Energy Supply and Demand Side Management in Industrial Microgrid Context

Alemayehu Addisu Desta

ÉCOLE DOCTORALE UNIVERSITÉ PARIS-EST

Mathématiques et STIC

THÈSE

en vue de l'obtention du titre de

Docteur

de l'Université Paris-Est

Spécialité : Informatique

Alemayehu Addisu Desta

Energy Supply and Demand Side Management in Industrial Microgrid Context

---

soutenue le 4 décembre 2017

Jury

Directeur :	**Laurent George**	–	ESIEE Paris (LIGM), France
Co-encadrant :	**Hakim Badis**	–	UPEM (LIGM), France
Rapporteurs :	**Maryline Chetto**	–	Université de Nantes (IRCCyN), France
	Leandro Soares Indrusiak	–	Université de York, Angleterre
Examinateurs :	**Pierre Courbin**	–	METRON, France
	Bruno Gaujal	–	INRIA, France
	Ye-Qiong Song	–	Université de Lorraine, France
Invité :	**David Menga**	–	EDF R&D, France

PhD prepared at
ESIEE Paris
2 boulevard Blaise Pascal,
Cité Descartes BP 99,
93162 Noisy-le-Grand Cedex, France

PhD in collaboration with
METRON
102 rue Réaumur,
75002 Paris, France

PhD in collaboration with
LIGM
Laboratoire d'Informatique Gaspard-Monge
Cité Descartes, Batiment Copernic
5 boulevard Descartes,
77454 Marne-la-vallée Cedex 2, France

To my wife and son,
To my parents,
To my grandparents, still there or already gone,
To my brother and many sisters,
To all my family,
To all my friends,
To all the people who have graced my life.

Because each encounter, even a brief one,
is an occasion to rediscover oneself
and to marvel.

À ma femme et mon fils,
À mes parents,
À mes grands parents, encore là ou déjà partis,
À mon frère et mes nombreux sœurs,
À toute ma famille,
À tous mes amis,
À toutes les personnes que j'ai eu le privilge de croiser.

Parce que chaque personne rencontre, même brièvement,
est une occasion exceptionnelle de se redécouvrir
et de s'émerveiller.

Acknowledgments

I would like to take this opportunity to wholeheartedly thank my thesis supervisor, *Prof Laurent George*. Since my master's thesis in ECE Paris, he has been an inspirational figure for me and I thank him for his invaluable help in both academic and private matters. I also wish to express my thanks to my co-supervisor, *Dr Hakim Badis*. My sincere gratitude also goes to *Dr Pierre Courbin* (technical director at METRON) whom I collaborated with for research works that gave fruitful results in this thesis. Around the start of my PhD, he told me to be "fier" (it means "proud" in English) of my thesis. After that moment, I have been in "strings attached" mode with my thesis. *Mrs Sylvie Cache* (secretary of UPEM's doctoral school) should also be thanked. I also thank my jury members *Prof Maryline Chetto*, *Dr Leandro Soares Indrusiak*, *Prof Ye-Qiong Song*, *Dr Bruno Gaujal* and *Mr David Menga* for willing to accept our invitations to be jury members of this thesis.

As this thesis is financed by METRON, my special thanks go to *Dr Vincent Sciandra* (CEO and co-founder of METRON) and *Mr David Tagliabue* (president of METRON) for their priceless support throughout my thesis. I must say thank you to my colleagues in METRON (order has no significance): *Julien Calderan*, *José Lopez*, *Nifemi Ogundare*, *Anthony Gadiou*, *Jérémie Rendolet*, *Victor Nicolas*, *Yann Le Bail*, *Jonathan Carmignac* and the new employees.

Now, it is time to thank my friends that have special places in my heart. Life could be difficult without their friendship and supports. Here they are: *Zerihun Gedeb*, *Getachew Tilahun*, *Wosen Eshetu*, *Yirgalem Bereka*, *Gelila Tafese*, *Emanuel Kassa*, *Ephrem Berhe*, *Bezaye Taye*, *Sergut Tekuamework*, *Varun Deshpande* and *Abdenour Kifouche*. If a friend of mine could not find his/her name in the list, please pardon me; it is not intentional at all.

Last but no least, I want to thank my wife *Meskerem*. She put her utmost effort to single-handedly raise my son *Leul* (it means "Prince" in English and French). I am very much indebted to you madame *Meskerem*. I also express my warm thanks to my parents, brother and many sisters. Lastly, I would like to thank the almighty *God* who I strongly believe gave me the power and the strength to accomplish this task. Praised be the lord! Amen.

Abstract

Energy Supply and Demand Side Management in Industrial Microgrid Context

Due to increased energy costs and environmental concerns such as elevated carbon footprints, centralized power generation systems are restructuring themselves to reap benefits of distributed generation in order to meet the ever growing energy demands. Microgrids are considered as a possible solution to deploy distributed generation which includes **D**istributed **E**nergy **R**esources (DERs) (e.g., solar, wind, battery, etc). In this thesis, we are interested in addressing energy management challenges in an industrial microgrid where energy loads consist of industrial processes. Our plan of attack is to divide the microgrid energy management into *supply* and *demand* sides.

In supply side, the challenges include modeling of power generations and smoothing out fluctuations of the DERs. To model power generations, we propose a model based on service curve concepts of **N**etwork **C**alculus (NC). Using this mathematical tool, we determine a minimum amount of power the DERs can generate and aggregating them will give us total power production in the microgrid. After that, if there is an imbalance between energy supply and demand, we put forward different strategies to minimize energy procurement costs. Based on real power consumption data of an industrial site located in France, significant cost savings can be achieved by adopting the strategies. In this thesis, we also study how to mitigate the effects of power fluctuations of DERs in conjunction with **E**nergy **S**torage **S**ystems (ESSs). For this purpose, we propose a Gaussian-based smoothing algorithm and compare it with state-of-the-art smoothing algorithms. We found out that the proposed algorithm uses less battery size for smoothing purposes when compared to other algorithms. To this end, we are also interested in investigating effects of *allowable* range of fluctuations on battery sizes.

In demand side, the aim is to reduce energy costs through **D**emand **S**ide **M**anagement (DSM) approaches such as **D**emand **R**esponse (DR) and **E**nergy **E**fficiency (EE). As industrial processes are power-hungry consumers, a small power consumption reduction using the DSM approaches could translate into crucial savings. This thesis focuses on DR approach that can leverage time varying electricity prices to move energy demands from peak to off-peak hours. To attain this goal, we rely on a queuing theory-based model to characterize temporal behaviors (arrival and departure of jobs) of a manufacturing system. After defining job arrival and departure processes, an effective utilization function

is used to predict workstation's (or machine's) behavior in temporal domain that can show its status (*working* or *idle*) at any time. Taking the status of every machine in a production line as an input, we also propose a DR scheduling algorithm that adapts power consumption of a production line to available power and production rate constraints. The algorithm is coded using **D**eterministic **F**inite **S**tate **M**achine (DFSM) in which state transitions happen by inserting a job (or not inserting) at conveyor input. We provide conditions for existence of feasible schedules and conditions to accept DR requests positively.

To verify analytical computations on the queuing part, we have enhanced **O**bjective **M**odular **Ne**twork **T**estbed in C++ (OMNET++) discrete event simulator for fitting it to our needs. We modified various libraries in OMNET++ to add *machine* and *conveyor* modules. In this thesis, we also setup a testbed to experiment with a smart DR protocol called **Open A**utomated **D**emand **R**esponse (OpenADR) that enables energy providers (e.g., utility grid) to ask consumers to reduce their power consumption for a given time. The objective is to explore how to implement our DR scheduling algorithm on top of OpenADR.

Contents

List of Figures

List of Tables

Part I

General Introduction and Concepts

Chapter 1

General Introduction

Contents

1.1 Introduction

Current power system is dominated by centralized generation in which electricity is distributed to consumers through a macrogrid (e.g., utility grid). Due to challenges such as increased energy costs and global emissions of greenhouse gases (e.g., carbon dioxide (CO_2), chlorofluorocarbons (CFCs), etc), the centralized generation systems need to be restructured in order to meet the ever growing energy demands [CN09]. A possible solution to these challenges is to deploy distributed generation systems including **D**istributed **E**nergy **R**esources (DERs) (e.g., solar, wind, battery, etc) which are normally small in generation capacity (from kW to a few MW capacity range). These resources can be installed close to consumers' premises. Being considered as an alternative to centralized generation, microgrids are emerging power systems to manage distributed generations.

Microgrids are relatively small-scale power systems that include electrical loads (any device that consumes electric power, e.g., refrigerators, industrial machines, etc), DERs and a control system (a formal definition of microgrid is given in section 2.2 of the next chapter). A microgrid can operate as a single system (*island* mode) or it can be connected to a utility grid (*grid-connected* mode). Due to introduction of Information and Communications Technologies (ICT) to microgrids, a two-way communication of energy data between producers and consumers is made possible [SW12]. Hence, informed decisions can be taken based on the information gathered on microgrid components.

According to Hayden [Hay13], microgrids are categorized into 5 types: *campus environment/institutional*, *remote off-grid*, *military base*, *community/utility* and *commercial and industrial (C&I)* microgrids. In this thesis, we consider an *industrial* microgrid where electrical loads comprise of industrial machines which are intensive power consumers. The motivation to consider this type of microgrid is that we have access to real power consumption data of an industrial plant located in France. Thanks to METRON's platform, we can exploit the data for analytical and simulation purposes throughout the thesis.

In this chapter, we discuss some potential benefits of microgrids in section 1.2. Then, section 1.3 highlights the challenges in microgrids. After that, we provide our motivations and contributions of this thesis in section 1.4. Finally, section 1.5 gives a *bird's eye view* of contents of the remaining chapters.

1.2 Why microgrids?

According to Parhizi et al. [Par+15], the installed microgrid capacity has estimated growth of 1.1 GW in 2012 to 4.7 GW in 2017 with an estimated market opportunity of $17.3 billion. The significant benefits associated with microgrids have led to vast efforts to expand their penetration in power systems. Based on a report in [VEZ14], potential benefits of microgrids include:

- *Renewable integration*: **R**enewable **E**nergy **S**ources (RESs) play major roles in satisfying some parts of global energy consumption. From Figure 1.1, we notice that the average growth rate of modern RESs (except biomass) is more than twice (around 4.7%) the rate of energy demand in the period between 2004 and 2014. In 2016, the RESs' share increased to 24.5% according to a report by REN21 (Renewable Energy Policy Network for the 21st Century) [ZL17] which is based in Paris, France. Hence, to reap the benefits of RESs, microgrids are becoming indispensable.

- *Increased reliability and resilience*: The ability of microgrids to *island* enables them to continue providing power to their consumers during events of power outage. The ability to *island* can also be effective in partitioning distribution feeders in order to isolate faults.

- *Relationship of the microgrid to the utility grid*: To create smart grids, microgrids can be viewed as fundamental building blocks. This is to say that future utility grids could be a collection of networked microgrids based on interactions among control systems that balance energy demand and supply at micro and macro levels.

- *Microgrids as a grid resource*: From the utility grid's view, microgrids can serve as a reliable energy resource, an ancillary service resource, a **D**emand

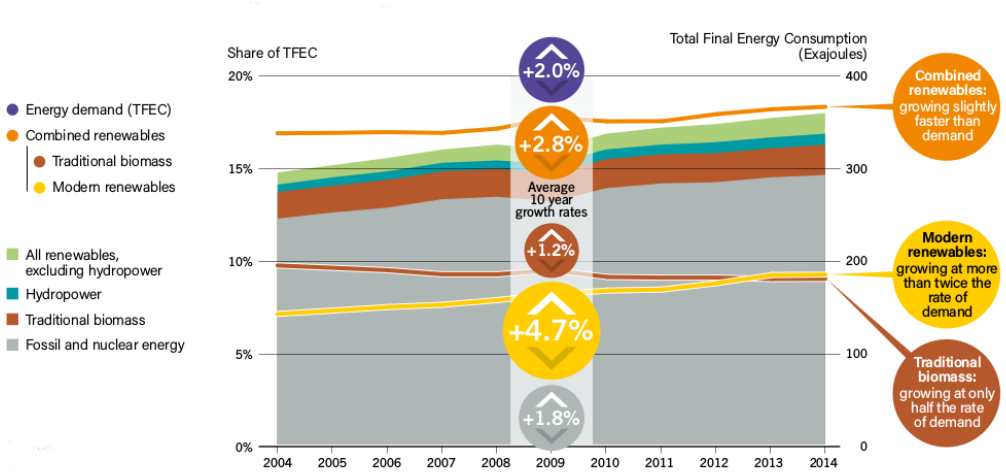


Figure 1.1 – Share of renewables in satisfying global energy consumption over 10 years (from 2004 to 2014) [ZL17].

Response (DR) (shedding loads in response to requests from utility grid, see formal definition in section 2.3 of chapter 2) resource or a power consumption resource (in case of excess generation). Microgrids can also participate in electricity wholesale markets in order to generate revenues by exporting surplus power productions.

1.3 Challenges in two sides of a microgrid

Based on the side of energy generation or consumption, we divide the industrial microgrid in two sides: *supply* and *demand* sides. In this thesis, we consider two RESs, namely, solar **P**hotovoltaics (PV) panels and wind turbines in the supply side. The other side consists of industrial loads. **E**nergy **S**torage **S**ystems (ESSs) can be in both sides: in supply side when discharging and in demand side when charging. The following sections describe the challenges we address in both sides of industrial microgrid energy management.

1.3.1 Supply side challenges

As mentioned above, RESs offer a considerable amount of energy in meeting the global energy consumption. However, these resources are intermittent in nature. For instance, the output of solar PV power changes frequently depending on the position of the sun and clouds. In the same way, wind power is subject to some of the same types of daily and seasonal variations. The variability in power productions of these resources poses challenges to seamlessly integrate them to

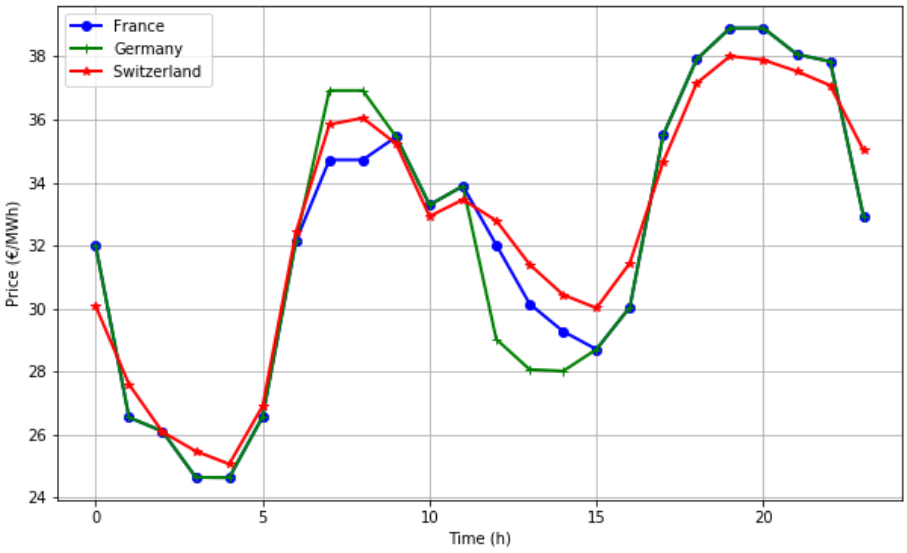


Figure 1.2 – Hourly electricity prices in France, Germany and Switzerland on 31/07/2017 (from EPEX spot [EPE17]).

a microgrid. The challenges can be categorized into two: *how to model RESs' power production* and *ways to reduce power fluctuations*.

Regarding power generation modeling of RESs, we can ask a question like *how to mathematically model the minimum amount of energy that an energy resource can provide?* Furthermore, there is also a question of supply and demand balance: *how can we minimize energy costs when demand is greater than supply?* To reduce power fluctuations, important questions include *how to smooth out the fluctuations using ESSs? What is the size of ESSs to smooth intermittency given allowable ranges of fluctuations?*

1.3.2 Demand side challenges

In the demand side, the concern is how to reduce energy costs. Based on a survey by the International Energy Agency [IEA16], the industrial plants are accounted for 42.5% of global electricity consumption in 2015. This will cost them dearly if cost minimization mechanisms are not put in place. The potential to reduce energy costs can be gained through implementing **D**emand **S**ide **M**anagement (DSM) approaches such as **E**nergy **E**fficiency (EE) and DR. The focus of this thesis is on DR that can leverage the varying electricity prices as shown in Figure 1.2. A cost saving can be achieved through DR mechanisms which include peak-shaving and load shifting for moving loads from peak to off-peak hours.

