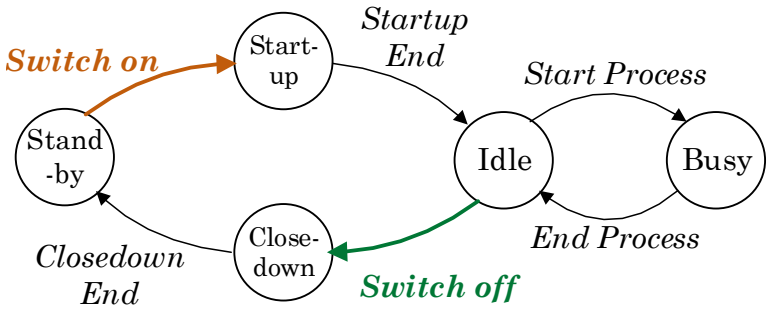


Figure 2.2: Machine state model under a switching off/on control policy.

generally treated as constant and uninterruptible, even though some exceptions are present. For example, Niu et al. [24] and Guo et al. [39] consider it as generally distributed and, in addition, the latter model it as interruptible.

- **Breakdown (Failed)** – The machine has experienced a failure and needs a repair to resume operations. The power consumption in this state can be null [40], equal to the power consumption in the idle state [16] or peculiar of this state [41, 42].

The general machine behavior under a state control is shown in figure 2.2, which depicts the aforementioned machine states and the transitions triggered either by uncontrollable events or by the implemented control policy. The common behavior of a not controlled machine tool provides for the switching between idle and busy states, depending on the execution of part processing. If a state control is applied, a switch off command can be issued when the machine is idle to trigger it into the standby state. Later, service can be resumed with the switch on command. Switch off/on transitions may require a certain amount of time, modeled by the sojourn in the closedown and startup transitory states.

2.3 Level of Analysis and Stochasticity Sources

In literature, EEC is addressed at two distinct levels of analysis:

- **Single Machine**
- **Production System**

Even though the early papers are focused on stand-alone machines, the most recent works concentrate on production systems (cf. figure 2.3). In fact,

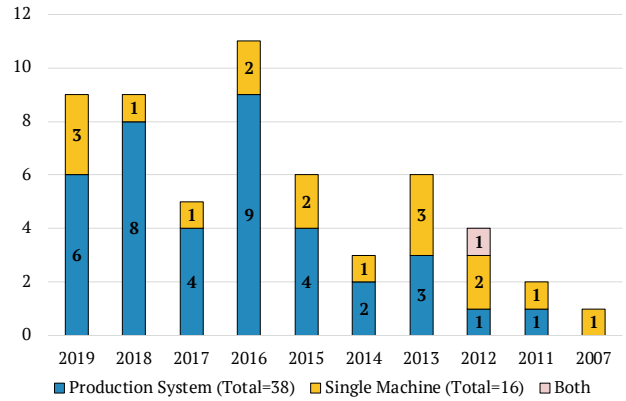


Figure 2.3: EEC papers classification according to the level of analysis.

examining the problem at a higher hierarchical level allows to achieve a greater energy efficiency.

2.3.1 Single Machine

In EEC literature, several papers address *stand alone machines*, without focusing on their interactions with the shop floor. In particular, a widespread assumption is that a *single machine* is never blocked, since parts can be always released to an infinite downstream buffer. Therefore, energy saving potentials arise from machine starvation, which is alternatively modeled in two ways:

- By considering stochastic arrivals and an input buffer in which parts accumulate. This buffer can either be finite [11, 36, 37] or infinite [34, 35, 39].

- By neglecting the presence of an input buffer and directly assuming a stochastic distribution for starvation times. Examples are provided by [27, 43–45].

Except for [11], failures are not explicitly modeled when dealing with *stand alone machine tools*. Therefore, in addition to the arrival process, stochasticity can be introduced in the system by assuming random processing times (which might also account for failures) and startup/closedown durations.

When the single machine is modeled as a $M/M/1$ or $M/G/1$ queue, both interarrivals and processing times are simultaneously stochastic. This is the case of [24, 34–37, 45, 46].

2.3.2 Production System

Due to their great industrial relevance, *production lines*, i.e. series of machines interspersed by finite buffers, are mainly analyzed in literature. Their control for energy savings purposes is particularly complex, since it is tough to assess how EEC policies affect system performance.

A *production line* is generally modeled considering a discrete flow of parts, but some researchers adopt continuous flow models as well [9, 10, 15, 16, 19, 21, 22, 47]. This choice allows to describe production dynamics by integral or differential equations.

Even though the majority of papers is focused on serial *production lines*, whose stations are characterized by a single machine tool, some interesting exceptions can be found. For example, Li et al. [48] consider a *production line* in which each station is composed by several machines connected in parallel. On the contrary, Wang et al. [28] introduce assembly modules with two upstream buffers and disassembly modules with two downstream buffers. Finally, Sun et al. [20] assume that a station could include some parallel substations or a series of manufacturing machines.

Depending on the scope of the analysis, the number of simultaneously controlled machines can range from one to all. Nevertheless, the risk of incurring in a deterioration of system performance increases with the number of controlled machines. Indeed, switching off a machine may respectively cause blocking and starvation to the upstream and downstream machines. According to Frigerio et al. [49], the interaction between two sequential machines might even cause system deadlocks, such that service cannot be resumed. When dealing with *unbalanced lines*, it is suggested to control all the machines except for the bottleneck. In fact, if the operational readiness of the bottleneck machine is not resumed in time, the line experiences a permanent production loss. On the contrary, controlling all the machines turns out to be successful for *balanced lines*.

In EEC literature, it is assumed that each machine of a *production line* is characterized by a deterministic processing time. This hypothesis is always reasonable for automatic systems. Except for [50] and [25], it is also supposed that the first machine is never starved and the last one is never blocked. Stochasticity is therefore introduced by means of unreliable machines, since the arrival process is generally not considered in this type of systems. Two reliability models are employed:

- **Bernoulli** – It is suitable for systems in which machine downtime is comparable to its cycle time. During each cycle a machine has a probability p_i to be up and $1 - p_i$ to be down, where p_i is referred as the efficiency of machine m_i. Examples are provided by [41, 42, 48, 51, 52].

In addition, Frigerio and Matta [49, 50, 53] and Renna [54] consider a Bernoulli reliability model with a machine downtime different from the cycle time.

- **MTBF and MTTR** – Failure metrics can be described by statistical distributions. While the time to repair is generally modeled by an exponential distribution, the time between failures can either follow exponential [10–12, 15, 16, 21, 22, 40] or Weibull distributions [9, 29]. When the time axis is slotted in intervals of fixed duration, a geometric reliability model is generally employed [13, 19, 28, 55, 56]. Moreover, Huang et al. [9] and Zou et al. [10] and also assume that system reliabilities slowly vary with time.

In addition to *production lines*, other system configuration are analyzed in EEC literature. In particular, Maccio and Down [30–33] focus on energy aware multiple servers queuing systems. Even though their research field is targeted on sever farms and data-centres, the employed model ($M/M/m$ queue) can be also transferred to manufacturing systems. Their scope is to evaluate system performance and derive some structural properties of the optimal control policies.

Finally, Zhang and Jiang [57] analyze a discrete manufacturing shop floor, realizing an energy-efficient control of different machine tools by exploiting the real time RFID data of jobs.

2.4 Model Type

In EEC papers, mathematical models are built to represent and study the effect of state control in terms of logical and quantitative relationships. If the model is simple enough, it is possible to directly work with its constituents to get analytical solutions and formulas (e.g. queuing theory). However, many production systems are highly complex, so that their dynamics cannot be expressed in analytical terms. These systems must therefore be analyzed by means of dynamic simulation models. In the followings some examples are reported:

- **Analytical Models** – They are always employed when the considered system is approximated to refer to the queuing theory framework [24, 30–34, 46] and when dealing with Bernoulli serial lines [14, 17, 18, 41, 42]. Frigerio et al. [36, 43–45] make use of a stochastic model to describe the machine tool behavior under several state control policies. Finally, analytical models allow to evaluate energy saving opportunities in real

time [10, 16, 19, 21–23, 47, 48, 55, 58, 59] or to predict production dynamics through a Markov chain [51, 56].

- **Simulation Models** - They are employed by Frigerio and Matta [36, 49, 50, 53] and Zou et al. [10] to find the optimal control parameters for production lines, since it is impossible to handle their complexity analytically. In addition, simulation models are used for performance evaluation as a surrogate of the real system, even if the control is based on analytical relationships or heuristic rules.

2.5 Control Method

Depending on the way in which energy efficiency is achieved, the EEC literature can be classified in two categories [59]:

- **Control Policies & Rules**
- **Real Time Decision Making**

The former category exploits quantitative policies or rules to describe the operation knowledge of machine/system for energy-efficient manufacturing. On the contrary, the latter is aimed at identifying energy control opportunities during the real time production process.

2.5.1 Policies & Rules

This category deals with the triggering of energy-efficient machine state transitions by exploiting ad hoc control rules, generally expressed in "IF..., THEN..." terms. Therefore, the focus lies on formulating appropriate control policies or rules and setting pertinent values for their parameters, which can be either optimized or not. Several types of information can be exploited for defining energy-efficient control policies (cf. figure 2.4), here outlined.

Time. These policies employ the knowledge of machine starvation periods to determine when service should be interrupted or resumed. Since they can be conveniently analyzed with stochastic models, time-based policies were the first to be developed [8, 24, 25, 34, 39, 45, 46, 60]. In particular, Frigerio and Matta [44] propose a framework that integrates different time-based control policies, whose core (*Switching Policy*) is exhaustively addressed in the prelude to chapter 3.

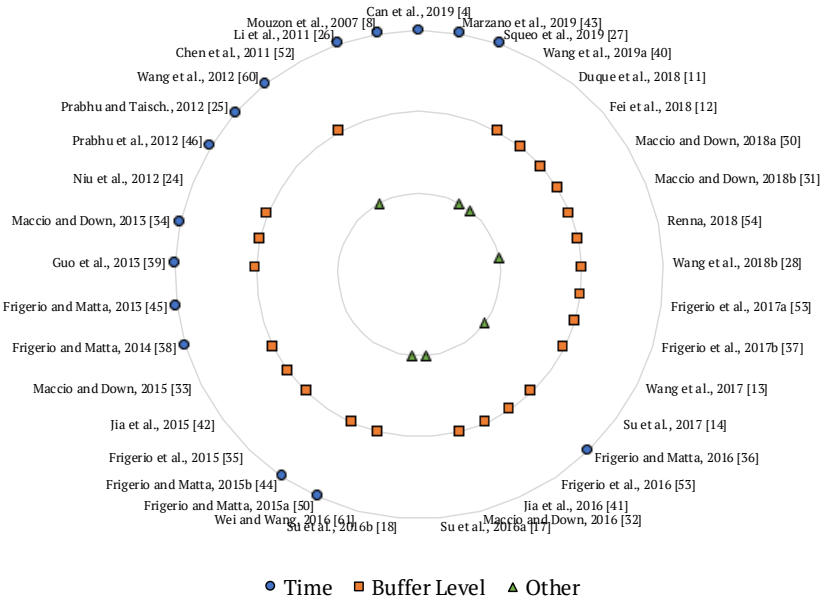


Figure 2.4: Classification of some EEC papers according to the control rule.

Buffer level. These policies employ the knowledge of the number of accumulated parts in the buffers. Control parameters are therefore related to thresholds on buffer levels, such that machines can be controlled during starvation periods, blocking periods or both. They are generally applied to production systems, since buffers are introduced to decouple the behavior of different connected machines. Nevertheless, some publications related to single machines can be found as well. According to Frigerio and Matta [36], a policy that uses the buffer level to control the machine is generally more effective than a time-based one. In literature, the levels of different buffers are used to control a machine:

- **Upstream** – Exploited both in case of single machines [11, 24, 35–37] and production systems [30–33, 50, 52, 53].
- **Downstream** – Exclusively exploited by Renna [54] and Wei and Wang [61]. The former analyzes a pull production control mechanism where customer orders are satisfied by a buffer, filled by a production line. The latter describe the state of a single machine with its production surplus, which can also be negative to account for the backlog of demand.
- **Upstream & Downstream** - Exploited only for production lines, some examples are provided by [12, 13, 28, 40–42, 49].

In particular, it is worth mentioning the work by Chen et al. [52], since they consider a time-dependent threshold on the buffer level to control a two machine Bernoulli serial line.

Time and buffer level. These policies simultaneously employ the knowledge of machine starvation times and the number of accumulated parts in the buffer. Different control rules work together to trigger transitions between machine energy states. For example, Maccio and Down [34] analyze a $M/M/1$ queue whose service is interrupted on the basis of starvation time and resumed after a given number of jobs accumulates in front of the server. On the contrary, Guo et al. [39] analyze a $M/G/1$ queue whose service is interrupted if no customers arrive during the closedown time and resumed when a given number of customers is waiting in the system. Finally, Frigerio and Matta [36] describe a three-parameter policy, i.e. the "TNT Policy", for energy oriented control of a single machine tool. This policy is applied in [50] to individually control the machines of a production line.

Other. Sometimes, different or additional information are employed to develop energy efficient control rules. Wang et al. [40] and Duoque et al. [11] consider the production rate of each machine in addition to information regarding buffer levels. Renna [54] also uses the customer demand and the final buffer level to control the stations of a pull production line.

Completely different parameters are instead employed by Su et al., which address the simultaneous optimization of energy consumptions and production rate in Bernoulli serial lines. In particular, they focus on the optimal allocation of machine capacity, which turns out to be the probability of each machine to be up at the beginning of a new cycle. The authors initially address a two machines line [17], then extend the policy to three and four machines lines, considering different configurations for buffer capacities and energy coefficients [18]. Finally, they propose a heuristic algorithm for larger systems [14].

2.5.2 Real Time Decision Making

The control parameters of the aforementioned policies and rules are generally optimized considering the statistical average behavior of the system or, even, not optimized at all. Therefore, they might not be optimal for a real time scenario, because system dynamics are not fully analyzed for decision making. To address this issue, a further category of papers focuses on the real time identification of energy control opportunities. In particular, data-driven methods are developed to systematically integrate manufacturing

monitoring and control with energy management. The advantage of these approaches is that they enable a real time data-driven prediction capability to more accurately estimate system performance, allowing for real time decision making.

In the followings, the energy *opportunity window* framework is described in details, since it is addressed by several papers and researchers. Later, additional real time decision making approaches are briefly mentioned.

Energy opportunity window. Several researches from EEC literature [9, 10, 16, 19, 21–23, 47, 48, 55, 56, 58, 59, 62] are focused on the identification of energy management opportunities for typical manufacturing systems with multiple machines and buffers. In particular, the *opportunity window* (OW) for a specific machine is defined as the time period during which the machine can be strategically shut down (or adjusted to other states for energy savings) without affecting the throughput of the entire line. Put differently, the OW is the longest possible downtime of a station that does not result in permanent production loss at the end-of-line station. The essence of *opportunity windows* is to aggregate shorter idle durations of a machine to a longer interval, by sleeping it for a suitable time length, which does not affect the throughput [55].

The algorithm to determine an energy OW descends directly from the methods for maintenance opportunities calculation. For perfectly reliable serial production lines the opportunity window is computed as the time it takes for the buffers between the considered station m and the slowest station M^* to become empty ($m < M^*$) or full ($m > M^*$) or, in other words, for the slowest station to become starved ($m < M^*$) or blocked ($m > M^*$) [47]. Therefore, the ability to capture real time buffer levels with sensors on the shop floor is crucial for identifying energy saving opportunities. Nevertheless, due to the intrinsic stochastic nature of production lines, simple deterministic algorithm may lead to inadequate solutions. Stochastic modeling and data-driven methods are therefore combined to calculate the OW under a stochastic scenario [19, 48, 55].

In order to reduce the risk of permanent production losses, machines are generally switched off for a time interval shorter than their OW. In addition, a minimum amount of time, referred to as recovery time [16, 19, 22], is required between two consecutive *opportunity window* exploitations.

Literature complexity in the energy *opportunity window* field has progressively increased: whilst the first papers only focus on OW calculation for a single machine, more recent publications try to determine the conditions for several machines to exploit their energy saving opportunities. Among the latest works in this research stream, Wang et al. [58, 59] propose a digi-

tal twin-based bidirectional operation framework for serial production lines, which estimates in real time energy OW for the idle machines with Max-plus Algebra. This mathematical tool allows to model discrete event systems in linear equations, allowing to reduce their computational complexity. On the contrary, Zou et al. [10] develop a data-driven adaptive control policy to improve profit and energy efficiency of a serial production line and deliver resilient performance against random disruption events. Controller parameters, which are lower bounds for the length of machine *opportunity windows*, are progressively updated to adapt to the slowing varying system reliabilities.

Other. Additional real time decision making approaches can be found in EEC literature, which do not make reference to the energy *opportunity window* framework. Some meaningful examples are here provided.

Li and Sun [29] develop an analytical model to establish a holistic view of energy efficiency for a tandem production line. When a machine is detected to be idle, the optimal action (i.e. turn it off, leave it alone or adjust its power level) is taken in real time by means of an approximate algorithm.

Sun et al. [20] propose a data-driven dynamic energy control method that is implemented at the beginning of each interval in which the production horizon is divided. At each station, the following data are collected for real time decision making: production target, completed production, level of downstream buffer and its required lower bound, remaining time to the end of the planning horizon and production rate.

Li et al. [51] develop an event-based control methodology to improve energy efficiency in a multistage manufacturing system. In particular, an analytical approach allows to quantitatively predict the system production loss which results from an energy saving control event (i.e. turning one or several machines to energy saving mode for a certain duration). The production horizon is then split in control periods, at the beginning of which the optimal energy saving control event is selected in real time.

2.6 Scope

The models found in EEC literature are not necessarily devoted to the optimization of energy consumptions. In fact, some of them are simply developed to assess system performance under state control. It is therefore possible to identify two different purposes related to EEC papers:

- **Optimization**
- **Performance Evaluation**

Once an optimization problem is solved, the effects of the optimal solution on system performance are always evaluated. Nevertheless, to facilitate the understanding, these categories are deliberately kept separate. Only some papers appear in both categories [8, 34, 36, 39, 43], since they carry out performance evaluation before the resolution of the optimization problem.

2.6.1 Optimization

This category deals with identifying, by means of appropriate optimization methodologies, the optimal state control parameters or control actions which maximize/minimize an objective function, respecting some constraints. In figure 2.5, papers are classified in terms of objective function, also indicating the presence of constraints on production performance.

Objective function. In EEC literature, the objective function of the optimization problem is always referred to energy consumption. Many researchers, such as Frigerio and Matta [27, 36, 38, 43–45, 49, 50, 53], Su et al. [14, 17, 18], Guo et al. [39] and Chen et al. [52], consider it alone. Nevertheless, it can also be inserted in a cost function, allowing to find a trade-off solution which accounts for other performance of interest. For example Huang et al. [9] minimize a cost function composed by energy cost, state switching cost, profit loss due to permanent production loss and integrated maintenance cost to identify an optimal strategy for switching machines on or off. On the contrary, Li et al. [51] use a simpler cost function, given by the difference between the gain and the cost of applying an energy saving control event.

Constraints on KPI. Constraints on performance indicators are sometimes introduced: they reduce the feasibility region for possible solutions and increase the optimization complexity. The throughput constraint is the most widespread in EEC literature, since energy savings should be achieved without major service level reductions. Examples are provided by [14, 17, 18, 27, 36, 38, 39, 43–45, 49, 50, 52, 53]. Other constraints regard the workforce [17], the number of issuable switch off commands [44] and the mean waiting time [34, 39].

Optimization methodology. According to the model type employed to describe the effect of state control, different optimization methodologies can be adopted, here outlined.

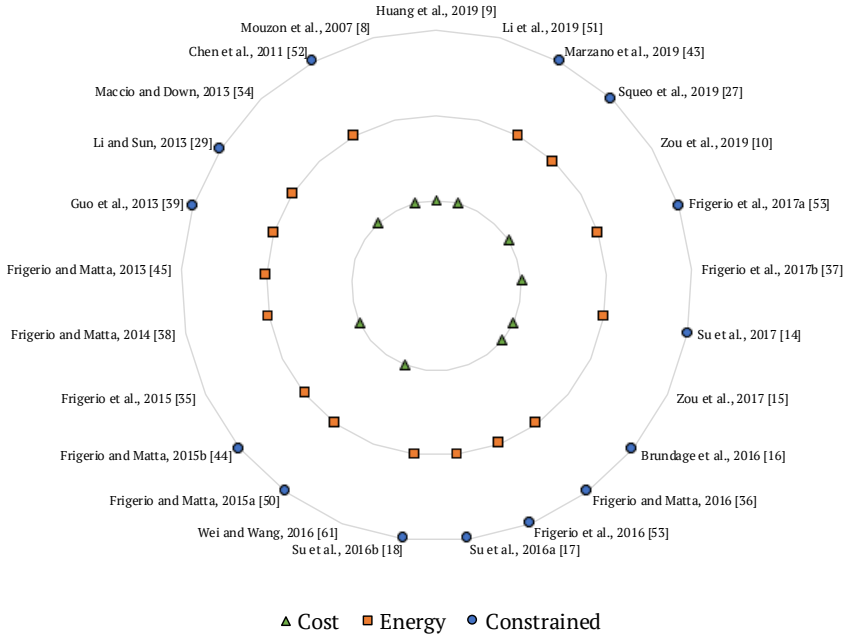


Figure 2.5: Objective function and constraints used in EEC optimization problems as in the literature.

- **Analytical** – It requires the objective function to be differentiable, so that convergence to the global optimum is guaranteed. Since this optimization methodology is derivative-based, analytical relationships and formulas are necessary. Depending on the way in which the optimum is identified, a further classification results:

  - **Exact** – The optimum is obtained in a closed-form, managing the objective function manually. Of course, the problem should be significantly simplified, so that it can be computationally tractable. For example, Guo et al. [39] identify in closed-form the optimal vacation and hysteresis time to achieve an energy-delay tradeoff in a single vacation $M/G/1$ queue. On the contrary, Maccio and Down [34] derive a closed-form expression for the optimal starvation time a $M/M/1$ queue has to wait before moving to a lower power consumption state, also considering a constraint on the expected response time. Frigerio and Matta [36, 44, 45] instead express the expected energy consumption of a single machine in closed-form, in the special case of exponentially distributed interarrivals. By deriving this expression, the optimal control can be exactly

identified. Finally, Su et al. [14, 17, 18] find a closed-form solution for a two-machine Bernoulli serial line.

- **Numerical** – The objective function is still differentiable, but the solution cannot be expressed in closed-form. Therefore, conventional solving methods, such as gradient descent, bisection method, Newton's method, subgradient projection methods and interior-point methods must be applied. Examples are provided by [17, 18, 27, 36, 38, 43–45, 47].

- **Heuristic** – Manufacturing systems are generally non-linear dynamic systems. Therefore, if their complexity is not reduced during the modeling phase, a challenging non-linear energy control problem must be addressed. In this case, an optimal or near-optimal solution can be identified with heuristic optimization techniques and algorithms. For example, Huang et al. [9] and Li et al. [51] exploit genetic algorithms, while Zou et al. [10] make use of a renewal particle swarm optimization algorithm. On the contrary, Frigerio et al. introduce a nested partition metaheuristic framework in [53]. These last one also employ OptQuest[1] to identify the optimal control of buffer-based policies [36, 49, 50]. Finally, Zou et al. [15] and Brundage et al. [16] develop pseudo-procedures to heuristically solve their optimization problems.

- **Dynamic Programming** – This methodology allows to solve stochastic problems in which a sequence of control decisions must be taken at different points in time. The goal is therefore to choose the sequence which causes the system to perform optimally, according to a certain objective function. In EEC literature, the complex interaction between energy control decisions and system state evolution can be modeled with a Markov Decision Process, which can be solved with dynamic programming. Examples are provided by [29, 35, 37, 61].

Optimization occurrence. In EEC literature, many optimization problems are solved only once, because all the required data are available with certainty since the beginning. These problems are addressed in papers which belong to the "Policies & Rules" category (cf. subsection 2.5.1) [8, 14, 17, 18, 27, 34–39, 44, 45, 49, 50, 52, 53, 61] and, by making reference to 1.3, they can be classified as off-line problems.

[1] OptQuest uses an implementation of scatter search (a population-based metaheuristic) as its primary search procedure, with tabu search and neural networks playing a secondary role.

On the contrary, a further class of papers requires optimization to be carried out several times during production progress, since the availability of real time data is required. This is the case of papers belonging to the "Real Time Decision Making" category (cf. subsection 2.5.2) [9, 10, 15, 16, 29, 51]. In addition, it is worth mentioning the work by Marzano et al. [43], which extend an off-line time-based policy by assuming a lack of knowledge on the starvation times distribution. Only they address an on-line problem, since the unknown distribution is progressively fitted in real time and iteratively employed to identify the optimal control.

2.6.2 Performance Evaluation

This category deals with studying the performance of a controlled production system (e.g. energy consumptions, throughput or mean waiting time) over a set of parameters, associated both to the control or to the system itself.

In particular, it includes several works related to the *opportunity window* framework [19, 21–23, 47, 48, 55, 56, 58, 59, 62]. Their scope is in fact to evaluate the energy savings and throughput losses which result from switching off particular machines during their OWs.

Finally, it is worth mentioning a stream of papers [11–13, 28, 40] which focus on performance evaluation of production systems controlled with fuzzy rules. In detail, fuzzy controllers employ real time data of buffers and machines in order to achieve energy efficient operation. The control policy is simply described by linguistic "IF..., THEN..." rules, which connect the real time data to the appropriate switch on/off decision. Wang et al. [40] extend this model by developing a dynamic adaptive fuzzy reasoning Petri net which, starting from the production information of a discrete stochastic manufacturing system, allows to control the transitions among machine energy states.

2.7 Contribution and Structure

Over the last decade, the topic of energy efficient state control (EEC) in manufacturing has gained an increasing prominence within the scientific community. In particular, this field is addressed at two different levels of analysis (i.e. single machine and production system), achieving energy saving either with ad hoc control rules or through a real time decision making. Several models are therefore developed to evaluate system performance under state control policies and to identify the optimal control parameters, which guarantee the NPE minimization.

2.7. Contribution and Structure

Based on the detailed review proposed in this chapter, three main needs emerge from the literature:

- **On-line algorithms.** The exploitation of real time learning allows to better cope with the intrinsic unknown stochasticity of production systems, thus reducing the barriers for a widespread implementation of sustainable solutions. However, only the works proposed by Marzano et al. [43, 63] address on-line control problems.

- **Constrained optimization.** The identification of trade-off solutions, which result from the introduction of productivity criteria (or others) into optimization problems, is fundamental to guarantee the tunability of EEC policies when applied in real environments.

- **Adaptive control.** The possibility to autonomously adapt the control in response to dynamic changes in the system or in the external environment is a clear requirement from industry, despite rarely addressed in EEC literature [9, 10, 25, 54].

In the light of the above needs, this thesis presents an on-line time-based policy, which enhances and extends the works proposed by Marzano et al. [43, 63]. In detail, the existing policy from literature provides the optimal state control parameters for a single machine tool while learning from a real time collection of part arrivals. Nevertheless, the deterioration of machine throughput is considered only in an approximate manner during problem resolution and the arrival process at the machine is assumed to be stationary. Therefore, the major contribution of this work is devoted to:

C1. The development of an efficient solving algorithm to exactly identify the optimal control when constraints limit the feasibility region for possible solutions. In particular, this algorithm is firstly employed to solve the optimization problem in presence of a condition on the expected throughput (**C1.1**). Later, since the policy is optimized considering the statistical average behavior of the machine, two additional constraints are introduced to mitigate the risk of deteriorating machine performance over the single service request (**C1.2**).

C2. The introduction of a change point detection method to identify variations in the part arrival process. This allows to adapt the control to the dynamic behaviour of a real manufacturing environment, thus overcoming a critical barrier for practical implementation.

Note that particular effort is also devoted to accomplish a significant response time improvement, in order to guarantee a real time shop floor applicability.

Several numerical cases are provided to assess the effectiveness and the computational effort of the proposed algorithms, which are entirely coded in Matlab environment.

The remainder of the thesis is organized as follows:

- The prelude to chapter 3 provides the basis to understand the analyzed control problem, presenting the off-line *Switching Policy* by Frigerio and Matta [44, 45].
- Chapter 3 illustrates and reviews the on-line algorithm by Marzano et al. [43, 63]. In particular, enrichments are introduced to enhance its effectiveness and robustness, while limiting the computational requirements. Contribution **C1.1** is here detailed.
- Chapter 4 widens the applicability of the on-line control policy, including considerations on risk and uncertainty. In addition to contribution **C1.2**, this chapter also provides a numerical approach to quantify the confidence of the fitting process carried out to achieve real time learning. The possible applications of this information in the control problem are also investigated.
- Chapter 5 adjusts the on-line control policy to make it effective when the upstream production process is non-stationary or decoupled from the machine by an input buffer. In addition to contribution **C2**, this chapter also introduces a modified objective function to identify an approximate optimal control when an input buffer collects parts in front of the machine. This because, when the on-line control policy is applied to this system, it may not converge towards a unique solution, due to the actual control effect on machine starvation.
- Chapter 6 summarizes the conclusions, while highlighting the possible future developments.

Prelude to Chapter 3

The scope of this prelude is to introduce the off-line control problem as in the literature. In detail, it supplies all the essential information to allow the understanding of the on-line policy proposed in chapter 3. Moreover, numerical examples are provided to get acquainted with the key results of the control problem and the performance indicators of interest.

This prelude is structured in three sections. In section P.1, the model of the considered system is described. In section P.2, the off-line *Switching Policy*, formalized by Frigerio and Matta [44, 45], is addressed. In particular, the control problem which needs to be solved to identify the optimal parameters is thoroughly reported. Finally, in section P.3 some illustrative examples are shown, describing the *scenarios* which will be used as demonstrators in chapter 3.

P.1 System Description

A single machine processing a single part type is considered. This system can represent both machines dedicated to one single part type or to a family of similar items and machines working large batches, while considering the single batch.

In more detail, the machine is perfectly reliable, it might be starving of raw parts and it is never blocked, therefore each part leaves the system immediately after process completion. Machine starvation is modeled considering an input mechanism that manages the arrival process: one part is sent to the machine only during its idle or not productive periods. Stochasticity is therefore introduced in the system considering randomly distributed machine starvation times. The machine has a "green" functionality, since it is possible to deactivate some of its modules for energy saving purposes. By considering a *cycle* as the time frame lasting from the departure of a part until the departure of the following one, the machine can be controlled by means of two parameters: they respectively define the instants wherein the machine tool is switched off/on in a *cycle*.

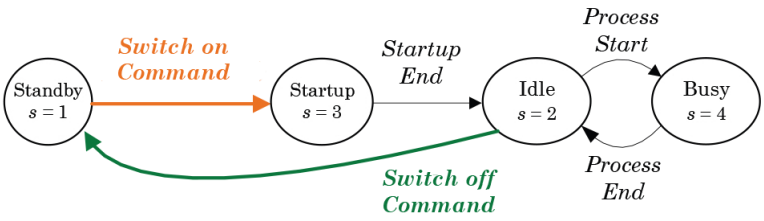


Figure P.1: Machine state model.

P.1.1 Machine Energy States

The machine can be in one of the following energy states, modeled with variable $s \in \mathcal{L}_S = \{1, 2, 3, 4\}$:

1. *Standby State*: $s = 1$. In this state some machine components are not active, such that a lower amount of power is consumed compared to other energy states. Being in a "sleeping" mode, the machine cannot process a part.

2. *Idle State*: $s = 2$. In this state the machine is ready to process a part upon its arrival, since all machine modules are active.

3. *Startup State*: $s = 3$. It is the transitory state the machine has to pass through to switch from the standby state to the idle state. While in startup, the control activates the machine components to achieve proper working conditions.

4. *Busy State*: $s = 4$. In this state, the machine is processing a part. It also includes any non-cutting operation with the part loaded on the machine.

The transition between two states can be triggered either by an uncontrollable event (eg. the part arrival or process completion) or by a controllable event, as described in section P.2. The proposed machine state model, commonly used in literature, is graphically represented in figure P.1.

P.1.2 Modeling Assumptions

The following assumptions are introduced:

1. Machine processing time t_p and machine startup duration t_{su} are finite, deterministic and constant. In particular, it is assumed that the user has already validated the startup time, which can be considered a constant input to the control policy.

2. The power consumed by a machine visiting a certain state w_s is a finite, deterministic and constant value that represents the average power absorbed in such a state. Power values are constrained such that $0 \leq w_1 < w_2 < w_3$ and $w_4 > w_2$. This assumption realistically represents manufacturing equipment that commonly require high power absorptions to resume the operational readiness of inactive components.

3. Machine control can only observe the starvation times, described by the random variable X and assumed to be independent. Machine starvation times are modeled by the stationary probability density function (pdf) $f_X(x)$, with mean t_a. The realization of X is denoted with x, which also constitutes the arrival time realization in a *cycle*. At the beginning of each *cycle* ($x = 0$) the machine is in the idle state, waiting for part arrival.

4. The random variable X is not affected by the applied energy efficient control policy. This assumption can represent a production case where machines are synchronized. If the assumption does not hold, the approach yields to approximate results.

5. When the machine is not working, the incoming part is immediately processed if the machine is ready, otherwise it has to wait until the end of the startup procedure. In this latter case, a fixed holding power w_q is absorbed by the auxiliary equipment (e.g. part-handling system or heated/cooled buffer) for keeping the part until machine operational readiness is resumed.

P.2 Control Policy

This section describes the *Switching Policy* (SP), i.e. an off-line time-based energy efficient state control policy developed by Frigerio and Matta [44, 45]. This policy can be used to control the state of the machine by activating a transition form the idle to the standby state – i.e. *Switch off command* – and from the standby to the startup state – i.e. *Switch on command*:

> **Switching Policy** - Switch off the machine after a time interval τ_{off} has elapsed from the last departure. Then, switch on the machine after a time interval τ_{on} has elapsed from the last departure, i.e., when $\tau_{on} - \tau_{off}$ has elapsed from the switch off command, or when a part arrives.

Starting from an a-priori knowledge of the starvation times distribution $f_X(x)$, the policy control parameters $\boldsymbol{\tau} = \{\tau_{off}, \tau_{on}\}$ have to be tuned in order to minimize the *expected value of the energy consumed in a cycle by the machine*.

P.2.1 Energy Consumption Model

When the SP is applied, the machine evolution may follow different paths in terms of states visited and transitions triggered, depending on the interaction between the random part arrival X and the control parameters $\boldsymbol{\tau}$. The *non-processing energy* (NPE) e_i^{path} consumed through each path is therefore the output of a stochastic process. It can be calculated as the conditional expectation of *power* $\times$ *time* for all the states visited through that path, given a certain event Λ_i representing the manifestation of path i over the feasible ones. The same approach can be applied to derive the *waiting time* q_i^{path} a part has to experience through each path before being processed. The busy state ($s = 4$) is not considered in the energy consumption model, since it does not affect the selection of the policy parameters. In fact, the service request cannot be avoided and the *processing energy* (PE) $w_4 t_p$ is constant. At each *cycle* four different events Λ_i may occur, here outlined.

1. *The part arrives before the machine is switched off*: $\Lambda_1 = \{0 \leq x \leq \tau_{off}\}$. The machine never switches from the idle state and the part is therefore processed immediately ($q_1^{path} = 0$). The energy consumption is:

$$e_1^{path} = \mathbb{E}[w_2 X \mid \Lambda_1] \tag{P.1}$$

2. *The part arrives when the machine is in standby*: $\Lambda_2 = \{\tau_{off} < x \leq \tau_{on}\}$. The machine, switched off at τ_{off}, is immediately triggered into the startup state by part arrival, so that processing can start only after the startup. The energy consumption is:

$$e_2^{path} = w_2 \tau_{off} + \mathbb{E}[w_1 (X - \tau_{off}) \mid \Lambda_2] + w_3 t_{su} \tag{P.2}$$

   Since the part has to wait the whole startup duration to be processed, the waiting time is:

$$q_2^{path} = t_{su} \tag{P.3}$$

3. *The part arrives when the machine is executing the startup procedure*: $\Lambda_3 = \{\tau_{on} < x \leq \tau_{on} + t_{su}\}$. The machine, switched off at τ_{off}, is switched on again after a time interval τ_{on} has elapsed from the last departure and it is executing the startup procedure when the part

arrives. Processing can start as soon as the transition to the idle state is completed. The energy consumption is:

$$e_3^{path} = w_2\tau_{off} + w_1(\tau_{on} - \tau_{off}) + w_3 t_{su} \tag{P.4}$$

Since the part has to wait for the startup completion to be processed, the waiting time is:

$$q_3^{path} = \mathbb{E}[(\tau_{on} + t_{su} - X) \mid \Lambda_3] \tag{P.5}$$

4. *The part arrives when the machine has completed the startup procedure*: $\Lambda_4 = \{x > \tau_{on} + t_{su}\}$. The machine, switched off at τ_{off}, is switched on again at τ_{on} and, after the completion of the startup procedure, is in the idle state when the part arrives. Therefore, the part is processed immediately ($q_4^{path} = 0$). The energy consumption is:

$$e_4^{path} = w_2\tau_{off} + w_1(\tau_{on} - \tau_{off}) + w_3 t_{su} + \mathbb{E}[w_2(X - \tau_{on} - t_{su}) \mid \Lambda_4] \tag{P.6}$$

Since all these events are mutually exclusive and collectively exhaustive, by letting $\mathbb{P}(\Lambda_i)$ be the occurrence probability of event Λ_i, the *expected NPE consumed in a cycle* $\Phi(\boldsymbol{\tau})$ and the *expected waiting time* $H(\boldsymbol{\tau})$ result from the application of the Total Expectation Theorem [64]:

$$\Phi(\boldsymbol{\tau}) = \sum_{i=1}^{4} e_i^{path}\mathbb{P}(\Lambda_i) \tag{P.7}$$

$$H(\boldsymbol{\tau}) = q_2^{path}\mathbb{P}(\Lambda_2) + q_3^{path}\mathbb{P}(\Lambda_3) \tag{P.8}$$

These expectations are functions of the policy control parameters $\boldsymbol{\tau}$ and, given the pdf $f_X(x)$, they can be expressed as:

$$\begin{aligned}\Phi(\boldsymbol{\tau}) = w_1 &\left[\int_{\tau_{off}}^{\tau_{on}} (x - \tau_{off}) f_X(x)\, dx + (\tau_{on} - \tau_{off}) \int_{\tau_{on}}^{\infty} f_X(x)\, dx\right] + \\ + w_2 &\left[\int_{0}^{\tau_{off}} x f_X(x)\, dx + \tau_{off} \int_{\tau_{off}}^{\infty} f_X(x)\, dx + \right. \\ + &\left.\int_{\tau_{on}+t_{su}}^{\infty} (x - \tau_{on} - t_{su}) f_X(x)\, dx\right] + w_3 \left[t_{su} \int_{\tau_{off}}^{\infty} f_X(x)\, dx\right]\end{aligned} \tag{P.9}$$

$$H(\boldsymbol{\tau}) = t_{su} \int_{\tau_{off}}^{\tau_{on}} f_X(x)\, dx + \int_{\tau_{on}}^{\tau_{on}+t_{su}} (\tau_{on} + t_{su} - x) f_X(x)\, dx \tag{P.10}$$

The *expected value of the energy consumed in a cycle by the machine* $g(\boldsymbol{\tau})$ can be finally written as:

$$g(\boldsymbol{\tau}) = \Phi(\boldsymbol{\tau}) + w_q H(\boldsymbol{\tau}) \tag{P.11}$$

where w_q is the power consumed by the auxiliary equipment to keep the part waiting for machine readiness.

P.2.2 Optimization Problem

Once derived the analytical expression for the objective function $g(\boldsymbol{\tau})$, it is possible to formulate the *optimization problem* which needs to be solved to search for the optimal values of the *Switching Policy* control parameters $\boldsymbol{\tau}^* = \{\tau^*_{off}, \tau^*_{on}\}$.

Whenever this policy is applied to the machine, it may lead to a throughput reduction because, in some cases, the part has to wait until the end of the startup procedure for being processed. In fact, the *expected throughput* $\theta(\boldsymbol{\tau})$ depends on the *expected waiting time* $H(\boldsymbol{\tau})$:

$$\theta(\boldsymbol{\tau}) = \frac{1}{t_p + t_a + H(\boldsymbol{\tau})} \tag{P.12}$$

Contrariwise, if the machine is kept always on (AON policy), i.e. $\boldsymbol{\tau} = \{\infty, \infty\}$, its throughput is the maximum achievable, since parts are immediately processed upon arrival:

$$\theta_{AON} = \frac{1}{t_p + t_a} \tag{P.13}$$

Therefore, to avoid significant profit losses, a lower bound for the *expected throughput* $\theta(\boldsymbol{\tau})$ is considered in the *optimization problem*:

$$\boxed{\begin{aligned} \boldsymbol{\tau}^* = \quad & \arg\min_{\boldsymbol{\tau}} \; g(\boldsymbol{\tau}) && \text{(P.14)} \\ \text{Subject to:} \quad & \theta(\boldsymbol{\tau}) \geq (1-\epsilon)\theta_{AON} && \text{(P.15)} \\ & \tau_{on} > \tau_{off} && \text{(P.16)} \\ & \tau_{off}, \tau_{on} \in \mathbb{R}_0^+ && \text{(P.17)} \end{aligned}}$$

where $\epsilon \in [0, 1]$ is the maximum admissible throughput reduction with respect to the AON policy.

The *throughput constraint* (P.15) can be manipulated, by plugging in equations (P.12) and (P.13), to obtain an equivalent condition on the *expected waiting time* $H(\boldsymbol{\tau})$:

$$\frac{1}{t_p + t_a + H(\boldsymbol{\tau})} \geq (1-\epsilon)\frac{1}{t_p + t_a} \Rightarrow H(\boldsymbol{\tau}) \leq \frac{\epsilon}{1-\epsilon}(t_p + t_a) = H_{max} \tag{P.18}$$

where H_{max} is an upper bound for the *expected waiting time* $H(\boldsymbol{\tau})$, constant once defined ϵ and $f_X(x)$.

Finally, constraint (P.16) is introduced to guarantee the consistency of the control action, in fact the machine cannot be resumed to the idle state before being switched off. This constraint, combined with that in equation (P.17), allows to define the domain of decision variables.

P.2.3 Structural Properties

In order to provide a better understanding of the *optimization problem*, some significant properties from literature [44] are now outlined. They will be employed in chapter 3 to develop a solving strategy for effectively identifying the optimal control parameters $\boldsymbol{\tau}^*$.

Property P.1. Relaxing the *throughput constraint* (P.15), the objective function $g(\boldsymbol{\tau})$ is continuous on the closed and bounded set of values that τ_{off} and τ_{on} can jointly assume. In addition, presenting finite limits, it has both a maximum and a minimum on this interval. As a consequence, the optimal solution $\boldsymbol{\tau}^*$ can be found.

Theorem P.1. *Control parameters are independent, since the mixed partial derivative of $g(\boldsymbol{\tau})$ with respect to τ_{off} and τ_{on} is equal to zero. In particular, the objective function partial derivatives can be expressed as:*

$$\frac{\partial g(\boldsymbol{\tau})}{\partial \tau_{off}} = [1 - F_X(\tau_{off})][-(w_3 + w_q)t_{su}HR(\tau_{off}) + w_2 - w_1] \tag{P.19}$$

$$\frac{\partial g(\boldsymbol{\tau})}{\partial \tau_{on}} = [1 - F_X(\tau_{on})][r(\tau_{on})(w_2 + w_q) - w_2 + w_1] \tag{P.20}$$

where $F_X(x)$ is the cdf of the starvation times distribution, while $HR(\tau_{off})$ is the Hazard Rate[2] of X evaluated for $x = \tau_{off}$ and $r(\tau_{on}) = \frac{F_X(\tau_{on}+t_{su})-F_X(\tau_{on})}{1-F_X(\tau_{on})}$.

Theorem P.2. *Under the assumption of unimodal distribution modeling the starvation time, the objective function $g(\boldsymbol{\tau})$ has at most one critical point in addition to $\{\infty, \infty\}$ — which is always a critical point. For these distributions, the general Switching Policy degenerates into simpler strategies. In particular:*

[2]Given the pdf $f_X(x)$ and the cdf $F_X(x)$ of a random variable, the Hazard Rate (HR) is defined as follows:

$$HR(t) = \frac{f_X(t)}{1 - F_X(t)}$$

It represents the conditional probability that a part will arrive in the next moment if the machine is starved at time t.

- *For Decreasing Hazard Rate (DHR) distributions,* $\boldsymbol{\tau}^* = \{\tau^*_{off}, \infty\}$ *with* $\tau^*_{off} \in \mathbb{R}^+_0$.
- *For Increasing Hazard Rate (IHR) distributions,* $\boldsymbol{\tau}^* = \{0, \tau^*_{on}\}$ *with* $\tau^*_{on} \in \mathbb{R}^+$ *or* $\boldsymbol{\tau}^* = \{\infty, \infty\}$.

Property P.2. The *expected waiting time* $H(\boldsymbol{\tau})$ monotonically decreases over τ_{off} and monotonically increases over τ_{on}. Therefore, the maximum waiting time occurs for $\boldsymbol{\tau} = \{0, \infty\}$, since every part has to wait the whole startup transitory t_{su} before being processed.

P.3 Illustrative Examples

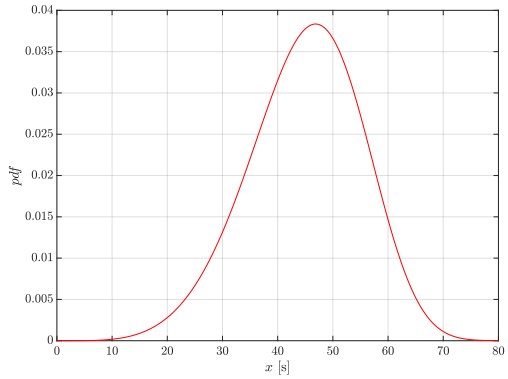
Since the optimal policy is case-dependent, in this section some examples are proposed to illustrate how the optimal solution $\boldsymbol{\tau}^*$ changes according to different operating situations and environments – i.e. machine parameters and starvation times distribution. The *scenarios* here presented will be used as demonstrators in Chapter 3.

Machine parameters. The three different machining centers from literature [44, 45, 63] are addressed, whose parameters are collected in table P.1. In detail, machine M2 features the same time parameters (i.e. t_{su} and t_p) of M1, but it is characterized by higher power absorptions. On the contrary, the energy performance of machine M3 are similar to that of M1, with a significantly shorter processing time duration.

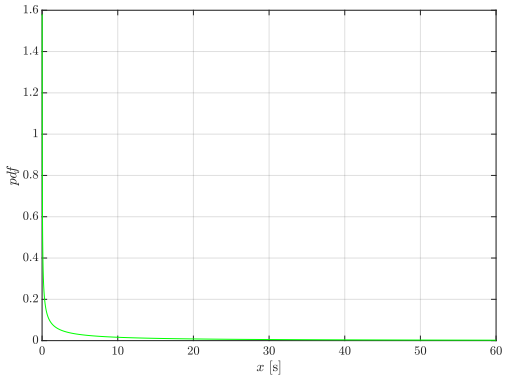
Table P.1: Machine parameters.

Machine	$w_1[kW]$	$w_2[kW]$	$w_3[kW]$	$w_q[kW]$	$t_{su}[s]$	$t_p[s]$
M1	0.52	5.35	6.08	1	24	168
M2	3.12	11	12.5	0	24	168
M3	0.85	4.5	6	1	20	64

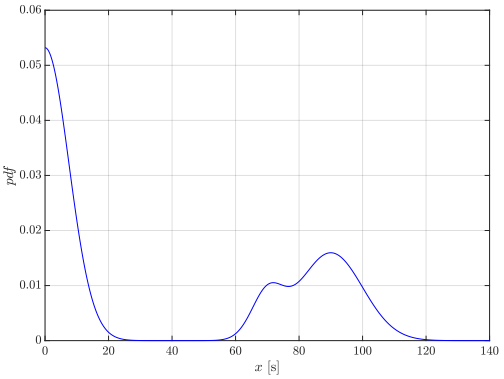
Starvation times distributions. Three starvation times distributions $f_X(x)$ are considered, whose trend is shown in figure P.2. For fixed machine tool parameters, they result in completely different shaped objective functions $g(\boldsymbol{\tau})$. In particular:



(a) D1 - Unimodal with IHR.



(b) D2 - Unimodal with DHR.



(c) D3 - Multimodal.

Figure P.2: Graphical display of the considered distributions.

- D1 (Unimodal with IHR) - Weibull distribution with shape $k = 5$ and scale $\beta = 49.011$ (i.e., mean is $t_a = 45\ s$).

- D2 (Unimodal with DHR) - Weibull distribution with shape $k = 0.45$ and scale $\beta = 15.73$ (i.e., mean is $t_a = 39\ s$).

- D3 (Multimodal) - Weighted sum of three normal densities, such their composition is characterized by a mode in $x = 0$ followed by a second bimodal peak. The pdf is here reported.

$$f_X(x) = \begin{cases} N(x, 7.5) + 0.1N(x - 70, 5) + 0.4N(x - 90, 10) & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{P.21}$$

Table P.2: Optimal control $\boldsymbol{\tau}^*$ for different *scenarios*. Related energy and throughput performance are also given.

Scenario	$\boldsymbol{\tau}^*$ [s]	g_{AON} [$kJ/part$]	$g(\boldsymbol{\tau}^*)$ [$kJ/part$]	$\Delta g(\boldsymbol{\tau}^*)$ [%]	θ_{AON} [$parts/h$]	$\theta(\boldsymbol{\tau}^*)$ [$parts/h$]	$\Delta\theta(\boldsymbol{\tau}^*)$ [%]
M1/D1/1	$\{0, 29.18\}$	240.75	175.94	26.92	16.90	16.22	-4.04
M1/D2/1	$\{15.91, \infty\}$	208.59	121.93	41.55	17.39	16.68	-4.07
M1/D3/1	$\{11.65, 72.87\}$	246.06	153.80	37.49	16.82	16.29	-3.18
M2/D2/1	$\{18.37, \infty\}$	428.87	295.55	31.09	17.39	16.73	-3.82
M3/D3/1	$\{12.12, 74.32\}$	206.96	141.89	31.44	32.73	31.17	-4.77

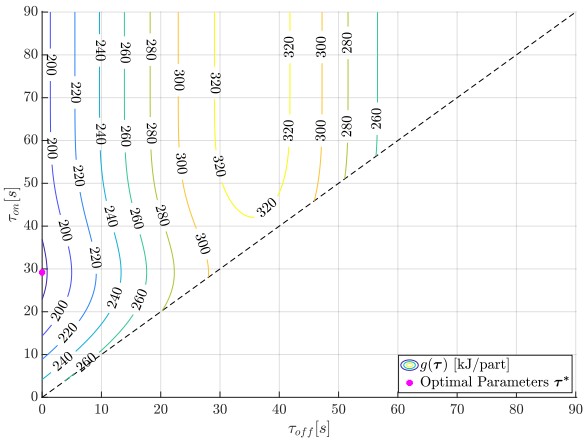
Off-line results. In the followings, the term *scenario* is employed to refer to a specific combination of machine tool, starvation times distribution and *throughput constraint* setting. For the sake of simplicity, each *scenario* is addressed with a notation "M/D/ϵ". The formulation of a *scenario* exhaustively provides the necessary inputs to solve the *optimization problem* introduced in subsection P.2.2. In particular, table P.2 shows the optimal control $\boldsymbol{\tau}^*$ for some combinations of machine parameters and starvation times distribution, obtained by relaxing the *throughput constraint*[3] in equation (P.15) (*unconstrained optimization problem* - $\epsilon = 1$). For each analyzed *scenario*, the performance of the optimal control is described providing the expected energy saving percentage $\Delta g(\boldsymbol{\tau}^*)$ and the expected throughput reduction percentage $\Delta\theta(\boldsymbol{\tau}^*)$ with respect to the AON condition:

$$\Delta g(\boldsymbol{\tau}^*) = 100 \cdot \frac{g_{AON} - g(\boldsymbol{\tau}^*)}{g_{AON}} \tag{P.22}$$

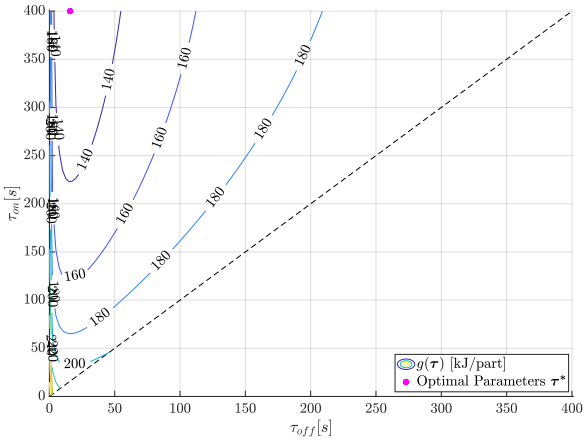
$$\Delta\theta(\boldsymbol{\tau}^*) = 100 \cdot \frac{\theta(\boldsymbol{\tau}^*) - \theta_{AON}}{\theta_{AON}} \tag{P.23}$$

The proposed illustrative examples show that the benefits achievable by implementing the control policy are meaningful, even though a throughput reduction always results, which may sometimes be unacceptable. In addition, it is clear how for unimodal distributions (i.e. D1 and D2) the optimal control parameters $\boldsymbol{\tau}^*$ lead to a simpler strategy, in accordance with theorem P.2. Finally, the objective function $g(\boldsymbol{\tau})$ shapes for some *scenarios* are graphically shown in the contour plots of figure P.3.

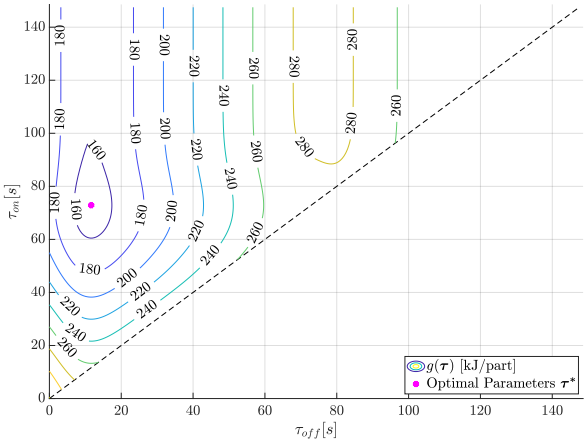
[3]Scenarios with $\epsilon < 1$ will be specifically addressed in Chapter 3.



(a) M1/D1/1 - $\boldsymbol{\tau}^* = \{0, 29.18\}$.



(b) M1/D2/1 - $\boldsymbol{\tau}^* = \{15.91, \infty\}$.



(c) M1/D3/1 - $\boldsymbol{\tau}^* = \{11.65, 72.87\}$.

Figure P.3: Contour plots of $g(\boldsymbol{\tau})$ for different *scenarios*.

Chapter 3

On-Line Control Policy: Extensions and Improvements

The scope of this chapter is to present an on-line time-based policy, which is able to control a single machine while learning from a real time collection of part arrivals. The proposed algorithm improves and extends an already existing policy by Marzano et al. [43, 63] which, in turns, adapt the off-line *Switching Policy* formalized by Frigerio and Matta [44, 45] to an innovative on-line framework.

Particular effort is devoted to the development of a solving algorithm to account for an upper bound on the machine service level reduction (contribution **C1.1**). In addition, a significant response time improvement is accomplished, allowing to achieve a real time shop floor applicability. Specifically, each of the phases from literature is analyzed in details, introducing enrichments to enhance the algorithm effectiveness and robustness, while limiting its computational requirements.

This chapter is structured in seven sections. In section 3.1, the on-line algorithm is described as in the literature. In section 3.2, the Kernel Density Estimation method is introduced, comparing the performance of three bandwidth estimation methods. In section 3.3, a solving algorithm is proposed to effectively handle the *optimization problem*, exploiting its structural properties. In section 3.4, a robust policy to manage the implementation of control parameters is described. In section 3.5, the algorithm effectiveness and computational requirements are evaluated with numerical cases. Finally, in section 3.6, some conclusive remarks are drawn.

3.1 On-Line Control Policy

This section briefly describes the energy efficient on-line algorithm as in the literature. In particular, the off-line control problem formulated in P.2 is extended by keeping the same system and assumptions, except for the knowledge level of the starvation times distribution $f_X(x)$:

- The off-line *Switching Policy* assumes $f_X(x)$ to be known.
- The on-line approach assumes $f_X(x)$ to be unknown. Nevertheless, the machine is able to observe the real time starvation times data, progressively collected in vector $\boldsymbol{x} = \{x_1, \ldots, x_n\}$.

The resulting on-line control problem is addressed by adopting a Separated Estimation and Optimization (SEO) approach, which consists in two phases: the first focuses on the estimation of stochasticity, whilst the second handles the objective function optimization. In addition, since data acquisition is on-line, the risk of implementing a control based on biased estimates might be high, especially with few observations available. Therefore, a third phase is introduced to reduce the risk of incurring in an increase of energy consumptions.

In order to exploit the greater amount of information which becomes progressively available, the on-line control problem is solved by means of an iterative algorithm, which performs the aforementioned phases in sequence. In particular, at each iteration k, the on-line algorithm is characterized by:

1. A *Learning Phase* where, starting from the collected data $\boldsymbol{x}$, an estimate of the starvation times distribution $\hat{f}_k(x)$ is made.
2. An *Optimization Phase*, where the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}} = \{\hat{\tau}_{off,k}, \hat{\tau}_{on,k}\}$ are identified.
3. An *Implementation Phase*, which ensures that the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ replace the current ones $\boldsymbol{\tau_{impl}} = \{\tau_{off,impl}, \tau_{on,impl}\}$ only if the estimated advantage is significant. This phase also allows to minimize the number of control changes, reducing the variability induced in the downstream production process.

To limit the computational burden, the algorithm is iterated only after the collection of Δn new observations. Therefore, at iteration k the number of available data to estimate $\hat{f}_k(x)$ is $n = k\Delta n$. The schematic representation of the algorithm main phases is shown by the flow chart in figure 3.1.

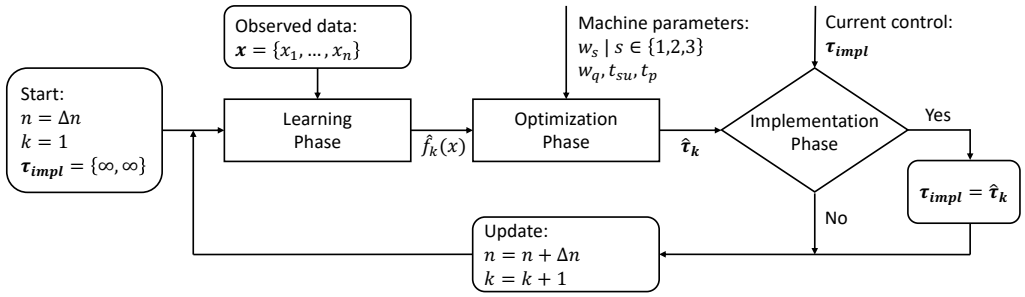


Figure 3.1: Algorithm main phases.

3.2 Learning Phase

This section describes the first phase of the algorithm, which is aimed at estimating the starvation times pdf $\hat{f}_k(x)$ on the basis of collected real time data $\boldsymbol{x}$. Since $\hat{f}_k(x)$ is the basis of the whole on-line algorithm, the *Learning Phase* is crucial to guarantee its effectiveness and robustness and to limit the response time. Therefore, a deeper analysis of the estimation methods is provided.

In literature, two different methods based on a frequentist inference are alternatively employed: the Maximum Likelihood Estimation (MLE) and the Kernel Density Estimation (KDE). Since the former allows to estimate the parameters of a known distribution type, it is not further addressed. In fact the distribution type is generally unknown in practice. Conversely, the latter learns automatically the shape of the density from the data. The KDE effectiveness is mostly affected by the choice of the bandwidth parameter h, which controls the smoothness of the resulting pdf.

Therefore, three bandwidth estimation methods are compared in terms of goodness of the estimate and computational requirements, as a function of the dataset size. The scope of the proposed sensitivity analysis is to identify the method which provides satisfactory and robust results, while ensuring low computational requirements for an on-line implementation.

3.2.1 Kernel Density Estimation Method

The Kernel Density Estimation method, also known as the Parzen's window [65], is one of the most common non-parametric approaches to estimate the probability density function (pdf) of a random variate starting from a set of n observations, without assuming the distribution type.

Let $\boldsymbol{x} = \{x_1, \ldots, x_n\} \in \mathbb{R}^n$ be a univariate, independent and identically distributed random sample from an unknown distribution with pdf $f_X(x)$.

The Kernel Density Estimator $\hat{f}_{X|H}(x|h)$ of the pdf can be expressed as:

$$\hat{f}_{X|H}(x|h) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.1}$$

where $K : \mathbb{R} \to \mathbb{R}$ is a smooth function called the *kernel function*, $h \in \mathbb{R}^+$ is the smoothing parameter or *bandwidth* and the "hat" symbol denotes the KDE dependence from sample $\boldsymbol{x}$. Therefore, given vector $\boldsymbol{x}$, the KDE is affected by the choice of kernel function $K(\cdot)$ and the bandwidth h. In particular, once selected a kernel function $K(\cdot)$, the KDE is the *conditional density function* of X given $H = h$.

Several kernel functions are of common use: uniform, triangular, biweight, triweight, Epanechnikov and Gaussian. Gaussian kernel function is very frequent due to its convenient mathematical properties. Herewith, the following Gaussian kernel is considered:

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \tag{3.2}$$

Therefore, equation (3.1) becomes:

$$\begin{aligned} \hat{f}_{X|H}(x|h) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-x_i}{h})^2} = \\ = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}h} e^{-\frac{(x-x_i)^2}{2h^2}} = \frac{1}{n}\sum_{i=1}^{n} N(x - x_i, h) \end{aligned} \tag{3.3}$$

where $N(x - m, v)$ is the normal distribution with mean m and standard deviation v. It can be noticed that the KDE $\hat{f}_{X|H}(x|h)$ is obtained as the average of n normal densities, each centered on one observation x_i and with standard deviation h. Since the bandwidth controls the width of the kernel, it acts as a tuning parameter for the KDE, determining how each observation is spread over the surrounding space. If the bandwidth $h = h(n)$ is chosen as a function of the observation number n such that:

$$\lim_{n\to\infty} h(n) = 0 \tag{3.4}$$

then the estimates defined by equation (3.1) are asymptotically unbiased as n increases.

3.2.2 Alternative Methods for Bandwidth Selection

The goodness of the KDE is related to the choice of bandwidth h:

- High values of bandwidth h lead to under-fitting (or oversmoothing), where the structure in the data is washed out by a kernel which is too wide.
- Low values of bandwidth h lead to over-fitting (or undersmoothing), where data are used only locally and the KDE is very variable.

Therefore, the selection is not trivial. The purpose of this section is to minimize the estimation error of the target density. Let us define the Integrated Square Error (ISE) as global error criterion [66]:

$$ISE(h) = \int [\hat{f}_{X|H}(x|h) - f_X(x)]^2 \, dx \tag{3.5}$$

which represents the squared distance among the pdf $f_X(x)$ and the KDE $\hat{f}_{X|H}(x|h)$. The ISE is a random variable which, by the way, depends on the particular realization of n observations. Therefore, for the purpose of bandwidth selection, the Mean Integrated Squared Error (MISE) is examined, which is the average of the ISE over these realizations:

$$MISE(h) = \mathbb{E}[ISE(h)] = \mathbb{E}\left[\int [\hat{f}_{X|H}(x|h) - f_X(x)]^2 \, dx\right] \tag{3.6}$$

where $\mathbb{E}$ denotes the expected value with respect to the sample. In particular, the value h^* minimizing the $MISE$ should be selected:

$$h^* = \arg\min_{h>0} \{MISE(h)\} \tag{3.7}$$

Function $MISE(h)$ is unknown, thus h^*. Therefore, several methods have been proposed in the literature for bandwidth selection. Herewith, we compare the three most common methods.

Silverman Rule of Thumb. This method, proposed by Silverman [67], approximates the $MISE(h)$ in equation (3.6) with its asymptotic approximation (AMISE) obtained with the truncated Taylor expansions. It can be proved that the optimal bandwidth minimizing the AMISE decreases with the sample size n proportionately to $n^{-\frac{1}{5}}$. Furthermore, considering the case in which $f_X(x)$ is a normal density, it is possible to obtain h^*_{AMISE} as follows:

$$h^*_{\text{AMISE}} = \left(\frac{4}{3n}\right)^{\frac{1}{5}} s_X = 1.059 \cdot s_X \cdot n^{-\frac{1}{5}} \tag{3.8}$$

where s_X denotes the standard deviation of the sample data $\boldsymbol{x}$. Equation (3.8) provides a practical choice of bandwidth which requires very little calculation. Nevertheless, this approach tends to introduce oversmoothing if applied to non-normal data, above all when they are strongly skewed or multimodal.

In order to reduce the risk of oversmoothing, Silverman proposes the following Rule of Thumb:

$$h^*_{rot} = 0.9 \min \left\{ s_X, \frac{IQR}{1.349} \right\} n^{-\frac{1}{5}} \tag{3.9}$$

where $IQR = \hat{q}_{0.75} - \hat{q}_{0.25}$ is the interquantile range of the sample data $\boldsymbol{x}$. In fact, s_X tends to overestimate the spread when the data present thick tails.

Least-Squares Cross Validation. This method, proposed by Rudemo and Bowman [68, 69], aims to estimate the optimal bandwidth with the minimizer of $ISE(h)$, approximating this function with a cross-validation approach. In more details, equation (3.5) can be also expressed as:

$$ISE(h) = \int \hat{f}^2_{X|H}(x|h)\, dx - 2 \int f_X(x) \hat{f}_{X|H}(x|h)\, dx + \int f^2_X(x)\, dx \tag{3.10}$$

Since the minimizer of $ISE(h)$ does not depend on the unknown quantity $\int f^2_X(x)\, dx$, the last term can be neglected. The approximation lies in the second term because, being dependent on function $f_X(x)$, it is unknown and needs to be estimated. In general, an integral with respect to $f_X(x)$ is an expectation with respect to random variable X. Since the true expectation is unknown, it can be estimated by taking the sample average, suggesting [70]:

$$\int f_X(x) \hat{f}_{X|H}(x|h)\, dx \approx \frac{1}{n} \sum_{i=1}^{n} \hat{f}_{X|H}(x_i|h) \tag{3.11}$$

However $\hat{f}_{X|H}(x_i|h)$ is itself a function of observation x_i, therefore the cross-validation methodology is employed to replace $\hat{f}_{X|H}(x|h)$ with the leave-one-out estimator $\hat{f}_{X|H,-i}(x|h)$, constructed from the data without observation x_i:

$$\hat{f}_{X|H,-i}(x|h) = \frac{1}{(n-1)h} \sum_{\substack{j=1 \\ j \neq i}}^{n} K\left(\frac{x - x_j}{h} \right) \tag{3.12}$$

Since the first term in equation (3.10) can be directly calculated, $ISE(h)$ can be approximated as [71]:

$$ISE(h) \approx \int \hat{f}^2_{X|H}(x|h) dx - \frac{2}{n} \sum_{i=1}^{n} \hat{f}_{X|H,-i}(x_i|h) \tag{3.13}$$

Nevertheless, Bowman replaces the first term of expression (3.13) with its asymptotic estimate:

$$ISE(h) \approx \frac{1}{n}\sum_{i=1}^{n}\int \hat{f}^2_{X|H,-i}(x|h)dx - \frac{2}{n}\sum_{i=1}^{n}\hat{f}_{X|H,-i}(x_i|h) \tag{3.14}$$

He indeed demonstrates that the expectation of expression (3.14) is the MISE of $\hat{f}_{X|H}(x|h)$ based on $n-1$ observations, omitting the $\int f_X^2(x)\,dx$ term. The value that minimizes this expression therefore provides an estimate of the optimal smoothing parameter. For Gaussian kernel function, the integrals in expression (3.14) can be easily calculated, leading to a simple function to be minimized [72]:

$$\begin{aligned} h^*_{ls} = \arg\min_{h>0}\Bigg[&\frac{1}{n-1}N(0,\sqrt{2}h) + \frac{n-2}{n(n-1)^2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}N(x_i-x_j,\sqrt{2}h) + \\ &-\frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}N(x_i-x_j,h)\Bigg] \end{aligned} \tag{3.15}$$

Leave-One-Out Cross Validation. This method, proposed by Habbema et al. and Duin [73, 74] and employed in the on-line algorithm from literature, suggests to choose h so that the pseudo-likelihood function $L(h) = \prod_{i=1}^{n}\hat{f}_{X|H}(x_i|h)$ is maximized. In particular, $L(h)$ gives the probability (likelihood) of obtaining the observed data as a function of the selected bandwidth h. Therefore, the bandwidth that maximizes $L(h)$, is the one which "best explains" the observed data.

Since $L(h)$ has a trivial maximum for $h = 0$, the cross validation principle is invoked by replacing $\hat{f}_{X|H}(x_i|h)$ with its leave-one-out estimator $\hat{f}_{X|H,-i}(x_i|h)$ defined in equation (3.12). Therefore, the bandwidth value h^*_{loo} is:

$$h^*_{loo} = \arg\max_{h>0}\ \tilde{L}(h) = \arg\max_{h>0}\left[\prod_{i=1}^{n}\frac{1}{(n-1)h}\sum_{\substack{j=1\\ j\neq i}}^{n}K(\frac{x_i-x_j}{h})\right] \tag{3.16}$$

Finally, by switching to logarithms to guarantee the feasibility of the maximization process:

$$h^*_{loo} = \arg\max_{h>0}\left\{\sum_{i=1}^{n}\log\left[\sum_{\substack{j=1\\ j\neq i}}^{n}K\left(\frac{x_i-x_j}{h}\right)\right] - n\log\left[(n-1)h\right]\right\} \tag{3.17}$$

3.2.3 Estimate Normalization

As mentioned above, while starvation times are always positive, the Gaussian kernel function is defined over an infinite domain. For this reason the *conditional density function* $\hat{f}_{X|H}(x|h)$ cannot be directly employed in the algorithm *Optimization Phase*. An approximation must be therefore introduced to ensure a null probability of having negative starvation times. In particular, the estimate $\hat{f}_{X|H}(x|h)$ has to be truncated and normalized over the positive domain:

$$\hat{f}^{norm}_{X|H}(x|h) = \begin{cases} \frac{\hat{f}_{X|H}(x|h)}{\int_0^\infty \hat{f}_{T|H}(t|h)\,dt} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3.18}$$

In conclusion, at each algorithm iteration k, once identified the optimal bandwidth h^*_k, the estimated starvation times distribution is:

$$\hat{f}_k(x) = \hat{f}^{norm}_{X|H}(x|h^*_k) \tag{3.19}$$

3.2.4 Numerical Analyses

The scope is to compare method performance in terms of goodness of the estimate and computational effort as functions of dataset size n. Methods are tested on data sets of increasing size (up to $n = 10000$), generated from the three distributions introduced in the Preludio (cf. section P.3). Five independent samples are used as replications of the learning procedure. Therefore, the different estimation methods are tested employing common random numbers. The first case (D1) represents an easy case, whilst others are known to be more difficult from the literature.

Goodness-of-fit is assessed using a Kolmogorov-Smirnov (K-S) test [75] and significance level $\alpha_{K-S} = 0.05$. The comparison is performed creating for each distribution 10 independent samples of 2500 observations each.

D1 (unimodal with IHR). From figure 3.2, it is possible to assess how the behaviors of the different estimation methods are comparable, showing a decreasing trend of h^* which is proportionate to $n^{-1/5}$. This is aligned with Silverman's results in equation (3.9). Method performance is similar in this case and figure 3.3 shows how the different KDEs are overlapping as the observation number n increases. Further, the results of the K-S test (table 3.1) do not show any significant difference. It is possible to conclude that unimodal IHR distributions with low densities near the origin can be easily estimated by all methods, even though Silverman Rule of Thumb provides the smoothest trend of h^* over n.

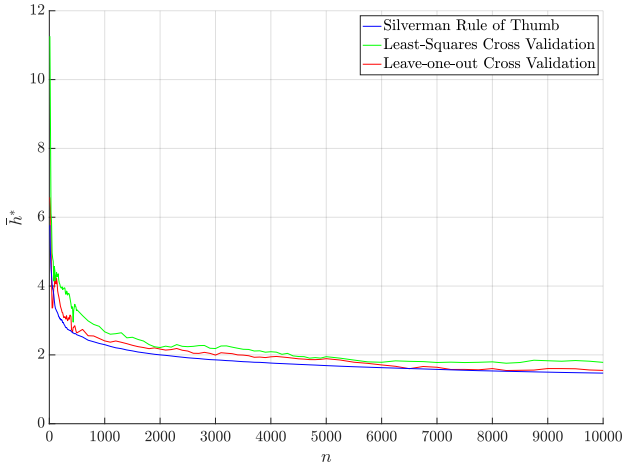
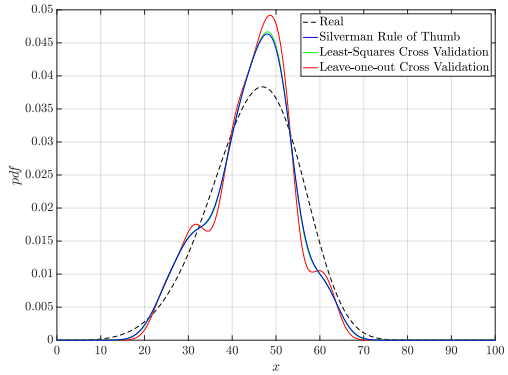
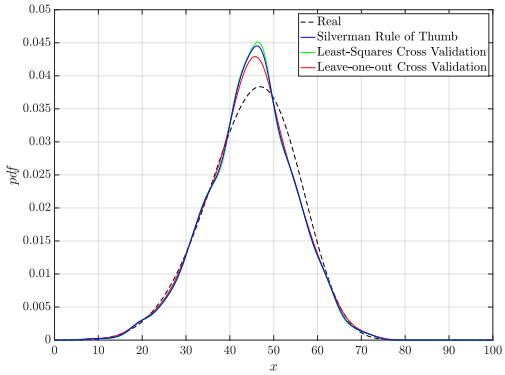


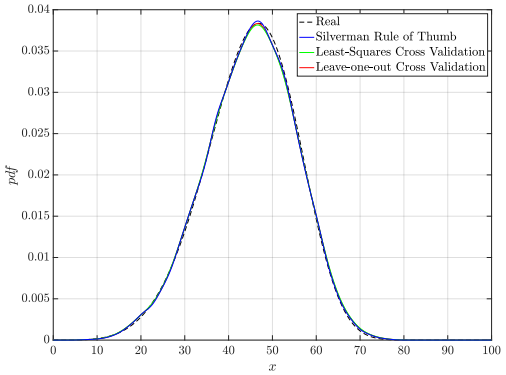
Figure 3.2: D1, $t_a = 45\ s$ - Selected bandwidth h^* for different methods as a function of the number of observations n (mean of 5 replications).



(a) KDEs for $n = 100$.



(b) KDEs for $n = 1000$.



(c) KDEs for $n = 10000$.

Figure 3.3: D1, $t_a = 45\ s$ - Comparison of KDEs for increasing sample size n (1 replication).

Table 3.1: D1, $t_a = 45$ s - K-S results as mean probability of acceptance as n increases ($\alpha_{K-S} = 0.05$, 10 replications).

Method	Observation number n				
	100	500	1000	5000	10000
Silverman Rule of Thumb	0.10	0.48	0.60	0.86	0.88
Least Squares Cross Validation	0.12	0.40	0.46	0.82	0.90
Leave-One-Out Cross Validation	0.10	0.50	0.60	0.84	0.88

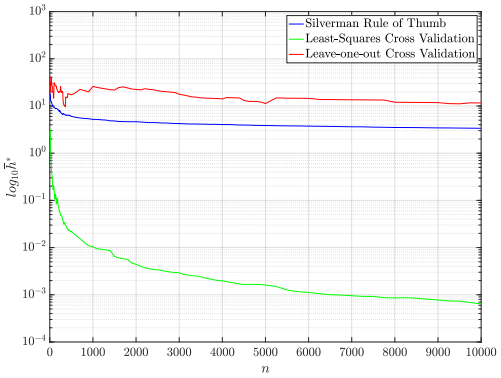
D2 (unimodal with DHR). Due to the asymptote in $x = 0$, the distribution is extremely difficult to handle. Figure 3.4a shows how methods perform differently: Least-Squares Cross Validation provides too small bandwidths (overfitting), whereas Leave-One-Out Cross Validation and Silverman Rule of Thumb tend to introduce oversmoothing, leading to biased estimates. The results of K-S are not reported because the null hypothesis is rejected in all cases. It is possible to conclude that it might happen that KDE method fails in providing adequate estimates. However, the obtained estimation might be good enough for the purpose of the algorithm *Optimization Phase*.

D3 (multimodal). This distribution is characterized by high probability in the neighborhoods of the origin, which makes the estimate more tricky. Figure 3.5 shows how Leave-One-Out and Least-Squares Cross Validation methods have overfitting issues. On the contrary, the bandwidth selected by Silverman Rule of Thumb has oversmoothing issues, especially for small sample sizes. From results of K-S test in table 3.2, the null hypothesis with Silverman Rule of Thumb is always rejected.

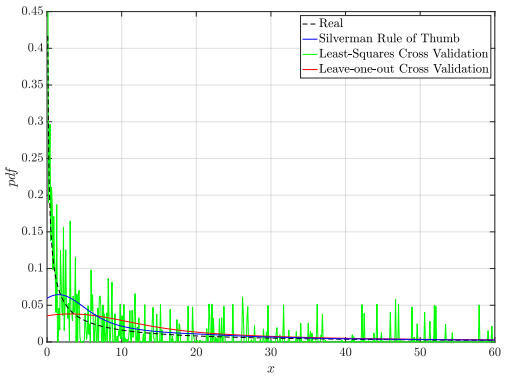
Table 3.2: D3 - K-S results as mean probability of acceptance as n increases ($\alpha_{K-S} = 0.05$, 10 replications).

Method	Observation number n				
	100	500	1000	5000	10000
Silverman Rule of Thumb	0.00	0.00	0.00	0.00	0.00
Least Squares Cross Validation	0.00	0.18	0.28	0.72	0.76
Leave-One-Out Cross Validation	0.00	0.10	0.10	0.74	0.74

Computational effort. Results are obtained with Matlab R2018b on a MacBook with 2.4 GHz Intel Core i5 and 8 GB 1600 MHz DDR3 of RAM.

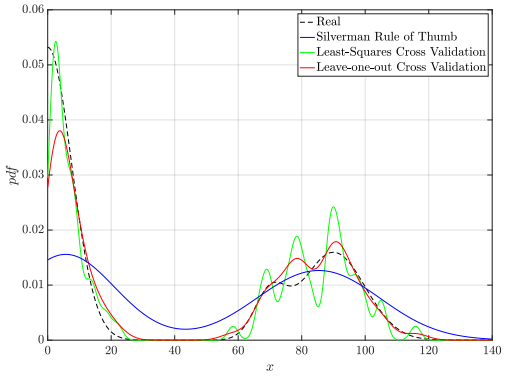


(a) Selected bandwidth h^* by increasing sample size n (mean of 5 replications).

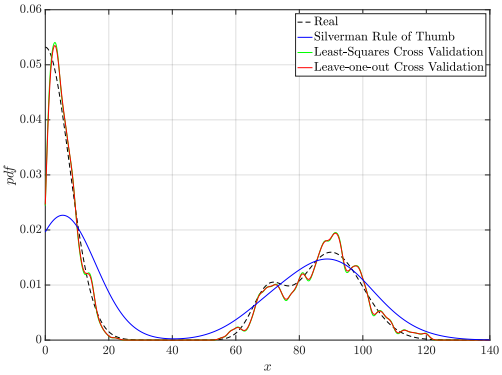


(b) KDEs obtained with $n = 10000$ (1 replication).