Considering DR mechanisms, a scheduling problem could arise: *which machine*

to turn ON or OFF according to a given power threshold? What are the conditions for existence of feasible schedules? Conditions to accept a DR positively? Another question could be *how to characterize temporal behaviors (arrival and departure of jobs) of a manufacturing system?*

1.4 Motivations and contributions of the thesis

Our motivations are two fold. First, to address the challenges detailed above, we believe that concepts from different disciplines should come together. In the beginning of this thesis, our area of expertise includes Computer Science, Network Communication and Telecommunication. From these fields, we rely on important concepts such as **N**etwork **C**alculus (NC) and queuing theory to address the challenges in energy domain which broaden our field of expertise. Second, nowadays researches on renewables and energy cost reduction are hot topics as a means of securing our energy future. Hence, we want to contribute our part to this global cause.

Contributions of this thesis are classified according to supply and demand sides of industrial microgrid energy management. They are listed below:

- **Supply side**:
  - To attain the minimum power generation of DERs, we proposed a model based on service curves of NC. Based on actual power consumption data of a factory, we also proposed different strategies for minimizing energy costs.
  - To mitigate power fluctuations using ESSs, a Gaussian-based smoothing algorithm is proposed. The algorithm attains lower ESSs size when compared to other smoothing algorithms.
- **Demand side**:
  - For characterizing a production system in a temporal domain, we proposed a queuing theory-based model. The model is used to describe system temporal behaviors such as job/task arrivals and departures and utilization of a station (or machine). To verify our analytical works, we have developed multiple modules in **O**bjective **M**odular **N**etwork **T**estbed in C++ (OMNET++) discrete event simulator.
  - For respecting available power and production rate constraints, a graph activity model is proposed. This model adapts power consumption of industrial processes when a utility grid emits a DR signal to consumers in order to reduce power consumption. We also provide conditions of accepting DR requests positively and existence of feasible schedules.

1.5 Thesis organization

The remaining of this thesis consists of four chapters. Contents of each chapter are briefly introduced as follows:

Chapter 2 presents relevant concepts to understand this work. In section 2.2, we discuss basic concepts of microgrid and models of microgrid elements such as solar and wind powers, ESSs and electrical loads. Then, section 2.3 details DR concepts together with DR program types, benefits of DR, approaches and mathematical problems in DR and a DR protocol which is called **Open Automated Demand Response** (OpenADR). After that, we present concepts of service curves of NC that are used to model power generation of RESs. Finally, for an objective of modeling a manufacturing system, we describe a queuing model with its notations in section 2.5.

Chapter 3 details our works in the supply side of industrial microgrid energy management. The works in the chapter are modeling of DERs and smoothing power production of RESs. In section 3.3, we provide our model of DERs based on concepts of service curves of NC. In this context, we use service curves to obtain minimum power production of the energy sources. Then, to reduce power generation fluctuations of RESs, we propose a smoothing algorithm in section 3.4. We compare its performances against other smoothing algorithms.

Chapter 4 proposes a queuing model of manufacturing systems and a DR scheduling algorithm. These two works are categorized under the demand side of industrial microgrid energy management. In section 4.3, we detail modeling of a **S**ynchronous **P**roduction **L**ine (SPL) system based on a queuing model. We use the model to define SPL system's temporal characteristics such as job arrival and departure processes to/from a machine. Then, section 4.4 presents our DR scheduling algorithm that adapts SPL's power consumption to constraints of available power and production rate. We also provide our experiments with OpenADR in section 4.6.

Chapter 5 concludes the works of this thesis by highlighting important points that are raised in each chapter and the corresponding publications. Perspectives of the thesis are also provided in the chapter.

In this chapter, we provided research contexts of this thesis and our motivations to pursue energy management in industrial microgrid. The next chapter discusses essential concepts such as microgrid, service curves of **N**etwork **C**alculus (NC), **D**emand **R**esponse (DR) and queuing theory. These concepts will be used in chapter 3 and 4.

Chapter 2

General Concepts and Models

Contents

2.1 Introduction

This chapter discusses fundamental concepts including microgrid, **D**emand **R**esponse (DR), service curves and queuing theory. These concepts are relevant to our works in the following chapters. We begin with the concept of microgrid in section 2.2. For understanding the characteristics of microgrids, we provide generic models on **D**istributed **E**nergy **R**esources (DERs) such as wind turbines, solar **P**hotovoltaics (PV) panels and **E**nergy **S**torage **S**ystems (ESSs). Then, in section 2.3, we detail concepts of DR focusing on DR programs, DR mathematical problems, and an automated DR enabler technology. Focusing on DR concepts, chapter 4 presents a DR scheduling algorithm that adapts production in a manufacturing system according to available power and production rate constraints. After that, section 2.4 discusses concepts of service curves of **N**etwork **C**alculus (NC) together with basic mathematical notations and theories. Based on the service curve concepts, chapter 3 details modeling of energy generations of DERs. Next, in section 2.5, we provide an overview of queuing theory by highlighting performance measures of a relevant queuing model which is used to characterize temporal evolution of a manufacturing system in chapter 4. Finally, section 2.6 summarizes this chapter by mentioning the main points discussed in the chapter.

2.2 Microgrid Concept

The term *microgrid* was first coined by Lasseter and Paigi [LP04]. Afterwards, different definitions are given in the literature. We adopt a definition of microgrid from U.S. Department of Energy (DoE) as given below.

Definition 2.1 (Microgrid [DOE11])**.**
A microgrid is a group of interconnected loads and DERs within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid and that connects and disconnects from such grid to enable it to operate in both *grid-connected* (a microgrid supplies or draws power to/from a utility grid) or *island* mode (a microgrid is disconnected from a utility grid).

Based on this definition, a microgrid is characterized by the following three distinct features [Par+15]:

- DERs installations must have clearly defined boundaries, i.e., they should be bounded to only one microgrid,
- total power generations need to exceed peak demands so that it could be disconnected from a utility grid, i.e., it can be in *island* mode, and
- it must contain computer systems that monitor, control and balance energy demand, supply and storage in response to changing energy needs.

Hence, these characteristics show that a microgrid is a small-scale power system with ability of self-healing when there is power interruption in a utility grid.

In the following sections, we first present an architecture of *industrial* microgrid where energy demand includes industrial loads (e.g., manufacturing processes). Then, we provide generic models of wind, solar and energy storage in section 2.2.2. In section 2.2.3, we give notations of a manufacturing system which is relevant to chapter 4. At last, a description of spot market is given in section 2.2.4.

2.2.1 Microgrid Architecture

An architecture of a microgrid is shown in Figure 2.1. The main microgrid components include solar PV panels, wind turbines, ESSs, loads (industrial loads in our case), energy spot markets, and a microgrid controller. Through a **P**oint of **C**ommon **C**oupling (PCC) circuit breaker, it is possible to connect or disconnect the microgrid from the utility grid. Under normal conditions, the microgrid is connected to the utility grid for the purpose of energy transactions. However, when there is fault (e.g., power outage, low power quality, etc) in the utility grid, the PCC disconnects the microgrid to be an autonomous system, i.e., it is in *island* mode. In this case, local generations in the microgrid should support the loads of the microgrid.

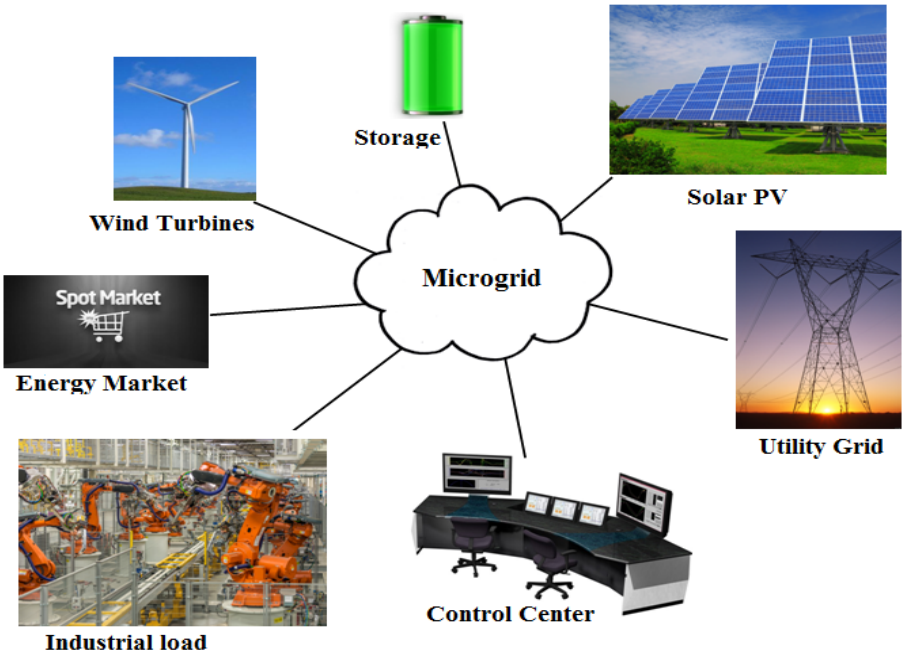


Figure 2.1 – Architecture of a microgrid.

Control and management of microgrids can be established in centralized or distributed manner. As shown in Figure 2.1, centralized control mechanism relies on a central controller and it coordinates the DERs in terms of energy

generation scheduling and protection from over-current due to short circuits. For global optimality, the centralized control manner can have the advantage of high efficiency according to Liang and Zhuang [LZ14]. Decentralized control and operation could be useful in case where distributed control is required (e.g., in remote areas). In such areas, communication network between the DERs and the central controller can be interrupted due to geographical distance or unreliable network connections.

2.2.2 Models of Wind, Solar and Storage

This section discusses models of basic microgrid components such as wind turbines, solar PV panels and ESSs.

2.2.2.1 Wind power

Wind turbines generate electrical power by extracting kinetic energy from air flow using rotors and blades (refer to Figure 2.2.1). If the turbines are installed in locations with strong and sustainable winds, the generated power could be of a significant amount to meet some energy demands.

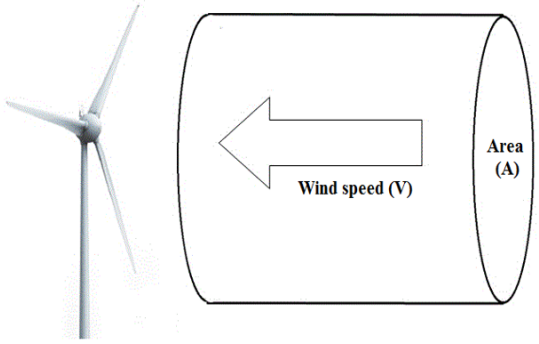
A typical wind turbine is characterized by its *power curve* [Car+13] as shown in Figure 2.2.2. The power curve relates wind power to wind speed. The power P_{wind} (W) extracted from wind speed is proportional to the density of the air, the rotor area, and the cube of the wind speed as in [SPK03]:

$$P_{wind} = \frac{\rho}{2} * A_w * c_p(\lambda_w, \theta) * v^3 \tag{2.1}$$

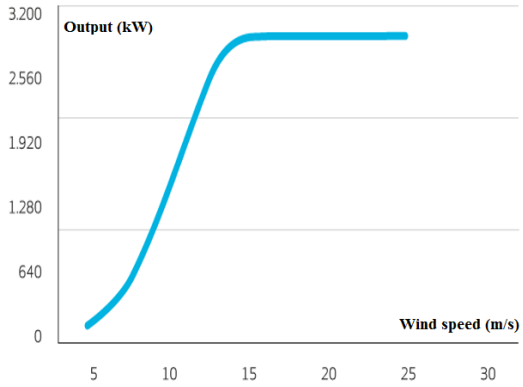
where

- ρ is air density (kg/m^3),
- c_p is performance or power coefficient,
- λ_w is ratio v_t/v_w (ratio between blade tip speed $v_t(m/s)$ and wind speed at hub height upstream the rotor $v_w(m/s)$),
- θ is angle of the blade chord to the plane of rotation (or pitch angle), and
- A_w is area covered by rotor of wind turbine (m^2).

A conventional way of characterizing the ability of a wind turbine to capture wind power is to use the power coefficient c_p which is a function of tip speed ratio (λ_w) and pitch angle (θ). According to Betz's limit [RR11], the maximum achievable value of c_p is 16/27 (59.3%). This upper-bound applies for any type of wind turbine, i.e., no wind turbine can extract kinetic energy from wind speed higher than this coefficient. The power coefficient of modern commercial wind turbines reaches values between 40 to 50% according to [EBL08].

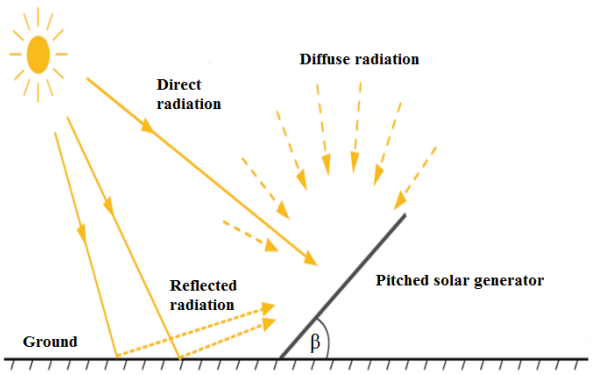


2.2.1: Air flow through rotor area A (m^2) at speed V (m/s)

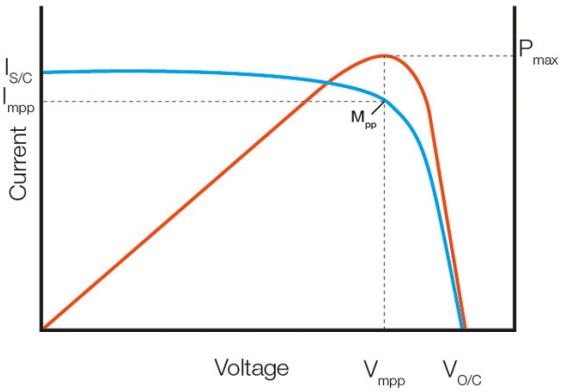


2.2.2: Power curve of a 3MW wind turbine [Ves]

Figure 2.2 – Characteristic of wind power production.



2.3.1: Solar radiation on PV array



2.3.2: PV power curve

Figure 2.3 – Characteristic of solar power production, adapted from [Mer13].

2.2.2.2 Solar PV power

Sunlight is an ambient energy source that is available almost everywhere and can be used to satisfy some parts of energy demand. Solar panels convert the sunlight into electrical energy in a PV system. Solar panels are composed of a number of solar cells that contain semiconducting materials which exhibit photovoltaic effects. To increase conversion efficiency, the PV panel has to operate at its **M**aximum **P**ower **P**oint (MPP) on its PV power curve [EC07]. Figure 2.3 shows a PV array and its power curve characteristic.

In a PV system, the electricity generated by solar cells is given as [RY07]:

$$P_{PV} = SR * \cos\phi * \eta_m * A_p * \eta_p, \tag{2.2}$$

where

- SR is solar radiation (W/m^2),
- ϕ is angle of incidence calculated by considering $\beta = 45°$,
- η_m is efficiency of the **M**aximum **P**ower **P**oint **T**racking (MPPT),
- A_p is area of the PV panel (m^2),
- η_p is efficiency of the PV panel.

According to MacKay [Mac08], typical solar panels have efficiency (η_p) of about 10%; expensive ones with tracking device can perform up to 20%. The MPPT efficiency (η_m) can have a value of 96% according to [RY07].

2.2.2.3 Energy storage systems

As mentioned in the previous chapter, there is a huge potential in **R**enewable **E**nergy **S**ources (RESs) to produce clean electricity and decrease greenhouse gas emissions by reducing our dependence on fossil fuels as primary energy resources. However, the variability of RESs has led to concerns regarding the reliability of an electric grid that derives a large fraction of its energy from these sources as well as the cost of reliably integrating large amounts of variable generation into the power system. Due to these effects, there has been an increased call for the deployment of ESSs as an essential component of future energy systems that use large amounts of variable renewable resources.