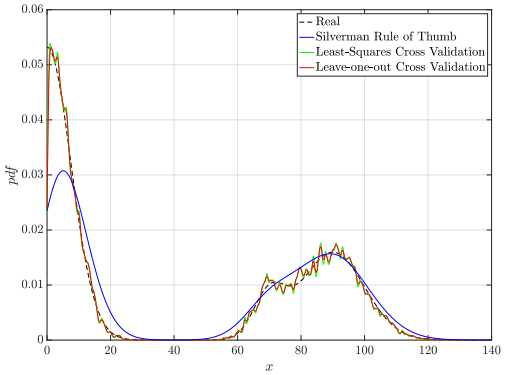
Figure 3.4: D2, $t_a = 39\ s$ - Results of comparison.



(a) KDEs for $n = 100$.



(b) KDEs for $n = 1000$.



(c) KDEs for $n = 10000$.

Figure 3.5: D3 - Comparison of KDEs for increasing sample size n (1 replication).

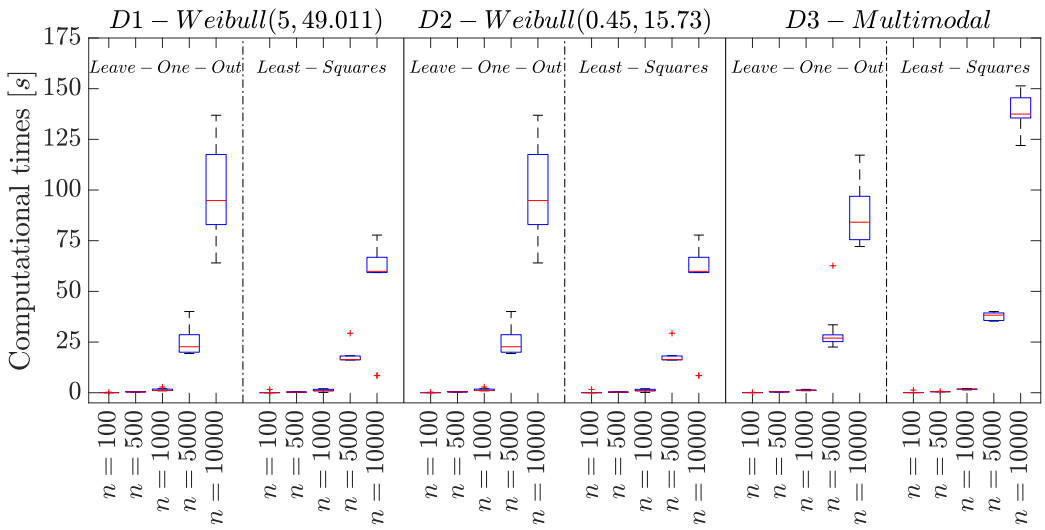


Figure 3.6: Computational time required by bandwidth selection methods as the sample size n increases (boxplots are obtained over 10 sample paths).

Computational times are reported in figure 3.6, showing a significant increment as the sample size increases. Silverman's method requires a negligible amount of time due to the simple computation of the method.

3.2.5 Remarks

First of all, Silverman Rule of Thumb ensures the best trade-off between computational times and results stability, especially when dealing with large sample sizes. However, this methods has significant oversmoothing problems with multimodal distributions, being a robust extension of an approach oriented to normal densities. In multimodal cases, which are common in industrial applications, it is necessary to switch towards Least-Squares or Leave-One-Out Cross Validation. The obtained results are coherent with the existing literature regarding KDE. In particular, it is proven that Least Squares Cross Validation is very sensitive to the presence of clustering in the data, resulting in very small smoothing parameters [72]. On the contrary, it is known that Leave-One-Out Cross Validation produces smoothing parameters that are too large if the underlying distribution has a long tail or if outliers are present [68]. This explains why both methods provide poor results with the DHR pdf (D2). This distribution is in fact characterized by high densities in the neighborhoods of the origin and a long tail towards infinity.

On the basis of the proposed sensitivity analysis, it results that no bandwidth estimation method can be singly implemented in a black-box version

of the algorithm *Learning Phase*. Therefore the following suggestions are proposed for practical application:

- For small sample sizes ($n < 1000$) execute in parallel Leave-One-Out Cross Validation and Silverman Rule of Thumb, since the computational requirements are negligible. Use h^*_{loo} to estimate the starvation times distribution $\hat{f}_k(x)$, unless $h^*_{loo} >> h^*_{rot}$. In this case, keep Silverman Rule of Thumb as the only estimation method (e.g. D2).
- When a significant amount of observation is collected ($n \approx 1000$), if $h^*_{rot} \approx h^*_{loo}$ switch to Silverman Rule of Thumb to avoid an increase in computational requirements (e.g. D1). On the contrary, if $h^*_{rot} > h^*_{loo}$ Silverman Rule of Thumb cannot be employed, since it may lead to an oversmoothing issue (e.g. D3).

3.3 Optimization Phase

This section addresses the second phase of the algorithm, which is aimed at identifying the optimal control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ on the basis of the estimated starvation times pdf $\hat{f}_k(x)$. Once introduced the *on-line optimization problem* which needs to be addressed at each iteration k, a solving algorithm to identify its exact solution is proposed. Finally, numerical examples are reported to show how the optimal control changes according to different *throughput constraint* settings.

3.3.1 On-line Optimization Problem

At each algorithm iteration k, once identified the starvation times pdf $\hat{f}_k(x)$ as in equation (3.19), the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ result from the resolution of the *on-line optimization problem*. Even though this problem has the same structure of its off-line counterpart (cf. subsection P.2.2), the formulation is based on the estimated starvation times pdf $\hat{f}_k(x)$. In particular, denoting respectively $\hat{g}_k(\boldsymbol{\tau})$ and $\hat{\theta}_k(\boldsymbol{\tau})$ the *expected value of the energy consumed in a cycle by the machine* and the *expected throughput* obtained with $\hat{f}_k(x)$, the optimization problem becomes:

$$\begin{aligned}
\hat{\boldsymbol{\tau}}_{\boldsymbol{k}} = \quad & \arg\min_{\boldsymbol{\tau}} \ \hat{g}_k(\boldsymbol{\tau}) && (3.20)\\
\text{Subject to:} \quad & \hat{\theta}_k(\boldsymbol{\tau}) \geq (1-\epsilon)\hat{\theta}_{AON,k} && (3.21)\\
& \tau_{on} > \tau_{off} && (3.22)\\
& \tau_{off}, \tau_{on} \in \mathbb{R}^+_0 && (3.23)
\end{aligned}$$

Since the estimate $\hat{f}_k(x)$ is asymptotically unbiased (cf. subsection 3.2.1), as $n \to \infty$ the control problem approaches that in equations (P.14)-(P.17), resulting in $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}} \to \boldsymbol{\tau}^*$.

3.3.2 Solving Algorithm

In literature the *on-line optimization problem* is slightly simplified. In particular, the *throughput constraint* in equation (3.21) is relaxed and a fictitious power absorption is tuned to penalize the *expected waiting time* in the objective function. Nevertheless, this approach leads to approximate results, since the constraint is not explicitly formulated.

Therefore, a new solving algorithm is proposed to identify the exact solution of the *on-line optimization problem*, which also guarantees a significant response time improvement. In fact, the *Optimization Phase* strongly impacts the whole algorithm computational requirements. Since this solving algorithm can be indifferently applied to any starvation times distribution, it is described considering a generic pdf $f_X(x)$, whose related optimal parameters are $\boldsymbol{\tau}^*$.

Unconstrained Optimization Problem

In order to illustrate the developed solving algorithm, the *unconstrained optimization problem*, obtained by relaxing the *throughput constraint*, is firstly addressed. The core idea is to efficiently identify all the potential objective function minima over the domain of feasible solutions and select the one associated with the lowest energy consumption. The algorithm is decomposed in four steps, exploiting the structural properties of $g(\boldsymbol{\tau})$.

Step U1. Identify the local minima $\tilde{\tau}_{off,i}$ of $g(\{\tau_{off}, \cdot\})$ among the solutions of:

$$\frac{\partial g(\boldsymbol{\tau})}{\partial \tau_{off}} = 0 \tag{3.24}$$

and collect them in set $\mathcal{T}_{off}$. Add $\tau_{off} = 0$ to the set to account for potential solutions on the domain boundary.

Step U2. Identify the local minima $\tilde{\tau}_{on,j}$ of $g(\{\cdot, \tau_{on}\})$ among the solutions of:

$$\frac{\partial g(\boldsymbol{\tau})}{\partial \tau_{on}} = 0 \tag{3.25}$$

and collect them in set $\mathcal{T}_{on}$.

Step U3. Since control parameters are independent (cf. theorem P.1), collect in set $\mathcal{D}_{unc}$ all the combinations $\{\tilde{\tau}_{off,i}, \tilde{\tau}_{on,j}\}$ with $\tilde{\tau}_{off,i} \in \mathcal{T}_{off}$, $\tilde{\tau}_{on,j} \in \mathcal{T}_{on}$ and $\tilde{\tau}_{on,j} > \tilde{\tau}_{off,i}$. Solution $\{\infty, \infty\}$ represents the always on policy and must be included in the set of candidates.

Step U4. Select, among the candidate solutions $\in \mathcal{D}_{unc}$, the combination associated with the lowest energy consumption.

This solving algorithm requires less computational effort than the numerical one-step calculation of $\boldsymbol{\tau}^*$, since the optimal parameters are identified independently. In addition, it can be effectively employed with any kind of starvation times distribution, even a multimodal one. The latter is in fact frequently encountered in industrial applications, especially when a machine is inserted in a production line.

Constrained Optimization Problem

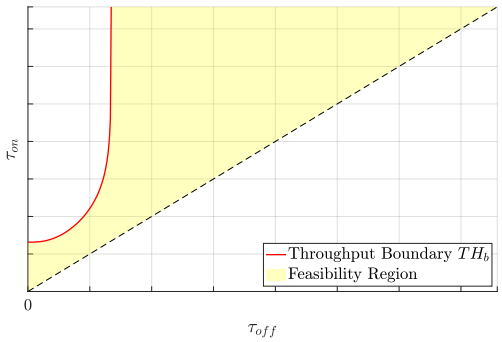
The complete solving algorithm for the *constrained optimization problem* is now addressed. Since control parameters are independent, the core idea is to select the minimum among:

1. The potential solutions of the *unconstrained optimization problem* (i.e. $\in \mathcal{D}_{unc}$), which satisfy the *throughput constraint*.
2. The constrained minimum of $g(\boldsymbol{\tau})$ along the set of points which satisfy the *throughput constraint* at equality, referred to as *throughput boundary* TH_b. With reference to equation (P.18), the *throughput boundary* can be expressed as:

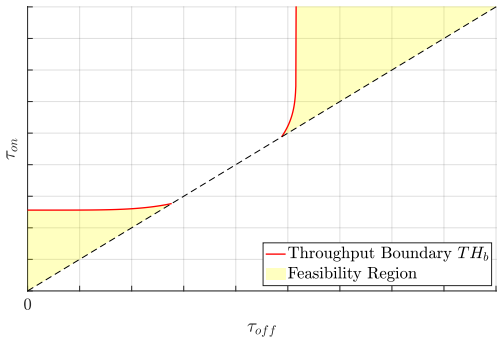
$$TH_b = \{\boldsymbol{\tau} \in \mathbb{R}^2 \mid H(\boldsymbol{\tau}) = H_{max},\ \tau_{on} > \tau_{off} \geq 0\} \tag{3.26}$$

   where H_{max} is an upper bound for the *expected waiting time* $H(\boldsymbol{\tau})$.

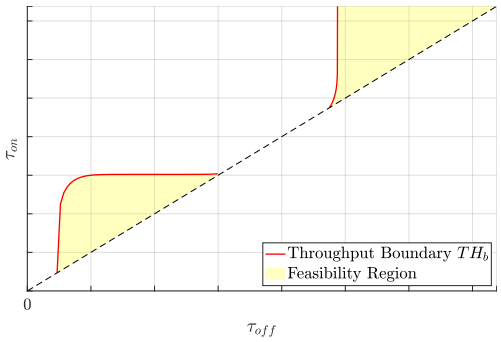
Before outlining the solving algorithm, the *throughput boundary* TH_b needs to be analyzed in more details. In particular, its representation on a (τ_{off}, τ_{on}) plane is generally given by a curve, whose characteristic vary significantly according to the considered *scenario*. Of course, all the points that lie above the TH_b do not satisfy the *throughput constraint*. From the qualitative examples in figure 3.7 it is possible to notice how the TH_b starts either from side $\tau_{off} = 0$ or $\tau_{on} = \tau_{off}$ and it is always characterized by a vertical asymptote. In addition, it can either be continuously defined or intersect one or several times side $\tau_{on} = \tau_{off}$, resulting in a piecewise curve.



(a) Continuous and starting from $\tau_{off} = 0$.



(b) Piecewise and starting from $\tau_{off} = 0$.



(c) Piecewise and starting from $\tau_{on} = \tau_{off}$.

Figure 3.7: Different shapes for the *throughput boundary* TH_b.

It is now possible to illustrate the solving algorithm, which is decomposed in four steps here outlined.

Step C1. Relaxing the constraint, define set $\mathcal{D}_{unc}$ accordingly to the previously detailed algorithm (cf. steps U1-U3).

Step C2. Conforming to property P.2:

- If $t_{su} \leq H_{max}$, the *throughput constraint* does not limit the domain. Therefore proceed with step U4 to find the solution.
- Otherwise, identify the combinations $\in \mathcal{D}_{unc}$ that satisfy the *throughput constraint* and collect them in set $\mathcal{D}_{con}$:

$$\mathcal{D}_{con} = \{\boldsymbol{\tau} \in \mathcal{D}_{unc} \mid H(\boldsymbol{\tau}) \leq H_{max}\} \tag{3.27}$$

Step C3. Identify the constrained minimum $\boldsymbol{\tau}_{\boldsymbol{b}}^*$ of $g(\boldsymbol{\tau})$ along the *throughput boundary* TH_b and add it to set $\mathcal{D}_{con}$. In particular, solve the following minimization problem:

$$\begin{aligned} \boldsymbol{\tau}_{\boldsymbol{b}}^* = \quad & \arg\min_{\boldsymbol{\tau}} \; g(\boldsymbol{\tau}) && (3.28)\\ \text{Subject to:} \quad & H(\boldsymbol{\tau}) = H_{max} && (3.29)\\ & \tau_{on} > \tau_{off} && (3.30)\\ & \tau_{off}, \tau_{on} \in \mathbb{R}_0^+ && (3.31) \end{aligned}$$

Since it is not possible to derive a closed form expression that links the control parameters such that $H(\boldsymbol{\tau}) = H_{max}$, a two-variables minimization problem must be handled.

Step C4. Select, among the candidate solutions $\in \mathcal{D}_{con}$, the combination associated with the lowest energy consumption.

If $\boldsymbol{\tau}_{\boldsymbol{b}}^*$ turns out to be the optimal solution, the *throughput constraint* is said to be *binding*, since $H(\boldsymbol{\tau}^*) = H_{max}$. On the contrary, the *throughput constraint* is *non-binding* because it holds as a strict inequality.

3.3.3 Numerical Analyses

The effectiveness of the proposed solving algorithm is tested with different *scenarios*, generated combining machine M1 with the starvation times distributions defined in subsection P.3. A progressively tighter *throughput constraint* is introduced to assess how it affects the optimal parameter selection and the related machine performance. In the proposed examples the *throughput constraint* always turns out to be *binding*, anyway this is not always the case. Complete results are shown in table 3.3.

Scenarios M1/D1/ϵ. Distribution D1 is unimodal with IHR therefore, according to theorem P.2, the optimal solution of the *unconstrained optimization problem* (i.e. $\epsilon = 1$) lies on the τ_{on} axis: it is optimal to switch off the machine at the beginning of each *cycle*. Figures 3.8a and 3.8b show that the optimal solution of the *constrained optimization problem* still lies on the τ_{on} axis, but the switch on command must be anticipated as the lower bound for $\theta(\boldsymbol{\tau})$ is progressively increased. In fact, switching on the machine in advance allows to reduce the probability that an incoming part finds the machine in the standby or startup state, thus reducing the *expected waiting time*.

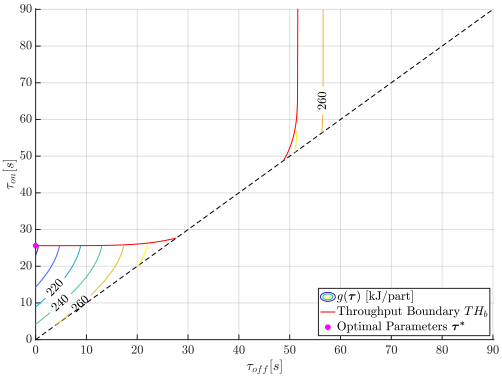
Table 3.3: Optimal control $\boldsymbol{\tau}^*$ for different *scenarios*. Related energy and throughput performance are also given. *Scenarios* with $\epsilon = 1$ are reported for completeness.

Scenario	$\boldsymbol{\tau}^*$ $[s]$	$g(\boldsymbol{\tau}^*)$ $[kJ/part]$	$\Delta g(\boldsymbol{\tau}^*)$ [%]	$\theta(\boldsymbol{\tau}^*)$ $[parts/h]$	$\Delta\theta(\boldsymbol{\tau}^*)$ [%]
$M1/D1/1$	$\{0, 29.18\}$	175.94	26.92	16.22	-4.04
$M1/D1/0.03$	$\{0, 25.61\}$	177.21	26.39	16.39	-3
$M1/D1/0.01$	$\{0, 15.89\}$	195.23	18.91	16.73	-1
$M1/D2/1$	$\{15.91, \infty\}$	121.93	41.19	16.70	-4.07
$M1/D2/0.03$	$\{29.16, \infty\}$	125.04	39.69	16.89	-3
$M1/D2/0.01$	$\{113.71, \infty\}$	155.96	24.78	17.24	-1
$M1/D3/1$	$\{11.65, 72.87\}$	153.80	37.49	16.29	-3.18
$M1/D3/0.02$	$\{13.04, 65.48\}$	156.32	36.47	16.49	-2
$M1/D3/0.01$	$\{14.91, 57.56\}$	165.66	32.68	16.65	-1

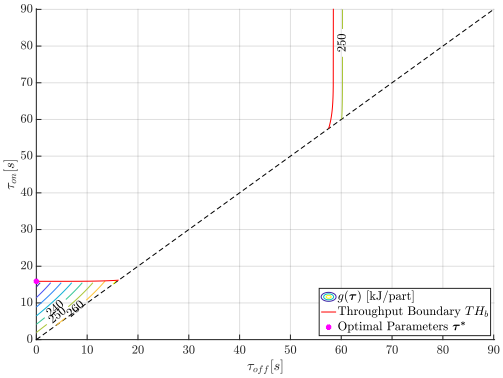
Scenarios M1/D2/ϵ. Distribution D2 is unimodal with DHR therefore, according to theorem P.2, the optimal solution of the *unconstrained optimization problem* is of the form $\boldsymbol{\tau}^* = \{\tau^*_{off}, \infty\}$: once triggered the machine to the standby state it is optimal to switch it on only at part arrival. Figures 3.8c and 3.8d show that optimal solution of the *constrained optimization problem* has the same formulation, even though the switch off command must be delayed as ϵ is progressively reduced.

Scenarios M1/D3/ϵ. Distribution D3 is multimodal and the optimal solution of the *unconstrained optimization problem* lies inside the domain. In particular, the optimal control suggest to switch off the machine in correspondence of the valley between the two peaks, since the probability of a part arrival is low. Figures 3.8e and 3.8f show that the introduction of a *throughput constraint* modifies both the control parameters, progressively decreasing the gap among them. In fact, reducing the machine standby period in correspondence of the pdf valley leads to a lower *expected waiting time*.

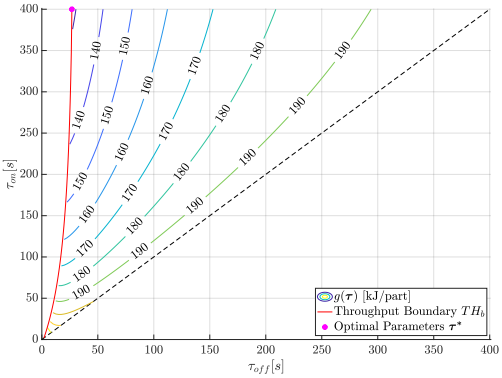
A note on implementation. In order to identify the optimal control for each *scenario*, the proposed solving algorithm is implemented on Matlab R2018b. The solutions of equations (3.24) and (3.25) are identified with function *fzero* (bisection, secant and inverse quadratic interpolation methods). On the contrary, the two-variables minimization problem in equations (3.28)-(3.31) is solved with function *fmincon* (gradient-based method).



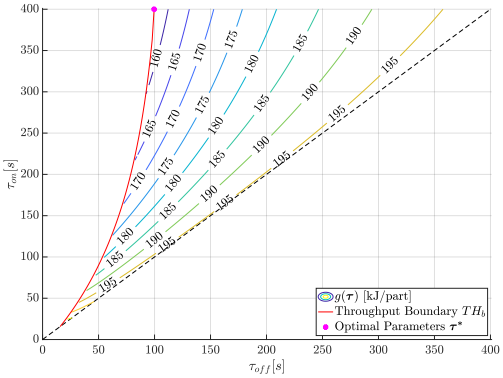
(a) M1/D1/0.03 - $\boldsymbol{\tau}^* = \{0, 25.61\}$.



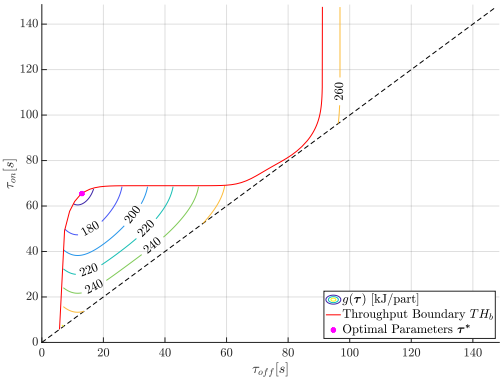
(b) M1/D1/0.01 - $\boldsymbol{\tau}^* = \{0, 15.89\}$.



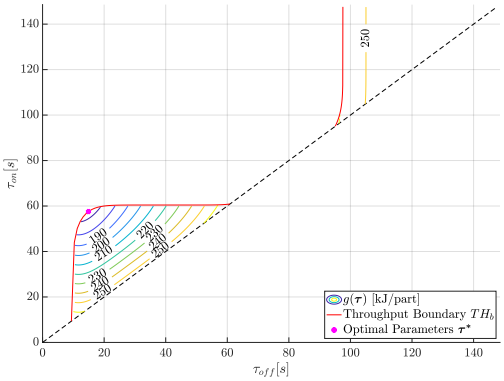
(c) M1/D2/0.03 - $\boldsymbol{\tau}^* = \{29.16, \infty\}$.



(d) M1/D2/0.01 - $\boldsymbol{\tau}^* = \{113.71, \infty\}$.



(e) M1/D3/0.02 - $\boldsymbol{\tau}^* = \{13.04, 65.48\}$.



(f) M1/D3/0.01 - $\boldsymbol{\tau}^* = \{14.91, 57.56\}$.

Figure 3.8: Contour plots of $g(\boldsymbol{\tau})$ for different *scenarios*.

3.4 Implementation Phase

This section describes the third phase of the algorithm, which ensures that a change in the control parameters occurs only in presence of a statistically evident machine energy performance improvement. In particular, at each iteration k the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ are compared with the ones currently in place $\boldsymbol{\tau}_{\boldsymbol{impl}}$ and the best configuration is selected. The *Implementation Phase* serves two different purposes:

1. It allows to avoid an increase of energy consumptions resulting from implementing biased control parameters, i.e. estimated on the basis of a low number of observations.

2. It allows to avoid that control parameters are changed too frequently overtime. In fact, the variability induced in the downstream production process should be possibly minimized.

In literature, $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ and $\boldsymbol{\tau}_{\boldsymbol{impl}}$ are compared in terms of objective function employing a paired-t test. Nevertheless, its normality assumption is generally far from being satisfied.

Therefore, an alternative policy for implementing the control parameters, i.e. the *Bootstrap CI Implementation Policy*, is developed in order to guarantee a greater robustness. Although the core structure remains unchanged, this policy employs a bias corrected and accelerated (BC_a) bootstrap confidence interval (CI) in place of the traditional t-interval. In addition, some secondary adjustments are proposed to facilitate parameters implementation under some favorable conditions. Finally, an extension is provided to guarantee parameters convergence when a constraint must be satisfied.

In order to address the proposed policy, it is necessary to introduce and describe the *energy output function* (3.32) and the *implementation cost function* (3.34).

3.4.1 Energy Output Function

Let random variable E be *machine energy consumption in a cycle* and denote its realization with e. The *energy output function* $e = \phi(x, \boldsymbol{\tau})$ links the occurrence x of starvation time X to the resulting consumption e, when the machine is controlled with parameters $\boldsymbol{\tau}$:

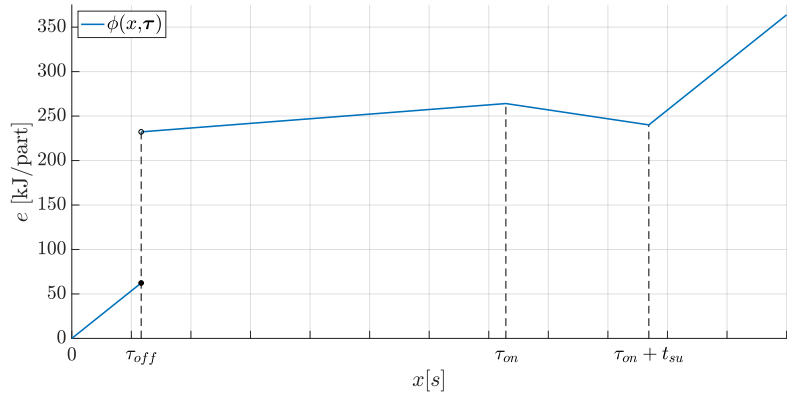


Figure 3.9: General *energy output function* for machine M1 ($\tau_{on} > \tau_{off} > 0$).

$$\phi(x,\boldsymbol{\tau}) = \begin{cases} w_2 x & \text{if } x \leq \tau_{off} \\ w_2\tau_{off} + w_1(x-\tau_{off}) + (w_3+w_q)t_{su} & \text{if } \tau_{off} < x \leq \tau_{on} \\ w_2\tau_{off} + w_1(\tau_{on}-\tau_{off}) + w_3 t_{su} + \\ \quad + w_q(\tau_{on}+t_{su}-x) & \text{if } \tau_{on} < x \leq \tau_{on}+t_{su} \\ w_2\tau_{off} + w_1(\tau_{on}-\tau_{off}) + w_3 t_{su} + \\ \quad + w_2(x-\tau_{on}-t_{su}) & \text{if } x > \tau_{on}+t_{su} \end{cases} \tag{3.32}$$

For ease of understanding, figure 3.9 shows a general $\phi(x,\boldsymbol{\tau})$ for machine M1:

- $x \leq \tau_{off}$ – $\phi(x,\boldsymbol{\tau})$ has the steepest slope w_2, since the machine never switches from the idle state.
- $\tau_{off} < x \leq \tau_{on}$ – It is possible to notice a discrete step, which accounts for the energy required during the startup ($w_3 t_{su}$) and the whole part holding energy ($w_q t_{su}$). $\phi(x,\boldsymbol{\tau})$ increases linearly with slope w_1, since the machine is switched to the standby state.
- $\tau_{on} < x \leq \tau_{on}+t_{su}$ – $\phi(x,\boldsymbol{\tau})$ decreases linearly with x, since the part holding energy is progressively reduced. In particular, for $x = \tau_{on}+t_{su}$ it results $\phi(\tau_{on}+t_{su},\boldsymbol{\tau}) = \phi(\tau_{on},\boldsymbol{\tau}) - w_q t_{su}$, as the part can be processed immediately avoiding any waiting time.
- $x > \tau_{on}+t_{su}$ – $\phi(x,\boldsymbol{\tau})$ gets back to increase with slope w_2, since the machine has resumed the operational readiness.

The *energy output function*, also allows to derive a more compact expression for the *expected value of the energy consumed in a cycle by the machine* $g(\boldsymbol{\tau})$:

$$g(\boldsymbol{\tau}) = \mathbb{E}[E] = \mathbb{E}[\phi(X,\boldsymbol{\tau})] = \int_0^\infty \phi(x,\boldsymbol{\tau}) f_X(x)\, dx \tag{3.33}$$

In the followings, two examples are introduced to analyze how the *energy output function* $\phi(x, \boldsymbol{\tau})$ changes according to the considered control parameters $\boldsymbol{\tau}$.

Example 1. Considering *scenario* M1/D1/1, figure 3.10 shows the comparison among the *energy output function* of the AON policy $\phi(x, \{\infty, \infty\}) = w_2 x$ and the one evaluated for $\boldsymbol{\tau}^* = \{0, 29.18\}$. In particular, the optimal control policy improves the machine energy performance for $x > \tilde{x}$. Nevertheless, there is a 14.3% probability that an incoming part incurs in higher energy consumptions. In addition, it is worth highlighting how the *energy output function* section characterized by the lowest consumptions is located in correspondence of the pdf mode. Finally, for $x > \tau_{on} + t_{su}$, the gap between $\phi(x, \{\infty, \infty\})$ and $\phi(x, \boldsymbol{\tau}^*)$ is constant, since machine operational readiness is resumed.

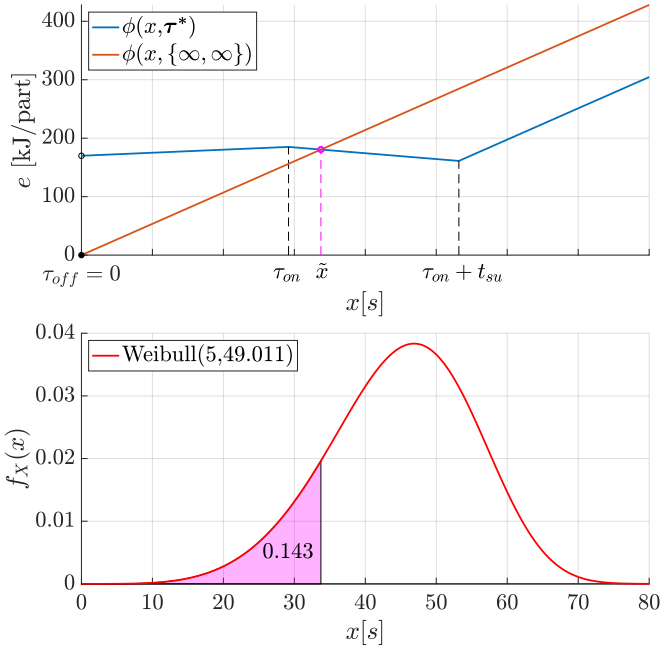
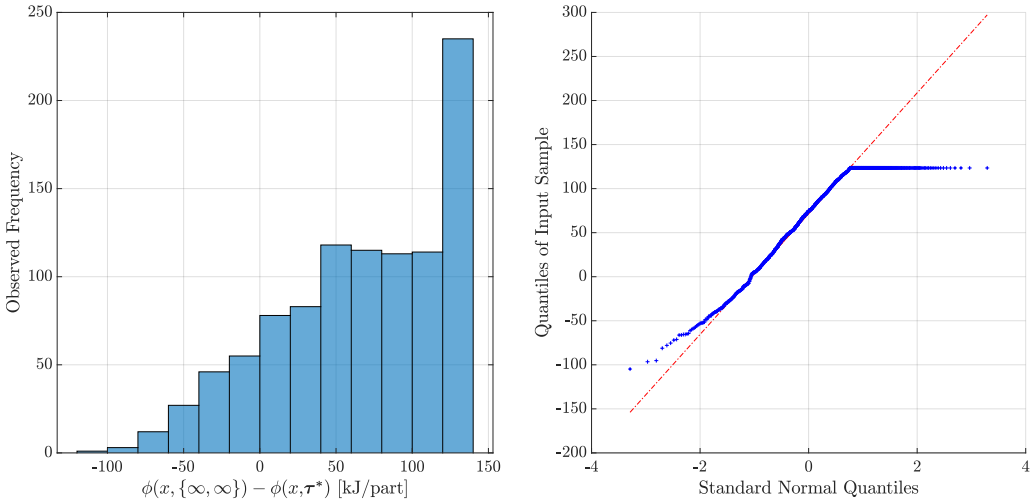


Figure 3.10: M1/D1/1 - $\phi(x, \boldsymbol{\tau}^*)$ and $\phi(x, \{\infty, \infty\})$.

Starting from a set of 1000 observations, it is possible to analyze the paired differences distribution $\phi(x_i, \{\infty, \infty\}) - \phi(x_i, \boldsymbol{\tau}^*)$. In particular, figure 3.11a clearly shows how the considered sample is neither symmetric nor, consequently, normally distributed. This is further confirmed by the QQ-Plot in figure 3.11b. It results that the normality hypothesis of the paired-t test employed in literature is difficult to be satisfied.



(a) Frequency histogram.

(b) QQ-Plot of sample data vs standard normal.

Figure 3.11: M1/D1/1 - Paired differences distribution (1000 samples).

Example 2. Figure 3.12 compares the *energy output function* of machine M1 for two generic control parameters ($\boldsymbol{\tau_1}$ and $\boldsymbol{\tau_2}$), considering the particular eventuality in which $\tau_{off_1} = \tau_{off_2} \neq 0$ and $\tau_{on_2} > \tau_{on_1}$. It is possible to notice how the two functions overlap for $0 \leq x \leq \tau_{on_1}$. This peculiarity has to be considered in the *Implementation Phase*, since the energy performance coincide for a set of starvation times values.

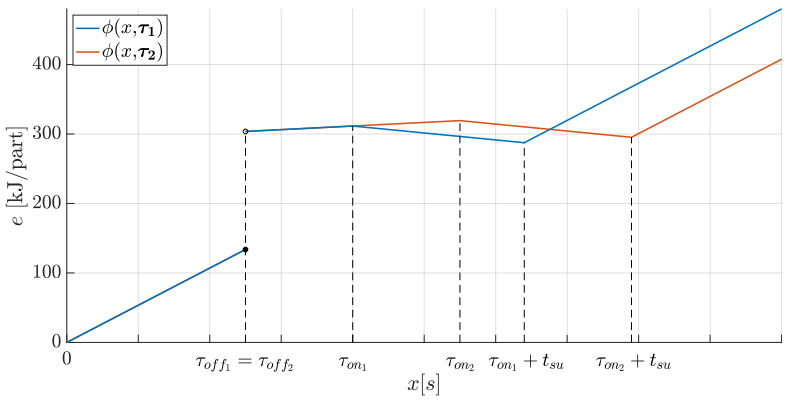


Figure 3.12: $\phi(x, \boldsymbol{\tau_1})$ and $\phi(x, \boldsymbol{\tau_2})$ for M1.

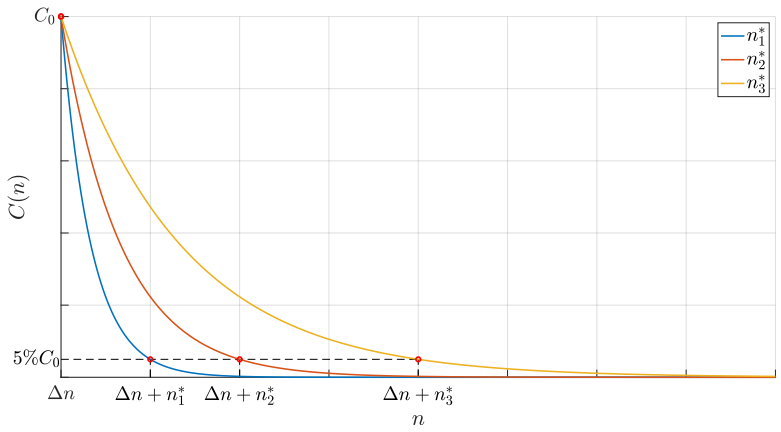


Figure 3.13: *Implementation cost function* for $n_1^* < n_2^* < n_3^*$.

3.4.2 Implementation Cost Function

The proposed policy considers an implementation cost C_0, introduced to account for the risk aversion of manufacturers and to reduce the number of changes in the applied control. In addition, this parameter is progressively discounted to account for the greater confidence on the estimates, resulting from an increasing number of observations. Therefore, at each iteration k, the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ are implemented only if they statistically guarantee a surplus of energy saving greater than the corresponding *implementation cost function* $C(n)$ value. In particular, this function is shaped on the basis of a negative exponential. In order to ensure that the implementation cost C_0 is wholly considered at the first algorithm iteration ($k = 1$ and $n = \Delta n$), it is necessary to set $C(\Delta n) = C_0$. This leads to the following expression for the *implementation cost function*:

$$C(n) = C_0 e^{-\frac{n-\Delta n}{\iota}} \quad \text{with} \quad n \geq \Delta n \tag{3.34}$$

Finally, in order to select a value for the constant ι, it is required to force the function to pass for a point of predefined coordinates. For example, ι can be derived by specifying the observation interval n^* required by the function to reach approximately the 5% of its initial value. In fact, $y = 5\%$ is conveniently reached for $n^* = n - \Delta n = 3\iota$. Therefore, the constant ι can be expressed as:

$$\iota = \frac{n^*}{3} \tag{3.35}$$

The definitive shape of the *implementation cost function* $C(n)$ for different values of n^* is reported in figure 3.13. Of course a higher n^* implies higher energy saving requirements for implementing the control parameters.

3.4.3 Bootstrap CI Implementation Policy

The *Bootstrap CI Implementation Policy* is now outlined. In particular, it consists of five steps to be performed in sequence at each algorithm iteration k.

Step 1. Calculate the paired differences D_i:

$$D_i = \phi(x_i, \boldsymbol{\tau_{impl}}) - \phi(x_i, \hat{\boldsymbol{\tau}}_{\boldsymbol{k}}) \quad i = 1, \ldots, n \tag{3.36}$$

and gather them in set $\boldsymbol{D} = \{D_i, \ldots, D_n\}$.

Step 2. Collect the non-zero paired differences in set $\boldsymbol{D_0}$:

$$\boldsymbol{D_0} = \{D_i \in \boldsymbol{D} \mid D_i \neq 0\}. \tag{3.37}$$

If the number of non-zero paired differences is greater than or equal to a safety threshold ST (i.e. if $|\boldsymbol{D_0}| \geq ST$), then set $\boldsymbol{D} = \boldsymbol{D_0}$.

Step 3. Formulate the null hypothesis H_0:

$$H_0 \ : \ Mean(\boldsymbol{D}) \leq C(n) \tag{3.38}$$

where $C(n)$ is the *implementation cost function* from equation (3.34).

Step 4. Test the null hypothesis, define a $(1-\alpha)\%$ BC_a confidence interval for the paired differences mean μ, following the procedure here hinted.

4.1) Starting from the empirical distribution $\boldsymbol{D}$, create B bootstrap resamples[1].

4.2) For each resample, compute the sample mean $\hat{\mu}^*$.

4.3) Sort the statistics in an increasing order, thus creating the bootstrap distribution of the estimate $\hat{\mu}$:

$$\hat{\mu}^*_{(1)} \leq \hat{\mu}^*_{(2)} \leq \cdots \leq \hat{\mu}^*_{(B)} \tag{3.39}$$

[1]The bootstrap resample of an empirical distribution is an equally sized sample generated by sampling with replacement from the empirical distribution. A single data point can appear multiple times, since at each sampling it has the same probability of being drawn [76].

4.4) Define the confidence interval:

$$CI = \left[\hat{\mu}^*_{(\alpha_1)}, \hat{\mu}^*_{(\alpha_2)}\right] = \left[\hat{\mu}_{low}, \hat{\mu}_{up}\right] \tag{3.40}$$

where α_1 and α_2 depend on the significance level α and on two additional parameters, i.e. the acceleration parameter $\hat{a}$ and the bias-correction factor $\hat{z}_0$, which are autonomously estimated. The exhaustive procedure is described in [77].