As described in [Den+10], ESSs play an important role in electric grid. The significant impact of ESSs is on *energy arbitrage* in which energy is purchased during low-cost off-peak periods and sold back during expensive peak periods. This reduces use of peaking plants (power plants that run only when there is a high demand) and can lower fuel costs. Other roles of ESSs include smoothing of renewable energy generation, operating reserves for electricity regulation, load following to follow longer term (hourly) changes in electricity demand, to black-start a system after system-wide failure (blackout), etc.

Common energy storage technologies in use today include mechanical, thermodynamic, electromagnetic and electrochemical [Che+09]. Mechanical energy storage devices are classified into three types: pumped hydroelectric storage (PHS), compressed air energy storage (CAES) and flywheels. Thermodynamic energy storage (e.g., combined heat and power (CHP)) allows excess thermal energy to be collected for later use. Electromagnetic storage devices such as capacitors and super-capacitors store energy in the magnetic field created by the flow of direct current in a superconducting coil. Among electrochemical storage technologies, the most common **B**attery **E**nergy **S**torage **S**ystems (BESSs) are lead-acid and lithium-ion batteries.

The BESSs have less cost per kWh and commonly used in different applications such as microgrids and electric vehicles. Since we consider BESSs size/capacity determination in chapter 3, we detail their important characteristics from [GFKR15] in the following list:

- *Battery storage capacity (B)*: it is the maximum amount of electric charge a BESSs can store. Storage capacity is measured in Amp-hour (Ah). The fraction of the stored charge that a BESS can deliver depends on factors such as BESSs type, ambient temperature, charge/discharge rate, terminal voltage, etc.

- ***S**tate of **C**harge (SoC) (0⩽SoC⩽100%)*: it describes how full a storage device is. We say that a BESSs is fully charged when its SoC is 100%.

- ***D**epth of **D**ischarge (DoD) (0⩽DoD⩽100%)*: it describes how deeply the battery is discharged. For instance, if a battery is fully charged, its DoD is 0%. Moreover, if a battery is empty, its DoD is 100%. The relationship between DoD and SoC can be described as SoC = 100% - DoD. For some battery types such as lithium-ion, it is not advisable to discharge them 100% DoD, as such discharge could shorten battery life. For this reason, maximum DoD (DoD_{max}) can be set to lower value than 100% (e.g., 90%).

- ***S**tate of **H**ealth (SoH) (0⩽SoH⩽100%)*: This factor reflects the general condition of the storage device with respect to its initial condition.

- *Efficiency (0⩽ η ⩽100%)*: each unit of energy stored is reduced by η units that can be used at a later time. This happens due to inefficiencies in BESSs materials. The efficiency η can be charge efficiency η_c or discharge efficiency η_d. In our case, we consider $\eta = \eta_c = \eta_d$.

- *Self-discharge (in W)*: it is due to non-current-producing *side* chemical reactions when there is no load attached to the BESSs. In most cases, self-discharge amount ranges from 8 to 20% per year at room temperature.

- *Life cycle*: number of full charge/discharge cycles before the capacity of BESSs is reduced to 80%.

Charging and discharging models A BESSs charging/discharging process is described as [CGW12]:

$$b(t+\Delta t) = \begin{cases} b(t) + \Delta t * P^c(t) & \text{charging,} \\ b(t) - \Delta t * P^d(t) & \text{discharging,} \end{cases} \tag{2.3}$$

where $b(t)$ represents state of the BESSs at time t, Δt is time step, and $P^c(t)$ and $P^d(t)$ are charging and discharging rates at time t, respectively.

For normal operations of BESSs, constraints should be imposed on power and energy limits. For example, stored energy cannot be greater than its predefined battery capacity (B) and cannot be lower than its minimum battery capacity (B_{min}), i.e., $B_{min} \leqslant b(t) \leqslant B$, where B_{min} = (1 - DoD_{max})*B. Furthermore, charging/discharging rate constraints are $0 \leqslant P^c(t) \leqslant P^c_{max}(t)$ and $0 \leqslant P^d(t) \leqslant P^d_{max}(t)$, where $P^c_{max}(t)$ is maximum charging rate and $P^d_{max}(t)$ is maximum discharge rate at time t.

2.2.3 Industrial loads and manufacturing types

Industrial load consists of electrical load demands by manufacturing plants or industries. According to Zhang et al. [Zha+16], most manufacturing plants have already installed smart meters and control infrastructures which are necessary for DR. In this section, we discuss a specific manufacturing type that we use for our work on DR in chapter 4. Furthermore, we provide a brief description of other manufacturing types (just for comparison of their working principles) as depicted in Figure 2.4.

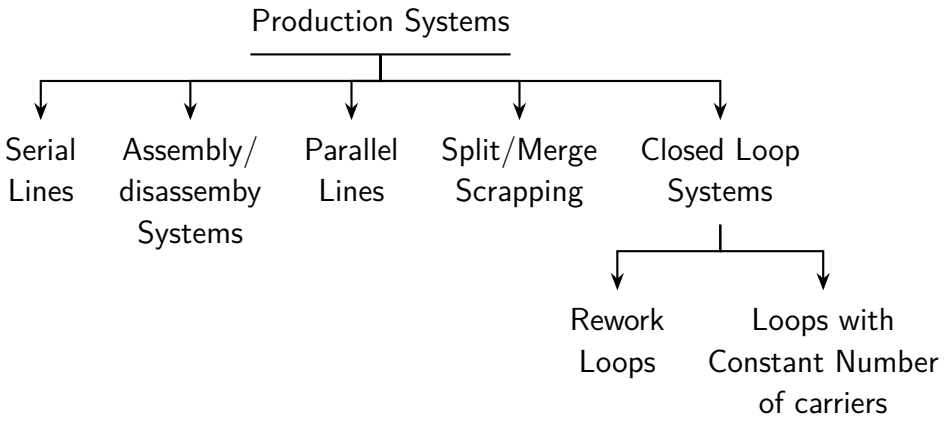


Figure 2.4 – Classification of production systems, adapted from [Li+09].

2.2.3.1 Serial production (or transfer) lines

Since we use notations and concepts of serial production lines in chapter 4, we give more emphasis on these manufacturing types. Serial lines are the most practical production systems in many manufacturing plants. Serial lines have been classified according to the type of part transfers: continuous, asynchronous and synchronous [GF99]. In continuous part transfer lines, the movement of parts is continuous with constant speed. In systems with asynchronous part transfer, each part moves independently of other parts which results in cycle variations between workstations/machines. Parts move simultaneously between machines in

systems with synchronous part transfer. Focusing on the **S**ynchronous **P**roduction **L**ine (SPL) shown in Figure 2.5, the following assumptions are defined [DG92; Li+09]:

- the production line consists of M machines in series that are connected by a conveyor system,
- each part is processed on a machine for some fixed duration, i.e., its processing time,
- the machines are reliable, i.e., there is no failure among the machines,
- the conveyor is blocked until the machine with the maximum processing time finishes,
- machine M_i consumes p_i amount of power when it processes a part,
- the production rate, i.e., number of finished parts per time unit, depends on the processing rate of the last machine(M_M) in the system.

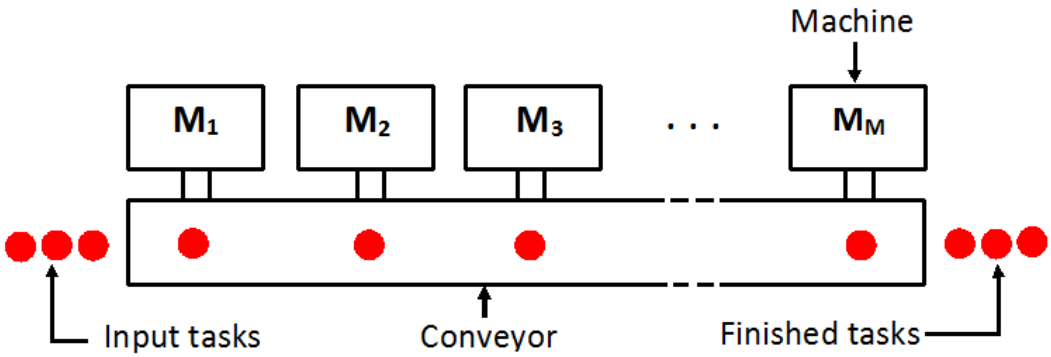


Figure 2.5 – A synchronous production line.

Regarding the working principle of the serial production line, we assume that the machines start processing at the same time but may not necessarily finish equally because the processing times could be different. In the system, unfinished tasks are processed first at machine M_1, then at machine M_2, and so on until the last machine M_M after which they leave the system. All the tasks have to be processed for a duration of C_i at machine M_i. When the conveyor is moving, all the machines wait until an unfinished task arrives in their respective slots. Hence, the system uses **B**locking **A**fter **S**ervice (BAS) mechanism to handle synchronization between the machines and the conveyor.

2.2.3.2 Assembly/disassembly lines

In assembly systems, the assembly machine will only process a part when upstream buffers are not empty, i.e., parts have to wait not only for the machine to become available but also for the other parts of the assembly to arrive before the machine can begin processing. This imposes synchronization constraints at assembly stations and this introduces dependencies between the machines. More details and models are provided by Li, Alden, and Rabaey [LAR05].

2.2.3.3 Parallel lines

To increase production, parallel lines have been used in manufacturing system. The machines at each stage could have identical processing times and aggregating the machines at the stage can form a serial production line to simplify analysis and modeling. Details can be found in [Li04].

2.2.3.4 Split/merge system

Split and merge operations are typically used to increase production capacity and variety, improve product quality, and implement product control and scheduling policies. Merge operations load parts at the same time to compose them to a single part when parts exist in the upstream buffers. Split operations split a part into multiple parts when downstream buffers are not full [LH05].

2.2.3.5 Closed-loop lines

Closed-loop lines are serial production systems where one or more loops are attached to the line. The loops can be used for *rework* of defective parts or quality improvement. Frein, Commault, and Dallery [FCD96] and Gershwin and Werner [GW07] give more details on this manufacturing type.

2.2.4 Spot market

An electricity spot market can be regarded as a market where the electricity can be sold or purchased at varying prices throughout a day. Wholesale transactions (bids and offers) in electricity are managed by the market operator. In Europe, EPEX (European Power EXchange) handles these transactions. It operates in France, Germany, the United Kingdom, the Netherlands, Belgium, Austria, Switzerland and Luxembourg. EPEX spot manages two types of spot markets, namely, day-ahead and real-time markets.

In *day-ahead spot markets*, an auction process is organized by EPEX spot. It determines market price based on the intersection of supply and demand curves as shown in Figure 2.6. Once the prices are determined for each hour of the

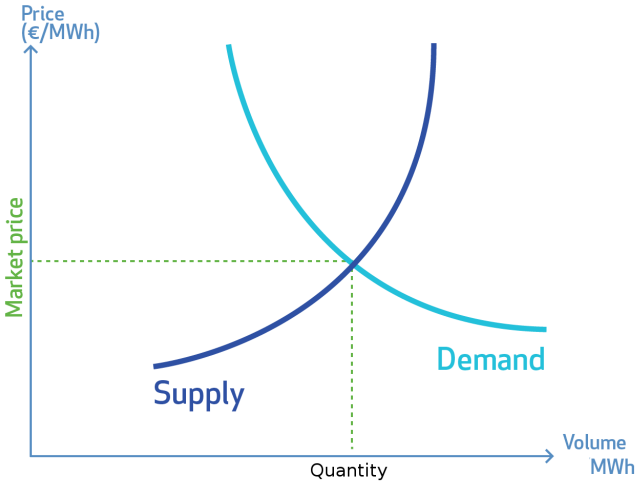


Figure 2.6 – EPEX spot energy supply and demands curves.

following day, the results are published to buyers and sellers from 11:10 am (in Switzerland) and 12:55 pm (in all other markets) [EPE17].

For *real-time spot markets*, the transaction of energy is performed close to real-time (e.g., in 15 minutes) based on contracts. These contracts help to accommodate intermittent energy resources by responding to intra-hour variations of energy production and consumption.

In this section, we discussed concepts of microgrids including the architecture and basic components of an industrial microgrid. The following section presents an important concept: DR.

2.3 Demand Response (DR)

In legacy power systems, the main focus has been improving power supply based on evolution of electricity demands. However, due to recent emergence of smart grids, a **D**emand **S**ide **M**anagement (DSM) also plays a crucial role by managing flexible loads. According to a World Bank report prepared by River [Riv05], DSM is defined as follows.

Definition 2.2 (DSM [Riv05])**.**
DSM encompasses systematic utility and government activities designed to change the amount and/or timing of the customer's use of electricity for the collective benefit of the society, the utility and its customers.

Under DSM, there are two concepts: **E**nergy **E**fficiency (EE) and DR. Referring to [YK05], a definition of EE is given below.

Definition 2.3 (EE [YK05])**.**
EE involves technology measures that produce the same or better levels of energy

services (e.g., light, space conditioning, motor drive power, etc.) using less energy. The technologies that comprise efficiency measures are generally long-lasting and save energy across all times when the end-use equipment is in operation. Depending on the timing of equipment use, EE measures can also produce significant reductions in peak demand.

According to this definition, EE programs involve replacing existing consumers' devices with new devices that are energy efficient (e.g., replacing old incandescent lamps with new fluorescent lamps). These programs offer financial incentives to encourage customers to acquire, install and adopt more energy efficient technologies. EE actions incorporate both short-term conservation actions and long-term investments in EE. Besides investing in EE, customers can also participate in DR programs (see section 2.3.1). The objective of this section is to pursue in the direction of DR because the concepts we discuss here lay the groundwork for our work in chapter 4.

In a power system, DR is regarded as an effective approach to increase electricity grid performance and consumer benefits. According to US Federal Energy Regulatory Commission [Kat+12], DR is defined as follows.

Definition 2.4 (DR [Kat+12])**.**
DR refers to changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized.

In the following sections, we first discuss different categories of DR programs. Then, discussions of DR potential benefits and mathematical problems and approaches in DR are provided. The concluding section of DR portion of this chapter gives a brief description of a smart DR enabling technology called **Open** **A**utomated **D**emand **R**esponse (OpenADR).

2.3.1 DR programs

DR programs throttle energy demands of different loads such as industrial, commercial and residential for adjusting demands to available power productions. DR programs are mainly divided into two branches: price- and incentive-based DR programs. The following sections discuss the two DR programs citing references such as [Riv05; Qdr06; Den+15; ZG16].

2.3.1.1 Price-based DR programs

In price-based DR (PDR) programs, the price of electricity varies over time so that customers are motivated to adjust their power consumption patterns. The price of electricity may differ from peak to off-peak times significantly. Hence,

customers would pay the highest prices during the peak periods and the lowest prices during the off-peak periods. The prices can be communicated to customers a day in advance or in real-time based on pricing mechanisms. The pricing mechanisms are classified into 4 types [Riv05; Qdr06; Den+15]:

- ***T**ime **O**f **U**se (TOU)*: customers are charged with different prices when they consume electricity at different time intervals of a day or different seasons of a year. In most cases, the time intervals are divided into block of 1 hour. TOU rates charge customers based on tariffs for off-peak and peak time blocks. Electricity prices at the peak time blocks are much higher than that at off-peak time blocks.

- ***C**ritical **P**rice **P**eaking (CPP)*: in this pricing mechanism, customers are on TOU rates for most hours of the year. However, they face additional charges during a small number of critical hours when system reliability is jeopardized or very high prices are encountered in wholesale markets because of extreme weather conditions. During non-CPP periods, CPP customers typically receive a price discount. The *Tempo* tariff of EDF (Électricité de France) is an example of CPP tariff. With tempo tariff, a year is divided into 22 red, 43 white and 300 blue days, and each day has a peak and an off-peak periods and corresponding tariffs [Alt+11].

- ***R**eal-**T**ime **P**ricing (RTP)*: in this tariff type, the electricity price usually varies at different time intervals of a day close to real-time, i.e., every 15 minutes or every hour. RTP prices are typically known to customers on a day-ahead, hour-ahead or 15 minutes ahead basis (see an illustration in Figure 2.7b).

- ***I**nclining **B**lock **R**ate (IBR)*: rate structure of this tariff has two levels: lower and higher blocks (see Figure 2.7c). If customers' hourly/daily/monthly energy consumption exceeds a predefined threshold, electricity prices climb up to the higher block. Several energy providers in USA such as Pacific Gas and Electric (PG&E), Southern California Edison (SCE), San Diego Gas and Electric (SDG&E) have been using IBR tariffs since the 1980s [Bor08].

A graphical summary of some of the price-based DR programs is shown in Figure 2.7. In the figure, the TOU rates are divided into three time block types: off-peak, mid-peak, and on-peak. The electricity price at the on-peak time block is much higher than that at the mid-peak and off-peak time blocks so that customers shift their loads over another time horizon.

2.3.1.2 Incentive-based DR programs

Incentive-based DR (IDR) programs reward participating customers for reducing their electricity usage in response to DR requests from grid operators. IDR

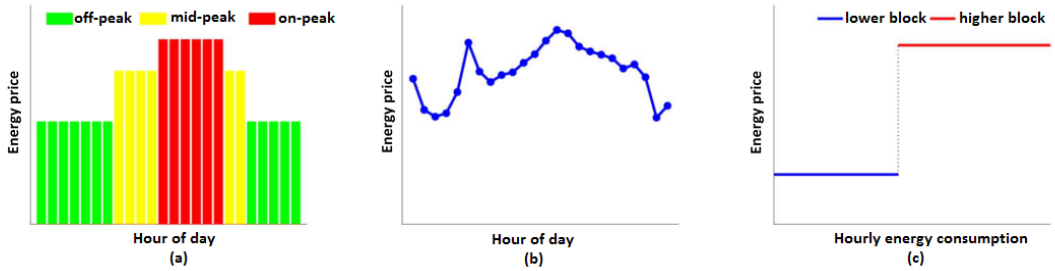


Figure 2.7 – Illustration of PDRs: (a) TOU, (b) RTP, and (c) IBR [Den+15].

programs diversify the ways in which demand side management contributes to reliable and efficient grid operations because these programs can be tailored according to specific requirements. The programs in this category are listed as follows [Qdr06; ZG16]:

- *Direct load control* is a program in which customers receive incentive payments for allowing the utility operator some degree of control over their equipments such as air conditioners and water heaters. While direct load control programs are primarily offered to residential and small commercial customers, it usually cannot be applied to industrial customers due to safety concerns.

- *Interruptible/curtailable service* is used to provide customers with a discount rate or bill for agreeing to reduce load on request. These services have traditionally been offered only to the largest industrial and commercial customers.

- *Demand Bidding/Buyback Programs* encourage large customers to bid into a wholesale electricity market by providing load reduction prices at which they are going to be curtailed.

- *Emergency Demand Response Programs* provide incentive payments for customers to reduce loads during reliability-triggered events. Enrolled customers that do not respond to this program may incur penalties.

- *Capacity Market Programs* give incentive payments for customers that can commit to providing pre-specified load reductions when system contingencies arise. These programs typically entail significant penalties for customers that do not respond when asked for load reduction.

- *Ancillary Services Market Programs* provide incentive payments for customers from the grid operator for committing load curtailment to be standby as operating reserves. If load curtailment is requested from the operator, the customers may get paid according to prices on the spot market.

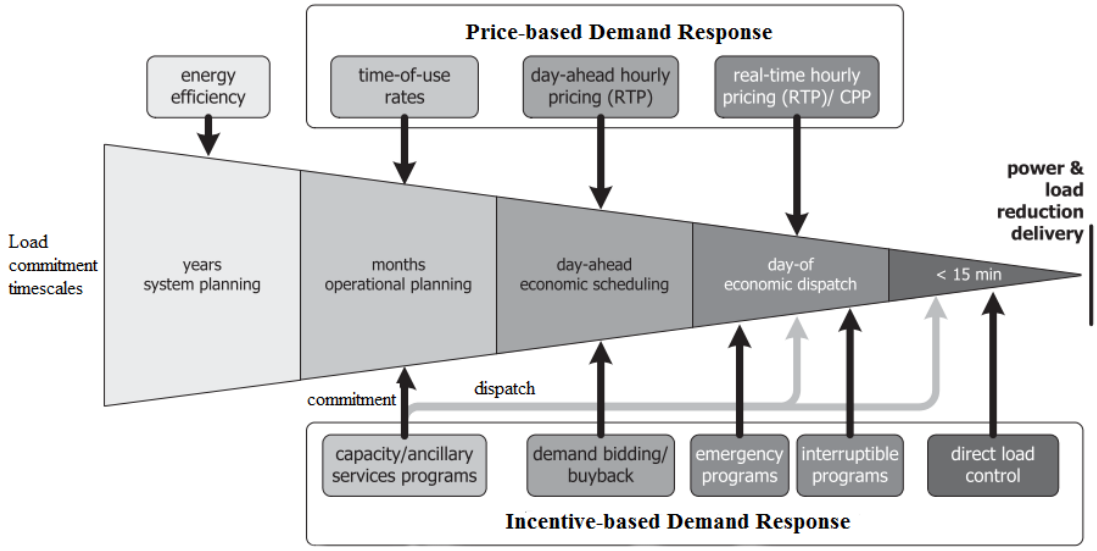


Figure 2.8 – Time scales of DR programs [Qdr06]

Figure 2.8 shows time scales of both price- and incentive-based DR programs. Price-based DR programs can be incorporated into utilities' planning at different time scales that can include TOU, CPP and RTP rates. Incentive-based DR programs may be introduced at virtually all time scales which can notify customers in hourly, daily, and real-time fashions based on the programs. In chapter 4, we consider the incentive-based DR in which power consumption is upper bounded by the request of the utility grid.

2.3.2 Potential benefits of DR

DR has a broad range of potential benefits primarily as resource savings (e.g., reduction in peaking plants) that improve the efficiency of power systems. In practice, these benefits depend on many factors including purpose, design, and performance of the implemented DR programs, the enabling technologies and the structure of the electricity markets [Qdr06]. The benefits of demand response can be classified in terms of whether they are interesting directly to participants or to some or all groups of electricity consumers as follows:

- *Participant bill savings*: electricity bill savings and incentive payments for customers willing to adjust their loads in response to current supply costs or other incentives.

- *Bill savings for other customers*: DR results in lower wholesale market prices which in turn reduces supply costs to retailers that provide electricity to non-DR costumers.

- *Reliability benefits*: refers to reduced probability of power outages and not incurring higher financial costs and inconvenience for customers.
- *Market performance*: DR prevents the exercise of market power by electric power producers, i.e., it encourages fair marketers (e.g., prevents over reduction of prices to influence the market).
- *Improved choices*: more options for customers to manage their electricity costs by choosing energy providers according to their will.
- *Power system security*: grid operators are endowed with more flexible means to meet contingencies and reduce costs.

2.3.3 Mathematical problems and approaches in DR

Based on a survey by Deng et al. [Den+15], this section provides a list of existing mathematical problems in DR and different approaches for solving them. The major DR mathematical problems include *utility maximization*, *cost minimization*, *price forecast*, and *renewable energy integration*.

2.3.3.1 Utility maximization

In utility maximization problem, the aim is to first define *utility functions* that quantify levels of customer satisfactions as a function of customer's power consumption [Sam+10; LCL11]. Then, the objective function is to maximize customer's welfare respecting varies constraints including power consumption profiles of devices, available power, etc. Several approaches such as *convex optimization* [Sam+10] and *dynamic programming* [JL11] are proposed in the literature to tackle this problem.

2.3.3.2 Cost minimization

This problem defines an energy consumption scheduling where the objective is to minimize energy costs for the customers. Potential approaches include *game theory* and *convex optimization* [Sam+10].

2.3.3.3 Price forecast

In price prediction problem, the goal is to include a dynamic price predictor in an energy consumption scheduler. The dynamic price predictor computes prices that can reflect the actual wholesale price at the time of consumption and type of a day (working or weekend). Mohsenian-Rad and Leon-Garcia [MRLG10] proposed a *linear programming (LP)* approach for the price prediction problem.

2.3.3.4 Renewable energy integration

This problem considers integration of DR with intermittent RESs such as wind, solar, etc. It incorporates energy generation of the resources into an optimization problem (e.g., minimizing cost by using locally generated power when electricity price is high or storing for later use). This type of DR problem was considered by Jiang and Low [JL11].

2.3.4 OpenADR - a DR tool

OpenADR [All13] was developed at Demand Response Research Center (DRRC) of Lawrence Berkeley National Laboratory (LBNL) [LBN] and its first specification was released in 2009. To enhance the development, adoption and compliance of OpenADR standards throughout the energy industry, a collaboration between industry stakeholders was initiated to form an OpenADR Alliance [Ope] in 2010. The alliance was created to standardize, automate and simplify DR, to meet growing cost effective energy demand of utilities, and to allow customers controlling the way they use energy. In the version 1.0 of the specification [Pie+09], the LBNL described OpenADR as follows.

Definition 2.5 (OpenADR [Pie+09])**.**
OpenADR is a communication data model designed to facilitate sending and receiving DR signals from a utility or independent system operator to electric customers. The intention of the data model is to interact with building and industrial control systems that are pre-programmed to take action based on a DR signal, enabling a demand response event to be fully automated, with no manual intervention. The OpenADR specification is a highly flexible infrastructure design to facilitate common information exchange between a utility or Independent System Operator (ISO) and their end-use participants. The concept of an open specification is intended to allow anyone to implement the signalling systems, providing the automation server or the automation clients.

The OpenADR 1.0 specification was based on OASIS (Organization of Structured Information Standards) Energy Interoperation (EI) standard [OAS]. Under OASIS' EI, OpenADR 2.0 profile builds on OpenADR 1.0 and it handles more complex interactions of DR and DERs, while keeping in mind the requirements of diverse markets and stakeholders needs. In this section, we discuss OpenADR architecture, some services and implementations of OpenADR 2.0.

2.3.4.1 OpenADR architecture

In OpenADR architecture, there are two communicating nodes that act as clients or servers:

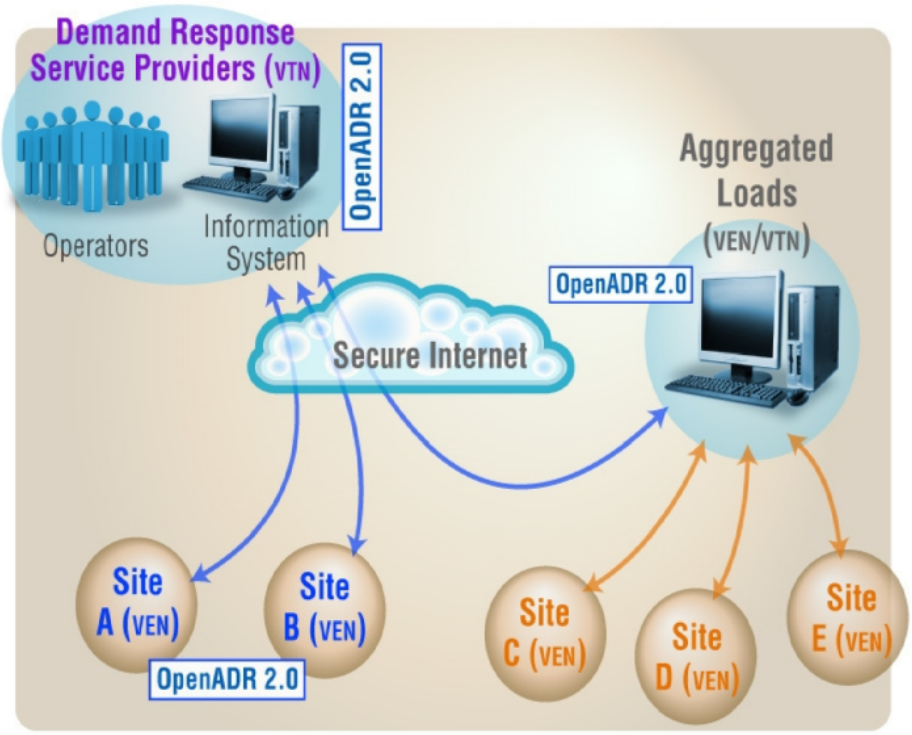


Figure 2.9 – OpenADR architecture [Haa13].

- nodes that publish and communicate event information to other nodes (e.g., utilities). These nodes are called **V**irtual **T**op **N**ode (VTN) and act as OpenADR servers,
- nodes that receive, process and respond to the information (e.g., electricity consumers). These nodes are called **V**irtual **E**nd **N**ode (VEN) and act as OpenADR clients.

Figure 2.9 shows an architecture of OpenADR with different combination of VTN and VEN interactions. In the figure, a VTN (i.e., DR service provider) can communicate events (e.g., electricity prices, grid reliability, etc) directly to an end customer (e.g., site A) or to an aggregator VEN. Then, the aggregator becomes VTN for another VENs such as site C, D, and E. Based on pair-wise relationship of VTN–VEN, a complex structure could be implemented that can address complex interactions in DR such as shown in Figure 2.10.

In Figure 2.10, certain nodes (B, E and G) act as both VTN and VEN. The arrows from VTN to its VENs could model a DR event initiated by the utility grid A that can invoke an operation on its second level VTNs (B to E) which are group of aggregators. For example, the second level VTN B can invoke services on its VENs F, G and H, which represent their customers. The customers might be industrial parks with multiple facilities, real estate developments with multiple tenants, or a company headquarters with facilities in many different geographical

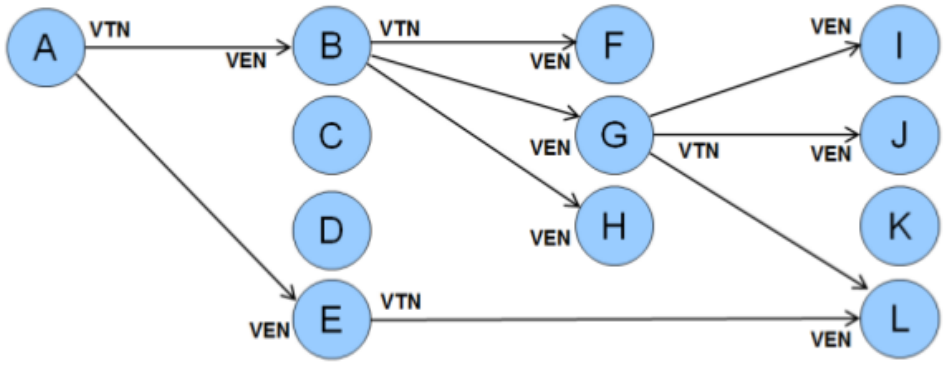


Figure 2.10 – Example of interactions between VTN and VEN [Haa13].

areas.

Regarding to communication protocols, the VTN and VEN can communicate using HTTP (HyperText Transfer Protocol) either in PUSH mode where the VTN initiates communication or in a PULL mode where the VEN continuously requests VTN for information [All13]. **X**ML **M**essaging and **P**resence **P**rotocol (XMPP) can also be used as transport mechanism between VTN and VEN. Both HTTP and XMPP use XML (eXtensible Markup Language) format for a standardized data representation during OpenADR message communication. According to the protocol specification, the VTN should support both HTTP and XMPP protocols, but the VEN should support either of the two.

2.3.4.2 OpenADR services

To facilitate common information exchanges between VTNs and VENs, OpenADR 2.0 provides the following 8 services [All13]:

- *EiRegisterParty*: used by VENs to perform in-band registration to VTNs. In this service, VENs and VTNs agree on parameters such as transport mechanisms (HTTP or XMPP), OpenADR profiles (2.0a or b), etc.
- *EiEnroll*: used by VENs to enroll their resources to participate in DR.
- *EiMarketContext*: used to discover DR program rules, standard reports, etc. Since market information rarely changes, it is not necessary to send market information every time.
- *EiEvent*: used by VTNs to convey DR events to VENs to indicate whether resources are going to participate in the event. It is a core function in the information models of OpenADR.
- *EiQuote*: used by VTNs to communicate complex dynamic prices such as block and tier tariffs to VENs.

- *EiReport*: used by VTNs and VENs to exchange historical, telemetry, and forecast reports. Using this service, VENs can report their resource status including availability, forecast, real-time energy and curtailment readings.
- *EiOpt*: used by VENs to communicate Opt-in and Opt-out (temporary availability to accept DR) schedules to VTNs. The VEN sends Opt-in message to the VTN if it accepts the DR request or Opt-out otherwise.
- *EiAvail*: used by VENs to indicate when an event may or may not be accepted. VENs execute *EiAvail* with respect to a Market Context. If VTNs know the availability and opt information of VENs, they can improve their ability to estimate VEN's response to events or requests.