This confidence interval is very simple to form and not necessarily symmetric, easily indulging for the distribution skewness.

Step 5. Evaluate the relative positioning of CI with respect to $C(n)$:

- If $\hat{\mu}_{low} \leq C(n)$, keep the current parameters $\boldsymbol{\tau_{impl}}$ (H_0 cannot be rejected with significance level α).
- Otherwise, implement the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$, since there is statistical evidence that the objective function improves (H_0 is rejected with significance level α).

Step 2 is introduced to solve an issue that arises when there is a partial overlapping between the *energy output function* evaluated for $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ and $\boldsymbol{\tau_{impl}}$ (cf. figure 3.12). In particular, if for a large amount of observations $\phi(x_i, \boldsymbol{\tau_{impl}}) - \phi(x_i, \hat{\boldsymbol{\tau}}_{\boldsymbol{k}}) = 0$, it is less likely to reject H_0 and to implement control $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$, even though it would ensure higher energy savings. The proposed solution is to simply exclude the paired differences D_i equal to 0, before applying the test. Nevertheless, in order to ensure that $\boldsymbol{D}$ has a sufficient size to guarantee the test effectiveness, a safety threshold ST on the minimum number of non-zero paired differences must be introduced.

Since the proposed *Bootstrap CI Implementation Policy* only tests the energy performance, it is no longer effective when a constraint needs to be satisfied. Therefore, in this case, a further initial step must be introduced to allow the convergence towards the optimal control:

Step 0. If, on the basis of latest pdf estimate $\hat{f}_k(x)$, the implemented control parameters $\boldsymbol{\tau_{impl}}$ no longer satisfy the constraint, directly implement $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$. The last identified parameters are indeed certainly feasible according to the new observations.

Parameter selection. Before its application, the *Bootstrap CI Implementation Policy* requires the selection of five parameters: C_0, n^*, α, ST and B. In particular, the first three are related to the user risk-adversity. In fact the policy becomes more conservative as C_0 increases, α decreases and n^* increases. In addition, it is suggested to set $ST = 10$ to assure an acceptable sample size for CI construction and $B = 10000$, which is a good trade off between accuracy and execution time.

3.5 Results

In this section, numerical analyses are performed to evaluate the on-line algorithm effectiveness and computational requirements. The machine energy consumption, the throughput, the number of observations before implementing the control and the mean number of control changes are the investigated performance indexes (KPIs). Three different *scenarios* are addressed, generated with the machines and the distributions from subsection P.3. In detail:

- *Scenarios* M1/D1/1 and M2/D2/1 are employed to compare the algorithm performance with that of the existing policy from literature [43, 63].
- *Scenario* M3/D3/0.02 is instead reported to assess the algorithm behaviour when a *throughput constraint* is set.

Moreover, since the *Implementation Phase* parameters strongly impact the investigated KPIs, a sensitivity analysis on their effect is performed. Unfortunately, due to the peculiarities of the analyzed model and the considered simplifying assumptions, additional comparisons with other control policies from literature cannot be carried out.

Experiments are designed by varying the *Implementation Phase* parameters according to the full factorial plans in tables 3.4, 3.8 and 3.9. In detail, $\alpha = 1$ specifies an experiment in which the *Implementation Phase* does not take place and, at each iteration, the estimated parameters are directly implemented. Moreover, $n^* = \infty$ indicates an experiment with a constant *implementation cost function* value $C(n) = C_0 \; \forall \; n$. The number of designed experiments for each *scenario* is 15, because some combinations are equivalent. The full factorial plans are replicated and balanced, since each experiment is composed by 10 replications to achieve satisfactory estimates. In addition, common random numbers are used to guarantee comparability.

Focusing on the single replication, the algorithm is initially iterated every $\Delta n = 10$ observations. Upon completion of the 500th part, this interval is

finally increased to $\Delta n = 50$. Each replication terminates when 5000 parts are processed. Nevertheless, it is unlikely that a real industrial process shows a stationary behaviour in such a long production run.

Finally, note that the results obtained assuming a complete knowledge of the starvation time distributions (off-line problem), referred to as "Real Values", are used as benchmarks. Complete results can be found in appendix A.

3.5.1 Scenario M1/D1/1

In this *scenario*, as reported in table P.2, the AON policy obtains an objective function value of $g_{AON} = 240.75$ kJ/part and an expected throughput of $\theta_{AON} = 16.90$ parts/h. The AON is used as reference policy. The optimal control is $\boldsymbol{\tau}^* = \{0, 29.18\}$, allowing to achieve 26.92% of savings on the objective function, with an expected throughput reduction of -4.04%.

Distribution D1 is unimodal with IHR, therefore Silverman Rule of Thumb is employed in the algorithm *Learning Phase*. In fact, according to section 3.2, it provides the best trade-off between computational times and results stability.

Implementation Phase settings are reported in table 3.4. In particular, the proposed non-zero implementation costs C_0 respectively correspond to 15% and 20% of g_{AON}.

Table 3.4: M1/D1/1 - *Implementation Phase* settings.

Factor	Level 1	Level 2	Level 3
C_0 $[kJ/part]$	0	36	48
α	1	0.20	0.05
n^*	∞	2500	500

Performance comparison and goodness of the estimate. The KPIs obtained with the proposed algorithm are similar to those of the existing policy from literature. From table 3.5 it indeed results that, under the same settings[2] for C_0 and α, the confidence intervals for the energy consumption per part always show some degree of overlap. The same consideration holds for the machine throughput, which is reported in table A.1. Focusing on the *Implementation Phase*, the number of acquired observations before starting the EEC is, with few exceptions, equal (cf. table 3.6). Moreover, when C_0 is non-zero, the control is never changed after the first implementation.

[2]Since the original algorithm does not provide for an implementation cost discount, only the experiments with $n^* = \infty$ should be considered.

Table 3.5: M1/D1/1 - Sample-based comparison of on-line algorithm energy performance for different settings (95% CI, 10 replications). Experiments with $n^* = \infty$ are reported.

C_0 [kJ/part]	α	Algorithm	Energy [kJ/part]		
			$n = 100$	$n = 250$	$n = 500$
0	0.05	Bootstrap CI	182.46 ± 1.15	178.74 ± 0.67	177.40 ± 0.54
		Paired-t	182.29 ± 1.21	178.71 ± 0.80	177.41 ± 0.62
	0.20	Bootstrap CI	182.21 ± 1.16	178.56 ± 0.68	177.30 ± 0.55
		Paired-t	182.22 ± 1.27	178.52 ± 0.68	177.27 ± 0.55
36	0.05	Bootstrap CI	191.01 ± 4.26	182.15 ± 1.69	179.16 ± 0.97
		Paired-t	193.32 ± 3.47	183.19 ± 1.52	179.76 ± 1.03
	0.20	Bootstrap CI	184.25 ± 1.42	179.62 ± 0.69	178.01 ± 0.58
		Paired-t	184.52 ± 1.38	179.94 ± 0.85	178.37 ± 0.78
48	0.05	Bootstrap CI	206.07 ± 7.97	188.92 ± 4.36	182.69 ± 2.37
		Paired-t	205.30 ± 8.60	188.66 ± 4.59	182.53 ± 2.48
	0.20	Bootstrap CI	194.42 ± 6.17	183.69 ± 2.63	180.01 ± 1.58
		Paired-t	194.60 ± 6.04	183.94 ± 2.45	180.29 ± 1.45

Table 3.6: M1/D1/1 - Number of observations collected before implementing the solution. For $C_0 = 0\ kJ/part$ the control is always implemented at the first iteration.

C_0 [kJ/part]	α	Algorithm	Replication									
			1	2	3	4	5	6	7	8	9	10
36	0.05	Bootstrap CI	**10**	30	**10**	20	30	30	20	50	20	30
		Paired-t	**20**	30	**30**	20	30	30	20	50	20	30
	0.20	Bootstrap CI	10	20	10	20	20	10	10	10	10	10
		Paired-t	10	20	10	20	20	10	10	10	10	10
48	0.05	Bootstrap CI	20	40	50	50	30	50	50	120	**30**	40
		Paired-t	20	40	50	50	30	50	50	120	**20**	40
	0.20	Bootstrap CI	20	30	10	30	30	40	10	60	20	30
		Paired-t	20	30	10	30	30	40	10	60	20	30

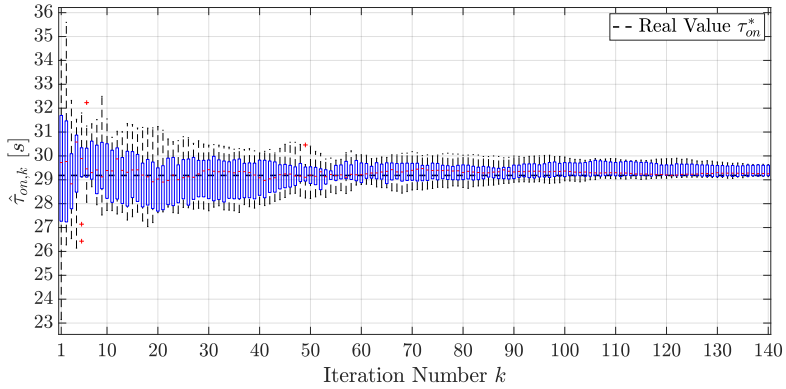
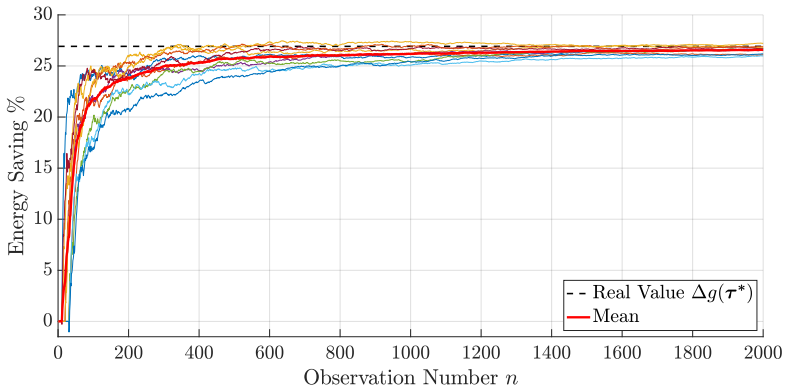


Figure 3.14: M1/D1/1 - Estimated switch on control parameter $\hat{\tau}_{on,k}$ at each algorithm iteration k (box plot of 10 replications).

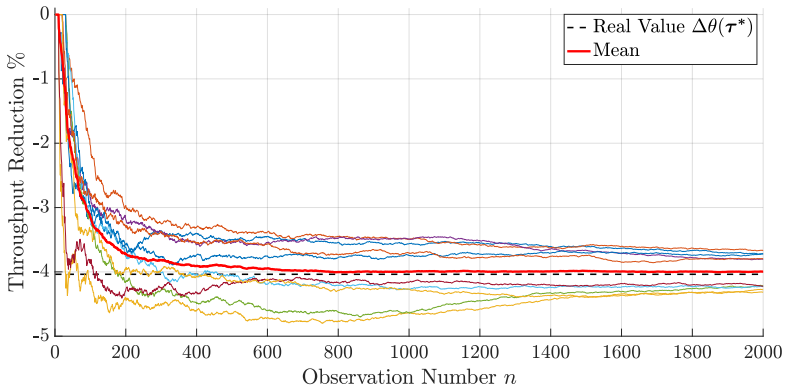
Switching to computational requirements, the proposed algorithm guarantees a significant response time reduction. Even though the burden progressively increases with the amount of observed data, a single iteration is always executed in less than six seconds.

This response time enhancement allows to analyze the long-range convergence of the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ towards the off-line optimal solution $\boldsymbol{\tau}^*$. In particular, figure 3.14 shows how the range of change for the estimated switch on parameter $\hat{\tau}_{on,k}$ gets progressively tighter, approaching the off-line optimal value τ^*_{on} (dashed line). It is worth highlighting that the algorithm identifies the exact value $\tau^*_{off} = 0$ since the first iteration.

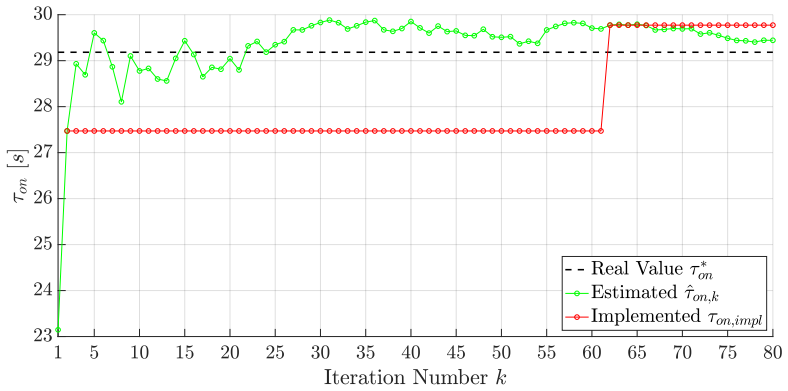
Analysis of a single experiment. In order to clarify the algorithm functioning and to get acquainted with the investigated KPIs, the experiment with $C_0 = 36\ kJ/part$, $\alpha = 0.05$ and $n^* = 500$ is specifically addressed. In particular, figures 3.15a and 3.15b show the sample-based energy saving and throughput reduction with respect to the AON policy: as the number of collected observations n increases, both trends approach the off-line control problem results $\Delta g(\boldsymbol{\tau}^*)$ and $\Delta\theta(\boldsymbol{\tau}^*)$. Focusing on a single replication, the main *Implementation Phase* effects can be instead appreciated from figure 3.15c. In detail, for each algorithm iteration k, it displays the estimated switch on parameter $\hat{\tau}_{on,k}$ (green line) and the implemented one $\tau_{on,impl}$ (red line). It is possible to notice that control implementation is delayed to the second iteration, since the first estimate does not guarantee a statistically evident improvement in energy performance. Moreover, just one additional control change occurs, allowing to approach the off-line optimal value τ^*_{on}.



(a) Sample-based energy saving (10 replications).



(b) Sample-based throughput reduction (10 replications).



(c) Estimated $\hat{\tau}_{on,k}$ (green line) and implemented $\tau_{on,impl}$ (red line) switch on control parameters (1 replication).

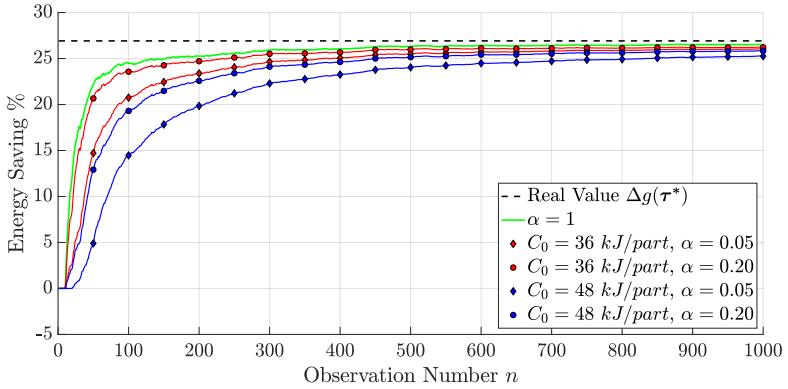
Figure 3.15: M1/D1/1 - Experiment with $C_0 = 36\ kJ/part$, $\alpha = 0.05$ and $n^* = 500$. Only the first 2000 observations are reported, i.e. 80 iterations.

Table 3.7: M1/D1/1 - Effects of factors C_0, α and n^* over the mean number of observations before implementing the control parameters $n_{initial}$ and the mean number of changes.

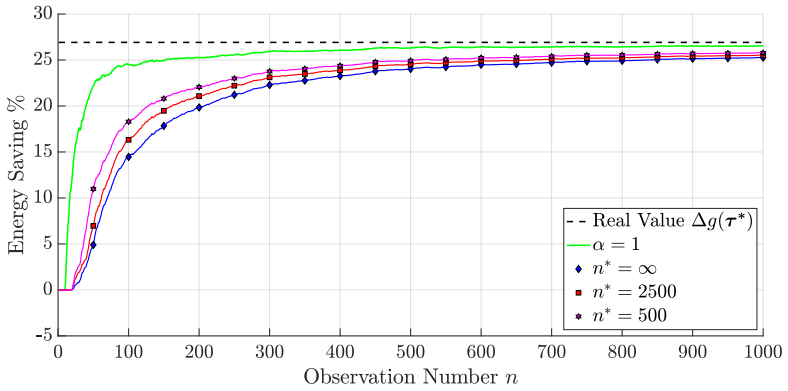
C_0 [kJ/part]	α	n^*	$n_{initial}$	Mean Number of Changes
0	0.05	/	10	2.20
	0.20	/	10	3.10
36	0.05	∞	25	1.0
		2500	25	1.7
		500	21	2.0
	0.20	∞	13	1.0
		2500	13	1.8
		500	13	2.3
48	0.05	∞	48	1.0
		2500	39	1.4
		500	32	1.5
	0.20	∞	28	1.0
		2500	28	1.4
		500	24	2.3

Sensitivity analysis. A sensitivity analysis is now introduced to evaluate how different *Implementation Phase* settings affect the investigated KPIs. Specifically, table 3.7 shows how the control implementation is progressively delayed in time when C_0 increases, α decreases and n^* increases. A similar behavior can be also identified for the occurrence of changes in the control, which decrease when more conservative parameters are set. In addition, once set a significance level α, an upper bound for the mean number of changes can be identified from the experiment characterized by a null implementation cost C_0. The effect of n^* on the mean number of observations before implementing the control $n_{initial}$ is more evident in a situation with low α. On the contrary, n^* has a larger influence on the mean number of control changes when a higher α is set.

Being responsible for the occurrence and the number of control changes, the selected *Implementation Phase* parameters also affect the achievable energy savings. This is clearly evident from the results reported in figure 3.16. In particular, top panel (figure 3.16a) represents four settings where the implementation cost C_0 is not discounted. A situation with high C_0 and/or low α, being more conservative, results into lower energy savings. Bottom panel (figure 3.16b) represents three cases with fixed C_0 and α, but different discounts. A situation with a smaller n^* implies a progressive greater confidence on the estimates, resulting in higher energy savings.



(a) Effects of C_0 and α with $n^* = \infty$.



(b) Effect of n^* with $C_0 = 48\ kJ/part$ and $\alpha = 0.05$.

Figure 3.16: M1/D1/1 - Sample-based mean energy saving for different settings (10 replications). Green line is obtained without the *Implementation Phase*.

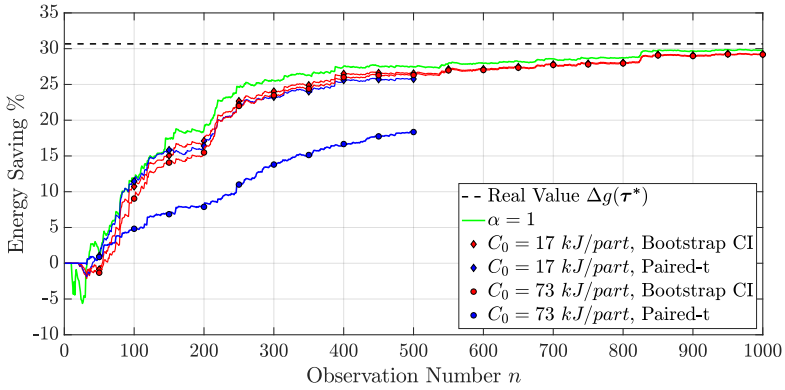
3.5.2 Scenario M2/D2/1

This additional numerical case is introduced to account for a *scenario* in which the proposed algorithm provides better energy performance with respect to the existing policy from literature. In detail, as reported in table P.2, in this *scenario* the AON policy obtains an objective function value of $g_{AON} =$ 428.87 kJ/part and an expected throughput of $\theta_{AON} = 17.39$ parts/h. The optimal control is $\boldsymbol{\tau}^* = \{18.37, \infty\}$, allowing to achieve 31.09% of savings on the objective function, with an expected throughput reduction of -3.82%.

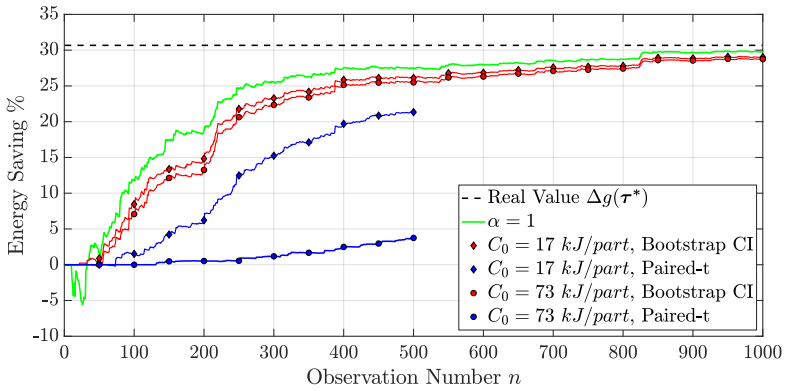
Silverman Rule of Thumb is employed in the algorithm *Learning Phase* and different settings for the *Implementation Phase* are tested, according to the full factorial plan in table 3.8. In particular, the proposed non-zero implementation costs C_0 respectively correspond to 4% and 17% of g_{AON}.

Table 3.8: M2/D2/1 - *Implementation Phase* settings.

Factor	Level 1	Level 2	Level 3
C_0 $[kJ/part]$	0	17	73
α	1	0.35	0.05
n^*	∞	5000	500



(a) Results with $\alpha = 0.35$.



(b) Results with $\alpha = 0.05$.

Figure 3.17: M2/D2/1 - Algorithm comparison in terms of sample-based mean energy saving for different settings (10 replications, $n^* = \infty$). Green line is obtained without the *Implementation Phase*.

Considering $n^* = \infty$, figure 3.17 compares the mean energy savings achieved by the algorithms for different settings of C_0 and α. Specifically, in a situation with low C_0 and high α (figure 3.17a - $C_0 = 17\ kJ/part$, $\alpha = 0.35$) the algorithms perform similarly while, with more conservative settings, the original once faces the implementation issue previously described in subsection 3.4.3 (figure 3.17b - $C_0 = 73\ kJ/part$, $\alpha = 0.05$). In this latter situation, the proposed adjustments effectively allow to obtain a significant improvement in energy performance.

3.5.3 Scenario M3/D3/0.02

A last numerical case is finally employed to analyze the algorithm effectiveness when an upper bound on the machine service level reduction is introduced. Specifically, in this *scenario* the AON policy obtains an objective function value of $g_{AON} = 206.96$ kJ/part and an expected throughput of $\theta_{AON} = 32.73$ parts/h. If a *throughput constraint* is set to limit its maximum admissible reduction to $\epsilon = 2\%$, the optimal control is $\boldsymbol{\tau}^* = \{14.44, 62.20\}$, allowing to achieve 28.62% of savings on the objective function.

Leave-One-Out Cross Validation is employed in the algorithm *Learning Phase*, since Silverman Rule of Thumb has significant oversmoothing problems with distribution D3 (multimodal). Even though the selected method shows slight overfitting issues (cf. subsection 3.2.4), it provides suitable estimates to the *Optimization Phase*. This because, in general, it is more convenient to handle well-defined but slightly noisy peaks rather than smoothed estimates: the former clearly indicate where the densities are concentrated, while the latter tend to spread the probability over the a larger interval.

Different combinations for the *Implementation Phase* parameters are tested, according to the full factorial plan in table 3.9. In particular, the proposed non-zero implementation costs C_0 respectively correspond to 12% and 25% of g_{AON}.

Table 3.9: M3/D3/0.02 - *Implementation Phase* settings.

Factor	Level 1	Level 2	Level 3
C_0 $[kJ/part]$	0	25	52
α	1	0.20	0.05
n^*	∞	2500	500

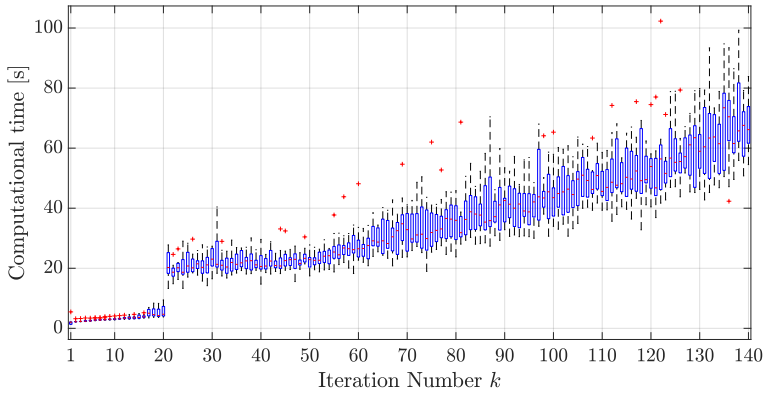
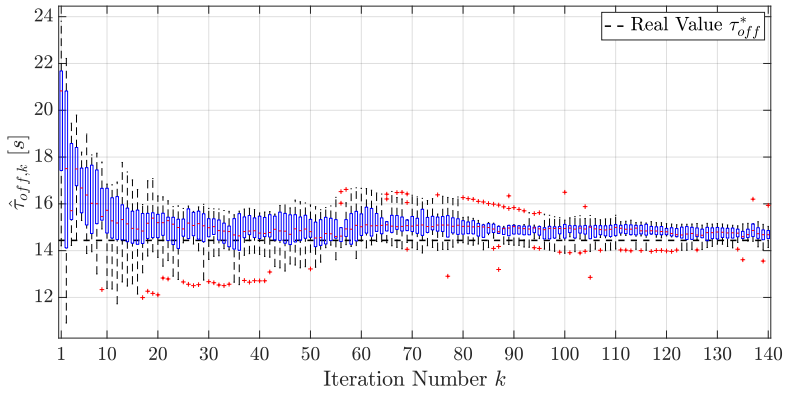
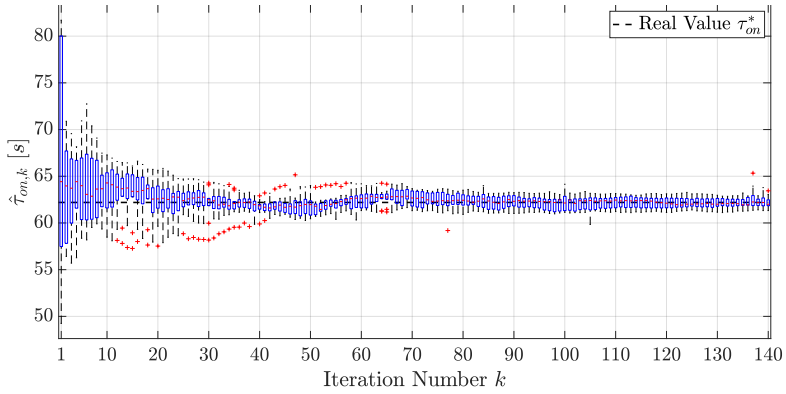


Figure 3.18: M3/D3/0.02 - Total time required to complete each iteration (box plot of 10 replications).



(a) Switch off parameter $\hat{\tau}_{off,k}$.



(b) Switch on parameter $\hat{\tau}_{on,k}$.

Figure 3.19: M3/D3/0.02 - Estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ at each algorithm iteration k (box plot of 10 replications).

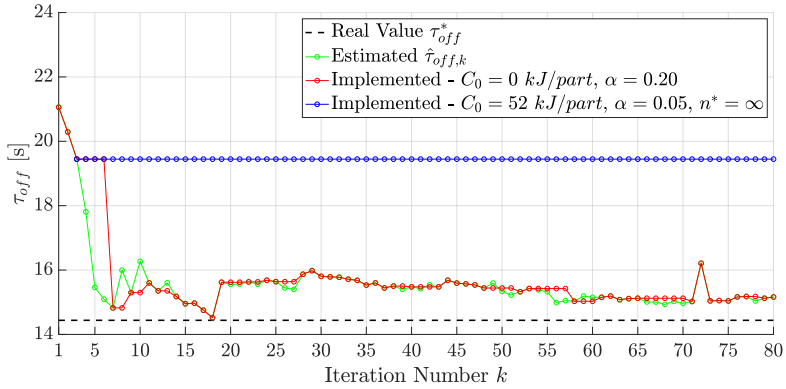
Performance analysis and goodness of the estimate. In this *scenario*, the algorithm computational requirements are higher with respect to the previous examples. In fact, Leave-One-Out Cross Validation and the solving algorithm developed for the *constrained optimization problem* are characterized by severe computational efforts. Nevertheless, the real-time algorithm applicability is still guaranteed. In this regard, figure 3.18 shows the box plot of the total time required to complete a single iteration, which is always lower than two minutes.

The long-range convergence of the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ towards the off-line optimal solution $\boldsymbol{\tau}^*$ is ensured even in this more challenging *scenario*. In fact, figure 3.19 shows how the estimate ranges of change get progressively tighter, approaching the corresponding optimal values (dashed lines).

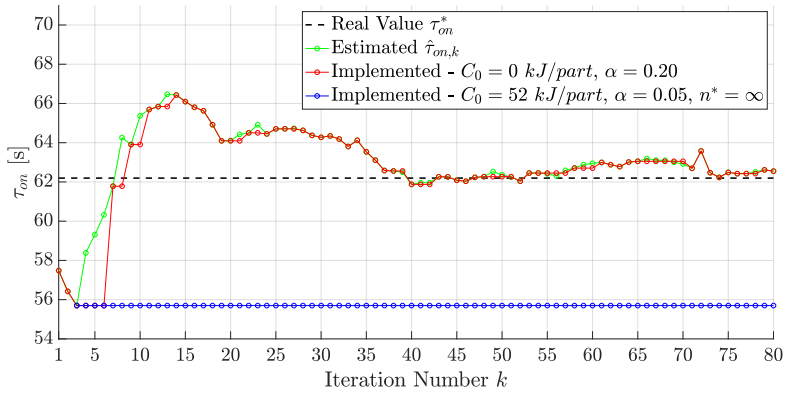
Sensitivity analysis. Dealing with a constrained control problem, **Step 0** of the *Bootstrap CI Implementation Policy* (cf. subsection 3.4.3) acts as a safety measure against the risk of excessive service level reduction. It indeed guarantees that, at each iteration k, the implemented control parameters $\boldsymbol{\tau}_{\boldsymbol{impl}}$ always satisfy the *throughput constraint*. This may obviously lead to an increase in the number of control changes, as reported in table 3.10.

Table 3.10: M3/D3/0.02 - Effects of factors C_0, α and n^* over the mean number of observations before implementing the control parameters $n_{initial}$ and the mean number of changes.

C_0 [kJ/part]	α	n^*	$n_{initial}$	Mean Number of Changes
0	0.05	/	12	77.9
	0.20	/	11	88.6
25	0.05	∞	25	6.5
		2500	25	23.3
		500	24	54.3
	0.20	∞	14	6.4
		2500	14	25.9
		500	14	59.2
52	0.05	∞	27	6.3
		2500	26	17.6
		500	26	51.1
	0.20	∞	26	6.4
		2500	26	20.3
		500	26	57.9



(a) Switch off parameter $\tau_{off,impl}$.

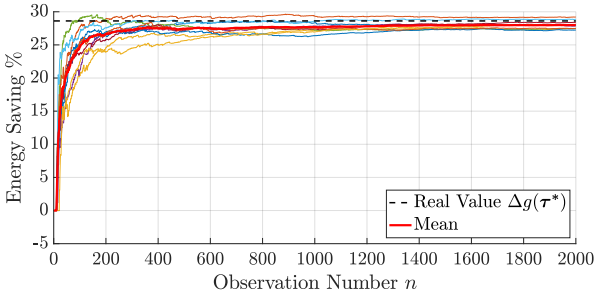


(b) Switch on parameter $\tau_{on,impl}$.

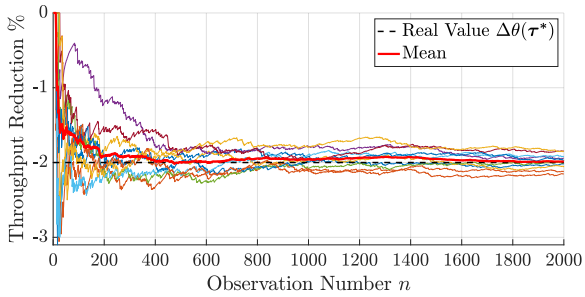
Figure 3.20: M3/D3/0.02 - Implemented control parameters $\boldsymbol{\tau_{impl}}$ for different *Implementation Phase* settings (1 replication). Only the first 2000 observations are reported, i.e. 80 iterations.

Focusing on a single replication, the above consideration can be visualized in figure 3.20. In particular, the effect of **Step 0**, combined with risk-prone *Implementation Phase* settings, leads to a large number of control changes (red line). Nevertheless, this allows to fully exploit the energy saving potential within the limit given by the *throughput constraint*, as reported in figure 3.21.

On the contrary, risk-averse *Implementation Phase* settings may cause the machine to be controlled with biased parameters, which in practice largely satisfy the *throughput constraint* (blue line). Even though more accurate control parameters are progressively estimated, they are not implemented due to the high risk aversion of the considered experiment. This result in partial energy savings and throughput reductions lower than the maximum admissible (cf. figure 3.22).

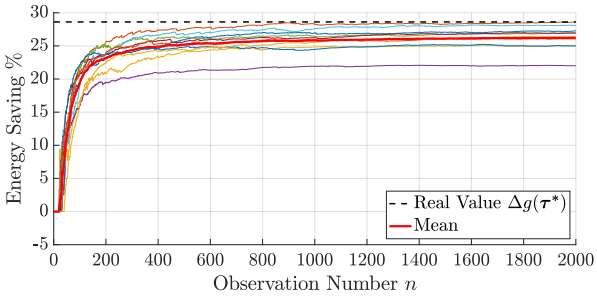


(a) Sample-based energy saving.

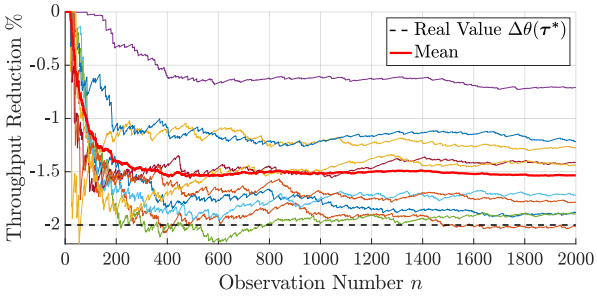


(b) Sample-based throughput reduction.

Figure 3.21: M3/D3/0.02 - $C_0 = 0\ kJ/part$ and $\alpha = 0.20$ (10 replications).



(a) Sample-based energy saving.



(b) Sample-based throughput reduction.

Figure 3.22: M3/D3/0.02 - $C_0 = 52\ kJ/part$, $\alpha = 0.05$ and $n^* = \infty$ (10 replications).

3.6 Conclusions

In this chapter, a time-based energy efficient policy is illustrated, which works on-line while acquiring information from the real system. The proposed algorithm improves and extends that from literature [43, 63]. In particular, each of the three algorithm phases is addressed in details, describing and justifying the proposed enrichments.

With regard to the *Learning Phase*, a brief literature review on the Kernel Density Estimation (KDE) method is reported. Three bandwidth estimation methods are illustrated and compared. Numerical analyses show that no method always provides acceptable results, suggesting to employ a mixture of them in a black-box version of the algorithm. Focusing on the *Optimization Phase*, an efficient solving algorithm is proposed step by step, assessing its goodness numerically. Moving to the *Implementation Phase*, the *Bootstrap CI Implementation Policy* is introduced. It guarantees a greater robustness with respect to the paired-t test proposed in literature and it can also be employed when a constraint must be satisfied.

The on-line algorithm effectiveness and computational requirements are tested on the basis of some numerical cases. In particular, the control problem is firstly addressed without considering any *throughput constraint* (*scenarios* M1/D1/1 and M2/D2/1). Results show that the algorithm performance, evaluated considering the investigated KPIs, are similar or better than that of the existing policy from literature. In addition, a significant response time enhancement is achieved, thus laying the foundations for a real time shopfloor applicability. The sensitivity analysis proves that more conservative *Implementation Phase* settings (i.e. higher C_0, lower α and higher n^*) entail a progressive delay in implementing the EEC and a lower number of control changes, leading to reduced energy savings. These outcomes are coherent with the existing literature. Finally, *scenario* M3/D3/0.02 is employed to asses the effectiveness of the algorithm extension to account for a *throughput constraint*. Results confirm the possibility to achieve energy efficiency while respecting a user-defined maximum admissible throughput reduction.

Chapter 4

Analysis on Risk and Uncertainty

The scope of this chapter is to widen the applicability of the on-line control policy proposed in chapter 3, including considerations on risk and uncertainty. Firstly, since the control is selected considering the statistical average behavior of the machine, the *optimization problem* from literature is enriched with two additional constraints in order to mitigate the risk of deteriorating machine performance over the single *cycle* (contribution **C1.2**). Then, the issue of uncertainty on density estimation is addressed, discussing how to include this information in the control problem.

This chapter is structured in five sections. In section 4.1, the probability distributions of the *machine energy consumption in a cycle* and the *waiting time in a cycle* are derived. In section 4.2, the complete *risk optimization problem* is formulated and the solving algorithm from subsection 3.3.2 is properly adjusted to identify its exact solution. In section 4.3 the goodness of the proposed risk mitigation measure is assessed numerically. In section 4.4, the probability density function of the KDE bandwidth h is obtained, examining the resulting potential applications in the control problem. Finally, section 4.5 concludes the analysis.

4.1 Machine Performance Density Functions

This section illustrates the procedure to express the distribution functions of the *machine energy consumption in a cycle* and the *waiting time in a cycle*, which are respectively described by random variables E and Q. Since these functions are conditioned to the starvation times distribution $f_X(x)$, the *Method of Transformations* (cf. appendix B.1) is applied for this purpose, being the general approach to find the distribution of a function of a continuous random variable [78].

4.1.1 Energy Probability Density Function

The *machine energy consumption in a cycle* E is here addressed. In particular, with reference to subsection 3.4.1, it depends on random variable X via the *energy output function*:

$$E = \phi(X, \boldsymbol{\tau}) \tag{4.1}$$

Given a starvation times distribution $f_X(x)$ and a control $\boldsymbol{\tau}$, the *Method of Transformations* can be therefore exploited to derive the analytical expression for its probability density function $f_E(e|\boldsymbol{\tau})$:

$$f_E(e|\boldsymbol{\tau}) = \begin{cases} \displaystyle\sum_{i=1}^{m} \frac{f_X(x_i)}{|\phi'(x_i, \boldsymbol{\tau})|} & \text{if } \exists\ x_i,\ i = 1, \ldots, m \mid e = \phi(x_i, \boldsymbol{\tau}) \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

where $\phi'(x_i, \boldsymbol{\tau})$ is the *energy output function* derivative with respect to x.