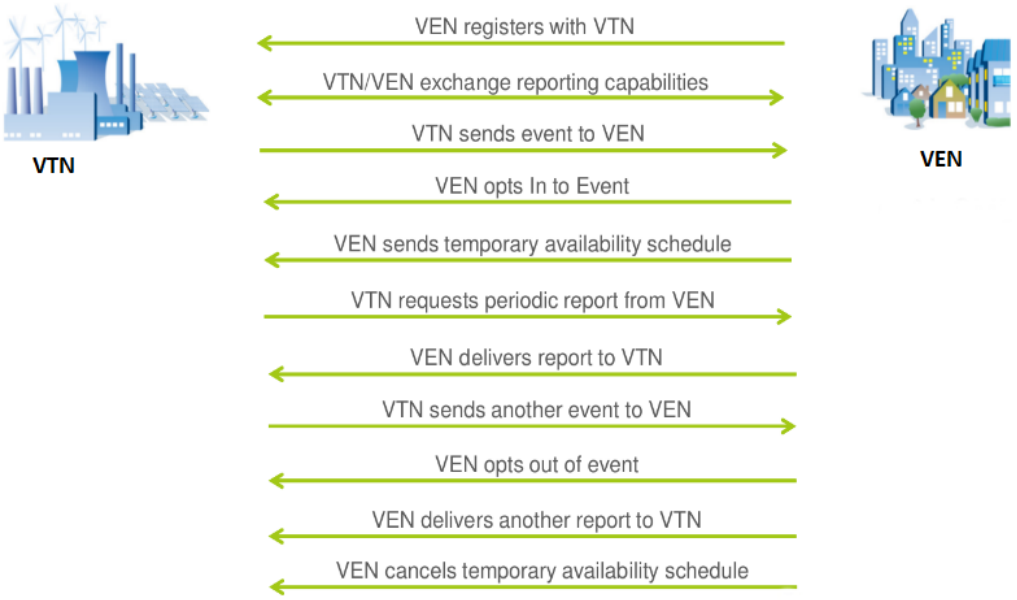


Figure 2.11 – VTN and VEN communications using OpenADR services [Haa13].

As shown in Figure 2.11, VTN and VEN commences communication by using *EiRegisterParty* service which is sent from VEN to VTN, assuming that enrollment and market information exchange were done a priori. After agreeing on OpenADR parameters in *EiRegisterParty*, VEN and VTN exchange report capabilities using *EiReport* service. Then, the VTN can send DR event to VEN by specifying DR start time, DR duration, amount/percentage of load shedding, etc in *EiEvent* service. At this moment, the VEN not only responds with *EiOpt* but also with temporary availability schedule. The communication continues with exchange of periodic reports to verify if the DR event has resulted in the targeted load shed. The VTN and VEN can repeat the same process in other times to notify another DR event.

2.3.4.3 OpenADR implementations

Existing implementations of OpenADR use either HTTP or XMPP as transport mechanisms. OpenADR Alliance certified OpenADR implementations are mostly provided by two organizations: EPRI (Electric Power Research Institute) and EnerNOC. EPRI selected a simple HTTP implementation and uses *C#* programming language to implement the OpenADR services in VEN side and *Ruby on Rails* web-application framework for VTN side. Source codes for VTN and VEN are available from [EPR]. EnerNOC provides XMPP-based OpenADR implementation and use *Java* and *Python* programming languages for VEN and *Grails/Groovy* web-application framework for VTN. Source codes are available in EnerNOC's GitHub [Ene].

Since the OpenADR protocol specification is based on XML, the availability of Java language specific tools for creating, manipulating and binding XML documents to program elements encourages us to choose the EnerNOC implementations of OpenADR. As both Java and Python languages are frequently used in the enterprise METRON, the integration of OpenADR to METRON's architecture was done with relative ease. Since the source code of EnerNOC only provide the *EiEvent* service, we modified the code to add the other services. In chapter 4, we provide a use case of OpenADR in METRON.

In the next section, we detail relevant concepts of service curve which will be used in chapter 3 to model power generation of DERs.

2.4 Service curves of Network Calculus

NC is a system theory that has been developed to give a theoretical framework for analyzing performance guarantees in computer networks based on min-plus and max-plus algebras. Specifically, NC is used to compute worst case bounds on delay and buffer requirements in a network. It was initially introduced by Cruz [Cru91] for communication network elements where the analysis of delay bounds and buffer requirements followed a non-probabilistic approach. After this initial work on NC, two works were published around the beginning of 2000. In 2000, the book of Chang [Cha00] gave formal presentation of the operators of the NC and their use in a case where time and data are discrete values. Then, in 2001, Le Boudec and Thiran [LBT01] described all the necessary tools of NC as well as the conditions of application and provided several illustrating examples of application on different problems in their book.

NC consists of two branches: *stochastic* and *deterministic* NC. In *Stochastic Network Calculus* (SNC), computations of worst case performance bounds are based on probabilistic distributions, i.e., some violations of the deterministic bounds are tolerable with small probability. More details on SNC can be found

in [JL08]. In this work, we focus on the *deterministic Network Calculus* (DNC) where deterministic worst case performance bounds are computed. Under DNC, NC provides basic concepts including arrival, service curves and shapers. Since we don't rely on arrival curves and shapers, a reader may want to refer to [LBT01] for more details on the two concepts.

In the following sections, focusing on the service curve concepts, we discuss notations and theories which are relevant to our work on modeling of energy generation of DERs in chapter 3. First, a brief overview of min-plus and max-plus algebras is given in section 2.4.1. Next, section 2.4.2 details the service curve concepts with few demonstrating examples. Finally, some applications of NC are provided in section 2.4.3.

2.4.1 Min-plus and Max-plus Algebras

Before defining the concepts of service curves, we first introduce the two important underlying mathematical concepts: min-plus and max-plus algebras. Since the min-plus algebra is frequently used in service curves, we give more emphasis on the concepts of the min-plus algebra and detail only few notations on the max-plus algebra concepts.

2.4.1.1 Min-plus Algebra

In conventional algebra, addition and multiplication are the two most common operations on real $\mathbb{R}$ and integer $\mathbb{Z}$ numbers. The particular properties of the two operators make the algebraic structure ($\mathbb{Z}$, +, x) as a commutative ring and ($\mathbb{R}$, +, x) as a commutative field [LBT01]. In min-plus algebra, the multiplication operator is replaced by the addition operator and the addition operator is replaced by the infimum (or minimum if it exists) operator ($\wedge$). Min-plus operators can also be applied to $+\infty$ that results in another algebraic structure ($\mathbb{R} \cup \{+\infty\}, \wedge, +$). Algebraic properties of the structure such as *associativity*, *commutativity*, etc are detailed in the book of Le Boudec and Thiran [LBT01](page 105). Based on the properties of the min-plus algebra, we discuss important features such as property of wide-sense increasing functions and convolutions that are necessary to define service curve concepts.

Wide-sense increasing functions A function f is said to be wide-sense increasing if and only if $f(s) \geqslant f(t)$, $\forall s \geqslant t$. We adopt the following notations:

- G: a set of non-negative wide-sense increasing functions,
- F: a set of non-negative wide-sense increasing functions such that $f(t) = 0$ if $t < 0$.

The range of F and G functions is $\mathbb{R}^+ = [0, +\infty]$. Some of these functions can be used as service curves (see section 2.4.2.1).

Pseudo-inverse of wide-sense increasing functions For an invertible function f, there exists a function f^{-1} such that $f^{-1}(f(t)) = t$ for all t. However, wide-sense increasing functions are not always invertible. Hence, a pseudo-inverse f^{-1} of a function $f \in F$ is defined as:

$$f^{-1}(x) = \inf_{t \geqslant 0}\{t : \ \ f(t) \geqslant x\}. \tag{2.4}$$

Concave and convex functions Concave and convex functions are important functions of the min-plus algebra. A function $f : \mathbb{R} \mapsto \mathbb{R}$ is convex if and only if

$$f(ux + (1-u)y) \leqslant uf(x) + (1-u)f(y), \ \ \forall x, y \in \mathbb{R}, u \in [0,1], \tag{2.5}$$

and, a function $f : \mathbb{R} \mapsto \mathbb{R}$ is concave if and only if $-f$ is convex or if and only if

$$f(ux + (1-u)y) \geqslant uf(x) + (1-u)f(y), \ \ \forall x, y \in \mathbb{R}, u \in [0,1]. \tag{2.6}$$

Some functions such as rate-latency functions are convex functions and some piece-wise linear functions are concave functions.

Min-plus convolution In classical system theory, the convolution of two functions $f(t)$ and $g(t)$ is defined as:

$$(f * g)(t) = \int_0^t f(t-s)g(s)\,\mathrm{d}s, \ \ t > 0. \tag{2.7}$$

Changing the addition into infimum and the multiplication into addition, the following definition of the min-plus convolution is given.

Definition 2.6 (Min-plus convolution)**.**
Assuming that $f, g \in F$, the min-plus convolution of f and g is:

$$(f \otimes g)(t) = \inf_{0 \leqslant s \leqslant t}\{f(t-s) + g(s)\}, \tag{2.8}$$

given that $(f \otimes g)(t) = 0$ if t < 0.

An important property of min-plus convolution is that if f and g are convex and piece-wise linear functions, their convolution $f \otimes g$ is obtained by putting end-to-end the different linear pieces of the individual functions, sorted by increasing slopes. This property is very useful during concatenation of two functions as will be shown in section 2.4.2.4.

Min-plus deconvolution The dual operation of the min-plus convolution is defined as follows.

Definition 2.7 (Min-plus deconvolution)**.**
Assuming that two functions f and g are in F, their min-plus deconvolution is given by:

$$(f \oslash g)(t) = \sup_{u \geqslant 0}\{f(t+u) - g(u)\}. \tag{2.9}$$

Illustrating examples and proofs are provided in the work of Le Boudec and Thiran [LBT01], and Van Bemten and Kellerer [VBK16]. These references can be consulted if more details are required.

2.4.1.2 Max-plus Algebra

In max-plus algebra, the algebraic structure is changed into $(\mathbb{R} \cup \{-\infty\}, \vee, +)$ by replacing the infimum operator by supremum operator $\vee$ and $+\infty$ by $-\infty$. Since we don't use the max-plus algebra in this work, interested readers may want to refer to the book of Le Boudec and Thiran [LBT01]. The following section discusses the concepts of service curves.

2.4.2 Service curve concepts

The concept of service curve is used to abstract the details of packet scheduling which requires a network or node to offer some guarantees to flows. The following definition of a service curve gives a general service curve concept used in NC. Referring to [LBT01], a service curve is defined as follows.

Definition 2.8 (Service curve [LBT01])**.**
Consider a system S and a flow through S with input and output functions R and R^*, respectively. We say that S offers to the flow a service curve β if and only if for all $t \geqslant 0$, there exists some $s \geqslant 0$, with $s \leqslant t$, such that

$$R^*(t) - R(s) \geqslant \beta(t-s), \tag{2.10}$$

or equivalently,

$$R^* \geqslant R \otimes \beta, \tag{2.11}$$

or alternatively,

$$R^*(t) \geqslant \inf_{s \leqslant t}\{R(s) + \beta(t-s)\}, \tag{2.12}$$

where $\beta(0) = 0$, β is wide-sense increasing, and $R(t)/R^*(t)$ are cumulative number of input/output bits in interval $(0,t]$.

Figure 2.12 illustrates the concept of service curve. In the figure, the output R^* of the system must always be greater than the convolution of the input R

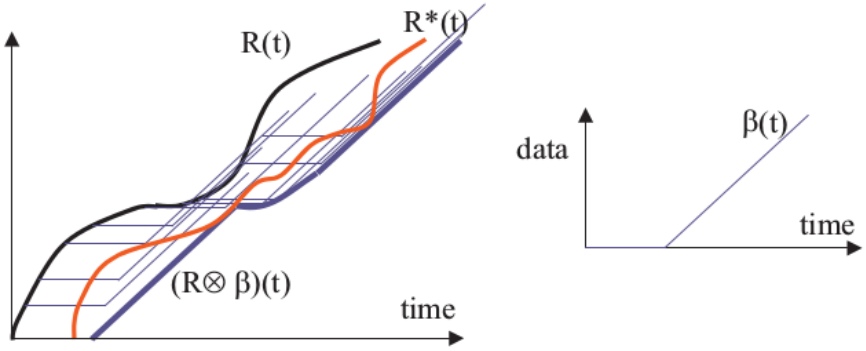


Figure 2.12 – An illustration of service curve concept [LBT01].

with the service curve β of the system. The output R^* lies in the area between the input R and $R \otimes \beta$. From the figure, we can also see that the service curve concept provides a lower bound on the service a system can offer. Based on this general definition of service curves, the following sections detail three types of service curves: functions as service curves, strict, and maximum service curves. Furthermore, we investigate concatenation of service curves.

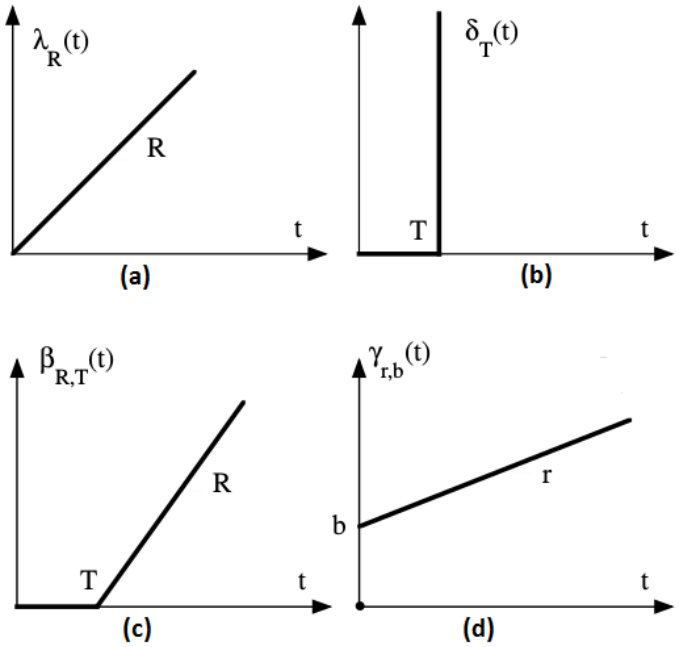


Figure 2.13 – Common functions as service curves: (a) peak-rate (b) burst-delay (c) rate-latency, and (d) affine functions. Adapted from [LBT01].

2.4.2.1 Common functions as service curves

The functions most commonly used as service curves include *peak-rate*, *burst-delay*, *rate-latency*, and *affine* functions. They are defined as follows:

- *peak-rate function*: for some rate $R > 0$, a peak-rate function is defined as: (see Figure 2.13a):

$$\lambda_R(t) = \begin{cases} Rt & \text{if } t > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.13}$$

- *burst-delay function*: referring to Figure 2.13b, a burst-delay function with delay time $T > 0$ is given as:

$$\delta_T(t) = \begin{cases} +\infty & \text{if } t > T, \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

  By convolving any function with δ_T, it can be shifted to the right by T.

- *rate-latency function*: with delay $T > 0$ and rate $R > 0$, a rate-latency function can be expressed as (see Figure 2.13c):

$$\beta_{R,T}(t) = R[t-T]^+ = \begin{cases} R(t-T) & \text{if } t > T, \\ 0 & \text{otherwise.} \end{cases} \tag{2.15}$$

  The rate-latency service curve is usually used to model a generalized process sharing (GPS) scheduler where each flow gets a share of at least R.

- *affine function*: for some rate $r > 0$ and burst $b > 0$, an affine function is defined as (see Figure 2.13d):

$$\gamma_{r,b}(t) = \begin{cases} rt + b & \text{if } t > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.16}$$

2.4.2.2 Strict service curve

Definition 2.9 (Strict service curve)**.**
A system S offers a strict service curve β to a flow if, during any backlogged period of duration d, the output R^* of the system is at least equal to $\beta(d)$. Mathematically, for any backlogged period $(s, t]$,

$$R^*(t) - R^*(s) \geqslant \beta(t-s). \tag{2.17}$$

In strict service curve property, a flow is guaranteed a service during any backlogged period, which is not possible with the general service curve. Since the strict service curve is stricter than the general service curve, if a node offers β service curve to a flow, then it also offers the same curve β as a general service curve to the flow. For example, a GPS node offers a strict service curve in the form of $\beta = rt$ because it will always serve the backlogged data at least at the promised rate r.

2.4.2.3 Maximum service curve

Definition 2.10 (Maximum service curve)**.**
Considering a system S with input R and output R^* functions, we say that S offers a maximum service curve γ to a flow if and only if $\gamma \in F$ and

$$R^* \leqslant R \otimes \gamma, \tag{2.18}$$

or alternatively,

$$R^*(t) \leqslant R(s) + \gamma(r - s), \ \ \forall t, s \leqslant t, \tag{2.19}$$

or equivalently with consideration of backlog $B(\cdot)$,

$$R^*(t) - R^*(s) \leqslant B(s) + \gamma(r - s), \ \ \forall t, s \leqslant t. \tag{2.20}$$

In the maximum service curve concept, a system is allowed to provide an upper bound on the service it can give to the flows. For example, a node is said to offer a maximum service curve of δ_T if and only if it imposes a minimum virtual delay of T. In the same way, the virtual delay $d(t)$ satisfies $d(t) \leqslant D$ for all t if a flow traverses a node with maximum service curve γ such that $\gamma(D) = 0$.

2.4.2.4 Concatenation and aggregation of service curves

In the above section, we have discussed several service curves for a standalone system. Now, we would like to concatenate the service curves and find a service curve that represents a global view of the systems concatenated in series. Assume that a flow traverses two systems S_1 and S_2 in series and the systems offer service curves β_1 and β_2 to the flow, respectively. Hence, for the input R at S_1, the output R^* at S_2 can be expressed as:

$$R^* \geqslant (R \otimes \beta_1) \otimes \beta_2 = R \otimes (\beta_1 \otimes \beta_2). \tag{2.21}$$

According to Van Bemten and Kellerer [VBK16], $\beta_1 \otimes \beta_2$ is not necessary strict even if β_1 and β_2 are strict service curves. Figure 2.14 shows the concatenation of two rate-latency service curves β_{R_1,T_1} and β_{R_2,T_2} of systems S_1 and S_2, respectively. Generally, for n rate-latency service curves β_{R_i,T_i}, $i \in [1, 2, ..., n]$, the concatenation becomes $\beta_{min\{R_1,...,R_n\},T_1+...+T_n}$ which is also a rate-latency service curve.

For aggregation property, a single service curve can be obtained by summing service curves of the systems. For example, if there are n service curves, the aggregated system service curve becomes $\beta_{R_1+...+R_n,T_1+...+T_n}$. A similar expression is also provided in [Wan+12]. We rely on this aggregation property in chapter 3 to model a composite power production curve of DERs.

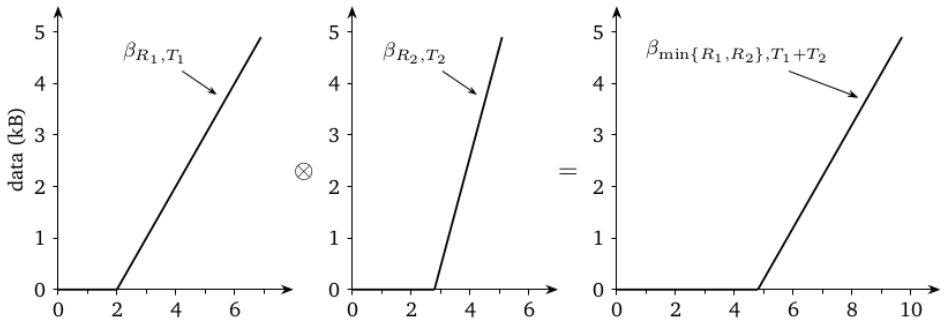


Figure 2.14 – Concatenation of service curves [VBK16].

2.4.3 Applications of Network Calculus

Most of the applications of NC are in the context of computer and communication networks. For instance, in [GDR11], the authors relied on network calculus concepts to determine whether a switched network may satisfy timing constraints of a real-time application. They computed upper bound on end-to-end delay in an Ethernet switch by setting service curves for fair queuing and weighted round robin scheduling algorithms.

Focusing on wireless networks, a work in [AZLB12] used NC to analyse delay and backlog performance metrics for multi-hop fading communication channels. Other applications on NC include delay and queue size analysis in AFDX (Avionics Full-Duplex Switched Ethernet) [FFG06] and guaranteed service networks [LB98].

In our work, we extend the service curve concepts of NC to energy domain and we will describe it in chapter 3. The next section discusses basic queuing theories which are relevant to chapter 4.

2.5 Queuing theory overview

In general, queues (waiting lines) are parts of our daily life. For example, once we finished shopping in a supermarket, we wait at checkouts if there are people ahead of us. What happens if we arrive at the checkouts during peak hours? How many checkout points to activate or deactivate based on the number of customers in the queue? The very same questions can be asked in queuing systems such as manufacturing systems, data communication, computer systems, etc. To analyze such situations in queuing systems, a queuing theory-based model is particularly a useful tool.

The first queuing theory problem was raised by calls in telephone exchange in which Erlang [Erl09] was the first to address congestion problems of calls at the beginning of 20^{th} century. Engineers and mathematicians were inspired by his work to further extend the results in computer science, operation research,

telecommunication, traffic engineering, etc. Here, our objective is to rely on some queuing theory models for performance analysis of a manufacturing system which is given in chapter 4. Hence, we only mention relevant theories from queuing models to attain our goal.

We start with discussions of Kendall's notations and Little's theorem in sections 2.5.1 and 2.5.2, respectively. Then, section 2.5.3 highlights average-based performance measures such as mean queue length and average waiting time. After that, temporal evolution of a queuing system is provided in section 2.5.4. Finally, some research works on application of queuing theory to manufacturing systems are given in section 2.5.5.

2.5.1 Kendall's notation for queues

Figure 2.15 shows basic components of a queuing system. In the figure, the small circles represent customers or parts and the small boxes represent servers or processing stations in the system. Kendall's notation provides a standard way to describe and classify queuing systems.

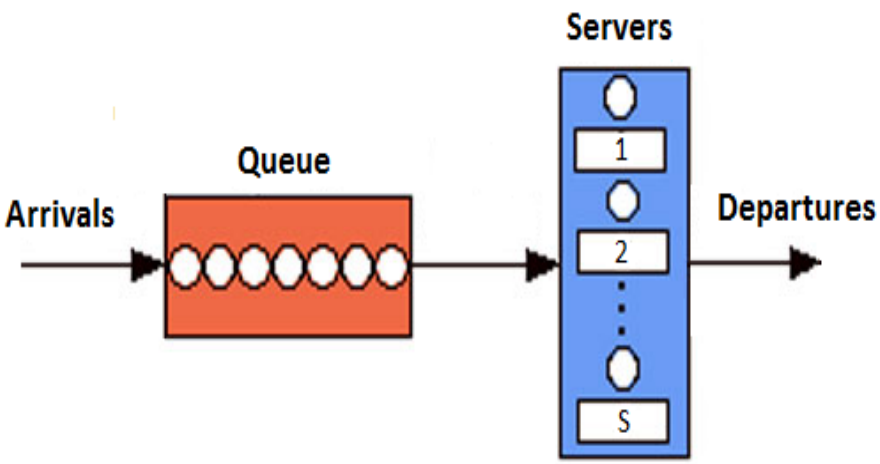


Figure 2.15 – A queuing system.

In Kendall's notation, a queuing system is denoted by [Szt12]:

$$\mathbb{A}/\mathbb{B}/m/\mathbb{K}/n/\mathbb{S},$$

where

- $\mathbb{A}$: Arrival process
  In arrival process, we assume that interarrival times are **I**ndependent and **I**dentically **D**istributed (IID). Most commonly used distributions are M (exponential or memoryless/Markovian), D (deterministic), G (general) and E_k (Erlang with shape parameter k).

- $\mathbb{B}$: Service time distribution
  Service times are also assumed to be IID and they can have the same distributions as the arrival process.

- m: Number of servers
  Available servers to serve customers. For example, the $G/G/1$ queue has single server while the $G/G/c$ queue has c servers.

- $\mathbb{K}$: Number of places in the system
  It represents the maximum number of customers allowed in the system including those in service. Incoming customers may be refused to enter the system if this value is the maximum number the system can hold.

- n: Client/costumer source size
  n indicates a population size from which customers arrive to a queuing system. For instance, in the $M/M/1$ notation, the population size is considered to be unlimited or infinite.

- $\mathbb{S}$: Service discipline
  Servers serve customers based on a specific service discipline. The service disciplines in queuing systems include **F**irst **I**n **F**irst **O**ut (FIFO), LIFO (Last in First Out), PS (Process Sharing), SIRO (Service In Random Order), and SPT (Shortest Processing Time first). The default service discipline is FIFO if it is not mentioned explicitly.

As an example of Kendall's notation, the shorthand $M/M/m/K/n$ stands for a system where arrival and service processes are exponential distributed, the service is carried out with m servers to serve n customer sources and system capacity of K slots with FIFO service discipline. In our work, we use a $D/D/1$ queue where both arrival and service processes are deterministic and a single server serves unlimited number of customers with FIFO service discipline. The queuing system is used to represent a processing station (or machine) in a manufacturing system. Consequently, production line of a manufacturing system can be viewed as a tandem of $D/D/1$ queues. Hence, we limit ourselves to give details on performance measures of $D/D/1$ queues.

2.5.2 Little's Formula

Little's formula or theorem was proposed by John Little [Lit61] to prove a queuing formula that relates average number of customers in a queue to average waiting time and arrival rate of the customers. The theorem is defined as follows.

Definition 2.11 (Little's theorem, adapted from [Lit61])**.**
Under steady state conditions, the long-term average number of customers in a

stable system is equal to the mean arrival rate multiplied by the average time a customer spends in the system.

Based on this definition, the stationary mean queue length $\mathbb{E}[Q]$ of queue Q is formulated as follows:

$$\mathbb{E}[Q] = \hat{\lambda}\mathbb{E}[D_Q] \tag{2.22}$$

where $\hat{\lambda}$ is mean arrival rate and $\mathbb{E}[D_Q]$ represents the mean delay of customer in queue Q from the time it enters to the queue until its service is completed. In section 2.5.3.3, we use this formula to obtain the average delay $\mathbb{E}[D_Q]$ after computing the average queue length $\mathbb{E}[Q]$.

2.5.3 Average-based performance measures of $D/D/1$ queue

A $D/D/1$ queuing system is the simplest queuing type (but very useful) where arrival rates (λ) have deterministic IID inter-arrival times and service times (μ) are also distributed in the same way. To avoid that the queue eventually grows to infinity, we have to require that $\lambda \leqslant \mu$, with the equality case also stable in deterministic queues. Important performance measures in the queuing system include mean waiting times and average number of customers in the system (or mean queue length).

In this section, we provide server utilization and the average-based performance measures such as the mean waiting time and mean queue length in the $D/D/1$ queue. Now, we recall some performance measures of the $D/D/1$ queue from works of Sztrik [Szt12] and Zukerman [Zuk13]. Assume that if an arrival and a departure occur at the same time, the departure takes place first. This assumption is not required for Markovian queuing systems such as $M/M/1$, $G/M/1$, and $M/G/1$ queues since occurrences arrival and departure events are probabilistic. In the following sections, analytical results of the performance measures of the $D/D/1$ queue are provided.

2.5.3.1 Server Utilization

The server utilization of the $D/D/1$ queue can be considered in two cases. If we consider the case where $\lambda > \mu$, the queue size grows to infinity as $t \to \infty$. Hence, there is always a customer in the queue waiting for service which makes the server always busy. The result is that the utilization is equal to one. However, in this case, the system becomes unstable due its queue size.

In the other case where $\lambda < \mu$, the system is always stable. Let us assume that the first customer arrives at $t = 0$. Its service time ends at $t = 1/\mu$. Then, the second arrival will occur at $t = 1/\lambda$ and it will be served at $t = 1/\mu + 1/\lambda$. This continues for all arrivals and it forms a deterministic cyclic process where the queue size is either 0 or 1. Thus, the cycle contains three parts: a period of customer arrival of $1/\lambda$, then there is a customer to be served for a period of

$1/\mu$, and a period where there is no activity ($1/\lambda$ - $1/\mu$). Therefore, the server utilization becomes $U = (1/\mu)/(1/\lambda) = \lambda/\mu$.

2.5.3.2 Mean queue length

If $\lambda < \mu$ (i.e., when the system is stable), customers entering the system are served before the next one arrives. This means that the queue length alternates between 1 or 0 with $1/\lambda$ time spent in state 1 and ($1/\lambda$ - $1/\mu$) time spent in state 0. Assume that the probability of having n customers in the queue Q at a randomly chosen point in time is $\mathbb{P}(Q = n)$, the mean queue length is defined as [Zuk13]:

$$\mathbb{E}[Q] = \sum_{n=0}^{\infty} n\mathbb{P}(Q = n) \tag{2.23}$$

Now, the probability that there is one customer in the queue Q is $\mathbb{P}(Q = 1) = (1/\mu)/(1/\lambda)$, and the probability that there is no customer is $\mathbb{P}(Q = 0) =$ 1 - $(1/\mu)/(1/\lambda)$. Based on equation (2.23), the mean queue length is $E[Q] =$ $0^*\mathbb{P}(Q = 0) + 1^*\mathbb{P}(Q = 1) = \lambda/\mu$, which is the same as the server utilization given above.

2.5.3.3 Mean waiting time

Using the Little's formula (see section 2.5.2), the mean waiting time is $\mathbb{E}[D_Q] = \mathbb{E}[Q]/\lambda = (\lambda/\mu)/\lambda = 1/\mu$. According to Sztrik [Szt12], every other queuing system ($M/M/1$, $G/G/1$, etc) has greater mean waiting time than $D/D/1$.

2.5.4 Temporal evolution of arrivals and departures

The performance measures discussed above are time-averaged values under steady state conditions. The steady state measures are independent of the initial conditions of the system. To characterize and predict the behavior (transient and steady state) of the system at any time t, we define two processes: arrival $Arr(t)$ and departure $Dep(t)$ processes. The process $Arr(t)$ is defined as the total number of arrivals and $Dep(t)$ is defined as the total number of departures during the interval [0,t]. For a system in which arrivals and departures occur one at a time (e.g., $D/D/1$ queue), a realization of $Arr(\cdot)$ and $Dep(\cdot)$ is displayed in Figure 2.16. From these two processes, we can compute the average values such as mean waiting time and average number of jobs in the queue.