The analysis can be also extended to account for the *difference in machine energy consumption with respect to the AON policy*, described with random variable ΔE:

$$\Delta E = \phi(X, \{\infty, \infty\}) - \phi(X, \boldsymbol{\tau}) \tag{4.3}$$

Denote the realization of ΔE with Δe. First and foremost, by making reference to figure 4.1, it is worth noting that $\Delta e_{min} < \Delta e \leq \Delta e_{max}$. Specifically:

- $\Delta e_{min} = -(w_3 + w_q)t_{su}$
- $\Delta e_{max} = w_2(\tau_{on} + t_{su}) - [w_2\tau_{off} + w_3 t_{su} + w_1(\tau_{on} - \tau_{off})]$

In addition, bottom panel of figure 4.1 also allows to appreciate that $\Delta e = 0$ and $\Delta e = \Delta e_{max}$ correspond to ranges of starvation times values. They may then become mass points for probability. In particular:

- $\Delta e = 0$ accounts for the probability that a part arrives before the machine is switched off ($x \leq \tau_{off}$):

$$\mathbb{P}(\Delta E = 0 \mid \boldsymbol{\tau}) = F_X(\tau_{off}) \tag{4.4}$$

- $\Delta e = \Delta e_{max}$ accounts for the probability that a part arrives after the machine operational readiness is resumed ($x > \tau_{on} + t_{su}$):

$$\mathbb{P}(\Delta E = \Delta e_{max} \mid \boldsymbol{\tau}) = 1 - F_X(\tau_{on} + t_{su}) \tag{4.5}$$

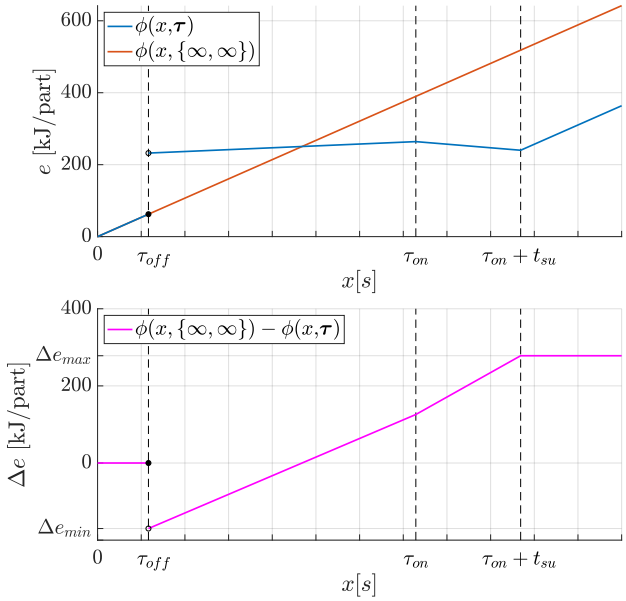


Figure 4.1: Comparison between $\phi(x,\{\infty,\infty\}) = w_2 x$ and a general $\phi(x,\boldsymbol{\tau})$ for machine M1.

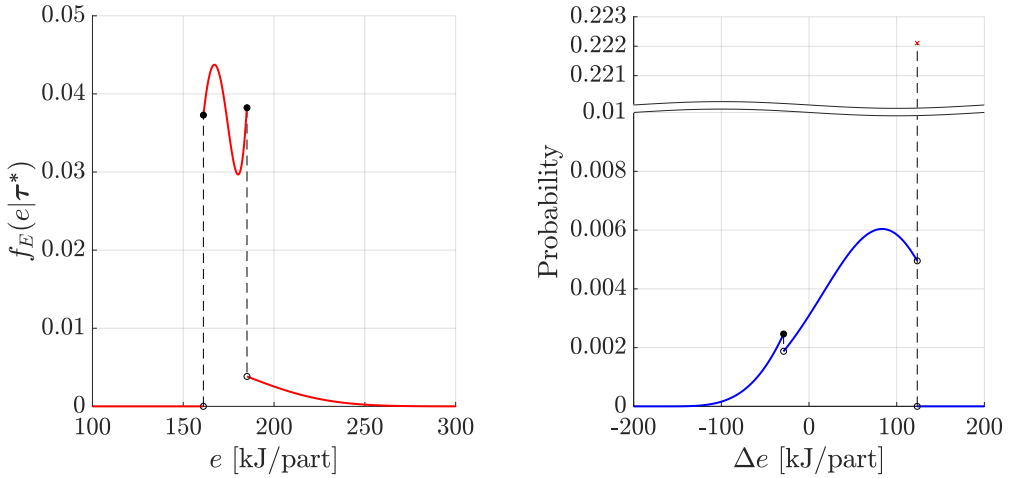
Due to presence of the mass points, the probability distribution of ΔE is mixed. Therefore, it can be uniquely expressed with the cumulative distribution function $F_{\Delta E}(\Delta e|\boldsymbol{\tau})$:

$$F_{\Delta E}(\Delta e|\boldsymbol{\tau}) = \begin{cases} \displaystyle\int_{\Delta e_{min}}^{e} f_{\Delta E}(t|\boldsymbol{\tau})\,dt & \text{if } \Delta e_{min} \le \Delta e < 0 \\ \displaystyle\int_{\Delta e_{min}}^{e} f_{\Delta E}(t|\boldsymbol{\tau})\,dt + \mathbb{P}(\Delta E = 0 \mid \boldsymbol{\tau}) & \text{if } 0 \le \Delta e < \Delta e_{max} \\ 1 & \text{if } \Delta e = \Delta e_{max} \end{cases} \tag{4.6}$$

where $f_{\Delta E}(\Delta e|\boldsymbol{\tau})$ is the pdf of the continuous part, whose analytical expression results from applying the *Method of Transformations*:

$$f_{\Delta E}(\Delta e|\boldsymbol{\tau}) = \begin{cases} \dfrac{f_X(x_i)}{w_2 - \phi'(x_i,\boldsymbol{\tau})} & \text{if } \exists\, x_i \in (\tau_{off}, \tau_{on}+t_{su}] \mid \Delta e = \\ & = \phi(x_i,\{\infty,\infty\}) - \phi(x_i,\boldsymbol{\tau}) \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Considering *scenario* M1/D1/1 (cf. figure 3.10), whose optimal control is $\boldsymbol{\tau}^* = \{0, 29.18\}$, the probability distribution of E is shown, for illustrative purposes, in figure 4.2a. In particular, since it is optimal to immediately switch off the machine at the beginning of each *cycle*, the startup procedure is



(a) $f_E(e|\boldsymbol{\tau}^*)$. (b) $f_{\Delta E}(\Delta e|\boldsymbol{\tau}^*)$ and mass point at Δe_{max}.

Figure 4.2: M1/D1/1 - Energy probability distributions for $\boldsymbol{\tau}^*$.

always executed and the lowest possible occurrence of E is $\phi(\tau_{on} + t_{su}, \boldsymbol{\tau}^*) = 161.10$ kJ/part. Moreover, focusing on the probability distribution of ΔE (cf. figure 4.2b), it is evident that most of the probability belongs to positive differences $\Delta e > 0$, resulting in significant energy savings. Finally, the mass point at $\Delta e_{max} = 123.43$ kJ/part indicates that, under the optimal control, more than 20% of the parts arrive at the machine once its operational readiness is resumed.

4.1.2 Waiting Time Probability Density Function

The discussion laid out in subsection 4.1.1 can also applied to the part *waiting time in a cycle* Q, whose realization is denoted with q. To achieve this goal, the *waiting time function* $q = \psi(x, \boldsymbol{\tau})$, which links q to the starvation times occurence x and the control $\boldsymbol{\tau}$, needs to be introduced:

$$\psi(x, \boldsymbol{\tau}) = \begin{cases} 0 & \text{if } x \leq \tau_{off} \\ t_{su} & \text{if } \tau_{off} < x \leq \tau_{on} \\ \tau_{on} + t_{su} - x & \text{if } \tau_{on} < x \leq \tau_{on} + t_{su} \\ 0 & \text{if } x > \tau_{on} + t_{su} \end{cases} \tag{4.8}$$

By looking at function $\psi(x, \boldsymbol{\tau})$ (cf. figure 4.3) it can be inferred that the probability distribution of Q is a mixed one. In particular, it is characterized by a continuous part $f_Q(q|\boldsymbol{\tau})$ combined with the following mass points:

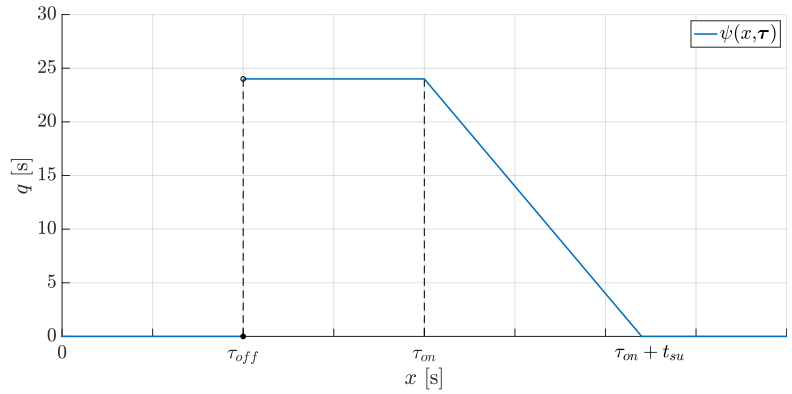


Figure 4.3: General *waiting time function* for machine M1 ($\tau_{on} > \tau_{off} > 0$).

- $q = 0$, which accounts for the probability that a part arrives when the machine is idle ($x \leq \tau_{off} \vee x > \tau_{on} + t_{su}$):

$$\mathbb{P}(Q = 0 \mid \boldsymbol{\tau}) = F_X(\tau_{off}) + [1 - F_X(\tau_{on} + t_{su})] \tag{4.9}$$

- $q = t_{su}$, which accounts for the probability that a part arrives when the machine is in standby ($\tau_{off} < x \leq \tau_{on}$), since it has to wait the whole startup duration to be processed:

$$\mathbb{P}(Q = t_{su} \mid \boldsymbol{\tau}) = F_X(\tau_{on}) - F_X(\tau_{off}) \tag{4.10}$$

Here again, the probability distribution of Q can be uniquely expressed with the cumulative distribution function $F_Q(q|\boldsymbol{\tau})$:

$$F_Q(q|\boldsymbol{\tau}) = \begin{cases} \mathbb{P}(Q = 0 \mid \boldsymbol{\tau}) & \text{if } q = 0 \\ \mathbb{P}(Q = 0 \mid \boldsymbol{\tau}) + \displaystyle\int_0^q f_Q(t|\boldsymbol{\tau})\, dt & \text{if } 0 < q < t_{su} \\ 1 & \text{if } q = t_{su} \end{cases} \tag{4.11}$$

where the analytical expression of the continuous part $f_Q(q|\boldsymbol{\tau})$ results from applying the *Method of Transformations*:

$$f_Q(q|\boldsymbol{\tau}) = \begin{cases} f_X(x_i) & \text{if } \exists\ x_i \in (\tau_{on}, \tau_{on} + t_{su}] \mid q = \psi(x_i, \boldsymbol{\tau}) \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

Focusing once more on *scenario* M1/D1/1, figure 4.4 shows the probability distribution of Q. In particular, it results that more than 75% of parts experience a waiting time before being processed.

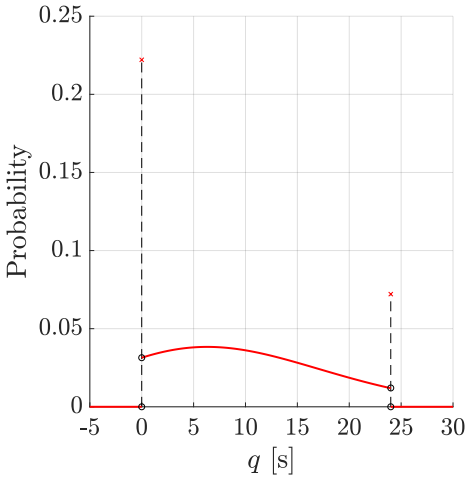


Figure 4.4: M1/D1/1 - $f_Q(q|\boldsymbol{\tau}^*)$ and mass points for $\boldsymbol{\tau}^*$.

4.2 Optimization with Risk Tail Constraints

Since control parameters are optimized based on expectations, they might sometimes lead to a deterioration of machine performance over the single *cycle*. Therefore, in this section, a measure is proposed to select the optimal control while mitigating the risk of increasing the machine energy consumption and the machine *cycle time* with respect to AON policy.

To serve this purpose, the focus is on the *Optimization Phase* of the on-line algorithm from chapter 3. In detail, once introduced the *risk optimization problem*, adjustments to the solving algorithm introduced in subsection 3.3.2 are proposed to identify its exact solution.

4.2.1 Risk Optimization Problem

To account for the possible negative influence of the control over the machine performance in the single *cycle*, the *optimization problem* from literature (cf. subsection P.2.2) is enriched with two additional constraints. The resulting *risk optimization problem* has the following complete formulation:

$$\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}} g(\boldsymbol{\tau}) \tag{4.13}$$

$$\text{Subject to: } F_{\Delta E}(\Delta e_{target}|\boldsymbol{\tau}) \leq \delta_e \tag{4.14}$$

$$1 - F_Q(q_{target}|\boldsymbol{\tau}) \leq \delta_q \tag{4.15}$$

$$\tau_{on} > \tau_{off} \tag{4.16}$$

$$\tau_{off}, \tau_{on} \in \mathbb{R}_0^+ \tag{4.17}$$

where $F_{\Delta E}(\Delta e|\boldsymbol{\tau})$ and $F_Q(q|\boldsymbol{\tau})$ are the cumulative distribution functions introduced in equations (4.6) and (4.11).

Constraint (4.14), referred to as *energy risk constraint*, is introduced to mitigate the risk of incurring in higher consumptions with respect to the AON policy:

$$\mathbb{P}[\phi(X, \{\infty, \infty\}) - \phi(X, \boldsymbol{\tau}) \leq \Delta e_{target}] \leq \delta_e \tag{4.18}$$

Risk-averse users may in fact prefer to give up the maximum saving on the objective function, if this implies a high probability of deteriorating machine energy performance in a single *cycle*. This constraint requires the setting of two parameters:

- A target difference $\Delta e_{target} < 0$.
- A probability $\delta_e \in [0, 1]$.

Constraint (4.15), referred to as *waiting time risk constraint*, is instead introduced to mitigate the risk of increasing the machine *cycle time* with respect to the AON policy:

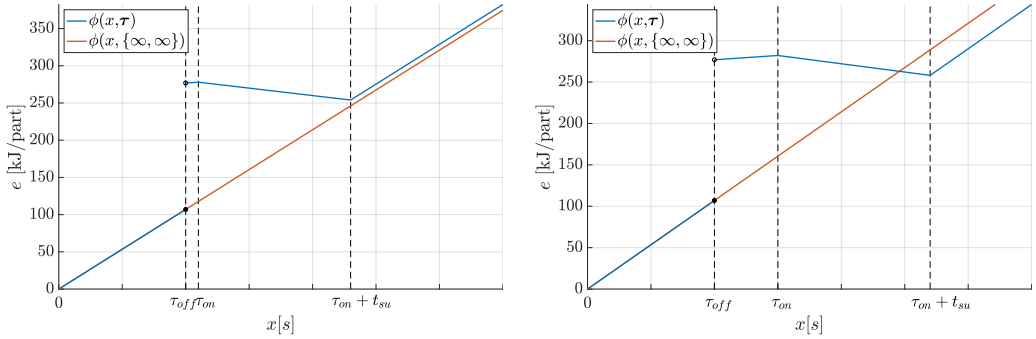
$$\mathbb{P}[\psi(X, \boldsymbol{\tau}) > q_{target}] \leq \delta_q \tag{4.19}$$

In fact, this increase actually results from the waiting time Q a part may experience before being processed. This constraint, which is conceived as an alternative to the condition on the *expected throughput* from equation (P.15), requires the setting of two parameters:

- A target waiting time $q_{target} \geq 0$.
- A probability $\delta_q \in [0, 1]$.

Since constraints (4.14) and (4.15) act on the tails of the aforementioned distribution functions, they are coherently termed *risk tail constraints*. Of course, smaller values of δ_e and δ_q imply a higher risk-aversity. Moreover, it is worth highlighting that these constraints should be perceived as safety measures, therefore they might be *non-binding* in reality.

Lastly, constraints (4.16) and (4.17) allow to define the domain of decision variables. Nevertheless, this set can be reduced thanks to the following theorem:



(a) $\tau_{on} < \tau_{off} + \xi$. (b) $\tau_{on} > \tau_{off} + \xi$.

Figure 4.5: Different control configurations for machine M1.

Theorem 4.1. *It exists a critical sojourn duration in the standby state ξ such that, for any feasible starvation times occurrence x:*

$$\phi(x, \{\infty, \infty\}) \leq \phi(x, \{\tau_{off}, \tau_{on} \leq \tau_{off} + \xi\}) \tag{4.20}$$

The critical sojourn duration is:

$$\xi = \frac{w_3 - w_2}{w_2 - w_1} \, t_{su} \tag{4.21}$$

Proof: The proof of theorem 4.1 is carried out graphically. In particular, figure 4.5a shows a control configuration for machine M1 which cannot guarantee any energy saving with respect to the AON policy. Indeed, the sojourn duration in the standby state is so short that $\phi(x, \{\infty, \infty\}) \leq \phi(x, \boldsymbol{\tau})$ for any feasible starvation times occurrence x.

It is therefore evident that, to guarantee the potential for energy saving, there must exist an interval of starvation times over which $\phi(x, \{\infty, \infty\}) > \phi(x, \boldsymbol{\tau})$. This can be appreciated from figure 4.5b.

The critical condition is obtained when $\phi(x, \{\infty, \infty\})$ and $\phi(x, \boldsymbol{\tau})$ overlap for $x > \tau_{on} + t_{su}$:

$$\phi(\tau_{on} + t_{su}, \{\infty, \infty\}) = \phi(\tau_{on} + t_{su}, \boldsymbol{\tau}) \tag{4.22}$$

Expanding equation (4.22), it results:

$$w_2(\tau_{on} + t_{su}) = w_2\tau_{off} + w_3 t_{su} + w_1(\tau_{on} - \tau_{off}) \tag{4.23}$$

Solving for τ_{on}:

$$\tau_{on} = \tau_{off} + \frac{w_3 - w_2}{w_2 - w_1} \, t_{su} \tag{4.24}$$

Therefore, the *critical sojourn duration in the standby state* is:

$$\xi \stackrel{\text{def}}{=} \frac{w_3 - w_2}{w_2 - w_1}\, t_{su} \tag{4.25}$$

□

Physically speaking, theorem 4.1 suggests that, the sojourn duration in the standby state ($\tau_{on} - \tau_{off}$) must be sufficiently long to ensure that the energy consumed by the machine in the standby and startup states is lower than that it would have absorbed if it had not been switched off:

$$w_3 t_{su} + w_1(\tau_{on} - \tau_{off}) < w_2(\tau_{on} + t_{su} - \tau_{off}) \tag{4.26}$$

The condition in equation (4.16) is henceforth replaced by:

$$\tau_{on} > \tau_{off} + \xi \tag{4.27}$$

Note that, in order to reduce the computational complexity of the problem, the *risk tail constraints* are addressed independently in the following. It is indeed extremely difficult to assess their combined influence on the domain of feasible solutions. The identification of a solving algorithm for the complete *risk optimization problem* is therefore left as a future development.

4.2.2 Energy Risk Constraint

The risk of incurring in higher consumptions with respect to the AON policy is firstly considered. In particular, the related *optimization problem* results from relaxing constraint (4.15). Since control parameters are independent (cf. theorem P.1), the optimal trade-off solution must be selected as the minimum among:

1. The potential solutions of the *unconstrained optimization problem*, which satisfy the *energy risk constraint*.

2. The constrained minimum of $g(\boldsymbol{\tau})$ along the set of points which satisfy the *energy risk constraint* at equality, referred to as ΔE_b:

$$\Delta E_b = \{\boldsymbol{\tau} \in \mathbb{R}^2 \mid F_{\Delta E}(\Delta e_{target}|\boldsymbol{\tau}) = \delta_e,\ \tau_{on} > \tau_{off} + \xi,\ \tau_{off} \geq 0\} \tag{4.28}$$

For the sake of clarity, two qualitative examples of general feasibility regions are shown in figure 4.6. In particular, it is possible to appreciate how the introduction of an *energy risk constraint* sections the domain into vertical feasibility bands. It is also evident that, once exceed a certain threshold (dotted line), the boundary ΔE_b goes straight towards $\tau_{on} = \infty$. This behavior is explained by the following theorem:

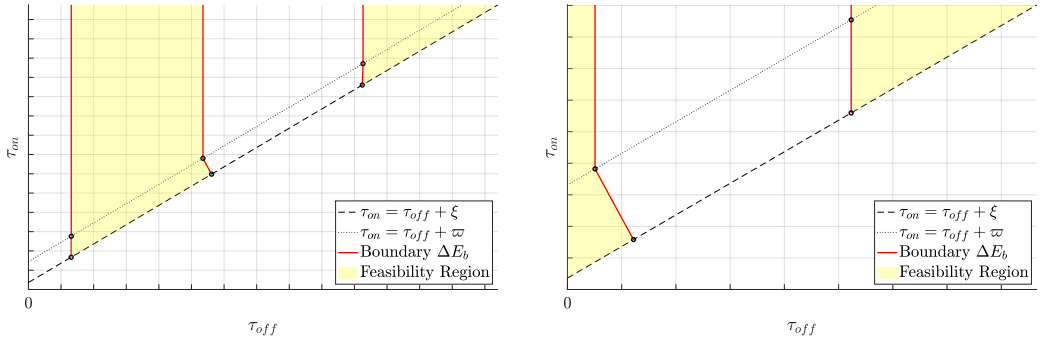


Figure 4.6: Different shapes for the boundary ΔE_b.

Theorem 4.2. *Once set a feasible target difference Δe_{target}, it exists a limit sojourn duration in the standby state ϖ such that:*

$$F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{on} > \tau_{off} + \varpi\}) = F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{off} + \varpi\}) \quad (4.29)$$

The limit sojourn duration is:

$$\varpi = \frac{(w_3 + w_q)t_{su} + \Delta e_{target}}{w_2 - w_1} \quad (4.30)$$

Proof: A graphical proof is provided in the appendix B.2. □

The trade-off solution under the *energy risk constraint* is finally identified by exploiting the solving algorithm introduced subsection 3.3.2. Nevertheless, the following steps are modified:

Step C2. Identify the combinations $\in \mathcal{D}_{unc}$ that satisfy the *energy risk constraint* and collect them in set $\mathcal{D}_{con}$:

$$\mathcal{D}_{con} = \{\boldsymbol{\tau} \in \mathcal{D}_{unc} \mid F_{\Delta E}(\Delta e_{target}|\boldsymbol{\tau}) \leq \delta_e\} \quad (4.31)$$

Step C3. Identify the constrained minimum $\boldsymbol{\tau_b^*}$ of $g(\boldsymbol{\tau})$ along the the boundary ΔE_b and add it to set $\mathcal{D}_{con}$. In particular, solve the following minimization problem:

$$\begin{aligned}
\boldsymbol{\tau_b^*} = \quad & \arg\min_{\boldsymbol{\tau}} \; g(\boldsymbol{\tau}) && (4.32)\\
\text{Subject to:} \quad & F_{\Delta E}(\Delta e_{target}|\boldsymbol{\tau}) = \delta_e && (4.33)\\
& \tau_{on} > \tau_{off} + \xi && (4.34)\\
& \tau_{off}, \tau_{on} \in \mathbb{R}_0^+ && (4.35)
\end{aligned}$$

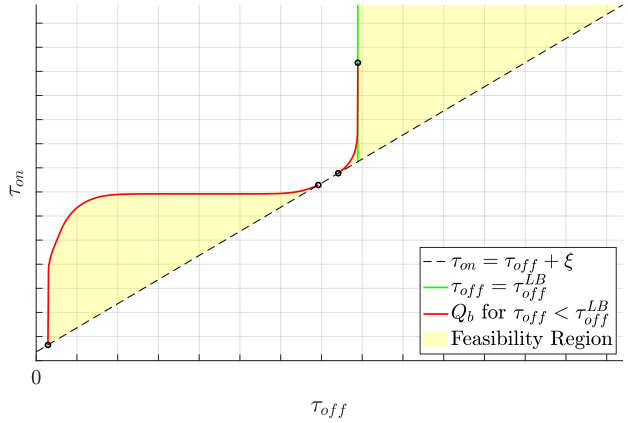


Figure 4.7: General shape for the boundary W_b.

4.2.3 Waiting Time Risk Constraint

The risk of increasing the machine *cycle time* with respect to the AON policy is now addressed. In particular, the related *optimization problem* results from relaxing constraint (4.14). Here again, the optimal trade-off solution must be selected as the minimum among:

1. The potential solutions of the *unconstrained optimization problem*, which satisfy the *waiting time risk constraint*.

2. The constrained minimum of $g(\boldsymbol{\tau})$ along the set of points which satisfy the *waiting time risk constraint* at equality, referred to as Q_b:

$$Q_b = \{\boldsymbol{\tau} \in \mathbb{R}^2 \mid F_Q(q_{target}|\boldsymbol{\tau}) = 1 - \delta_q,\ \tau_{on} > \tau_{off} + \xi,\ \tau_{off} \geq 0\} \quad (4.36)$$

A qualitative example of a general problem feasibility region is shown in figure 4.7. In particular, it is worth noting how the behavior of Q_b resembles that of the *throughput boundary* TH_b, previously depicted in figure 3.7. This is not surprising, since they both result from constraints which, albeit in different ways, address the part waiting time. It is also evident that, once exceed a certain threshold (green line), the *waiting time risk constraint* is always satisfied. This behavior is explained by the following theorem:

Theorem 4.3. *Once set a probability δ_q, it exists a lower bound for the switch off parameter τ_{off}^{LB} such that, for any feasible q_{target}:*

$$1 - F_Q(q_{target}|\boldsymbol{\tau}) \leq \delta_q\ \forall\ \boldsymbol{\tau} \mid \tau_{off} \geq \tau_{off}^{LB} \wedge \tau_{on} > \tau_{off} + \xi \quad (4.37)$$

In particular, τ_{off}^{LB} is the $100(1-\delta_q)$th quantile of $f_X(x)$.

Proof: A graphical proof is provided in the appendix B.3. □

The trade-off solution under the *waiting time risk constraint* is again identified by properly adjusting the solving algorithm from subsection 3.3.2:

Step C2. Identify the combinations $\in \mathcal{D}_{unc}$ that satisfy the *waiting time risk constraint* and collect them in set $\mathcal{D}_{con}$:

$$\mathcal{D}_{con} = \{\boldsymbol{\tau} \in \mathcal{D}_{unc} \mid 1 - F_Q(q_{target}|\boldsymbol{\tau}) \leq \delta_q\} \tag{4.38}$$

Step C3. Identify the constrained minimum $\boldsymbol{\tau}_{\boldsymbol{b}}^*$ of $g(\boldsymbol{\tau})$ along the the boundary Q_b and add it to set $\mathcal{D}_{con}$. In particular, solve the following minimization problem:

$$\begin{aligned} \boldsymbol{\tau}_{\boldsymbol{b}}^* = \quad & \arg\min_{\boldsymbol{\tau}} \; g(\boldsymbol{\tau}) && (4.39)\\ \text{Subject to:} \quad & F_Q(q_{target}|\boldsymbol{\tau}) = 1 - \delta_q && (4.40)\\ & \tau_{on} > \tau_{off} + \xi && (4.41)\\ & \tau_{off}, \tau_{on} \in \mathbb{R}_0^+ && (4.42) \end{aligned}$$

4.3 Results

In this section, numerical analysis are reported to assess the benefits brought by the *risk tail constraints* introduction. In particular, off-line control problems are firstly addressed to show how the optimal control changes according to different constraint settings. Later, the on-line applicability of the proposed risk mitigation measure is evaluated with two numerical cases.

4.3.1 Off-Line Control Problems

The possibility to effectively solve the *risk optimization problem* is first of all investigated in different off-line operating environments, generated with the machines and the starvation times distributions defined in subsection P.3. For each combination, addressed with notation "M/D", the *risk tail constraints* are applied independently, in order to assess how they influence the optimal parameters selection and the related machine energy performance. In the examples herewith provided, the constraints are designed to be always *binding*, considering two levels for their settable parameters.

Energy risk constraint. Focusing on the *energy risk constraint*, results from table 4.1 show the possibility to identify trade-off solutions which correctly mitigate the risk of incurring in higher energy consumptions with

Table 4.1: Optimal control $\boldsymbol{\tau}^*$ for different *energy risk constraint* settings. Related energy performance are also given. Results with $\delta_e = 1$ (*unconstrained optimization problem*) are reported for comparison.

M/D	Δe_{target} $[kJ/part]$	δ_e	$\boldsymbol{\tau}^*$ $[s]$	$g(\boldsymbol{\tau}^*)$ $[kJ/part]$	$\Delta g(\boldsymbol{\tau}^*)$ [%]	$F_{\Delta E}$ $(\Delta e_{target}\|\boldsymbol{\tau}^*)$
M1/D1	-20	1	$\{0, 29.18\}$	175.94	26.92	0.09
		0.08	$\{0, 25.94\}$	176.97	26.49	0.08
		0.05	$\{0, 14.41\}$	199.78	17.02	0.05
	0^-	1	$\{0, 29.18\}$	175.94	26.92	0.14
		0.11	$\{0, 21.41\}$	182.32	24.27	0.11
		0.08	$\{0, 12.79\}$	205.21	14.76	0.08
M2/D2	-100	1	$\{18.37, \infty\}$	295.55	31.09	0.14
		0.11	$\{24.19, \infty\}$	296.63	30.42	0.11
		0.07	$\{39.17, \infty\}$	304.11	28.66	0.07
	0^-	1	$\{18.37, \infty\}$	295.55	31.09	0.17
		0.15	$\{22.34, \infty\}$	296.09	30.55	0.15
		0.11	$\{32.46, \infty\}$	300.31	29.56	0.11
M3/D3	-120	1	$\{12.12, 74.32\}$	141.89	31.44	0.04
		0.02	$\{14.93, 74.32\}$	143.22	30.80	0.02
		0.01	$\{17.10, 74.32\}$	145.64	29.63	0.01
	0^-	1	$\{12.12, 74.32\}$	141.89	31.44	0.05
		0.03	$\{14.11\,74.32\}$	142.59	31.11	0.03
		0.02	$\{15.42, 74.32\}$	143.68	30.57	0.02

respect to the AON policy. As mentioned in subsection 4.2.1, a high risk-aversity (lower δ_e) leads to the loss of a greater amount of energy saving opportunities. In detail, as the constraint gets tougher:

- The switch on command must be anticipated for IHR distributions (D1).
- The switch off command must be delayed for DHR distributions (D2).

Finally, it is worth highlighting that the optimal switch on parameter τ^*_{on} for combination M3/D3 in not modified by the constraint introduction. This follows from theorems P.1 (independence of control parameters) and 4.2.

Waiting time risk constraint. Moving on to the *waiting time risk constraint*, results from table 4.2 show that, when identifying the optimal control, it is possible to effectively account for its effect on part waiting time. Here

Table 4.2: Optimal control $\boldsymbol{\tau}^*$ for different *waiting time risk constraint* settings. Related energy performance are also given. Results with $\delta_q = 1$ (*unconstrained optimization problem*) are reported for comparison.

M/D	q_{target} $[s]$	δ_q	$\boldsymbol{\tau}^*$ $[s]$	$g(\boldsymbol{\tau}^*)$ $[kJ/part]$	$\Delta g(\boldsymbol{\tau}^*)$ [%]	$1 - F_Q$ $(q_{target}\|\boldsymbol{\tau}^*)$
M1/D1	0	1	$\{0, 29.18\}$	175.94	26.92	0.78
		0.60	$\{0, 24.16\}$	178.51	25.85	0.60
		0.40	$\{0, 18.85\}$	187.47	22.13	0.40
	t_{su}^-	1	$\{0, 29.18\}$	175.94	26.92	0.07
		0.02	$\{0, 22.46\}$	180.66	24.96	0.02
		0.04	$\{0, 25.85\}$	177.04	26.46	0.04
M2/D2	t_{su}^-	1	$\{18.37, \infty\}$	295.55	31.09	0.34
		0.30	$\{23.76, \infty\}$	296.49	30.45	0.30
		0.20	$\{45.14, \infty\}$	307.74	27.81	0.20
M3/D3	0	1	$\{12.12, 74.32\}$	141.89	31.44	0.43
		0.30	$\{13.09, 67.44\}$	143.62	30.61	0.30
		0.20	$\{14.22, 60.89\}$	148.70	28.16	0.20
	t_{su}^-	1	$\{12.12, 74.32\}$	141.89	31.44	0.16
		0.05	$\{14.43, 65.75\}$	145.25	29.82	0.05
		0.01	$\{18.61, 60.46\}$	154.20	25.50	0.01

again, a high risk-aversity (lower δ_q) leads to the loss of a greater amount of energy saving opportunities. Considering combination M2/D2, it is worth noting how, under the optimal control $\boldsymbol{\tau}^* = \{\tau_{off}^*, \infty\}$, the *waiting time in a cycle* can only be $q = 0$ $(x \leq \tau_{off}^*)$ or $q = t_{su}$ $(x > \tau_{off}^*)$. Therefore, any feasible setting for q_{target} would lead to the same result. Finally, with regards to combination M3/D3, it clear how, unlike before, both control parameters are affected by the constraint introduction.

4.3.2 On-Line Control Problems

The possibility to pursue energy efficiency, while mitigating the risk of a deterioration of machine performance over the single *cycle*, is here evaluated in on-line scenarios. In particular, two cases are analyzed, assuming that the starvation times distribution $f_X(x)$ needs to be learnt from real time data:

- *Energy risk constraint* - M1/D1 with $F_{\Delta E}(0^-|\boldsymbol{\tau}) \leq 0.11$.
- *Waiting time risk constraint* - M3/D3 with $1 - F_Q(0|\boldsymbol{\tau}) \leq 0.20$.

The resulting on-line control problems are addressed with the iterative algorithm from chapter 3, taking into account the latest enrichments brought to its *Optimization Phase*. The investigated KPIs are the machine energy consumption and the proportion of observations in which the targets (i.e. Δe_{target} and q_{target}) are exceeded, respectively denoted with π_e and π_q:

$$\pi_e = \frac{1}{n}\sum_{i=1}^{n} I(\Delta e_i \leq \Delta e_{target}) \tag{4.43}$$

$$\pi_q = \frac{1}{n}\sum_{i=1}^{n} I(q_i > q_{target}) \tag{4.44}$$

where $I(\cdot)$ is the indicator function.

For each case, the *Implementation Phase* is set according to a generic configuration, i.e. $C_0 = 20\%\ g_{AON}$, $\alpha = 0.10$ and $n^* = 2500$. Moreover, an additional experiment is carried out to evaluate the effects of updating the control parameters at each iteration ($\alpha = 1$). All the experiments are composed by 10 replications, within which the algorithm is iterated every $\Delta n = 10$ new observations. Upon completion of the 500th part, this interval is increased to $\Delta n = 50$. The replication ends when 5000 parts are processed.

Note that the results of the off-line control problems (cf. subsection 4.3.1), referred to as "Real Values", are used as benchmarks.

M1/D1 with $\mathbf{F_{\Delta E}(0^-|\tau) \leq 0.11}$. Consider combination M1/D1, with which the AON policy obtains an objective function value of $g_{AON} = 240.75$ kJ/part. In order to achieve energy efficiency while limiting at $\delta_e = 0.11$ the whole probability of increasing the machine energy consumption with respect to the AON policy, the optimal control is $\boldsymbol{\tau}^* = \{0, 21.41\}$. In detail, it allows to obtain 24.27% of savings on the objective function. Since D1 is unimodal with IHR, Silverman Rule of Thumb is employed in the *Learning Phase*.

Starting from figure 4.8, it is possible to appreciate the algorithm effectiveness in identifying the optimal control: the range of change for the estimated switch on parameter $\hat{\tau}_{on,k}$ gets progressively tighter, approaching the off-line optimal value τ^*_{on}. Nevertheless, an accurate convergence is achieved only in the long range. This because the behavior of the boundary ΔE_b is strongly sensitive to the estimated starvation times distribution $\hat{f}_k(x)$.

Dealing with a constrained control problem, **Step 0** of the *Bootstrap CI Implementation Policy* (cf. subsection 3.4.3) guarantees that the *energy risk constraint* is always satisfied by the implemented control. This can be visually perceived by making reference to figure 4.9, in which the details of one replication are reported. In particular, at each iteration k, the implemented

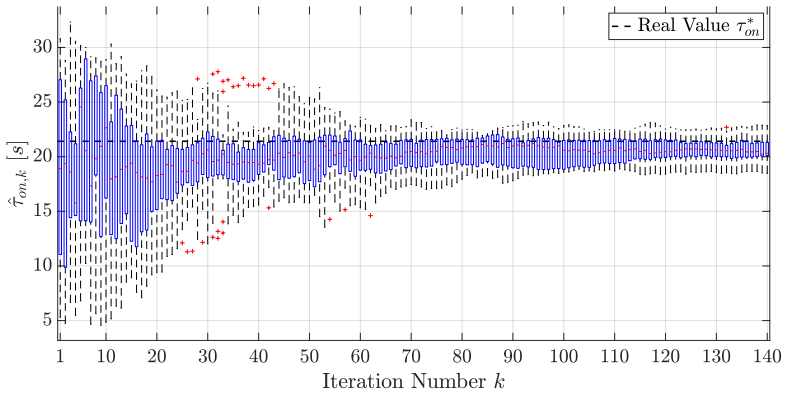


Figure 4.8: M1/D1 with $F_{\Delta E}(0^-|\boldsymbol{\tau}) \leq 0.11$ - Estimated switch on control parameter $\hat{\tau}_{on,k}$ at each algorithm iteration (box plot of 10 replications).

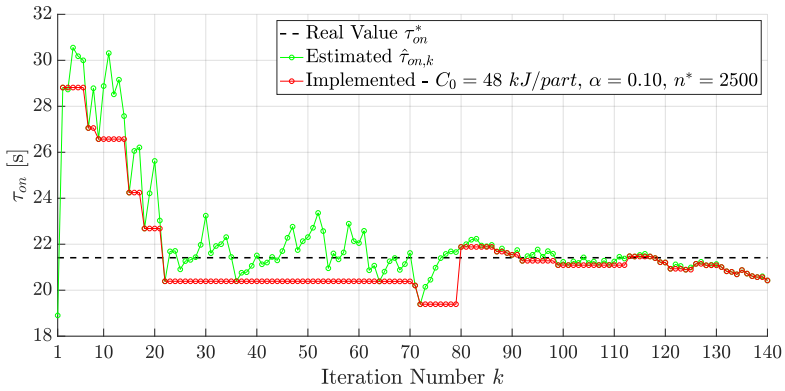
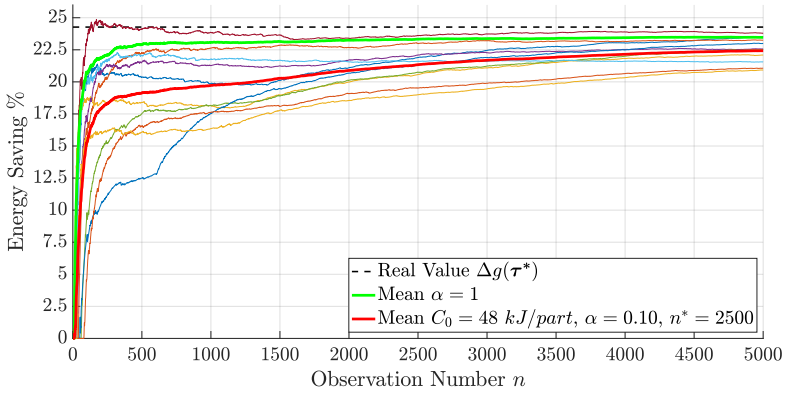


Figure 4.9: M1/D1 with $F_{\Delta E}(0^-|\boldsymbol{\tau}) \leq 0.11$ - Estimated (green) and implemented (red) switch on control parameters at each algorithm iteration k (1 replication).

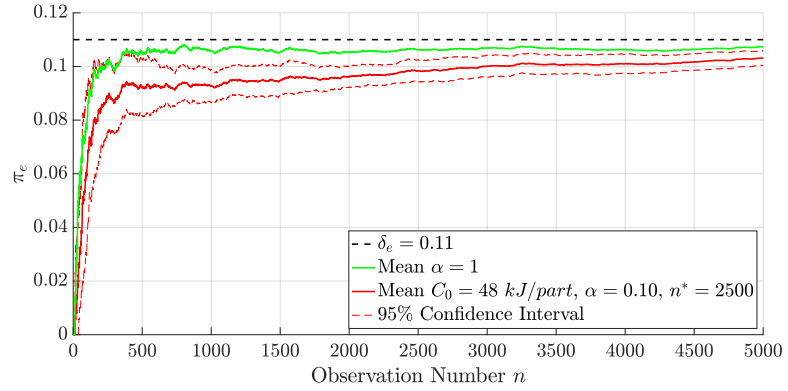
switch on parameter $\tau_{on,impl}$ (red line) is forced to be lower than or equal to the estimated one $\hat{\tau}_{on,k}$ (green line).

Focusing on the investigated performance indexes, figure 4.10a shows the on-line algorithm capability to exploit most of the energy saving potential, within the limit given by the *energy risk constraint*. In this regard, it is worth noting how the 95% confidence interval for the mean of π_e (proportion of observations in which $\Delta e_i \leq 0^-$), reported in figure 4.10b, is always lower than $\delta_e = 0.11$. This indicates that the risk of incurring in higher consumptions with respect to the AON policy is correctly mitigated.

Finally, results with $\alpha = 1$ prove that the actual machine energy consumption and the number of control changes still depend on the *Implementation Phase* settings. Nevertheless, a sensitivity analysis on their effect is not carried out, since it would not add value to the insights from section 3.5.



(a) Sample-based energy saving with $C_0 = 48$ $kJ/part$, $\alpha = 0.10$ and $n^* = 2500$. Green line is obtained without the *Implementation Phase* ($\alpha = 1$).

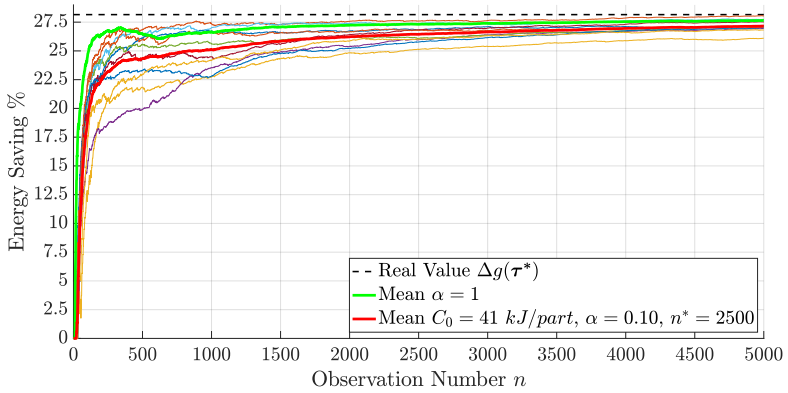


(b) Proportion of observations in which Δe_{target} is exceeded (π_e). Green line is obtained without the *Implementation Phase* ($\alpha = 1$).

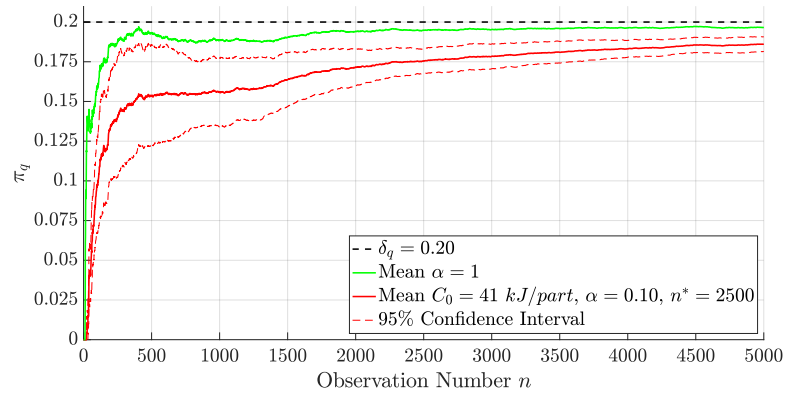
Figure 4.10: M1/D1 with $F_{\Delta E}(0^-|\boldsymbol{\tau}) \leq 0.11$.

M3/D3 with $1 - \mathbf{F_Q}(\mathbf{0}|\boldsymbol{\tau}) \leq \mathbf{0.20}$. The focus now shifts to combination M3/D3, with which the AON policy obtains an objective function value of $g_{AON} = 206.96$ kJ/part. In order to achieve energy efficiency, while limiting at $\delta_q = 0.20$ the probability of increasing the machine *cycle time* with respect to the AON policy, the optimal control is $\boldsymbol{\tau}^* = \{14.22, 60.89\}$. In detail, it allows to obtain 28.16% of savings on the objective function. According to subsection 3.2.4, Leave-One-Out Cross Validation is employed in the algorithm *Learning Phase*.

A brief analysis of the investigated KPIs is simply carried out for this case. In particular, figure 4.11a shows how, for each replication, the sample-based energy saving quickly approaches the off-line control problem result $\Delta g(\boldsymbol{\tau}^*)$. This indicates the algorithm effectiveness in identifying a promising control,



(a) Sample-based energy saving with $C_0 = 41$ $kJ/part$, $\alpha = 0.10$ and $n^* = 2500$. Green line is obtained without the *Implementation Phase* ($\alpha = 1$).



(b) Proportion of observations in which q_{target} is exceeded (π_q). Green line is obtained without the *Implementation Phase* ($\alpha = 1$).

Figure 4.11: M3/D3 with $1 - F_Q(0|\boldsymbol{\tau}) \leq 0.20$.

even when few observations are available. Finally, it is worth highlighting how the 95% confidence interval for the mean of π_q (proportion of observations in which $q_i > 0$) never overcomes $\delta_q = 0.20$ (cf. figure 4.11b). Here again, the risk of increasing the machine *cycle time* with respect to the AON policy is successfully controlled.

A note on computational requirements. Even though the computational complexity of the on-line algorithm significantly increases due to introduction of the *risk tail constraints*, its real time applicability is still guaranteed. In particular, focusing on the analyzed cases, the *Optimization Phase* is always executed in less than a minute, also when $n = 5000$ observations are collected.

4.4 Uncertainty on Density Estimation

Since, as discussed in section 3.2, the whole on-line algorithm effectiveness might be strongly affected by the estimated starvation times distribution $\hat{f}_k(x)$, this section is aimed at identifying the potential benefits of accounting for its uncertainty. In particular, the focus is on the bandwidth parameter h, which acts as a smoothness controller for the Kernel Density Estimation (KDE) method employed in the *Learning Phase* of the algorithm. Once introduced a numerical approach to express the bandwidth probability density function $f_H(h)$, an overview of the resulting potential applications is then reported.

4.4.1 KDE Bandwidth Probability Density Function

Even though the KDE bandwidth selection is a classical research problem in non-parametric statistics, all the existent approaches are aimed at identifying just one optimal value h^* (cf. subsection 3.2.2). This means that the resulting estimate $\hat{f}_{X|H}(x|h^*)$ does not contain any information on the goodness in approximating the real distribution. There is, then, no doubt that the knowledge of the entire bandwidth density function $f_H(h)$ would bring a significant added value to the estimate.

In this regard, the cross validated pseudo-likelihood function $\tilde{L}(h)$, introduced in equation (3.16), needs to be recalled:

$$\tilde{L}(h) = \prod_{i=1}^{n} \frac{1}{(n-1)h} \sum_{\substack{j=1 \\ j\neq i}}^{n} K(\frac{x_i - x_j}{h}) \tag{4.45}$$

Indeed, once normalized, it can actually be considered as the bandwidth density function $f_H(h)$, since it estimates the probability of obtaining the observed data as a function of h. Unfortunately, due to a numerical issue, function $\tilde{L}(h)$ cannot be directly employed for this purpose. Specifically, as the sample size n increases, $\tilde{L}(h)$ takes values lower than the Matlab machine epsilon, due to the effect of the $(n-1)$ term in the denominator of equation (4.45).

How to solve the numerical issue. The the aforementioned numerical issue can be solved by switching to logarithms. In fact, once identified the optimal bandwidth h^*_{loo} by means of equation (3.17), the order of magnitude γ for the maximum of $\tilde{L}(h)$ is also known:

$$\gamma = \log[\tilde{L}(h^*_{loo})] = \sum_{i=1}^{n} \log\left[\sum_{\substack{j=1 \\ j\neq i}}^{n} K\left(\frac{x_i - x_j}{h^*_{loo}}\right)\right] - n\log\left[(n-1)h^*_{loo}\right] \tag{4.46}$$

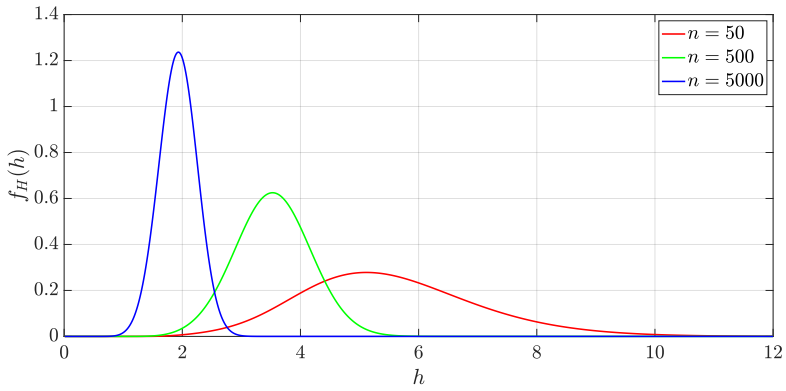


Figure 4.12: Bandwidth density function $f_H(h)$ of distribution D1 for different sample sizes.

As a result, the cross validated pseudo-likelihood function is bounded between:

$$0 \leq \tilde{L}(h) \leq 10^{\lceil \gamma \rceil} \tag{4.47}$$

In order to magnify this interval, a scale factor equal to $10^{-\lceil \gamma \rceil + 1}$ needs to be applied:

$$0 \leq 10^{-\lceil \gamma \rceil + 1} \tilde{L}(h) \leq 10 \tag{4.48}$$

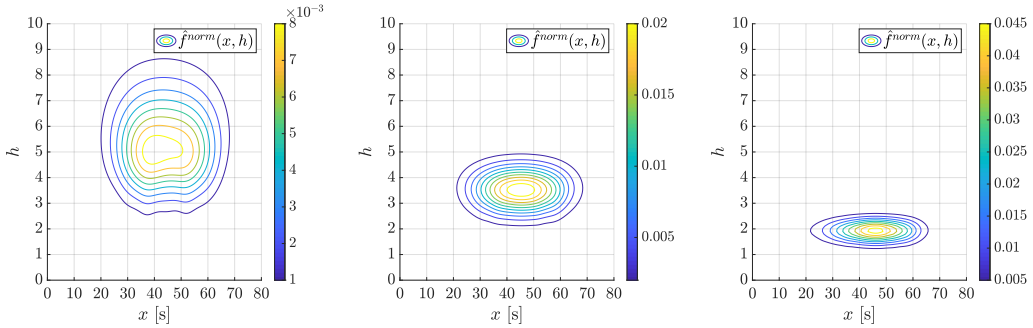
Once spread this scale factor on each of the n components which make up the product in equation (4.45), it results:

$$0 \leq \tilde{L}^{scaled}(h) = \prod_{i=1}^{n} \frac{10^{\frac{-\lceil \gamma \rceil + 1}{n}}}{(n-1)h} \sum_{\substack{j=1 \\ j \neq i}}^{n} K(\frac{x_i - x_j}{h}) \leq 10 \tag{4.49}$$

As mentioned above, the pdf $f_H(h)$ is finally obtained by normalizing function $\tilde{L}^{scaled}(h)$, since it can now be evaluated numerically:

$$f_H(h) = \frac{\tilde{L}^{scaled}(h)}{\int_0^{\infty} \tilde{L}_{scaled}(t)\, dt} \tag{4.50}$$

Considering distribution D1, the bandwidth density functions for different sample sizes ($n = 50$, 500 and 5000) are shown in figure 4.12. It is worth highlighting that, as the sample size increases, $f_H(h)$ shifts towards the origin, its variance progressively decreases and its shape approaches that of a Gaussian distribution. This is coherent with what introduced in equation (3.4) and it implies a gradual greater confidence on the estimate.



(a) $n = 50$. (b) $n = 500$. (c) $n = 5000$.

Figure 4.13: *Joint distribution function* $\hat{f}^{norm}(x,h)$ of D1 for different sample sizes.

4.4.2 Potential Applications in the Control Problem

Since the KDE can be seen as the *conditional density function* of X given $H = h$, starting from equations (3.18) and (4.50) it is now possible to express the starvation times *joint distribution function*:

$$\hat{f}^{norm}(x,h) = \hat{f}^{norm}_{X|H}(x|h) f_H(h) \tag{4.51}$$

The qualitative examples in figure 4.13 clearly show the added value brought to the estimate by the knowledge of the entire bandwidth density function $f_H(h)$. In addition, the progressive improvement resulting from an increasing sample size is also evident.

In the followings, an investigation over the possible applications of $f_H(h)$ is carried out, focusing both on the *Optimization* and *Implementation Phases*.

Optimization phase. When considering the bandwidth variability, the traditional objective function, i.e. the *expected value of the energy consumed in a cycle by the machine*, turns out to be a random variable:

$$\hat{g}(\boldsymbol{\tau}|h) = \int_0^\infty \phi(x,\boldsymbol{\tau}) \hat{f}^{norm}_{X|H}(x|h)\, dx \tag{4.52}$$

An obvious choice is therefore to minimize its expectation over the bandwidth pdf $f_H(h)$:

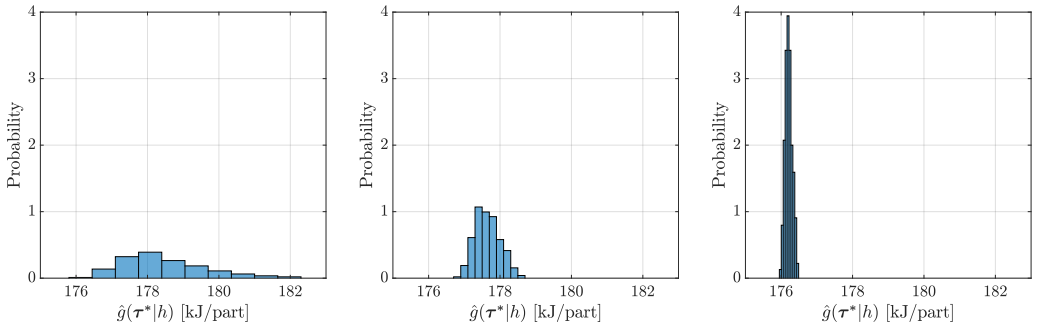
$$\mathbb{E}_h[\hat{g}(\boldsymbol{\tau}|h)] = \int_0^\infty \int_0^\infty \phi(x,\boldsymbol{\tau}) \hat{f}^{norm}(x,h)\, dx\, dh \tag{4.53}$$

Table 4.3: M1/D1/1 - Optimal switch on parameters and objective function estimates for different sample sizes (5 replications). Results with $\hat{g}(\boldsymbol{\tau}|h^*_{loo})$ are compared to those with $\mathbb{E}_h[\hat{g}(\boldsymbol{\tau}|h)]$. $\tau^*_{off} = 0$ is not shown.

Sample size n	Replication number	$\hat{g}(\boldsymbol{\tau}\|h^*_{loo})$		$\mathbb{E}_h[\hat{g}(\boldsymbol{\tau}\|h)]$	
		Switch on parameter [s]	Objective function [kJ/part]	Switch on parameter [s]	Objective function [kJ/part]
50	1	31.15	175.05	31.29	175.28
	2	28.72	175.18	29.15	175.57
	3	28.88	177.33	29.13	177.42
	4	29.41	176.52	29.70	177.10
	5	31.86	178.58	32.14	178.71
500	1	29.91	176.09	29.94	175.83
	2	27.94	175.83	27.96	175.51
	3	29.27	176.72	29.27	176.43
	4	29.66	176.33	29.70	176.09
	5	31.32	177.66	31.36	177.43

The potential benefits of this approach can be numerically assessed by making reference to *scenario* M1/D1/1, whose optimal control $\boldsymbol{\tau}^* = \{0, 29.18\}$ allows to obtain an objective function value of $g(\boldsymbol{\tau}^*) = 175.94$ kJ/part. The idea is to compare the optima obtained by minimizing both $\mathbb{E}_h[\hat{g}(\boldsymbol{\tau}|h)]$ and $\hat{g}(\boldsymbol{\tau}|h^*_{loo})$: while the former accounts for the the whole bandwidth pdf $f_H(h)$, the latter simply employs the optimal value h^*_{loo}. Five replications are carried out for the analyzed sample sizes ($n = 50$ and 500), whose results are reported in table 4.3. In detail, it can be noticed that the differences among the two solutions are very limited and, sometimes, accounting for the whole pdf $f_H(h)$ may even drive away from the real optimum. Finally, the minimization of $\mathbb{E}_h[\hat{g}(\boldsymbol{\tau}|h)]$ respectively raises the computational times from few seconds to few minutes ($n = 50$) or more than an hour ($n = 500$).

Summarizing, this approach does not bring any significant advantage, in fact the enormous increase in computational complexity is not counterbalanced by a greater accuracy in identifying the optimal control. Moreover, all the objective function properties from literature are lost.



(a) $n = 50$. (b) $n = 500$. (c) $n = 5000$.

Figure 4.14: M1/D1/1 - Normalized frequency histogram of $\hat{g}(\boldsymbol{\tau}^*|h)$ for different sample sizes. The theoretical value is $g(\boldsymbol{\tau}^*) = 175.94$ kJ/part.

Implementation phase. The knowledge of the bandwidth density function $f_H(h)$ also allows to appreciate the gradual greater confidence on the objective function estimate $\hat{g}(\boldsymbol{\tau}|h)$, resulting from an increasing number of observations. Unfortunately, the *Method of Transformations* cannot be employed for this purpose, therefore the distribution function of $\hat{g}(\boldsymbol{\tau}|h)$ must be calculated numerically from the bandwidth probability space. In this regard, focusing again on *scenario* M1/D1/1, figure 4.14 shows the normalized frequency histograms of $\hat{g}(\boldsymbol{\tau}^*|h)$ for increasing sample sizes ($n = 50$, 500 and 5000). It is clear that, while the estimate approaches the theoretical value $g(\boldsymbol{\tau}^*) = 175.94$ kJ/part, its dispersion is progressively reduced.

This awareness of the uncertainty on the objective function estimate enables to define a new *Bandwidth Implementation Policy*. In particular, at each iteration k, the idea is to compare the estimated control parameters $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ with the ones currently in place $\boldsymbol{\tau}_{\boldsymbol{impl}}$ in terms of objective function. To account for the uncertainty embedded in the observations, random variable D_h is defined:

$$D_h = \hat{g}(\boldsymbol{\tau}_{\boldsymbol{impl}}|h) - \hat{g}(\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}|h) \tag{4.54}$$

The new control $\hat{\boldsymbol{\tau}}_{\boldsymbol{k}}$ is implemented only if the following condition holds:

$$\mathbb{P}[D_h \leq C(n)] \leq \alpha_h \tag{4.55}$$

where $C(n)$ is the *implementation cost function* from equation (3.34) and $\alpha_h \in [0, 1]$ is a probability. Since the cdf of variable D_h must calculated numerically, the number of random samples drawn from $f_H(h)$ requires a detailed calibration, in order to guarantee a suitable trade-off between accuracy and computational requirements.

The introduction of the *Bandwidth Implementation Policy* in the on-line algorithm as well as the assessment of its efficacy and effectiveness is however left as a future development.

4.5 Conclusions

This chapter is aimed at extending the previously detailed on-line control policy with considerations on risk and uncertainty. First of all, the *Method of Transformation* [78] is employed to analytically derive the distribution functions of the *machine energy consumption in a cycle* and the *waiting time in a cycle*, thus increasing the amount of knowledge available in literature.

Later, on the basis of this additional information, the *risk optimization problem* is formulated. Specifically, assuming the AON policy as a reference, the introduction of the *risk tail constraints* allows to pursue energy efficiency while mitigating the risk of a deterioration of machine performance over the single *cycle*. Addressing each constraint independently, adjustments to the solving algorithm from subsection 3.3.2 are proposed to identify the exact trade-off solutions. The resulting benefits are assessed in different off-line operating environments, showing the constraint influence on the optimal control $\boldsymbol{\tau}^*$. Moving on to on-line problems, the possibility to mitigate the risk while learning from real time data is tested with two numerical cases. Results confirm the goodness of the approach: most of the energy saving potentials are exploited, within the limits given by the *risk tail constraints*.

Finally, focusing on the issue of estimates uncertainty, the KDE bandwidth probability density function $f_H(h)$ is numerically expressed. While its application to the *Optimization Phase* only involves an increase in computational complexity, the gradual greater confidence on the objective function estimate can be employed to define the new *Bandwidth Implementation Policy*.

Chapter 5

Adjustments towards Realistic Arrival Processes

The scope of this chapter is to adjust the on-line control policy from chapter 3 to make it effective when the upstream production process is non-stationary or decoupled from the machine by an input buffer.

This chapter is structured in three sections. In section 5.1 a *change point detection* method is proposed to identify changes in the part arrival process, thus allowing to adapt the control in response to variations in the operating environment (contribution **C2**). In section 5.2, a slightly different control problem is studied, assuming that the machine is able to observe the interarrivals of parts to its upstream input buffer, in which they accumulate before processing. Since the on-line control policy may not converge to a unique solution in this environment, a modified objective function is proposed to identify an approximate optimal control. Finally, in section 5.3 some conclusive remarks are drawn.

5.1 Non-Stationary Arrival Process

In this section, an algorithm enhancement to account for a non-stationary arrival process is presented. In particular, it is now assumed that the starvation times distribution $f_X(x)$ changes at unknown discrete points in time, due to variations in the upstream production process. If these changes were not systematically detected, the estimated control parameters would be biased, being computed from a sample with non-homogeneous subsets. Therefore, a *change point detection* method from literature is introduced and calibrated in order to ensure an effective control adaptation. The goodness of the proposed approach is then assessed numerically.

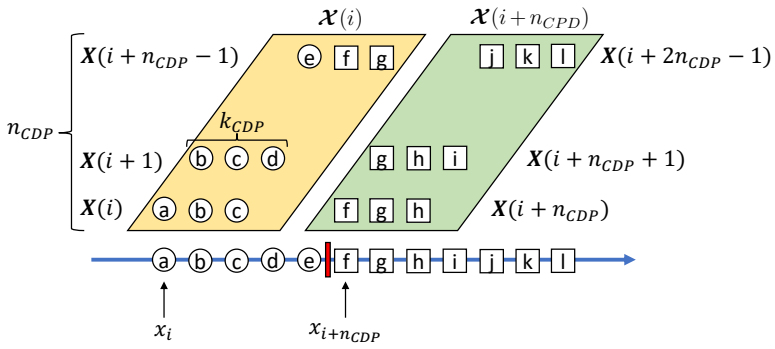


Figure 5.1: An illustrative example of RuLSIF method notation.

5.1.1 Change Point Detection Problem and Methods

A *change point* represents a transition between different states in a process that generates a time series data stream. Therefore, *change point detection* (CPD) can be defined as the problem of hypothesis testing between two alternatives, the null hypothesis H_0: "No change occurs" and the alternative hypothesis H_1: "A change occurs".

Over the last several decades, many CPD algorithms have been proposed in literature, which differ in terms of on-line applicability, computational efficiency and requirements on the input data [79]. Amongst them, particular attention should be placed on *unsupervised* methods, since they can be directly employed to discover patterns in unlabeled data, without requiring prior training for each situation. In fact, such algorithms identify *change points* based on the statistical features of the collected samples.

On the basis of a detailed review of *unsupervised* algorithms, the *relative unconstrained least-squares importance fitting* (RuLSIF) method is implemented [80]. Indeed, being on-line and non-parametric, it completely complies with the requirements of the considered problem. The core idea behind this approach is to analyze the probability distributions of the data before and after a candidate point, and identify it as a *change point* if the two distributions are significantly different. In particular, the *ratio* of these probability densities is directly estimated, since it is easier than estimating each distribution separately.

Descending further into the details (cf. figure 5.1), given x_i as a generic sample in the collected time series $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, let $\boldsymbol{X}(i)$ be the subsequence starting from x_i with length k_{CPD}:

$$\boldsymbol{X}(i) = [x_i, x_{i+1}, \ldots, x_{i+k_{CPD}-1}] \tag{5.1}$$

In addition, let $\boldsymbol{\mathcal{X}}(i) \in \mathbb{R}^{k_{CPD} \times n_{CPD}}$ be a set of n_{CPD} retrospective subsequence

samples starting from observation x_i:

$$\boldsymbol{\mathcal{X}}(i) = [\boldsymbol{X}(i)^T, \boldsymbol{X}(i+1)^T, \ldots, \boldsymbol{X}(i+n_{CPD}-1)^T] \tag{5.2}$$

For *change point detection*, the RuLSIF method considers two consecutive sets $\boldsymbol{\mathcal{X}}(i)$ and $\boldsymbol{\mathcal{X}}(i+n_{CPD})$ and computes the *symmetrized α_{CPD}-relative Pearson divergence* as a dissimilarity measure between them[1]. More specifically, this measure can be interpreted as a *change point score*: the higher the dissimilarity, the more likely point $x_{i+n_{CPD}}$ is a *change point*.

According to the authors, peaks of *change point score* should be perceived as detection alarms. Therefore, in order to filter out false alarms, a threshold η_{CPD} for the *change point score* is introduced. In conclusion, *change points* correspond to peaks of *change point score* whose value is greater than or equal to η_{CPD}.

In addition to the threshold η_{CPD}, the effectiveness of the RuLSIF method is affected by the choice of three main parameters:

- n_{CPD}, which controls the number of retrospective subsequences in each set $\boldsymbol{\mathcal{X}}$.
- k_{CPD}, which indicates the size of the sliding window employed to generate each subsequence.
- α_{CPD}, which acts as a smoothness controller for the *change point score*. In particular, the α_{CPD}-relative density-ratio, which is bounded above by $1/\alpha_{CPD}$, tends to be smoother as α_{CPD} gets larger.

Note that n_{CPD} should be large enough to store a sufficient number of data to represent the time series state, yet small enough to guarantee a responsive detection [79]. The RuLSIF method is indeed defined to be ε-real time, since it needs to look at $\varepsilon = n_{CPD} + k_{CPD}$ data points ahead of the candidate *change point*. Finally, $2n_{CPD} + k_{CPD} - 1$ observations should be collected to estimate the first *change point score*, which is referred to point $n_{CPD} + 1$.

5.1.2 Parameter Calibration

RuLSIF parameters are calibrated on the basis of a specific generating process for the starvation times data stream (labeled P1), which is characterized by two states. In detail, distribution D1 is employed in state 1, while distribution D3 is selected for state 2. A back and forth transition (D1-D3-D1)

[1]The Matlab implementation of the proposed method is provided by the authors, refer to [80] for the exhaustive procedure.

Table 5.1: Experimental plan for RuLSIF parameter calibration.

Factor	Level 1	Level 2	Level 3
α_{CPD}	0.01	0.05	
n_{CPD}	20	30	40
k_{CPD}	5	10	

between the states is considered, assuming that changes occur at intervals of 200 observations (cf. top panel of figure 5.2). As a consequence, two *change points* must be detected.

To appropriately manage the trade-off between accuracy and responsiveness, it is required to identify the parameter combination which allows to:

- Detect the *change points* with the least possible delay.
- Minimize the number false alarms.

In this regard, an experimental campaign is designed by varying α_{CPD}, n_{CPD} and k_{CPD} according to the factorial plan in table 5.1. In detail, each experiment is composed by 10 replications and comparability is guaranteed by the use of common random numbers. Based on the value of α_{CPD}, the detection alarms returned by each experiment are then filtered out with different threshold levels η_{CPD}. This because, with reference to subsection 5.1.1, the *change point score* is bounded above by $1/\alpha_{CPD}$. The investigated responses are the average probability of CPD, respectively denoted with $\bar{C}_1$ and $\bar{C}_2$, and the average number of false alarms $\bar{F}_a$.

To discriminate between correct detections and false alarms, the same convention employed in [80] is adopted. More specifically, a *change point detection* referred to observation x_i is regarded as correct if there exists a true *change point* in position j such that $i \in [j-10, j+10]$. In addition, to avoid duplication, an alarm is removed if it is less than 20 observations away from the previous one.

Results, which are thoroughly reported in tables 5.2 and 5.3, show that two parameter combinations provide the best performance in terms of accuracy and responsiveness:

- $\alpha_{CPD} = 0.01$, $n_{CPD} = 30$, $k_{CPD} = 5$ and $\eta_{CPD} = 25$
- $\alpha_{CPD} = 0.05$, $n_{CPD} = 30$, $k_{CPD} = 5$ and $\eta_{CPD} = 10$

Table 5.2: RuLSIF results with $\alpha = 0.01$.

(a) $n_{CPD} = 20$, $k_{CPD} = 5$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
20	1	1	1.6
25	1	0.7	0.9
30	0.6	0.5	0.4

(b) $n_{CPD} = 20$, $k_{CPD} = 10$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
20	0.7	0.7	0.8
25	0.5	0.4	0.6
30	0.4	0.2	0.2

(c) $n_{CPD} = 30$, $k_{CPD} = 5$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
20	1	1	0.1
25	**1**	**1**	**0**
30	1	0.9	0

(d) $n_{CPD} = 30$, $k_{CPD} = 10$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
20	0.7	0.8	0.5
25	0.6	0.6	0.2
30	0.4	0.4	0

(e) $n_{CPD} = 40$, $k_{CPD} = 5$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
20	1	1	0.1
25	1	1	0
30	1	1	0

(f) $n_{CPD} = 40$, $k_{CPD} = 10$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
20	1	1	0.2
25	0.8	1	0.1
30	0.6	0.7	0

Table 5.3: RuLSIF results with $\alpha = 0.05$.

(a) $n_{CPD} = 20$, $k_{CPD} = 5$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
7.5	1	1	1.5
10	0.8	0.8	0.4
12.5	0.5	0.2	0.3

(b) $n_{CPD} = 20$, $k_{CPD} = 10$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
7.5	0.7	0.9	1.1
10	0.1	0.1	0.3
12.5	0.1	0	0

(c) $n_{CPD} = 30$, $k_{CPD} = 5$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
7.5	1	1	0.3
10	**1**	**1**	**0**
12.5	1	0.7	0

(d) $n_{CPD} = 30$, $k_{CPD} = 10$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
7.5	1	1	0.4
10	0.5	0.2	0.1
12.5	0.2	0	0

(e) $n_{CPD} = 40$, $k_{CPD} = 5$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
7.5	1	1	0.2
10	1	1	0
12.5	1	0.9	0

(f) $n_{CPD} = 40$, $k_{CPD} = 10$

η_{CPD}	$\bar{C}_1$	$\bar{C}_2$	$\bar{F}_a$
7.5	1	1	0.2
10	1	0.6	0
12.5	0.3	0.2	0

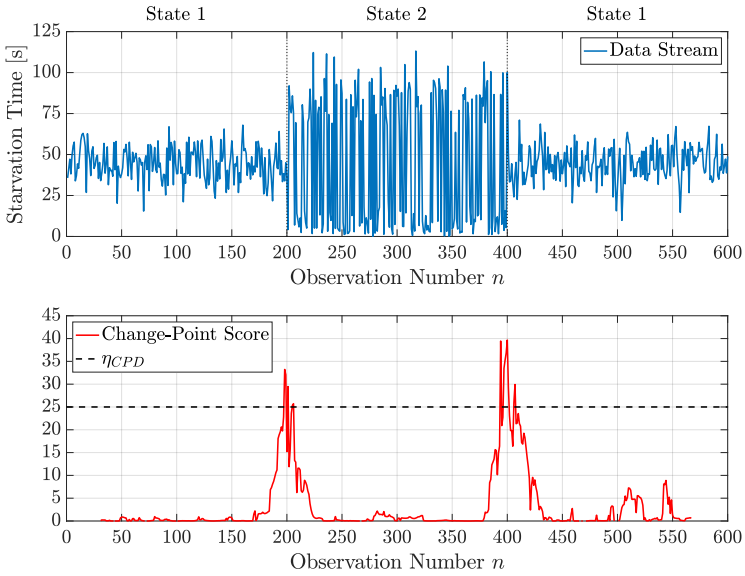


Figure 5.2: Starvation times data stream and change point score for a single replication - $\alpha_{CPD} = 0.01$, $n_{CPD} = 30$, $k_{CPD} = 5$ and $\eta_{CPD} = 25$. Note that the *change point score* is returned with a delay of 35 observations.

In particular, they ensure that *change points* are always identified with the least possible delay, i.e. $n_{CPD} + k_{CPD} = 35$, without returning any false alarm. For the sake of completeness, the *change point score* obtained for one replication with the first parameter combination is reported in figure 5.2, allowing to visually appreciate the effectiveness of the approach.

Finally, the following insights can be also extrapolated:

- Lower threshold levels η_{CPD} imply a higher probability of *change point detection* but also a greater number of false alarms.

- For fixed values of the other parameters, a higher n_{CPD} improves the accuracy of the method. In particular, it results in a higher probability of *change point detection* and in a lower number of false alarms. The drawback is a longer delay in identifying the *change point*.

- For fixed values of the other parameters, results with $k_{CPD} = 10$ are generally worse than those obtained with $k_{CPD} = 5$. Therefore, a higher sliding window size does not implicate a more accurate detection.

5.1.3 Results

The benefits of detecting changes in the upstream production process, while controlling the machine for energy saving purposes are here evaluated with two numerical cases. In this regard, an additional starvation times distribution $f_X(x)$ is introduced to define a second generating process for the starvation times data stream:

- D4 (Unimodal with IHR) - Erlang-3 distribution with rate $\lambda = 0.037$ (i.e., mean is $t_a = 81.08\ s$).

In detail, this new generating process (labeled P2) is always characterized by two states: distribution D2 is employed in state 1, while distribution D4 is selected for state 2. Here again, a back and forth transition (D2-D4-D2) between the states is considered, now assuming that changes occur at intervals of 300 observations.

In this numerical analysis, the *optimization problem* in equations (3.20)-(3.23) is considered, since it includes a condition on the *expected throughput*. Therefore, in conclusion, the following cases (denoted with notation "M/P/ϵ") are analyzed, obtained by combining a machine tool, a generating process for the starvation time occurrences and a *throughput constraint* setting ϵ:

- Case M1/P1/0.02 - *Change points* at $n = 201$ and $n = 401$.

- Case M2/P2/0.05 - *Change points* at $n = 301$ and $n = 601$.

In addition to the accuracy in identifying the *change points*, the machine energy consumption and the throughput are the investigated performance indexes (KPIs). Specifically, to achieve satisfactory estimates, the experiments are composed by 10 replications.

According to the analysis in subsection 5.1.2, the RuLSIF method is tuned with the following parameter combination: $\alpha_{CPD} = 0.01$, $n_{CPD} = 30$, $k_{CPD} = 5$ and $\eta_{CPD} = 25$. From an implementation perspective, every time the *change point score* exceeds the threshold η_{CPD}, the control is set to $\boldsymbol{\tau_{impl}} = \{\infty, \infty\}$. Once identified the peak, the following observations are then employed to estimate the new control parameters. Therefore, the algorithm is iterated every $\Delta n = 10$ new observations or at *change point detection*.

Finally, focusing on the *Learning Phase*, the suggestions provided in subsection 3.2.5 are here implemented, in order to always provide suitable estimates of the starvation times distribution.

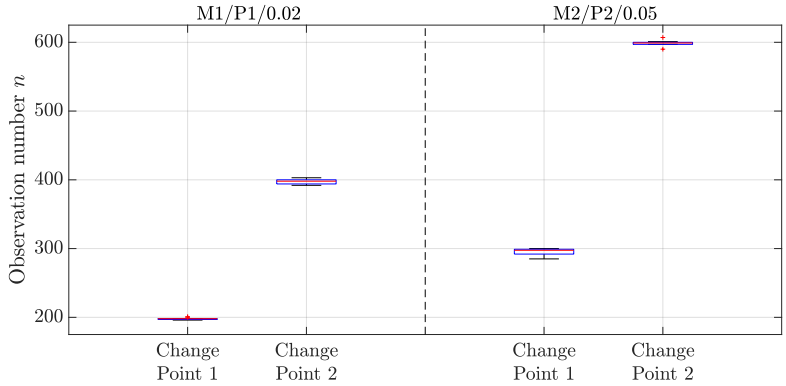


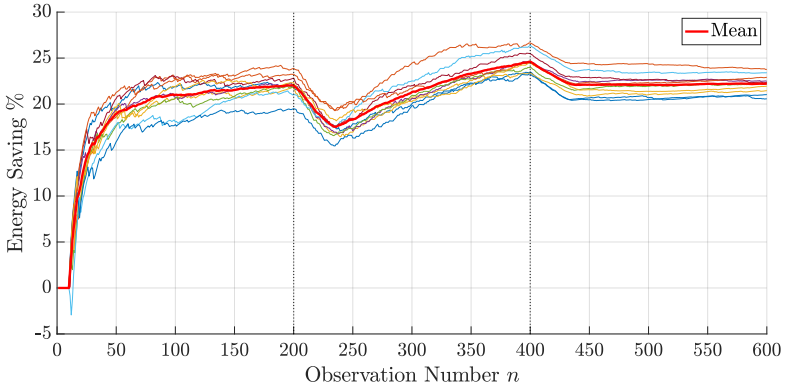
Figure 5.3: *Change point* locations returned by the RuLSIF method for cases M1/P1/0.02 and M2/P2/0.05 (boxplots of 10 replications).

Accuracy in *change point detection*. First of all, the accuracy of the RuLSIF method can be assessed by making reference to figure 5.3, which shows the boxplots of the returned *change point* locations for each of the analyzed cases. Even though the estimates vary according to the considered sample path, they are very close to the real locations.

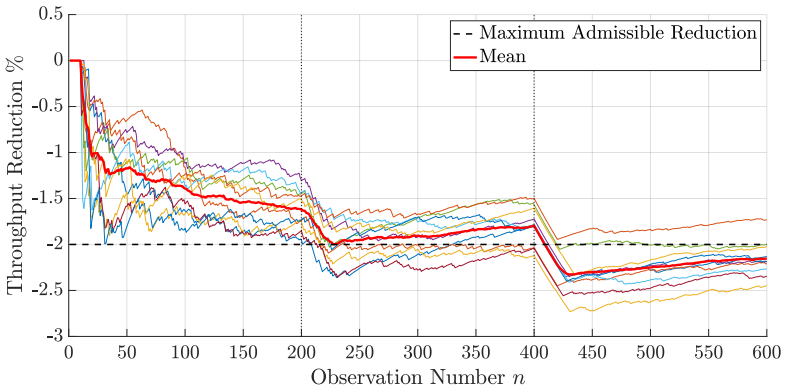
Case M1/P1/0.02. In order to reduce NPE consumptions while limiting the maximum throughput reduction at $\epsilon = 0.02$, the optimal control is $\boldsymbol{\tau}^* = \{0, 21.55\}$ for distribution D1 and $\boldsymbol{\tau}^* = \{13.04, 65.48\}$ for distribution D3. These parameters respectively allow to achieve 24.36% and 36.47% of savings on the objective function. In addition, the following *Implementation Phase* setting is employed: $C_0 = 10\ kJ/part$, $\alpha = 0.05$ and $n^* = 500$.

The goodness of the proposed approach clearly results from an analysis of machine energy performance (cf. figure 5.4a). Specifically, it is evident how the on-line detection of *change points* allows to reverse the initially declining trends of the sample-based energy savings, caused by the net variations in the part arrival process (vertical dotted lines).

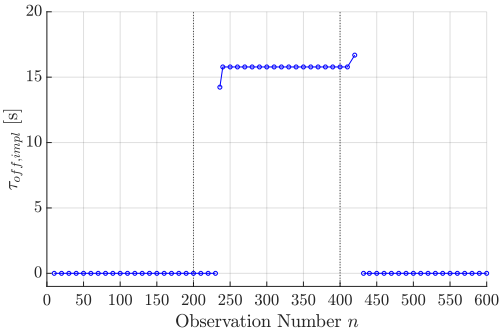
The steps to achieve an effective control adaptation can be appreciated from the details of one replication, reported in figures 5.4c and 5.4d. In particular, immediately after the first *change point* ($n = 200$), the implemented control is strongly biased, since the algorithm processes the whole starvation times dataset. Later, when the *change point score* exceeds the threshold η_{CPD}, the control is set to $\boldsymbol{\tau_{impl}} = \{\infty, \infty\}$. Finally, once properly detected the *change point* ($n \approx 235$), the earlier observations are discarded, allowing to adapt the control to the current starvation times distribution. Note that the same sequence occurs for the second *change point* ($n = 400$).