Now, let t_j^a and t_j^d are arrival and departure times of job j, respectively. Assuming that the system starts working at time t_a and finishes at time t_b, the average waiting times (WT) of jobs can be expressed as [CF10]:

$$WT_{t_a,t_b} = \frac{1}{n}\sum_{j=0}^{n}\left(t_j^d - t_j^a\right) = \mathbb{E}[D_Q], \tag{2.24}$$

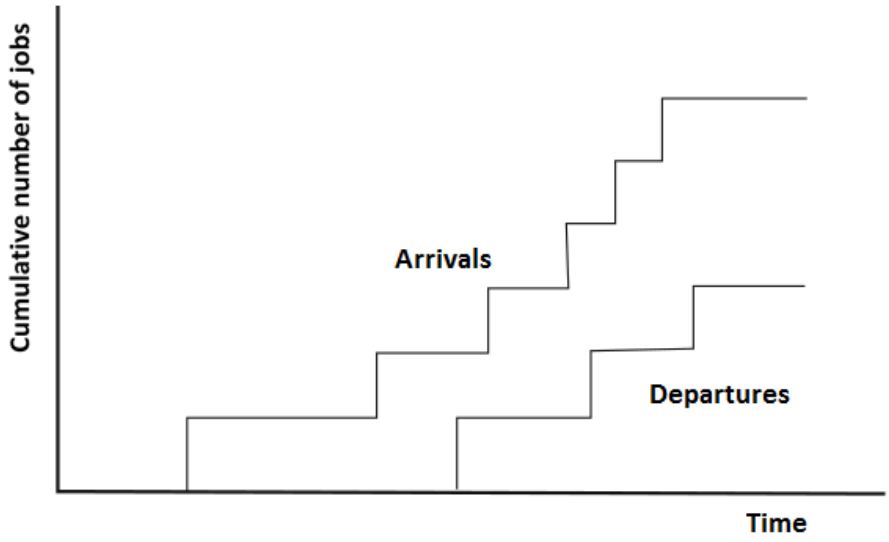


Figure 2.16 – A realization of arrival and departure processes of jobs where the left curve represents cumulative arrivals and the right curve represents cumulative departures [CF10].

where n is the number of job arrivals to the system during the interval (t_a,t_b). The area between $Arr(t)$ and $Dep(t)$ represents the integral of the number of jobs in the system at time t (see Figure 2.16). Hence, the average number of jobs waiting in the system during the time interval (t_a, t_b) is given by:

$$N_{t_a,t_b} = \frac{1}{t_b - t_a} \int_{t_a}^{t_b} \Big[Arr(t) - Dep(t)\Big] \mathrm{d}t = \mathbb{E}[Q]. \tag{2.25}$$

According to Curry and Feldman [CF10], Equation 2.25 is called *work-in-process* in manufacturing systems where jobs are either undergoing processing or waiting in a queue for processing. From Little's theorem, the relationship between the mean waiting time and the average number of jobs is given as:

$$N_{t_a,t_b} = \hat{\lambda} * WT_{t_a,t_b}, \tag{2.26}$$

where $\hat{\lambda} = n/(t_b - t_a)$ is the mean arrival rate of the jobs arriving to the system in the interval (t_a, t_b).

The above equations depend on the values of $Arr(\cdot)$ and $Dep(\cdot)$ to compute average-based performance measures. In chapter 4, we analytically compute the temporal arrival and departure processes for a manufacturing system. Then, we verify the analytical computations using simulations.

2.5.5 Applications of queuing theory to manufacturing

A considerable body of research has shown that queueing theory can be a useful analysis tool for several application domains. Besides its application in telecommunications and computer networks, queuing theory based models for manufacturing

systems have been gained much attention since three decades ago with initial works by Buzacott and Shanthikumar [BS92]. After this work, several applications of queuing models in manufacturing systems are provided in the literature.

Considering multi-stage assembly lines, Manitz [Man08] proposed a $G/G/1/K$ (general arrival and service processes, single server with FIFO service discipline and system capacity of K slots) queuing model for performance evaluation of the system with general processing times and finite buffers between machines. Another work on assembly lines with Bernoulli reliability model and finite buffers is given in [Jia+16]. Further surveys on queuing theory-based manufacturing systems are provided in [PH96], [GF99] and [Li+09].

2.6 Summary

In this chapter, we discussed several important concepts, models and theories which are relevant to the next chapters. The following list summarizes this chapter referring back to the sections:

- Section 2.2 presented the microgrid concept focusing on issues such as architecture of microgrid, models of wind/solar power and energy storages, industrial/manufacturing load types and spot market.
- In section 2.3, we detailed **D**emand **R**esponse (DR) programs (price- and incentive-based), benefits of DR, approaches and mathematical problems in DR and an automated DR protocol called **Open A**utomated **D**emand **R**esponse (OpenADR).
- Then, the service curve concepts of **N**etwork **C**alculus (NC) were discussed in section 2.4. Not only the underlying mathematical concepts such as min-plus and max-plus algebras but also concatenation and aggregation of service curves were provided.
- Finally, in section 2.5, we presented basic queuing theory notations such as Kendall's notations and Little's theorem. Moreover, some performance measures and temporal evolution of the $D/D/1$ queue were discussed.

Based on these relevant concepts, the following chapter details our work on energy supply side of the industrial microgrid. Specifically, we rely on service curve concepts to model power production of **D**istributed **E**nergy **R**esources (DERs). The next chapter also discusses on smoothing of power generation of wind and solar using **B**attery **E**nergy **S**torage **S**ystems (BESSs).

Part II

Energy Management in Supply and Demand Sides

Chapter 3

Modeling and Smoothing of DERs

Contents

3.1 Introduction

In this chapter, we present two of our works on the energy supply side of the industrial microgrid. In section 3.2, we provide a literature review on modeling of **D**istributed **E**nergy **R**esources (DERs) and smoothing of their power productions. Then, based on the service curve concepts defined in the previous chapter (section

2.4), we give a service curve model of the power generations of DERs in section 3.3. The model will help us know the minimum power produced by DERs so that we can do energy transactions on spot markets or with the utility grid if there is energy deficit or surplus. Section 3.4 discusses smoothing of the renewable resources using three smoothing algorithms. Performance measures of the algorithms are also provided in the section. After that, numerical results are given in section 3.5. Finally, section 3.6 concludes the chapter by invoking important results discussed in this chapter.

3.2 Related work

The intermittent nature of **R**enewable **E**nergy **S**ources (RESs) such as solar and wind cause significant power fluctuations and integrating them to power systems requires control mechanisms in the form of smoothing and modeling of these resources. This section provides a review of relevant research works on the two categories: modeling and smoothing of DERs in sections 3.2.1 and 3.2.2, respectively.

3.2.1 DER modeling

A challenging aspect of DERs in the microgrid is that the energy produced by the renewable resources is intermittent. Hence, individual components of DERs should be modeled in order to know system's total available energy and then energy transactions can be done to minimize energy costs. To model power generation of DERs, there are two type of models: simulation and analytical models. In [ZYF07] and [CS02], the authors considered simulations models to estimate the actual performance of **P**hotovoltaics (PV) modules under varying operating conditions. In simulation models, the models may not capture the system's dynamics and the results could be inaccurate and misleading due to inappropriate simulation parameters or other factors. However, analytical models can capture complex dynamics of the system better than simulation models. Following the description in [WJM12], generic analytical frameworks such as **N**etwork **C**alculus (NC) provide better performance modeling and evaluation of DERs. The research works mentioned above relied on complex probability functions to give probabilistic upper and lower bounds on energy production. In this work, we use the deterministic version of NC to give a deterministic lower bound on an energy DERs produce.

As noted in the preceding chapter (section 2.4.3), the application of NC theories were limited to performance analysis of computer and communication networks. There are only few research papers on NC's application in energy domain. For example, [WLL11] and [Wan+12] considered a stochastic NC framework for modeling DERs and energy demands. They used probabilistic

cumulative distribution functions to express energy arrival processes to an energy queue. Stochastic NC is also used in [WJM12] for performance modeling and evaluation of network systems powered with RESs. Le Boudec and Tomozei [LBT12] relied on the concept of service curves to ensure the existence of a feasible battery charging and discharging schedules given an energy demand curve. They computed a sufficient battery size so that the feasible online schedule can be ensured during the scheduling period.

Our motivation to extend NC theory to energy management in a microgrid is that the theory can provide an analytical framework for different scenarios and its applicability to different research domains. Among the concepts of network calculus theory, we rely on the concepts of service curves as described in section 2.4 of chapter 2. In our context, we use service curves to model DERs of the power system. After that, we obtain the total energy production which is used to satisfy some part or the total energy demand. Then, when the demand is greater than the supplies, we can purchase energy from either spot markets or the utility grid. Hence, we also discuss different strategies to minimize energy procurement costs by using locally produced energies as much as possible.

3.2.2 Smoothing of energy production

One way to reduce power fluctuations in RESs is to smooth power production using **E**nergy **S**torage **S**ystems (ESSs) such as electric double-layer capacitor [Kak+09], superconducting magnetic energy storage [TKF89], fuel cells [RT88], and **B**attery **E**nergy **S**torage **S**ystems (BESSs) [HGB10]. The ESSs can be used to store surplus energy and to shave peak demands. Further roles of ESSs in power systems are described in section 2.2.2.3 of the previous chapter. In this work, we use BESSs such as lithium-ion and lead-acid batteries for the purpose of smoothing.

For smoothing power production of RESs, different approaches have been proposed in the literature. A **S**imple **M**oving **A**verage (SMA) based smoothing was proposed by Ellis et al. [Ell+12], Johnson et al. [Joh+13], and Hund, Gonzalez, and Barrett [HGB10]. They used the SMA method to mitigate short-term fluctuations of PV power using BESSs. An **E**xponential **M**oving **A**verage (EMA) method with hydrogen storage system is used by Tesfahunegn et al. [Tes+11]. The EMA gives more weights on recent values. In both SMA and EMA, length of the averaging window determines how the storage systems charge or discharge. If the window is long, it requires the storage systems to cover the difference between the actual and smoothed powers, even if the fluctuation is not significant. In [Li+11], a fuzzy wavelet transform method is used to smooth out wind and solar power productions using BESSs.

Although the moving average methods (SMA and EMA) are easy to implement alongside the BESSs, they are affected by peaks and troughs during power

generations which could result in bigger battery sizes. In [Add+17c], we proposed a Gaussian-based smoothing algorithm that solves the pitfalls of the moving average methods. Gaussian filters have been extensively used in computer vision and image processing domains. In our work, we would like to investigate it in energy domain. Section 3.4.1 describes the Gaussian-based, SMA and EMA methods in more details. Then, in section 3.5.3, we compare performances of the three approaches using real and forecast datasets.

In the following section, we present one of the two works, namely, modeling energy generation of DERs using service curves.

3.3 DER modeling using Service Curves

This section details the modeling of DERs power productions using the concept of service curves. We first provide the service curve model of solar and wind powers in section 3.3.1. After having the two service curves, we modify the battery equation to reflect the changes by including the service curves and energy demands. Then, in section 3.3.2, we discuss a problem of energy supply and demand by incorporating the service curves of DERs. Finally, cost minimization strategies are provided in section 3.3.3.

3.3.1 Service curves of DERs

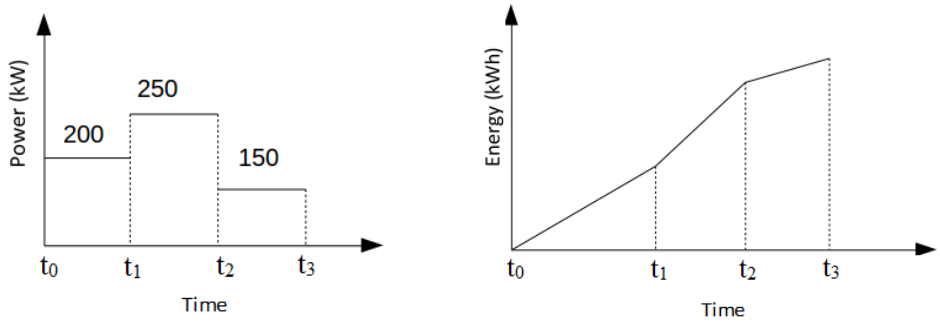
We consider a system where solar and wind powers provide energy and a battery system that can charge or discharge according to difference between energy supply and demand. Based on the description of service curves in chapter 2 (section 2.4), we define service curves to model the minimum amount of energy that the DERs can provide. A service curve $S_i(t)$ is defined as:

$$S_i(t) = \alpha_i + \beta_i * t \tag{3.1}$$

where (α_i, β_i) are service curve parameters of energy resource i. This is to say that the node of DERs provides the amount of energy which is represented by its service curve $S_i(t)$ that can be seen as a local minimum [Flo+13] amount of energy at time t. Equation (3.1) is similar to Equation (2.16) (defined in section 2.4.2.1 of chapter 2) which is one of the common functions as service curves.

3.3.1.1 Service curves of solar and wind

Performing mathematical integration on extracted power over a specific time interval will give us *energy* (in kWh). From these sets of data, we can obtain the service curve parameters such as α_i (in kWh) and β_i (in kW) where i is a component of DERs (solar or wind).



3.1.1: Solar/wind power generation 3.1.2: Corresponding service curve

Figure 3.1 – Service curve example for a node of DERs (solar or wind).

Figure 3.1 illustrates how to setup a service curve for a solar or wind power. For the sake of simplicity, assume that the power generation is constant over a short period of time as shown in Figure 3.1.1. Referring to Figure 3.1.1 and 3.1.2, in the time interval $[t_0, t_1]$, the generated power is 200kW and the corresponding linear line has a slope of 200. Therefore, the service curve in this interval is $S(t) = 200t$, $t \in [t_0, t_1]$, that is $\alpha = 0$ and $\beta = 200$. Then, when t is between t_1 and t_2, the new service curve becomes $S(t) = 200 + 250t$, that is $\alpha = 200$ and $\beta = 250$. Hence, the curve in Figure 3.1.2 is piece-wise linear curve that contains the different service curves in the interval $[t_0, t_3]$.

In the above example, we setup a service curve for a single energy resource. Now, we would like to aggregate the different service curves. The aggregation of heterogeneous power supply sources provide a single service curve that represents the total energy production. Consider a power system which consists of N renewable power producers in parallel. If a power producer n $(n = 1, 2, ..., N)$ provides an energy service curve S_n, then the power system provides a system service curve defined as:

$$S(t) = S_1(t) + S_2(t) + \cdots + S_N(t), \tag{3.2}$$

As noted in previous chapter (section 2.4.2.4), aggregated system service curve can be obtained by summation of the individual service curves of the energy resources. Next, we incorporate the system service curve into battery charging and discharging processes.

3.3.1.2 Energy storage

The energy storage (e.g., battery) is modeled by a discrete time process $b(t)$ (refer to section 2.2.2.3 of chapter 2), with maximum battery capacity B and it can be defined recursively. If the energy generated from the PV/wind hybrid system is greater than the load for a particular hour, then the surplus energy is stored in

the battery and the battery is charged as:

$$b(t) = \texttt{min}\Big\{B, b(t-1) + [S(t) - D(t)] * \eta_c\Big\}, \tag{3.3}$$

where $S(t)$ represents a system service curve as shown in equation 3.2, $D(t)$ is an energy demand at time t, and η_c is charge efficiency of the battery. When the battery reaches its maximum value B, any excess energy generated is wasted and hence the battery cannot be charged.

In the discharging case, if the energy demand is greater than the supply for a particular hour, then the battery is discharged in order to fill the void between energy supply and demand. The battery can be discharged as below:

$$b(t) = \texttt{max}\Big\{(1 - DoD_{max}) * B, b(t-1) - [D(t) - S(t)]/\eta_d\Big\}, \tag{3.4}$$

where η_d is the battery discharge efficiency. Due to physical constraints, the minimum battery level should be lower bounded with the value of **D**epth of **D**ischarge (DoD). For instance, if DoD_{max} has a value of 80% (for lead-acid/lithium-ion battery as shown in Table 3.1), then the minimum battery level (B_{min}) is determined as B_{min} = (1 - 0.8)*B = 0.2B. Hence, if the battery decreases to its minimum value B_{min}, then the energy deficit cannot be met by the battery.

3.3.2 Energy supply and demand balance

In the above section, we provided our approach of modeling the DERs using service curves of NC. In this section, we use the models to formulate our problem which is to balance demand and supply. We consider a constant energy demand D for a particular period and N energy resources. Suppose that each resource n = 1, 2, ..., N provides a service curve guarantee S_n. Then, the total energy demand is satisfied if:

$$S_1(t) + S_2(t) + \cdots + S_N(t) \geqslant Dt \tag{3.5}$$

Equation (3.5) can be interpreted as the aggregation of all the source service curves have to be greater than the demand curve.

As shown in Figure 3.2, the cumulative energy demand is given by straight line Dt which has a slope of D. When the summation of energy provided by wind and solar is greater than the demand line, we say that the demand is met. Otherwise, we discharge the battery if there is enough energy. If the battery is unable to cover the shortage, we need to buy the required energy from either spot market or utility grid based on costs of energy. In the following section, we outline different strategies that minimizes energy procurement costs and enable us to use as much as possible locally produced energy.