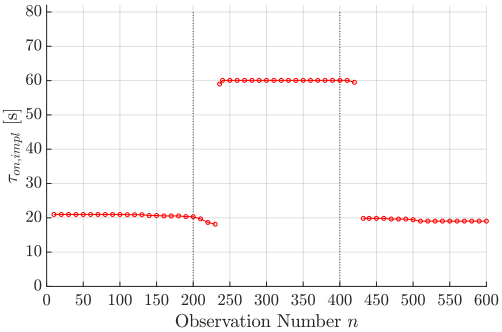
(a) Sample-based energy saving (10 replications).



(b) Sample-based throughput reduction (10 replications).

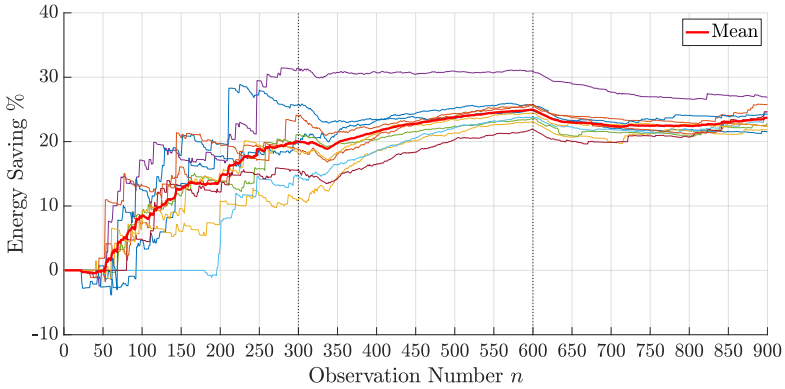


(c) Implemented switch off parameter $\tau_{off,impl}$ (1 replication).

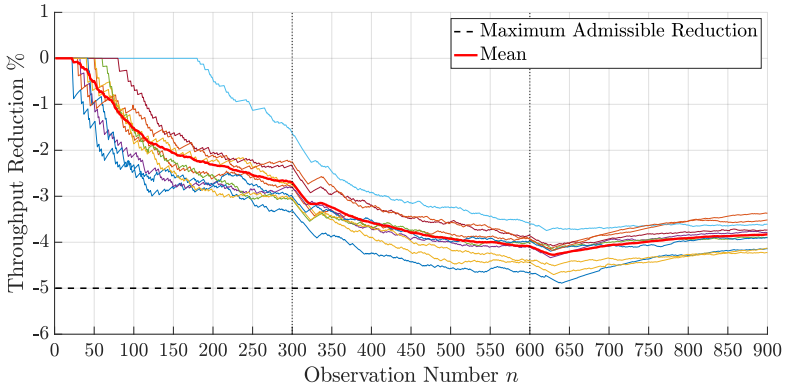


(d) Implemented switch on parameter $\tau_{on,impl}$ (1 replication).

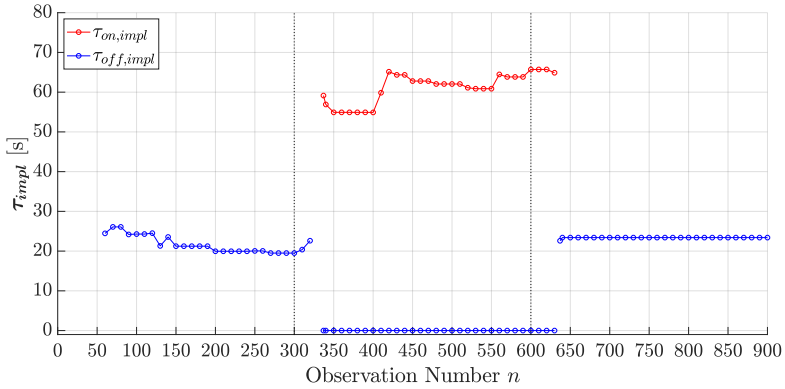
Figure 5.4: M1/P1/0.02 - $C_0 = 10\ kJ/part$, $\alpha = 0.05$ and $n^* = 500$. Dotted lines indicate the true *change point* locations.



(a) Sample-based energy saving (10 replications).



(b) Sample-based throughput reduction (10 replications).



(c) Implemented control $\boldsymbol{\tau_{impl}}$ (1 replication). Note that $\tau_{on,impl} = \infty$ for $n < 335 \wedge n > 635$.

Figure 5.5: M2/P2/0.05 - $C_0 = 20\ kJ/part$, $\alpha = 0.05$ and $n^* = 500$. Dotted lines indicate the true *change point* locations.

Switching towards machine throughput, figure 5.4b shows the capability of the algorithm to guarantee the requested service level. Indeed, the threshold is sometimes slightly exceed only after a change in the arrival process, before the control is again adapted to the new operating environment.

Case M2/P2/0.05. In this case, the *throughput constraint* turns out to be binding only when machine starvation times are modeled with distribution D4. Therefore, while the optimal control for distribution D2 is $\boldsymbol{\tau}^* = \{18.37, \infty\}$, the trade-off solution for distribution D4 is $\boldsymbol{\tau}^* = \{0, 65.74\}$. These parameters respectively allow to achieve 30.67% and 28.73% of savings on the objective function. In addition, the *Implementation Phase* is set according to the following generic configuration: $C_0 = 20\ kJ/part$, $\alpha = 0.05$ and $n^* = 500$.

Here again, the dynamic behavior of the operating environment is effectively handled. Specifically, focusing on a single replication, figure 5.5c shows how the implemented control is successfully modified in response to the net changes in the part arrival process (vertical dotted lines). Note that, for observations $335 \leq n \leq 600$, the variability of the implemented switch on parameter $\tau_{on,impl}$ only results from the need to satisfy the *throughput constraint*.

The benefits of control adaptation can be especially perceived from the clear adjustments in the sample-based energy saving trends near observation $n = 335$ (cf. figure 5.5a). On the contrary, the lack of confidence on the estimated control results in smoother trends after the detection of the second *change point* ($n \approx 635$).

Finally, focusing on machine throughput, figure 5.5b shows how the sharply declining trends after the first *change point* are then mitigated once identified the variation in the arrival process. A similar behavior can be also observed for the second *change point*: after its detection, the sample-based throughput reduction starts to increase.

5.2 Arrivals to an Upstream Buffer

This section finally addresses a slightly different control problem, which results from considering the presence of an upstream buffer in front of the machine, in which parts accumulate. First of all, the analyzed system is described, highlighting the considered modeling assumptions. Since in this environment the on-line control policy from chapter 3 may lead to a non-stationary control problem, a modified objective function is then proposed to guarantee the convergence towards a unique solution. In conclusion, the resulting benefits are assessed with a numerical analysis.

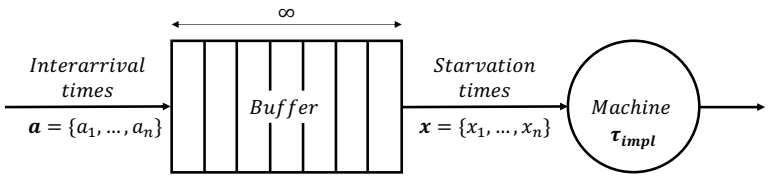


Figure 5.6: Graphical representation of the considered system.

5.2.1 System Description and Assumptions

A single machine processing a single part type is considered. In more detail, it is perfectly reliable, never blocked and some of its modules can be deactivated for energy saving purposes. This machine is still characterized by the energy states and parameters introduced in section P.1, nevertheless the part holding power w_q is no longer included in the model.

An infinite input buffer is now assumed to collect parts in front of the machine. Stochasticity can be therefore introduced considering i.i.d. interarrival times at the buffer, described by random variable A, whose realization is denoted with a. This arrival process, which is not affected by the applied energy efficient control policy, is modeled by the stationary pdf $f_A(a)$. Even though $f_A(a)$ in unknown to the machine, the real time interarrival and starvation times data can be collected in vectors $\boldsymbol{a} = \{a_1, \ldots, a_n\}$ and $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, respectively. A graphical representation of the described system is reported in figure 5.6.

Note that starvation times are now affected by the applied energy efficient control policy. Indeed, the possible occurrence of part waiting times, caused by the introduction of the startup procedure, entails an average decrease of machine starvation periods. This influence has two major consequences:

- The on-line control policy from chapter 3 results in a non-stationary control problem. Indeed, according to the previous consideration, the implementation of new control parameters modifies the behavior of the starvation times data stream, on the basis of which they were estimated. Therefore, the convergence towards an optimal solution cannot be guaranteed, since the control problem may diverge.

- The control effect on the throughput is negligible. Indeed, the occurrence of a part waiting time is counterbalanced by a subsequent reduction of machine starvation. As a consequence, the average cycle time stays basically unchanged. The *throughput constraint* in equation (3.21) can therefore be relaxed.

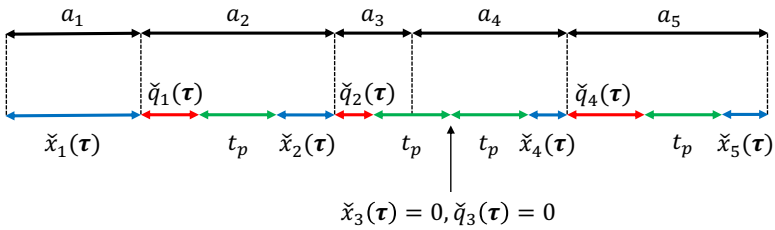


Figure 5.7: Illustrative example of the *recursive function* behavior. Note that $\check{x}_3(\boldsymbol{\tau}) = 0$ since a part is waiting in the buffer at process completion of part 2.

5.2.2 Sample-Based Objective Function

To guarantee the convergence towards an approximate optimal solution, the *traditional* objective function $\hat{g}_k(\boldsymbol{\tau})$ of the *on-line optimization problem* (cf. subsection 3.3.1) is properly modified, allowing to include the control effect on machine starvation. This complex dynamic interaction can be indeed modeled recursively. In particular, starting from the interarrivals time series $\boldsymbol{a}$, the following *recursive function* (cf. figure 5.7) allows to derive the corresponding machine starvation times $\check{\boldsymbol{x}}(\boldsymbol{\tau}) = \{\check{x}_1(\boldsymbol{\tau}), \ldots, \check{x}_n(\boldsymbol{\tau})\}$ when control $\boldsymbol{\tau}$ is implemented:

Recursive step.

$$\check{x}_i(\boldsymbol{\tau}) = \max\left[0, \sum_{j=1}^{i} a_j - \sum_{j=1}^{i-1}(\check{x}_j(\boldsymbol{\tau}) + \check{q}_j(\boldsymbol{\tau}) + t_p)\right] \tag{5.3}$$

$$\check{q}_i(\boldsymbol{\tau}) = \psi[\check{x}_i(\boldsymbol{\tau}), \boldsymbol{\tau}] \tag{5.4}$$

where $\check{q}_i(\boldsymbol{\tau})$ is the *waiting time in a cycle* and $\psi(x, \boldsymbol{\tau})$ is the *waiting time function* introduced in equation (4.8).

Base case.

$$\check{x}_1(\boldsymbol{\tau}) = a_1 \tag{5.5}$$

$$\check{q}_1(\boldsymbol{\tau}) = \psi[\check{x}_1(\boldsymbol{\tau}), \boldsymbol{\tau}] \tag{5.6}$$

On the basis of the the collected interarrivals $\boldsymbol{a} = \{a_1, \ldots, a_n\}$, this *recursive function* can then be employed to formulate a *sample-based* objective function $\check{g}_k(\boldsymbol{\tau})$:

$$\check{g}_k(\boldsymbol{\tau}) = \frac{\sum_{i=1}^{n} \phi[\check{x}_i(\boldsymbol{\tau}), \boldsymbol{\tau}]}{n} \tag{5.7}$$

where $\phi(x, \boldsymbol{\tau})$ is the *energy output function* introduced in equation (3.32). Physically speaking, $\check{g}_k(\boldsymbol{\tau})$ represents the average energy per part the machine would have historically consumed, if it had been controlled with parameters $\boldsymbol{\tau}$.

From an implementation perspective, at each algorithm iteration k, the *Learning Phase* is skipped and a genetic algorithm is employed to carry out the minimization task, since the *sample-based* objective function $\check{g}_k(\boldsymbol{\tau})$ cannot be expressed in closed-form. It is worth highlighting that the resulting control $\check{\boldsymbol{\tau}}_{\boldsymbol{k}}$ is approximate, being obtained from an optimization over the sample-path. Finally, focusing on the *Implementation Phase*, $\check{\boldsymbol{\tau}}_{\boldsymbol{k}}$ is implemented only if the following condition holds:

$$\check{g}_k(\boldsymbol{\tau_{impl}}) - \check{g}_k(\check{\boldsymbol{\tau}}_{\boldsymbol{k}}) \geq C(n) \tag{5.8}$$

where $C(n)$ is the *implementation cost function* from equation (3.34).

5.2.3 Results

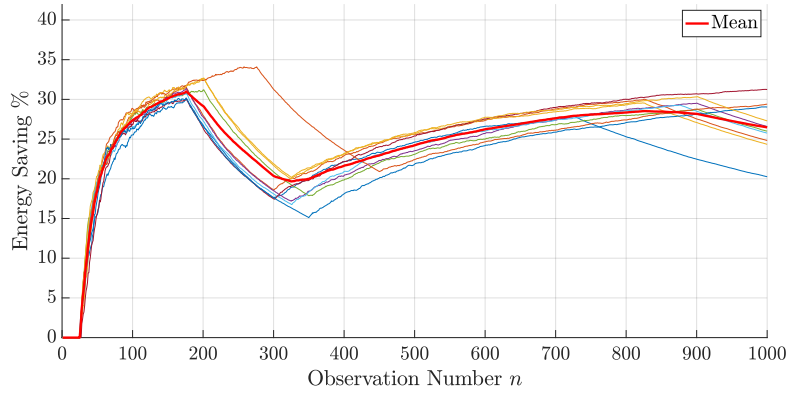
The benefits brought by the the introduction of the *sample-based* objective function are finally assessed numerically. In this regard, two production cases are considered, obtained by combining a machine tool and an interarrival times distribution $f_A(a)$. Whilst the former case is similar to those treated in the previous chapters, the latter is instead conceived to be a more faithful representation of a real manufacturing environment. For each case, the on-line algorithm is applied with both the *sample-based* objective function $\check{g}_k(\boldsymbol{\tau})$ and the *traditional* objective function $\hat{g}_k(\boldsymbol{\tau})$. In particular, experiments are composed by 10 replications and common random numbers are employed to guarantee comparability. The investigated KPIs are the machine energy consumption, the throughput and the average buffer level.

Case 1. Assume that the interarrival times at machine M1 ($t_p = 168\ s$) are modeled by distribution D1, delayed by the part processing time t_p:

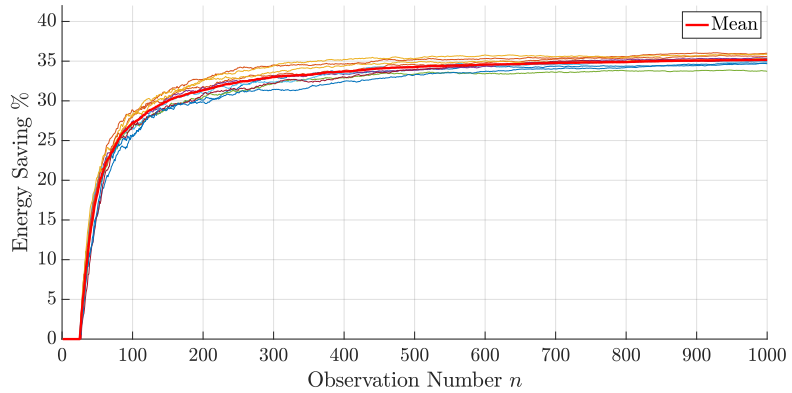
$$f_A(a) \sim t_p + Weibull(5, 49.011) \tag{5.9}$$

In this case, the considered machine is clearly underutilized. In particular, its upstream buffer would be always empty if the AON policy were adopted. To pursue energy efficient operation, the on-line algorithm is then iterated every $\Delta n = 25$ new observations, until 1000 parts are processed. In addition, the following *Implementation Phase* settings are considered[2]: $C_0 = 10\ kJ/part$, $\alpha = 0.05$ and $n^* = \infty$.

[2]Note that α is employed only with the *traditional* objective function.

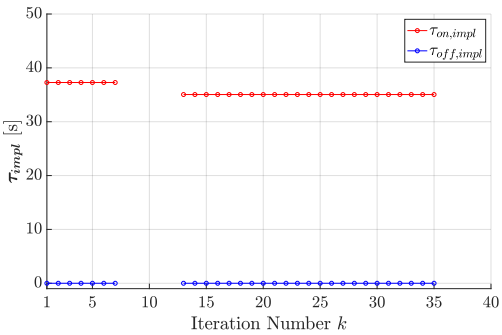


(a) *Traditional* objective function.

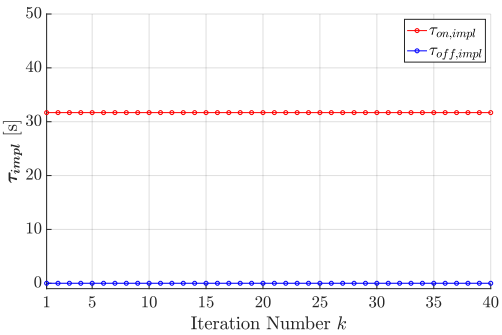


(b) *Sample-based* objective function.

Figure 5.8: Case 1 - Sample-based energy saving (10 replications).

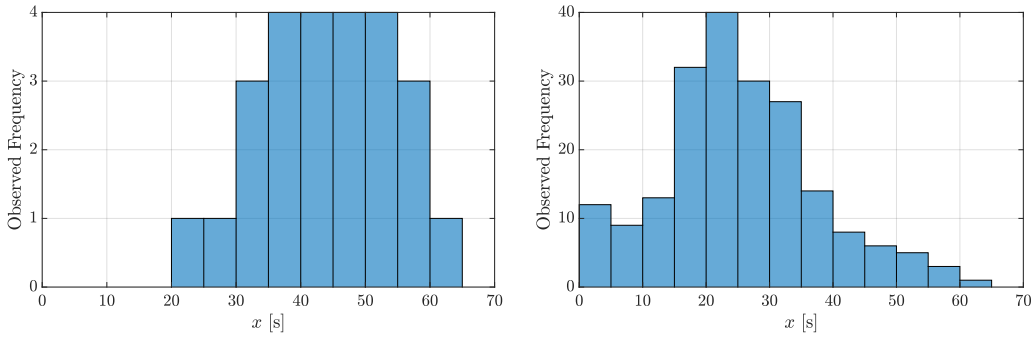


(a) *Traditional* objective function.



(b) *Sample-based* objective function.

Figure 5.9: Case 1 - Implemented control $\boldsymbol{\tau_{impl}}$ at each algorithm iteration $k = n/25$ (1 replication). Missing points correspond to $\boldsymbol{\tau_{impl}} = \{\infty, \infty\}$.



(a) $k = 1,\ n = 25$. (b) $k = 8,\ n = 200$.

Figure 5.10: Case 1 - Frequency histograms of $\boldsymbol{x}$ (1 replication).

Results, in terms of sample-based energy saving with respect to the AON policy, are respectively reported in figures 5.8a and 5.8b. While the savings are initially similar, an abrupt inversion in the trends obtained with the *traditional* objective function can be then identified. Focusing on the details of a single replication (cf. figure 5.9a), it can be indeed noticed how at iteration $k = 8$ (i.e. $n = 200$) the EEC is suddenly interrupted. This undesirable change actually results from the control influence on machine starvation times. In particular, the parameters implemented at the first iteration (i.e. $n = 25$) modify the behavior of the starvation times data stream, as shown in the frequency histograms of figure 5.10. Since, at iteration $k = 8$, a good portion of observations is located near the origin, the algorithm identifies that it is optimal to never switch off the machine (i.e. AON policy). Once set the control to $\boldsymbol{\tau_{impl}} = \{\infty, \infty\}$, machine starvation times get back to increase and the limit cycle starts again. On the contrary, the results obtained with the *sample-based* objective function are not affected by this issue, since it is formulated to directly account for the control effect on machine starvation. Specifically, focusing on the same single replication, figure 5.9b shows that no additional changes occur after the the first control implementation.

Moving on to the others KPIs, no throughput reductions were identified and the 95% confidence intervals for the average buffer level are respectively 0.078 ± 0.005 and 0.103 ± 0.006. These results are coherent with the considered arrival process, since the accumulation of parts is only due to the effect of the implemented control policy.

Case 2. In this last numerical analysis, a more realistic production case is finally addressed. It is indeed assumed that the arrival process at the buffer of machine M2 ($t_p = 168\ s$) is generated by three unreliable parallel machines

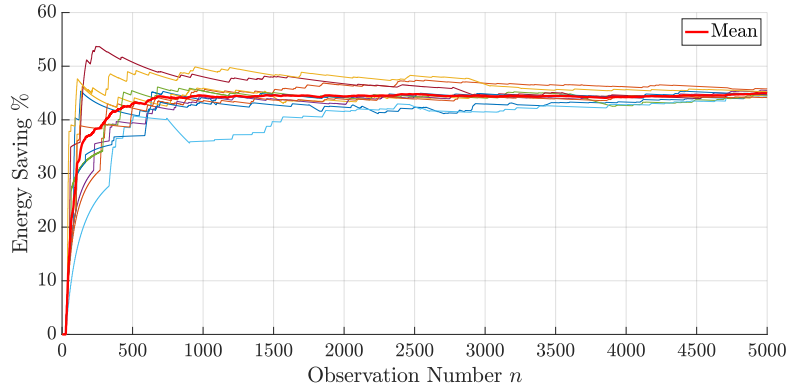


Figure 5.11: Case 2 - Sample-based energy saving (10 replications). Note that the *traditional* and the *sample-based* objective functions provide the exact same results.

with the same parameters. In detail, their processing time is $t_p = 564\ s$ and their failure metrics are described by exponential distributions with $MTTF = 10\ h$ and $MTTR = 30\ min$, i.e. the availability is 0.95. Therefore, the production rate in isolation of the upstream portion of the system is 18.24 $parts/h$, whilst that of the controlled machine is 21.43 $parts/h$. It is evident that the parallel machines constitute the bottleneck of the system. Focusing on the single replication, the algorithm is initially iterated every $\Delta n = 25$ observations. Upon completion of the 500th part, this interval is finally increased to $\Delta n = 100$. The replication terminates when 5000 parts are processed, therefore simulation time is approximately 233 hours. In addition, the *Implementation Phase* is set according to the following generic configuration: $C_0 = 10\ kJ/part$, $\alpha = 0.05$ and $n^* = \infty$.

In this second case, the *traditional* and the *sample-based* objective functions provide the exact same results. In particular, the identified optimal control is always $\boldsymbol{\tau} = \{0, \infty\}$. Therefore, at process completion, the machine is immediately switched off if starved and the startup procedure is then triggered by part arrival.

For what concerns machine energy performance, figure 5.11 shows that the EEC allows to reduce the NPE consumptions of approximately 45% in the long range. Specifically, at observation $n = 5000$, the 95% confidence interval for the mean energy saving with respect to the AON policy is $726.02 \pm 22.24\ MJ$. Assuming a gross industrial electricity price of 0.26 €/kWh [81], this corresponds to an economy of about 50 €. Of course, more noticeable monetary results would be achieved if the policy were applied to a more energy demanding production equipment.

It is also worth noting that energy efficiency is pursued without affecting

the production rate of the system. In particular, upon completion of the simulations, the 95% confidence interval for the average throughput of the controlled machine is 18.24 ± 0.08 $parts/h$, which exactly corresponds to the arrival frequency at the buffer.

Finally, the 95% confidence interval for the average buffer level is 0.44 ± 0.03. This results from the combined effect of the arrival process and the implemented control policy. Of course, a higher capacity utilization of the controlled machine would progressively reduce the energy saving potentials.

5.3 Conclusions

This chapter is aimed at making the on-line control policy effective when the upstream production process is non-stationary or decoupled from the machine by an input buffer. First of all, the possibility to systematically adapt the control in response to changes in the starvation times distribution is investigated. In this regard, a *change point detection* method (RuLSIF) from literature [80] is exploited. Indeed, being unsupervised, on-line and nonparametric, it completely complies with the requirements of the considered problem. The effectiveness of this method is affected by the choice of four main parameters, which must be calibrated addressing the trade-off between accuracy and responsiveness. On the basis of an experimental campaign, the following combination is suggested: $\alpha_{CPD} = 0.01$, $n_{CPD} = 30$, $k_{CPD} = 5$ and $\eta_{CPD} = 25$. Numerical results show the goodness of the proposed approach. In particular, the detection of *change points* allows to adapt the control to the new starvation times distribution, thus avoiding the deterioration of machine performance.

Later, a *sample-based* objective function is proposed to guarantee the convergence towards a unique approximate control, when an infinite input buffer is assumed to collect parts in front of the machine. In particular, it is recursively formulated on the basis of the collected interarrivals time series, allowing to account for the control effect on machine starvation. A first numerical analysis shows that the *traditional* objective function may result in an alternate switching between two control configurations, which strongly affects the achievable savings. On the contrary, the results obtained with the *sample-based* objective function are not affected by undesirable behavior. Finally, a more realistic production case is addressed, in which the arrival process is generated by three unreliable parallel machines. Considering the AON policy as a reference, the on-line algorithm allows to achieve approximately 45% of savings with both objective functions, without influencing the throughput of the system.

Chapter 6

Conclusions

The objective of this work is to move a step forward in the energy efficient control of machine states under uncertainty in part arrival. Specifically, the control problem is addressed in an innovative on-line framework, providing solutions in real time while acquiring information from the real system. In this regard, an on-line time-based policy is presented, which improves and extends an existing algorithm from literature [43, 63]. By means of deep analytical and numerical analyses, the following enrichments are introduced:

- *Learning Phase* - Being crucial to guarantee the effectiveness of the whole algorithm, three methods to estimate the KDE bandwidth parameter h are compared. In order achieve robust results, it is suggested to employ a mixture of them in a black-box version of the algorithm. In addition, the issue of uncertainty on density estimation is explicitly addressed by identifying the bandwidth probability density function $f_H(h)$. This information provides an additional unprecedented dimension to the problem, but increases its computational complexity.

- *Optimization Phase* - A solving algorithm is developed to exactly select the optimal control in presence of constraints on machine performance. In particular, a condition on the expected throughput allows to improve machine sustainability, while guaranteeing a user-defined service level. Moreover, two additional constraints are formulated to mitigate the risk of deteriorating machine performance over the single *cycle*. Their influence on the optimal control is studied numerically.

- *Implementation Phase* - The *Bootstrap CI Implementation Policy* is developed in order to provide a greater robustness with respect to the statistical paired-t test employed in literature. In addition, it also guarantees parameters convergence when a constraint must be satisfied.

Several numerical analyses are performed to evaluate the policy effectiveness and computational requirements. In particular, it allows to achieve significant savings with respect to the AON policy, within the limits given by the imposed constraints. Moreover, the shopfloor applicability is ensured by an overall enhancement in computational times.

The possibility to adapt the control in response to changes in the upstream production process is also investigated. In detail, a *change point detection* method (RuLSIF) [80] is included into the algorithm, in order to identify variations in the distribution of starvation times. After an appropriate calibration, numerical results show that the detection of *change points* allows to avoid the deterioration of machine performance, caused by a transition in the part arrival process.

In conclusion, a slightly different control problem is addressed, which results from considering the presence of an upstream buffer in front of the machine. In this environment, the proposed policy may not converge towards a unique solution due to the control influence on starvation times. Therefore, the collected interarrivals data are employed to formulate a *sample-based* objective function which, as shown by the numerical results, guarantees the implementation a stable optimal control.

Besides the scientific results, this work contributes to the removal of some critical barriers, which limit the practical implementation of machine state control in the industrial sector. The proposed policy can be indeed applied to a wide range of production cases, being flexible and highly autonomous.

Two main future research directions can be outlined:

- *Uncertainty on Density Estimation* - The potential applications of the bandwidth probability density function $f_H(h)$ in the control problem require a deeper investigation. For example, this additional information can be exploited to set a stopping criteria for assessing the algorithm convergence. In addition, a numerical analysis of the *Bandwidth Implementation Policy* can be also carried out.

- *Time Variant Behavior* - Future developments may be devoted to the application of *change point detection* methods to other performance of interest. This may allow to identify variations in the system or in the external environment in a more effective way. Moreover, it could be also possible to classify the different states of the process that generates the arrivals at the machine.

Appendix A

Detailed Simulation Results of Chapter 3

In this appendix, detailed simulation results from chapter 3 are collected as follows:

- Table A.1: M1/D1/1 - Sample-based comparison of on-line algorithm energy consumption and throughput for different settings.
- Table A.2: M1/D1/1 - Effects of factors C_0, α and n^* over the number of observations before implementing the control parameters and the mean number of changes.
- Table A.3: M2/D2/1 - Sample-based comparison of on-line algorithm energy consumption and throughput for different settings.
- Table A.4: M2/D2/1 - Effects of factors C_0, α and n^* over the number of observations before implementing the control parameters and the mean number of changes.
- Table A.5: M3/D3/0.02 - Sample-based energy consumption and throughput for different settings.
- Table A.6: M3/D3/0.02 - Effects of factors C_0, α and n^* over the number of observations before implementing the control parameters and the mean number of changes.

Table A.1: M1/D1/1 - Sample-based comparison of on-line algorithm energy consumption and throughput for different settings (95% CI, 10 replications).

C_0 [kJ/part]	α	Implementation Policy	n^*	Performance	Observation number n				
					100	250	500	1000	2500
AON Policy				Energy [kJ/part]	241.18 ± 3.68	239.88 ± 2.25	240.53 ± 1.69	240.41 ± 1.57	240.82 ± 0.80
				Throughput [parts/h]	16.90 ± 0.05	16.91 ± 0.03	16.90 ± 0.02	16.91 ± 0.02	16.90 ± 0.01
/	1	/	/	Energy [kJ/part]	182.06 ± 1.41	178.49 ± 0.67	177.20 ± 0.52	176.65 ± 0.18	176.19 ± 0.19
				Throughput [parts/h]	16.27 ± 0.05	16.24 ± 0.05	16.23 ± 0.04	16.22 ± 0.03	16.22 ± 0.02
0	0.05	Bootstrap CI	/	Energy [kJ/part]	182.46 ± 1.15	178.74 ± 0.67	177.40 ± 0.54	176.80 ± 0.18	176.28 ± 0.21
				Throughput [parts/h]	16.28 ± 0.10	16.25 ± 0.06	16.25 ± 0.05	16.25 ± 0.04	16.24 ± 0.03
		Paired-t	/	Energy [kJ/part]	182.29 ± 1.21	178.71 ± 0.80	177.41 ± 0.62		
				Throughput [parts/h]	16.27 ± 0.10	16.24 ± 0.09	16.23 ± 0.06		
	0.20	Bootstrap CI	/	Energy [kJ/part]	182.21 ± 1.16	178.56 ± 0.68	177.30 ± 0.55	176.74 ± 0.18	176.23 ± 0.20
				Throughput [parts/h]	16.31 ± 0.08	16.28 ± 0.06	16.26 ± 0.05	16.25 ± 0.04	16.23 ± 0.03
		Paired-t	/	Energy [kJ/part]	182.22 ± 1.27	178.52 ± 0.68	177.27 ± 0.55		
				Throughput [parts/h]	16.27 ± 0.10	16.24 ± 0.07	16.24 ± 0.05		
36	0.05	Bootstrap CI	∞	Energy [kJ/part]	191.01 ± 4.26	182.15 ± 1.69	179.16 ± 0.97	177.81 ± 0.48	176.82 ± 0.27
				Throughput [parts/h]	16.41 ± 0.11	16.08 ± 0.34	16.07 ± 0.28	16.26 ± 0.07	16.25 ± 0.07
			2500	Energy [kJ/part]	191.01 ± 4.26	182.15 ± 1.69	179.16 ± 0.97	177.81 ± 0.48	176.82 ± 0.27
				Throughput [parts/h]	16.41 ± 0.11	16.31 ± 0.08	16.27 ± 0.07	16.26 ± 0.07	16.25 ± 0.07
			500	Energy [kJ/part]	188.69 ± 3.41	181.24 ± 1.27	178.64 ± 0.77	177.47 ± 0.36	176.54 ± 0.25
				Throughput [parts/h]	16.36 ± 0.09	16.27 ± 0.07	16.24 ± 0.06	16.23 ± 0.06	16.23 ± 0.04
		Paired-t	/	Energy [kJ/part]	193.32 ± 3.47	183.19 ± 1.52	179.76 ± 1.03		
				Throughput [parts/h]	16.40 ± 0.12	16.28 ± 0.10	16.24 ± 0.10		

C_0 [kJ/part]	α	Implementation Policy	n^*	Performance	Observation number n				
					100	250	500	1000	2500
36	0.20	Bootstrap CI	∞	Energy [kJ/part]	184.25 ± 1.42	179.62 ± 0.69	178.01 ± 0.58	177.36 ± 0.47	176.81 ± 0.47
				Throughput [parts/h]	16.28 ± 0.11	16.22 ± 0.11	16.20 ± 0.10	16.19 ± 0.10	16.19±0.10
			2500	Energy [kJ/part]	184.25 ± 1.42	179.62 ± 0.69	178.01 ± 0.58	177.36 ± 0.28	176.81 ± 0.21
				Throughput [parts/h]	16.28 ± 0.11	16.22 ± 0.11	16.20 ± 0.10	16.19 ± 0.10	16.19±0.10
			500	Energy [kJ/part]	184.25 ± 1.42	179.62 ± 0.69	178.01 ± 0.58	177.19 ± 0.47	176.41 ± 0.47
				Throughput [parts/h]	16.28 ± 0.11	16.22 ± 0.11	16.20 ± 0.10	16.22 ± 0.08	16.22±0.04
		Paired-t	/	Energy [kJ/part]	184.52 ± 1.38	179.94 ± 0.85	178.37 ± 0.78		
				Throughput [parts/h]	16.22 ± 0.12	16.16 ± 0.12	16.14 ± 0.11		
48	0.05	Bootstrap CI	∞	Energy [kJ/part]	206.07 ± 7.97	188.92 ± 4.36	182.69 ± 2.37	179.66 ± 1.09	177.67 ± 0.62
				Throughput [parts/h]	16.49 ± 0.18	16.30 ± 0.12	16.23 ± 0.10	16.20 ± 0.09	16.18±0.09
			2500	Energy [kJ/part]	201.65 ± 4.90	186.56 ± 1.96	181.49 ± 1.19	179.06 ± 0.69	177.41 ± 0.50
				Throughput [parts/h]	16.44 ± 0.13	16.28 ± 0.11	16.22 ± 0.10	16.19 ± 0.09	16.18±0.09
			500	Energy [kJ/part]	196.87 ± 5.06	184.67 ± 2.09	180.53 ± 1.24	178.742 ± 0.54	176.91 ± 0.20
				Throughput [parts/h]	16.41 ± 0.12	16.28 ± 0.10	16.23 ± 0.10	16.23 ± 0.07	16.23±0.03
		Paired-t	/	Energy [kJ/part]	205.30 ± 8.60	188.66 ± 4.59	182.53 ± 2.48		
				Throughput [parts/h]	16.48 ± 0.18	16.30 ± 0.12	16.23 ± 0.10		
	0.20	Bootstrap CI	∞	Energy [kJ/part]	194.42 ± 6.17	183.69 ± 2.63	180.01 ± 1.58	178.30 ± 0.90	177.11 ± 0.63
				Throughput [parts/h]	16.35 ± 0.14	16.23 ± 0.10	16.19 ± 0.09	16.17 ± 0.09	16.16±0.09
			2500	Energy [kJ/part]	194.42 ± 6.17	183.69 ± 2.63	180.01 ± 1.58	178.30 ± 0.90	177.09 ± 0.63
				Throughput [parts/h]	16.35 ± 0.14	16.23 ± 0.10	16.19 ± 0.09	16.17 ± 0.09	16.17±0.08
			500	Energy [kJ/part]	191.14 ± 3.53	182.37 ± 1.60	179.36 ± 1.07	177.80 ± 0.52	176.67 ± 0.21
				Throughput [parts/h]	16.35 ± 0.13	16.24 ± 0.10	16.21 ± 0.09	16.21 ± 0.06	16.22±0.04
		Paired-t	/	Energy [kJ/part]	194.60 ± 6.04	183.94 ± 2.45	180.29 ± 1.45		
				Throughput [parts/h]	16.32 ± 0.15	16.20 ± 0.11	16.16 ± 0.10		

Table A.2: M1/D1/1 - Effects of factors C_0, α and n^* over the number of observations before implementing the control parameters and the mean number of changes.

C_0 [kJ/part]	α	Implementation Policy	n^*	Replication number										Mean Number of Changes
				1	2	3	4	5	6	7	8	9	10	
0	0.05	Bootstrap CI	/	10	10	10	10	10	10	10	10	10	10	2.2
		Paired-t	/	10	10	10	10	10	10	10	10	10	10	1.7
	0.20	Bootstrap CI	/	10	10	10	10	10	10	10	10	10	10	3.1
		Paired-t	/	10	10	10	10	10	10	10	10	10	10	2.0
36	0.05	Bootstrap CI	∞	10	30	10	20	30	30	20	50	20	30	1.0
			2500	10	30	10	20	30	30	20	50	20	30	1.6
			500	10	30	10	20	30	30	10	30	20	20	1.7
		Paired-t	/	20	30	30	20	30	30	20	50	20	30	1.0
	0.20	Bootstrap CI	∞	10	20	10	20	20	10	10	10	10	10	1.0
			2500	10	20	10	20	20	10	10	10	10	10	1.8
			500	10	20	10	20	20	10	10	10	10	10	2.3
		Paired-t	/	10	20	10	20	20	10	10	10	10	10	1.0
48	0.05	Bootstrap CI	∞	20	40	50	50	30	50	50	120	30	40	1.0
			2500	20	40	40	50	30	40	50	60	20	40	1.4
			500	20	40	30	30	30	40	20	60	20	30	1.5
		Paired-t	/	20	40	50	50	30	50	50	120	20	40	1.0
	0.20	Bootstrap CI	∞	20	30	10	30	30	40	10	60	20	30	1.0
			2500	20	30	10	30	30	40	10	60	20	30	1.4
			500	20	30	10	20	30	30	10	40	20	30	2.3
		Paired-t	/	20	30	10	30	30	40	10	60	20	30	1.0

Table A.3: M2/D2/1 - Sample-based comparison of on-line algorithm energy consumption and throughput for different settings (95% CI, 10 replications).

C_0 [kJ/part]	α	Implementation Policy	n^*	Performance	Observation number n				
					100	250	500	1000	2500
AON Policy				Energy [kJ/part]	376.96 ± 74.63	400.32 ± 40.23	413.12 ± 33.55	424.52 ± 22.67	424.72 ± 17.58
				Throughput [parts/h]	17.83 ± 0.58	17.62 ± 0.31	17.52 ± 0.26	17.43 ± 0.17	17.43 ± 0.13
/	1	/	/	Energy [kJ/part]	322.78 ± 41.63	300.25 ± 24.33	298.10 ± 14.98	297.66 ± 11.73	296.59 ± 7.24
				Throughput [parts/h]	17.21 ± 0.63	17.04 ± 0.34	16.93 ± 0.25	16.83 ± 0.18	16.80 ± 0.14
0	0.05	Bootstrap CI	/	Energy [kJ/part]	326.08 ± 49.42	303.93 ± 29.27	299.95 ± 18.71	298.73 ± 13.54	297.66 ± 7.87
				Throughput [parts/h]	17.57 ± 0.64	17.20 ± 0.34	17.02 ± 0.28	16.89 ± 0.18	16.84 ± 0.15
		Paired-t	/	Energy [kJ/part]	353.34 ± 62.99	331.57 ± 40.00	314.32 ± 23.66		
				Throughput [parts/h]	17.67 ± 0.62	17.24 ± 0.36	16.99 ± 0.29		
	0.35	Bootstrap CI	/	Energy [kJ/part]	323.60 ± 55.44	301.31 ± 29.75	298.75 ± 18.06	298.04 ± 12.28	296.78 ± 7.72
				Throughput [parts/h]	17.46 ± 0.64	17.14 ± 0.33	16.97 ± 0.26	16.85 ± 0.18	16.82 ± 0.14
		Paired-t	/	Energy [kJ/part]	319.70 ± 49.19	299.82 ± 26.66	289.04 ± 16.74		
				Throughput [parts/h]	17.43 ± 0.62	17.12 ± 0.34	16.95 ± 0.26		
17	0.05	Bootstrap CI	∞	Energy [kJ/part]	338.51 ± 49.36	312.34 ± 30.37	303.89 ± 18.79	301.05 ± 13.93	298.69 ± 7.97
				Throughput [parts/h]	17.61 ± 0.65	17.21 ± 0.34	17.02 ± 0.28	16.89 ± 0.19	16.85 ± 0.15
			5000	Energy [kJ/part]	338.51 ± 49.36	312.32 ± 30.33	304.08 ± 19.11	301.01 ± 13.88	298.64 ± 7.88
				Throughput [parts/h]	17.61 ± 0.65	17.21 ± 0.34	17.02 ± 0.28	16.89 ± 0.19	16.86 ± 0.15
			500	Energy [kJ/part]	338.51 ± 49.36	309.17 ± 27.20	302.50 ± 17.85	300.12 ± 12.95	298.13 ± 7.50
				Throughput [parts/h]	17.61 ± 0.65	17.20 ± 0.34	17.01 ± 0.28	16.88 ± 0.19	16.84 ± 0.15
		Paired-t	/	Energy [kJ/part]	368.04 ± 61.95	349.35 ± 35.27	323.58 ± 21.54		
				Throughput [parts/h]	17.80 ± 0.63	17.34 ± 0.37	17.04 ± 0.29		

C_0 [kJ/part]	α	Implementation Policy	n^*	Performance	Observation number n				
					100	250	500	1000	2500
17	0.35	Bootstrap CI	∞	Energy [kJ/part]	330.81 ± 53.04	308.87 ± 29.81	302.16 ± 18.14	300.07 ± 13.38	298.32 ± 7.97
				Throughput [parts/h]	17.48 ± 0.64	17.17 ± 0.34	17.01 ± 0.27	16.89 ± 0.17	16.85±0.14
			5000	Energy [kJ/part]	330.81 ± 53.04	308.89 ± 29.80	302.27 ± 18.45	299.99 ± 13.31	298.21 ± 7.96
				Throughput [parts/h]	17.48 ± 0.64	17.17 ± 0.34	17.00 ± 0.27	16.88 ± 0.18	16.84±0.14
			500	Energy [kJ/part]	329.99 ± 53.30	305.57 ± 28.62	301.21 ± 17.74	299.26 ± 12.33	297.47 ± 7.63
				Throughput [parts/h]	17.48 ± 0.64	17.16 ± 0.34	16.99 ± 0.27	16.86 ± 0.17	16.82±0.14
		Paired-t	/	Energy [kJ/part]	327.80 ± 51.33	331.15 ± 31.96	305.22 ± 18.59		
				Throughput [parts/h]	17.46 ± 0.66	17.14 ± 0.37	16.97 ± 0.29		
73	0.05	Bootstrap CI	∞	Energy [kJ/part]	344.24 ± 52.30	317.29 ± 33.20	306.48 ± 19.79	302.19 ± 14.51	299.31 ± 8.14
				Throughput [parts/h]	17.67 ± 0.65	17.24 ± 0.33	17.04 ± 0.27	16.90 ± 0.19	16.86±0.15
			5000	Energy [kJ/part]	344.24 ± 52.30	317.29 ± 33.20	306.48 ± 19.79	302.19 ± 14.51	299.31 ± 8.14
				Throughput [parts/h]	17.67 ± 0.65	17.24 ± 0.33	17.04 ± 0.27	16.90 ± 0.19	16.86±0.15
			500	Energy [kJ/part]	344.36 ± 52.27	314.68 ± 31.39	305.12 ± 19.09	301.60 ± 14.03	298.93 ± 7.99
				Throughput [parts/h]	17.66 ± 0.64	17.23 ± 0.33	17.04 ± 0.27	16.90 ± 0.18	16.86±0.14
		Paired-t	/	Energy [kJ/part]	376.96 ± 74.63	397.87 ± 38.82	396.58 ± 28.81		
				Throughput [parts/h]	17.83 ± 0.58	17.59 ± 0.34	17.43 ± 0.30		
	0.35	Bootstrap CI	∞	Energy [kJ/part]	337.07 ± 53.11	311.94 ± 30.01	303.12 ± 19.88	300.40 ± 14.52	298.20 ± 8.45
				Throughput [parts/h]	17.51 ± 0.65	17.16 ± 0.34	17.00 ± 0.26	16.88 ± 0.17	16.85±0.13
			5000	Energy [kJ/part]	337.07 ± 53.11	308.90 ± 30.25	301.61 ± 18.69	299.62 ± 13.63	297.94 ± 8.25
				Throughput [parts/h]	17.51 ± 0.65	17.16 ± 0.33	17.00 ± 0.26	16.88 ± 0.17	16.85±0.13
			500	Energy [kJ/part]	337.07 ± 53.11	308.69 ± 30.11	301.76 ± 18.88	299.54 ± 13.46	297.28 ± 7.89
				Throughput [parts/h]	17.51 ± 0.65	17.16 ± 0.34	16.99 ± 0.26	16.86 ± 0.18	16.82±0.14
		Paired-t	/	Energy [kJ/part]	352.70 ± 53.46	352.81 ± 29.22	333.40 ± 16.68		
				Throughput [parts/h]	17.74 ± 0.67	17.39 ± 0.43	17.13 ± 0.37		

Table A.4: M2/D2/1 - Effects of factors C_0, α and n^* over the number of observations before implementing the control parameters and the mean number of changes.

C_0 [kJ/part]	α	Implementation Policy	n^*	Replication number										Mean Number of Changes
				1	2	3	4	5	6	7	8	9	10	
0	0.05	Bootstrap CI	/	50	40	20	50	60	40	60	20	60	150	5.3
		Paired-t	/	50	220	60	50	50	190	70	70	120	180	2.7
	0.35	Bootstrap CI	/	20	30	20	20	60	30	40	10	60	130	8.0
		Paired-t	/	20	30	10	20	40	30	10	10	40	120	4.1
17	0.05	Bootstrap CI	∞	50	80	20	50	60	60	70	30	60	160	1.3
			5000	50	80	20	50	60	60	70	30	60	160	1.6
			500	50	80	20	50	60	60	70	30	60	150	3.3
		Paired-t	/	120	260	60	140	100	220	110	100	140	200	1.0
	0.35	Bootstrap CI	∞	20	40	20	20	60	40	60	30	60	150	1.5
			5000	20	40	20	20	60	40	60	30	60	150	2.5
			500	20	40	20	20	60	40	60	20	60	130	5.7
		Paired-t	/	20	40	20	20	40	40	60	30	40	160	1.4
73	0.05	Bootstrap CI	∞	50	80	40	110	60	70	70	50	70	180	1.0
			5000	50	80	40	110	60	70	70	50	70	180	1.2
			500	50	80	40	90	60	60	70	50	70	160	2
		Paired-t	/	/	/	/	130	360	/	440	490	390	440	/	0.6
	0.35	Bootstrap CI	∞	20	50	20	40	60	40	70	30	60	160	1.2
			5000	20	50	20	40	60	40	70	30	60	150	1.3
			500	20	50	20	40	60	40	70	30	60	150	3.6
		Paired-t	/	150	400	30	220	/	250	110	30	80	210	0.9

Table A.5: M3/D3/0.02 - Sample-based energy consumption and throughput for different settings (95% CI, 10 replications).

C_0 [kJ/part]	α	n^*	Performance	Observation number n				
				100	250	500	1000	2500
AON Policy			Energy [kJ/part]	205.55 ± 19.60	206.82 ± 7.32	207.59 ± 5.47	2016.17 ± 4.74	207.09 ± 2.35
			Throughput [parts/h]	32.92 ± 1.35	32.75 ± 0.50	32.70 ± 0.36	32.79 ± 0.31	32.72 ± 0.16
/	1	/	Energy [kJ/part]	153.88 ± 13.52	150.25 ± 5.76	150.35 ± 4.03	148.90 ± 3.31	148.94 ± 1.49
			Throughput [parts/h]	32.35 ± 1.37	32.13 ± 0.54	32.04 ± 0.37	32.14 ± 0.31	32.07 ± 0.15
0	0.05	/	Energy [kJ/part]	155.47 ± 12.89	151.23 ± 5.47	150.89 ± 4.03	149.25 ± 3.30	149.12 ± 1.51
			Throughput [parts/h]	32.38 ± 1.39	32.15 ± 0.55	32.05 ± 0.37	32.15 ± 0.31	32.07 ± 0.15
	0.20	/	Energy [kJ/part]	154.49 ± 13.54	150.71 ± 5.74	150.62 ± 4.04	149.08 ± 3.29	149.04 ± 1.50
			Throughput [parts/h]	32.36 ± 1.38	32.13 ± 0.55	32.04 ± 0.37	32.15 ± 0.31	32.07 ± 0.15
25	0.05	∞	Energy [kJ/part]	163.61 ± 13.63	156.70 ± 4.29	154.91 ± 2.71	152.71 ± 2.91	152.58 ± 2.69
			Throughput [parts/h]	32.52 ± 1.38	32.27 ± 0.60	32.19 ± 0.45	32.29 ± 0.37	32.22 ± 0.18
		2500	Energy [kJ/part]	163.61 ± 13.63	156.59 ± 4.44	154.34 ± 3.33	151.87 ± 3.33	150.47 ± 1.73
			Throughput [parts/h]	32.52 ± 1.38	32.27 ± 0.59	32.16 ± 0.41	32.26 ± 0.33	32.13 ± 0.15
		500	Energy [kJ/part]	162.89 ± 13.19	155.54 ± 5.30	153.15 ± 3.94	150.45 ± 3.28	149.61 ± 1.53
			Throughput [parts/h]	32.15 ± 1.39	32.24 ± 0.57	32.11 ± 0.38	32.18 ± 0.31	32.09 ± 0.15
	0.20	∞	Energy [kJ/part]	158.80 ± 13.53	155.02 ± 4.00	153.70 ± 3.53	151.61 ± 3.16	151.44 ± 2.28
			Throughput [parts/h]	32.47 ± 1.40	32.26 ± 0.59	32.16 ± 0.41	32.36 ± 0.33	32.18 ± 0.15
		2500	Energy [kJ/part]	158.80 ± 13.53	154.47 ± 4.93	153.42 ± 3.94	151.45 ± 3.34	150.27 ± 1.61
			Throughput [parts/h]	32.47 ± 1.40	32.25 ± 0.57	32.16 ± 0.40	32.25 ± 0.33	32.13 ± 0.15
		500	Energy [kJ/part]	158.80 ± 13.53	153.82 ± 5.34	152.25 ± 4.08	149.94 ± 3.27	149.38 ± 1.47
			Throughput [parts/h]	32.47 ± 1.40	32.22 ± 0.56	32.10 ± 0.38	32.17 ± 0.31	32.08 ± 0.15
52	0.05	∞	Energy [kJ/part]	165.18 ± 14.19	157.33 ± 4.70	155.22 ± 2.95	152.87 ± 2.90	152.64 ± 2.65
			Throughput [parts/h]	32.55 ± 1.37	32.28 ± 0.59	32.20 ± 0.45	32.29 ± 0.37	32.22 ± 0.18
		2500	Energy [kJ/part]	164.52 ± 13.64	157.06 ± 4.34	155.09 ± 2.71	152.63 ± 2.80	151.22 ± 1.79
			Throughput [parts/h]	32.53 ± 1.38	32.28 ± 0.60	32.19 ± 0.45	32.28 ± 0.36	32.16 ± 0.15
		500	Energy [kJ/part]	164.52 ± 13.64	156.65 ± 4.95	154.23 ± 3.62	151.08 ± 3.19	149.85 ± 1.48
			Throughput [parts/h]	32.53 ± 1.38	32.27 ± 0.58	32.14 ± 0.39	32.20 ± 0.31	32.09 ± 0.15
	0.20	∞	Energy [kJ/part]	164.52 ± 13.64	157.06 ± 4.34	155.09 ± 2.71	152.80 ± 2.80	152.61 ± 2.64
			Throughput [parts/h]	32.53 ± 1.38	32.28 ± 0.60	32.19 ± 0.45	32.29 ± 0.37	32.22 ± 0.18
		2500	Energy [kJ/part]	164.52 ± 13.64	157.06 ± 4.34	155.09 ± 2.71	152.63 ± 2.80	151.18 ± 1.82
			Throughput [parts/h]	32.53 ± 1.38	32.28 ± 0.60	32.19 ± 0.45	32.28 ± 0.36	32.16 ± 0.15
		500	Energy [kJ/part]	164.52 ± 13.64	156.51 ± 5.18	154.09 ± 3.72	150.97 ± 3.22	149.80 ± 1.49
			Throughput [parts/h]	32.53 ± 1.38	32.26 ± 0.57	32.13 ± 0.38	32.19 ± 0.31	32.09 ± 0.15

Table A.6: M3/D3/0.02 - Effects of factors C_0, α and n^* over the number of observations before implementing the control parameters and the mean number of changes.

C_0 [kJ/part]	α	n^*	Replication number										Mean Number of Changes
			1	2	3	4	5	6	7	8	9	10	
0	0.05	/	10	10	10	10	10	20	10	10	10	20	77.9
	0.20	/	10	10	10	10	10	10	10	10	10	20	88.6
25	0.05	∞	30	30	20	30	20	30	20	20	20	30	6.5
		2500	30	30	20	30	20	30	20	20	20	30	23.3
		500	30	20	20	30	20	30	20	20	20	30	54.3
	0.20	∞	10	10	10	10	10	20	20	10	20	20	6.4
		2500	10	10	10	10	10	20	20	10	20	20	25.9
		500	10	10	10	10	10	20	20	10	20	20	59.2
52	0.05	∞	30	30	20	30	30	30	20	20	20	40	6.3
		2500	30	30	20	30	30	30	20	20	20	30	17.6
		500	30	30	20	30	30	30	20	20	20	30	51.1
	0.20	∞	30	30	20	30	30	30	20	20	20	30	6.4
		2500	30	30	20	30	30	30	20	20	20	30	20.3
		500	30	30	20	30	30	30	20	20	20	30	57.9

Appendix B

Proofs of Chapter 4

B.1 Method of Transformations

Assuming that X is a continuous random variable with pdf $f_X(x)$ and $Y = \Omega(X)$ is a function of X, the *Method of Transformations* [78] allows to analytically express the pdf $f_Y(y)$. In order to guarantee an intuitive grasp of this approach, let function Ω satisfy the following properties:

- It is differentiable, with derivative Ω'.
- It is strictly increasing, therefore inverse function Ω^{-1} is well defined. That is, for each possible y, there exists a unique x_1 such that $\Omega(x_1) = y$ and $\Omega'(x_1) > 0$.

In this case, the cumulative distribution function (cdf) $F_Y(y)$ can be easily found:

$$F_Y(y) = \mathbb{P}(Y \leq y) = \mathbb{P}(\Omega(X) \leq y) = \mathbb{P}(X \leq \Omega^{-1}(y)) = F_X(x_1) \tag{B.1}$$

To express the pdf $f_Y(y)$, it is eventually required to differentiate:

$$f_Y(y) = F'_Y(y) = F'_X(x_1)\frac{dx_1}{dy} = \frac{f_X(x_1)}{\Omega'(x_1)} \tag{B.2}$$

The same argument can be repeated for the case where function Ω is strictly decreasing. Since $\Omega'(x_1)$ turns out to be always negative, its absolute value $|\Omega'(x_1)|$ needs to be introduced in equation (B.2).

These considerations can be finally extended to a more general case, in which function Ω is not monotone. The solution consists in partitioning its

domain into a finite number of intervals, over which Ω is strictly monotone and differentiable. The pdf of Y is therefore given by:

$$f_Y(y) = \sum_{i=1}^{n} \frac{f_X(x_i)}{|\Omega'(x_i)|} \tag{B.3}$$

where $x_1, x_2, \ldots, x_n$ are real solutions to $y = \Omega(x)$. Of course, $f_Y(y) = 0$ when $y = \Omega(x)$ does not have a solution.

B.2 Proof of Theorem 4.2

Theorem 4.2. *Once set a feasible target difference Δe_{target}, it exists a limit sojourn duration in the standby state ϖ such that:*

$$F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{on} > \tau_{off} + \varpi\}) = F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{off} + \varpi\}) \tag{B.4}$$

The limit sojourn duration is:

$$\varpi = \frac{(w_3 + w_q)t_{su} + \Delta e_{target}}{w_2 - w_1} \tag{B.5}$$

Proof: The proof for this theorem starts from an analysis of figure B.1, which shows the *energy output functions* for $\boldsymbol{\tau} = \{\infty, \infty\}$ and $\boldsymbol{\tau} = \{\tau_{off}, \infty\}$ as well as their difference. Once set a feasible target difference Δe_{target}, it is possible to identify the corresponding starvation time occurrence x_{target} such that:

$$\phi(x_{target}, \{\infty, \infty\}) - \phi(x_{target}, \{\tau_{off}, \infty\}) = \Delta e_{target} \tag{B.6}$$

Solving for x_{target}, it results:

$$x_{target} = \tau_{off} + \frac{(w_3 + w_q)t_{su} + \Delta e_{target}}{w_2 - w_1} = \tau_{off} + \varpi \tag{B.7}$$

According to figure B.1, the cdf value $F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \infty\})$ corresponds to the probability of part arrival between $x = \tau_{off}$ and $x = x_{target}$:

$$F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \infty\}) = F_X(\tau_{off} + \varpi) - F_X(\tau_{off}) \tag{B.8}$$

The same consideration can be extended to any control configuration of the type $\boldsymbol{\tau} = \{\tau_{off}, \tau_{on} > \tau_{off} + \varpi\}$, in fact:

- $\phi(x_{target}, \{\infty, \infty\}) - \phi(x_{target}, \{\tau_{off}, \tau_{on} > \tau_{off} + \varpi\}) = \Delta e_{target}$.
- $\phi(x, \{\infty, \infty\}) - \phi(x, \{\tau_{off}, \tau_{on} > \tau_{off} + \varpi\})$ is monotonic increasing for $x > \tau_{off}$.

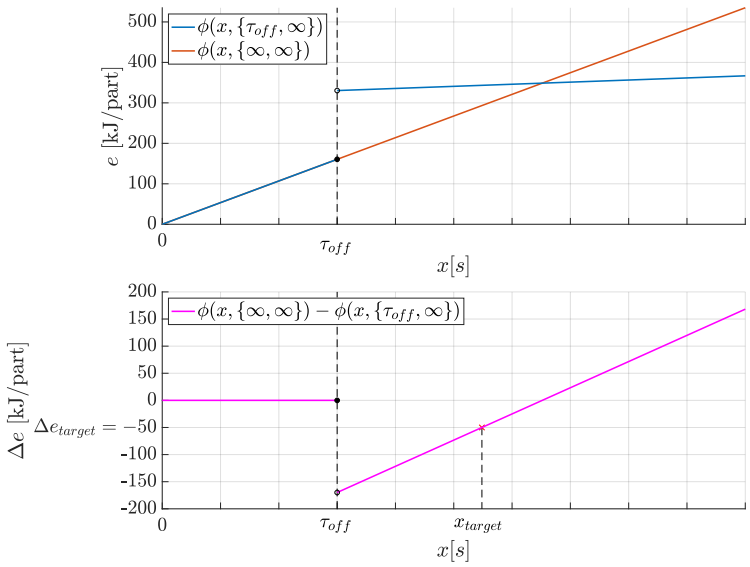


Figure B.1: $\phi(x,\{\infty,\infty\}) - \phi(x,\{\tau_{off},\infty\})$ for machine M1.

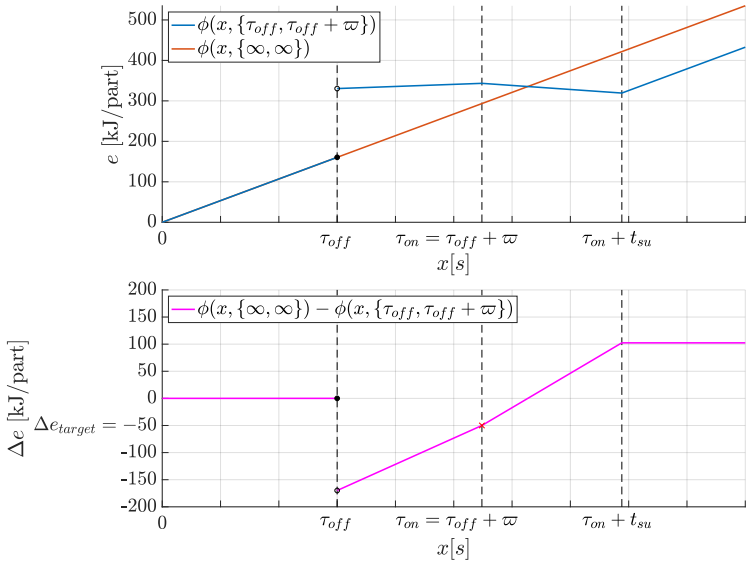


Figure B.2: $\phi(x,\{\infty,\infty\}) - \phi(x,\{\tau_{off},\tau_{off}+\varpi\})$ for machine M1.

It is therefore possible to state that:

$$F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{on} > \tau_{off} + \varpi\}) = F_X(\tau_{off} + \varpi) - F_X(\tau_{off}) \quad \text{(B.9)}$$

The control configuration $\boldsymbol{\tau} = \{\tau_{off}, \tau_{off} + \varpi\}$ is now addressed. In this case, as shown by figure B.2, the cdf value $F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{off} + \varpi\})$ accounts for the probability that a part arrives at the machine during its whole sojourn in the standby state:

$$F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{off} + \varpi\}) = F_X(\tau_{off} + \varpi) - F_X(\tau_{off}) \quad \text{(B.10)}$$

By comparing equations (B.9) and (B.10), it finally results:

$$F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{on} > \tau_{off}+\varpi\}) = F_{\Delta E}(\Delta e_{target}|\{\tau_{off}, \tau_{off}+\varpi\}) \quad \text{(B.11)}$$

□

B.3 Proof of Theorem 4.3

Theorem 4.3. *Once set a probability δ_q, it exists a lower bound for the switch off parameter τ_{off}^{LB} such that, for any feasible q_{target}:*

$$1 - F_Q(q_{target}|\boldsymbol{\tau}) \leq \delta_q \;\forall\; \boldsymbol{\tau} \mid \tau_{off} \geq \tau_{off}^{LB} \wedge \tau_{on} > \tau_{off} + \xi \quad \text{(B.12)}$$

In particular, τ_{off}^{LB} is the $100(1-\delta_q)th$ quantile of $f_X(x)$.

Proof: The proof for this theorem starts from an analysis of figure B.3, which shows a general *waiting time function* for machine M1. Given a feasible target $0 \leq q_{target} < t_{su}$, it is possible to identify the corresponding starvation time occurrence x_{target} such that:

$$\psi(x_{target}, \boldsymbol{\tau}) = q_{target} \quad \text{(B.13)}$$

Solving for x_{target}, it results:

$$x_{target} = \tau_{on} + t_{su} - q_{target} \implies \tau_{off} < \tau_{on} < x_{target} \leq \tau_{on} + t_{su} \quad \text{(B.14)}$$

According to figure B.3, $1 - F_Q(q_{target}|\boldsymbol{\tau})$ corresponds to the probability of part arrival between $x = \tau_{off}$ and $x = x_{target}$:

$$1 - F_Q(q_{target}|\boldsymbol{\tau}) = F_X(x_{target}) - F_X(\tau_{off}) \quad \text{(B.15)}$$

Let τ_{off}^{LB} be the $100(1-\delta_q)th$ quantile of $f_X(x)$. By assuming $\tau_{off} \geq \tau_{off}^{LB}$, it results:

$$1 - \delta_q \leq F_X(\tau_{off}) \leq F_X(x_{target}) \leq 1 \quad \text{(B.16)}$$

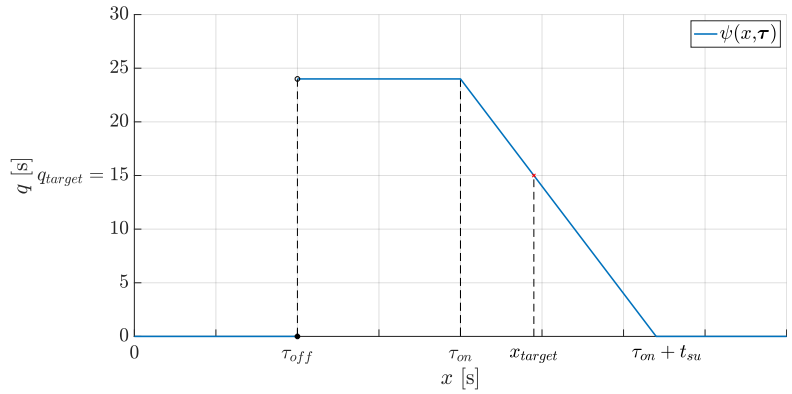


Figure B.3: *Waiting time function* $\psi(x, \boldsymbol{\tau})$ for machine M1.

Considering equations (B.15) and (B.16), it finally possible to conclude that:

$$0 \leq 1 - F_Q(q_{target} | \{\tau_{off} \geq \tau_{off}^{LB}, \tau_{on} > \tau_{off} + \xi\}) \leq \delta_q \tag{B.17}$$

In particular:

- The left boundary results from $F_X(x_{target}) = F_X(\tau_{off})$.
- The right boundary results from $F_X(x_{target}) = 1$ and $F_X(\tau_{off}) = 1 - \delta_q$.

□

Bibliography

[1] U.S. Energy Information Administration (EIA). *International Energy Outlook 2019 with projections to 2050*. 2019.

[2] U.S. Energy Information Administration (EIA). *Monthly Energy Review November 2019*. 2019.

[3] The International Organization for Standardization. *ISO Standards*. 2019. URL: `http://www.iso.org/iso/home/standards.htm`.

[4] A. Can et al. "A practical approach to reduce energy consumption in a serial production environment by shutting down subsystems of a machine tool." In: *Procedia Manufacturing*. Vol. 33. 2019, pp. 343–350.

[5] L. Hu et al. "Minimising the machining energy consumption of a machine tool by sequencing the features of a part." In: *Energy* 121 (2017), pp. 292–305.

[6] W. B. Powell and I. O. Ryzhov. *Optimal Learning*. Wiley Series in Probability and Statistics. Wiley, 2012. ISBN: 9780470596692.

[7] R. M. Karp. "On-Line Algorithms Versus Off-Line Algorithms: How Much is It Worth to Know the Future?" In: *Proceedings of the IFIP 12th World Computer Congress on Algorithms, Software, Architecture - Information Processing '92, Volume 1 - Volume I*. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1992, pp. 416–429.

[8] G. Mouzon, M. B. Yildirim, and J. Twomey. "Operational methods for minimization of energy consumption of manufacturing equipment." In: *International Journal of Production Research* 45.18-19 (2007), pp. 4247–4271.

[9] J. Huang et al. "A maintenance and energy saving joint control scheme for sustainable manufacturing systems." In: *Procedia CIRP*. Vol. 80. 2019, pp. 263–268.

[10] J. Zou et al. "Resilient adaptive control based on renewal particle swarm optimization to improve production system energy efficiency." In: *Journal of Manufacturing Systems* 50 (2019), pp. 135–145.

[11] E. T. Duque et al. "Energy Consumption Control of One Machine Manufacturing System with Stochastic Arrivals Based on Fuzzy Logic." In: *IEEE International Conference on Industrial Engineering and Engineering Management*. Vol. 2019-December. 2019, pp. 1503–1507.

[12] Z. Fei et al. "Fuzzy Petri Net Based Intelligent Machine Operation of Energy Efficient Manufacturing System." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2018-August. 2018, pp. 1593–1598.

[13] J. Wang et al. "Fuzzy decision of machine switch on-off for energy efficient operation of manufacturing system." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2017-August. 2018, pp. 1158–1162.

[14] W. Su et al. "Reducing energy consumption in serial production lines with bernoulli reliability machines." In: *International Journal of Production Research* 55.24 (2017), pp. 7356–7379.

[15] J. Zou et al. "Data-driven modeling and real-time distributed control for energy efficient manufacturing systems." In: *Energy* 127 (2017), pp. 247–257.

[16] M. P. Brundage et al. "Implementing a Real-Time, Energy-Efficient Control Methodology to Maximize Manufacturing Profits." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46.6 (2016), pp. 855–866.

[17] W. Su et al. "Improving energy efficiency in Bernoulli serial lines: An integrated model." In: *International Journal of Production Research* 54.11 (2016), pp. 3414–3428.

[18] W. Su et al. "An energy and productivity optimization model in Bernoulli serial lines." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2016-November. 2016, pp. 855–860.

[19] J. Zou et al. "Opportunity Window for Energy Saving and Maintenance in Stochastic Production Systems." In: *Journal of Manufacturing Science and Engineering, Transactions of the ASME* 138.12 (2016).

[20] Z. Sun et al. "Data driven production runtime energy control of manufacturing systems." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2015-October. 2015, pp. 243–248.

[21] M. P. Brundage et al. "Utilizing energy opportunity windows and energy profit bottlenecks to reduce energy consumption per part for a serial production line." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2014-January. 2014, pp. 461–466.

[22] M. P. Brundage et al. "Energy efficiency management of an integrated serial production line and HVAC system." In: *IEEE Transactions on Automation Science and Engineering* 11.3 (2014), pp. 789–797.

[23] Z. Sun and L. Li. "Opportunity estimation for real-time energy control of sustainable manufacturing systems." In: *IEEE Transactions on Automation Science and Engineering* 10.1 (2013), pp. 38–44.

[24] Z. Niu et al. "On energy-delay tradeoff in base station sleep mode operation." In: *2012 IEEE International Conference on Communication Systems, ICCS 2012*. 2012, pp. 235–239.

[25] V. V. Prabhu and M. Taisch. "Simulation modeling of energy dynamics in discrete manufacturing systems." In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. Vol. 14. 2012. Chap. PART 1, pp. 740–745.

[26] W. Li et al. "An investigation into fixed energy consumption of machine tools." In: *Glocalized Solutions for Sustainability in Manufacturing - Proceedings of the 18th CIRP International Conference on Life Cycle Engineering*. 2011, pp. 268–273.

[27] M. Squeo, N. Frigerio, and A. Matta. "Multiple sleeping states for energy saving in CNC machining centers." In: *Procedia CIRP*. Vol. 80. 2019, pp. 144–149.

[28] J. Wang et al. "Multi-state decision of unreliable machines for energy-efficient production considering work-in-process inventory." In: *International Journal of Advanced Manufacturing Technology* 102.1-4 (2019), pp. 1009–1021.

[29] L. Li and Z. Sun. "Dynamic energy control for energy efficiency improvement of sustainable manufacturing systems using markov decision process." In: *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans* 43.5 (2013), pp. 1–11.

[30] V. J. Maccio and D. G. Down. "Structural properties and exact analysis of energy-aware multiserver queueing systems with setup times." In: *Performance Evaluation* 121-122 (2018), pp. 48–66.

[31] V. J. Maccio and D. G. Down. "Asymptotic Performance of Energy-Aware Multiserver Queueing Systems with Setup Times." In: *Proceedings of the American Control Conference*. Vol. 2018-June. 2018, pp. 6266–6272.

[32] V. J. Maccio and D. G. Down. "Exact analysis of energy-aware multi-server queueing systems with setup times." In: *Proceedings - 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS 2016*. 2016, pp. 11–20.

[33] V. J. Maccio and D. G. Down. "On Optimal Control for Energy-Aware Queueing Systems." In: *Proceedings - 2015 27th International Teletraffic Congress, ITC 2015*. 2015, pp. 98–106.

[34] V. J. Maccio and D. G. Down. "On optimal policies for energy-aware servers." In: *Proceedings - IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS*. 2013, pp. 31–39.

[35] N. Frigerio, J. G. Shanthikumar, and A. Matta. "Dynamic Programming for Energy State Control of Machine Tools in Manufacturing with Part Admission Control." In: *11th Conference on Stochastic Models of Manufacturing and Service Operations (SMMSO 2017)*. 2017.

[36] N. Frigerio and A. Matta. "Analysis on energy efficient switching of machine tool with stochastic arrivals and buffer information." In: *IEEE Transactions on Automation Science and Engineering* 13.1 (2016), pp. 238–246.

[37] N. Frigerio, J. G. Shanthikumar, and A. Matta. "Dynamic programming for energy control of machine tools in manufacturing." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2015-October. 2015, pp. 39–44.

[38] N. Frigerio and A. Matta. "Energy efficient control strategy for machine tools with stochastic arrivals and time dependent warm-up." In: *Procedia CIRP*. Vol. 15. 2014, pp. 56–61.

[39] X. Guo et al. "Optimal wake-up mechanism for single base station with sleep mode." In: *Proceedings of the 2013 25th International Teletraffic Congress, ITC 2013*. 2013.

[40] J. Wang et al. "Energy saving operation of manufacturing system based on dynamic adaptive fuzzy reasoning petri net." In: *Energies* 12.11 (2019).

[41] Z. Jia et al. "Performance analysis for serial production lines with Bernoulli Machines and Real-time WIP-based Machine switch-on/off control." In: *International Journal of Production Research* 54.21 (2016), pp. 6285–6301.

[42] Z. Jia et al. "Performance analysis of Bernoulli serial production lines with switch-on/off machine control." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2015-October. 2015, pp. 477–482.

[43] L. Marzano, N. Frigerio, and A. Matta. "Energy efficient state control of machine tools: A time-based dynamic control policy." In: *IEEE International Conference on Automation Science and Engineering* 2019-August (2019), pp. 596–601.

[44] N. Frigerio and A. Matta. "Energy-efficient control strategies for machine tools with stochastic arrivals." In: *IEEE Transactions on Automation Science and Engineering* 12.1 (2015), pp. 50–61.

[45] N. Frigerio and A. Matta. "Machine control policies for energy saving in manufacturing." In: *IEEE International Conference on Automation Science and Engineering*. 2013, pp. 651–656.

[46] V. V. Prabhu, H. W. Jeon, and M. Taisch. "Modeling green factory physics - An analytical approach." In: *IEEE International Conference on Automation Science and Engineering*. 2012, pp. 46–51.

[47] Q. Chang et al. "Energy saving opportunity analysis of automotive serial production systems (March 2012)." In: *IEEE Transactions on Automation Science and Engineering* 10.2 (2013), pp. 334–342.

[48] Y. Li et al. "Event-Based Supervisory Control for Energy Efficient Manufacturing Systems." In: *IEEE Transactions on Automation Science and Engineering* 15.1 (2018), pp. 92–103.

[49] H. Su, N. Frigerio, and A. Matta. "Energy Saving Opportunities and Value of Information: A Trade-off in a Production Line." In: *Procedia CIRP*. Vol. 48. 2016, pp. 301–306.

[50] N. Frigerio and A. Matta. "Analysis of an energy oriented switching control of production lines." In: *Procedia CIRP*. Vol. 29. 2015, pp. 34–39.

[51] Y. Li, J. Wang, and Q. Chang. "Event-Based Production Control for Energy Efficiency Improvement in Sustainable Multistage Manufacturing Systems." In: *Journal of Manufacturing Science and Engineering, Transactions of the ASME* 141.2 (2019).

[52] G. Chen et al. "Feedback control of machine startup for energy-efficient manufacturing in Bernoulli serial lines." In: *IEEE International Conference on Automation Science and Engineering*. 2011, pp. 666–671.

[53] N. Frigerio, A. Matta, and Z. Lin. "Nested partitioning algorithm for the optimization of control parameters in energy efficient production lines." In: *Proceedings of the XIII AITeM Conference* (2017), pp. 1–10.

[54] P. Renna. "Energy saving by switch-off policy in a pull-controlled production line." In: *Sustainable Production and Consumption* 16 (2018), pp. 25–32.

[55] J. Wang et al. "Markov chain based idle status control of stochastic machines for energy saving operation." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2017-August. 2018, pp. 1019–1023.

[56] J. Wang et al. "Active energy saving strategy for sensible manufacturing systems operation based on real time production status." In: *IEEE International Conference on Industrial Engineering and Engineering Management*. Vol. 2016-December. 2016, pp. 1001–1005.

[57] C. Zhang and P. Jiang. "RFID-Driven Energy-Efficient Control Approach of CNC Machine Tools Using Deep Belief Networks." In: *IEEE Transactions on Automation Science and Engineering* (2019).

[58] J. Wang et al. "Event-driven online machine state decision for energy-efficient manufacturing system based on digital twin using Max-plus Algebra." In: *Sustainability (Switzerland)* 11.18 (2019).

[59] Y. Huang, J. Wang, and S. Li. "Max-Plus Algebra Based Machine Sleep Decision for Energy Efficient Manufacturing." In: *Proceedings 2018 Chinese Automation Congress, CAC 2018* (2019), pp. 3986–3991.

[60] J. Wang, S. Li, and J. Liu. "A multi-granularity model for energy consumption simulation and control of discrete manufacturing system." In: *19th International Conference on Industrial Engineering and Engineering Management: Assistive Technology of Industrial Engineering*. 2013, pp. 1055–1064.

[61] W. Wei and Z. Wang. "Integrated optimal production and energy control of a single machine and single product-type manufacturing system." In: *2016 13th International Workshop on Discrete Event Systems, WODES 2016*. 2016, pp. 343–348.

[62] C. Cronrath, B. Lennartson, and M. Lemessi. "Energy reduction in paint shops through energy-sensitive on-off control." In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2016-November. 2016, pp. 1282–1288.

[63] L. Marzano. "Online algorithms for a time-based state control policy for energy saving in manufacturing systems." Master thesis. Politecnico di Milano, 2019.

[64] G. Grimmett and D. Stirzaker. *Probability and random processes*. Third edition. Oxford University Press, 2001.

[65] E. Parzen. "On Estimation of a Probability Density Function and Mode." In: *Ann. Math. Statist.* 33.3 (Sept. 1962). Ed. by The Institute of Mathematical Statistics, pp. 1065–1076.

[66] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. A Wiley-interscience publication. Wiley, 1992.

[67] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986.

[68] A. W. Bowman. "An Alternative Method of Cross-Validation for the Smoothing of Density Estimates." In: *Biometrika* 71.2 (1984), pp. 353–360.

[69] M. Rudemo. "Empirical Choice of Histograms and Kernel Density Estimators." In: *Scandinavian Journal of Statistics* 9.2 (1982), pp. 65–78.

[70] B. E. Hansen. *Lecture Notes on Nonparametrics*. University of Wisconsin, 2009.

[71] S. J. Sheather. "Density Estimation." In: *Statistical Science* 19.4 (2004), pp. 588–597.

[72] A. W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford Statistical Science Series. OUP Oxford, 1997.

[73] J. D. F. Habbema, J. Hermans, and K. Van Den Broek. "A Stepwise Discriminant Analysis Program Using Density Estimation." In: *Compstat 1974: Proceedings in Computational Statistics* (1974).

[74] R. P. W. Duin. "On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions." In: *IEEE Transactions on Computers* C-25.11 (1976), pp. 1175–1179.

[75] A. M. Law. *Simulation Modeling & Analysis*. 5th ed. New York, NY, USA: McGraw-Hill, 2015.

[76] B. Efron. "Bootstrap Methods: Another Look at the Jackknife." In: *Ann. Statist.* 7.1 (Jan. 1979), pp. 1–26.

[77] N. E. Helwig. *Bootstrap Confidence Intervals*. University of Minnesota (Twin Cities), 2017.

[78] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002. ISBN: 9780534243128.

[79] S. Aminikhanghahi and D. Cook. "A Survey of Methods for Time Series Change Point Detection." In: *Knowledge and Information Systems* 51 (Sept. 2016).

[80] S. Liu et al. "Change-point detection in time-series data by relative density-ratio estimation." In: *Neural Networks* 43 (July 2013), pp. 72–83. ISSN: 0893-6080.

[81] Eurostat. *Electricity price statistics*. 2019.