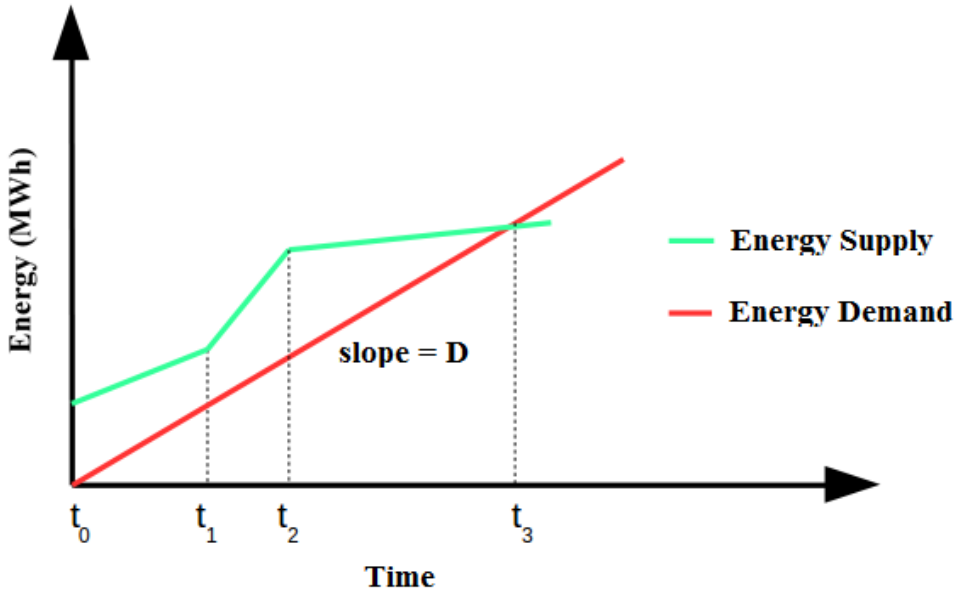


Figure 3.2 – Energy demand and supply curves [Add+15].

3.3.3 Cost minimization strategies

Before discussing different strategies for procurement cost minimization, we define a net energy procurement cost as follows:

$$Net_{cost} = \sum_{t \in year} \Big[Pr_{buy}(t) * E_{buy}(t) - Pr_{sell}(t) * E_{sell}(t) \Big] \tag{3.6}$$

where $Pr(\cdot)$ and $E(\cdot)$ are selling/buying prices in €/MWh and sold/bought energy in MWh, respectively. When we consume local energy from wind, solar or battery, we assume that the energy cost is zero. Furthermore, we assume that there is no upper bound on the amount of energy that can be bought from the spot market. However, there are minimum limits on energy volume that we want to sell or buy as discussed below.

In this section, we define three strategies that minimize the reliance on external energy sources and make use of local energy more often. We compare them based on their performances against the total net energy procurement cost defined by Equation (3.6). Algorithm 1 describes an implementation of the three cost minimization strategies. The following three strategies are taken from our work on service curves [Add+15].

3.3.3.1 Strategy 1 – Sell excess energy

In this strategy, we would like to sell the excess energy which is a leftover after a demand is met, i.e., when $S[t] > D[t]$ in Algorithm 1. According to EPEX Spot [EPE17] (see section 2.2.4 of chapter 2), the possibility to sell energy on the spot

Algorithm 1 Cost minimization strategies

Require: (S, D, $grid\_price$, $spot\_price$, B)

$net\_cost \leftarrow 0$ ▷ energy procurement net cost
$stored\_energy \leftarrow \epsilon$ ▷ stored energy in the battery
$T \leftarrow 8760$ ▷ Number of hours in a year
$buy\_when\_cheap \leftarrow false$ ▷ To buy energy from spot when it is cheap
$sell\_excess\_energy \leftarrow false$ ▷ To sell excess energy or not
$spot\_energy\_volume \leftarrow 1000$ ▷ Minimum energy to sell on spot in kWh
$max\_spot\_price \leftarrow \epsilon$ ▷ Max. spot price we wish to buy
$sold\_energy \leftarrow 0$ ▷ aggregated sold energy
$bought\_energy \leftarrow 0$ ▷ aggregated bought energy
while $t < T$ **do**
    $difference \leftarrow S[t]$ - $D[t]$
    if $difference \geqslant spot\_energy\_volume$ **then** ▷ Surplus energy
        if $sell\_excess\_energy == true$ **then** ▷ STRATEGY 1
            $sold\_energy \leftarrow sold\_energy + difference$
            $net\_cost \leftarrow net\_cost$ - $(spot\_price[t] * difference)/1000$
        else ▷ STRATEGY 2
            if $stored\_energy + difference \leqslant B$ **then** ▷ If battery is not full
                $stored\_energy \leftarrow stored\_energy + difference$
            else if $diff == stored\_energy + difference$ - $B \geqslant 1000$ **then**
                $stored\_energy \leftarrow B$
                $sold\_energy \leftarrow sold\_energy + diff$
                $net\_cost \leftarrow net\_cost$ - $(spot\_price[t] * diff)/1000$
    else if $0 < difference < spot\_energy\_volume$ **then**
        if $stored\_energy < B$ **then** ▷ Store the leftover
            $stored\_energy \leftarrow \min(B, stored\_energy + difference)$
    else ▷ There is no enough energy, BUY!
        $bought\_energy \leftarrow bought\_energy + difference$
        $net\_cost \leftarrow net\_cost + \min(spot\_price[t], grid\_price[t]) * difference/1000$
    if $spot\_price[t] < max\_spot\_price$ **then** ▷ STRATEGY 3
        $stored\_energy \leftarrow B$ ▷ Make the battery full
        $bought\_energy \leftarrow bought\_energy + (B - stored\_energy)$
        $net\_cost \leftarrow net\_cost + spot\_price[t] * (B - stored\_energy)/1000$

market depends on the minimum energy volume available to be sold. In EPEX Spot, the minimum volume is 1MWh (*spot_energy_volume* in the algorithm). Therefore, if we have an excess energy greater than 1MWh for some period, we can sell it on the spot market at market price. Otherwise (i.e., excess energy less than 1MWh), if the battery is not full, we store it for future use.

3.3.3.2 Strategy 2 – Store excess energy

Under this strategy, instead of selling the excess energy, we would like to store it for future use if the battery is not full. If the battery is full, there are two possibilities: either sell the excess energy if it is above 1MWh or otherwise dispose it. For bigger battery sizes, this condition cannot happen too often because they can store much of the surplus energy due to their sizes. However, for smaller battery sizes, if there are lower energy consumption, the user can opt to sell the excess energy. This could be another good strategy. Strategy 2 also minimizes the net energy cost by providing zero-cost local energy from solar, wind and battery. If the energy from battery, solar and wind cannot meet the load, we buy from either the utility grid or spot market whichever has lower energy price.

3.3.3.3 Strategy 3 – Use external energy to charge battery

The objective of this strategy is to make use of the varying prices of electricity in spot markets. The cost of energy on the spot market is cheap during some periods of the day. Charging the batteries during these periods can be a good strategy to minimize the net energy procurement cost defined by Equation 3.6. We assume that the battery can start charging at the beginning of a particular time slot (which has length of 1 hour) and it can be ready at the end of the slot. We give charging precedence for locally produced energy from solar and wind. Moreover, if the battery is not full yet, we can buy the energy from the spot market. For this, we set a price limit that we would like to buy from the market (*max_spot_price* in the algorithm). For example, If we set this limit to 20€/MWh, then we buy energy from the market whenever the price is under 20€/MWh. Under this strategy, we store more energy for future use.

In section 3.5.2, we compare the performance of the strategies using real datasets of energy demand of a factory, sport prices, solar and wind data. The section also provides discussions on payback periods which are the period of time required to recoup the funds expended in an investment on solar PV panels, wind turbines and batteries.

The above sections presented our work on modeling of DERs using concepts of service curve. The following section outlines the second part of this work: smoothing energy production of the RESs.

3.4 Smoothing renewable energy production

As discussed in section 3.2.2, there are ways to deal with fluctuations in renewable energy resources by implementing smoothing in conjunction with energy storage systems. In the literature, the commonly used approaches for smoothing (for instance, simple and exponential moving averages) result in a big battery size. Hence, we would like to come up with a smoothing algorithm which has more performance in terms of battery sizes, i.e., it has to attain lower battery sizes.

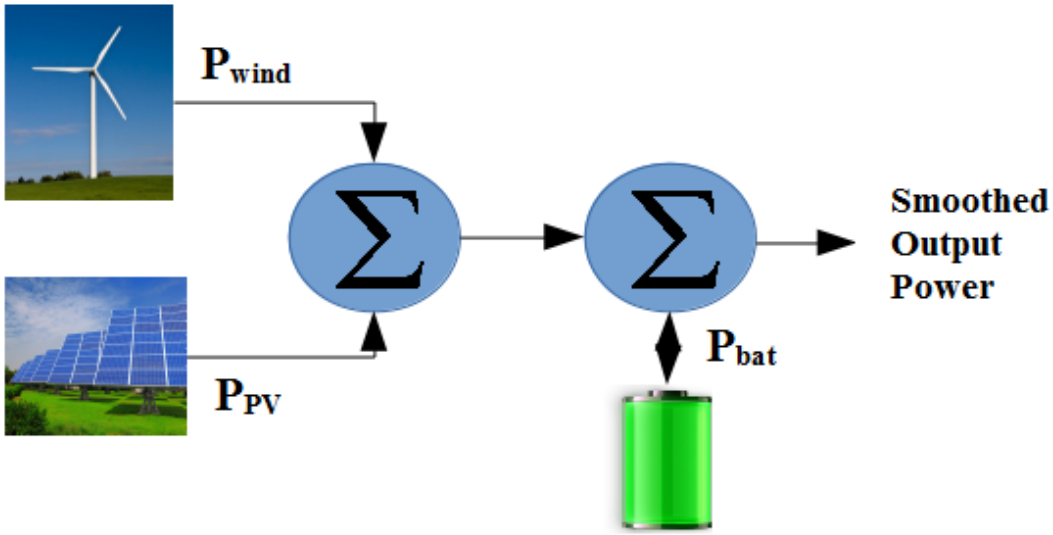


Figure 3.3 – Conceptual schematic representation of solar, wind and battery hybrid system [Add+17c].

We consider a scenario where the energy supply side of the microgrid has a combination of solar PV panels, wind turbines and battery. Figure 3.3 shows a conceptual schema of solar, wind and battery hybrid system. Based on the figure, we first aggregate the solar and wind power productions. Then, we apply smoothing algorithms to smooth out power fluctuations. After that, to make-up the difference between the actual power and the smoothed one, we charge or discharge the battery accordingly. The final smoothed power $P_{sm}(\cdot)$ is expressed as:

$$P_{sm}(t) = \begin{cases} P_{pv}(t) + P_{wind}(t) + P^{d}_{bat}(t) & \text{discharge battery (release power)} \\ P_{pv}(t) + P_{wind}(t) - P^{c}_{bat}(t) & \text{charge battery (absorb power),} \end{cases} \tag{3.7}$$

where $P_{pv}(t)$ and $P_{wind}(t)$ are solar and wind powers, and $P^{c}_{bat}(t)$ and $P^{d}_{bat}(t)$ are charging and discharging rates at time t, respectively. $P^{c}_{bat}(\cdot)$ and $P^{d}_{bat}(\cdot)$ should respect the charging and discharging constraints described in chapter 2 (section 2.2.2.3). The battery can be then charged or discharged based on the output of

the smoothing algorithms.

3.4.1 Smoothing algorithms

In this section, we present three smoothing algorithms, namely, simple moving average, exponential moving average, and Gaussian-based algorithm. With demonstrating examples, we show how parameters of the algorithms affect the smoothing of solar and wind power generations.

3.4.1.1 Moving average-based smoothing

A moving average (rolling average or running average) is a technique to get an overall idea of the trends in a data set. The averaging technique is based on a sliding window size. Under moving average, we consider two types: simple moving average and exponential moving average.

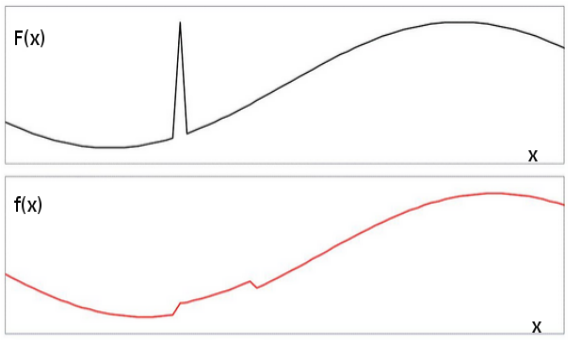
Simple Moving Average (SMA) According to Alessio et al. [Ale+02], a moving average method is a well-known low-pass filter for time series and it is defined as:

$$\tilde{y}(i) = \frac{1}{w} \sum_{k=0}^{w-1} y(i-k) \tag{3.8}$$

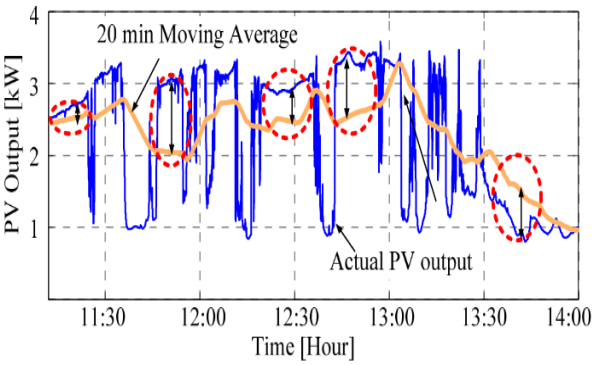
where $y(\cdot)$ is a time series data with window length w. Although it is simple and traditionally accepted way to reduce fluctuations in renewable resources [AMS14], it exhibits a memory effect which depends on the length of the averaging window. Numerically, a moving average with window length of w contains only $(1/w)\%$ of present values of a fluctuating variable. To see the effect, a demonstrating example is shown in Figure 3.4. In Figure 3.4.1, an input signal with a glitch and its smoothed output using a moving window of 10 samples are shown. The effect of the glitch lasts for the size of the averaging window.

In [AMS14], the authors also showed the memory effect of a moving average method on a PV power data obtained from National Renewable Energy Laboratory (NREL). Figure 3.4.2 shows an actual PV power and its 20 minutes moving average taken from [AMS14]. From the figure, we can see that the moving average values in the broken circles deviate significantly from the actual PV power due to memory effect of the moving average. However, there is no much fluctuations in power output during these periods. Hence, to accommodate the difference, the battery has to be charged or discharged and this forces a battery to operate frequently.

Exponential Moving Average (EMA) The moving average has an extension in the form of EMA. The EMA gives greater weights to recent changes in data, i.e., only one old value has to be remembered if it is of first order [BN02]. Hence,



3.4.1: An example to show limitation of moving average where F(x) is an original function and f(x) is a smoothed output



3.4.2: A 20-min moving average of an actual PV output [AMS14]

Figure 3.4 – Pitfalls of a moving average method.

based on Equation (3.8), a new estimation $\tilde{y}$ is calculated as follows:

$$\tilde{y}(i) = (1-\alpha)\tilde{y}(i-1) + \alpha y(i), \tag{3.9}$$

where $\alpha \in [0,1]$ is an exponential weight. The choice of α has two effects: with a large value the estimation follows the measurement truly, but does not suppress peaks, whereas with a small value peaks are suppressed but the estimation follows real changes too slowly.

As noted above, both SMA and EMA have the memory effects. To lessen these effects and to have more smoother power production curves of solar and wind, we rely on a Gaussian-base method as proposed in our work [Add+17d] and we detail it in the next section.

3.4.1.2 Gaussian-based smoothing

The Gaussian filter has been a de-facto standard for applications such as image processing and computer vision. In one dimension, a first order zero-mean Gaussian filter is given as [Wei02]:

$$G(y,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-y^2}{2\sigma^2}}, \tag{3.10}$$

where the parameter σ is the standard deviation of the Gaussian distribution and y is a one dimensional input variable. Typically σ is used as a parameter for smoothing. Hence, we change the values of σ to obtain different results for smoothing purpose.

In [RLK06], Gaussian filters are used to effectively suppress impulse noises in image processing. In our context, the impulse noises are glitches of wind and solar power productions. Suppressing the peaks and troughs could result in smooth

power generation from the renewable resources and helps to seamlessly integrate them into a power system. But, how do we measure the smoothness of power production curves? The answer is given in the next section.

3.4.2 Measure of smoothness

According to Adams and Van Deventer [AVD94], an expression for common mathematical definition of smoothness is written as:

$$Z = \int_0^T [f''(s)]^2 \mathtt{d}s, \tag{3.11}$$

for function $f(t)$, where $t \in [0, ..., T]$. Equation (3.11) is described as an integral of squared second-order differential of function $f(\cdot)$. The minimum value of Z corresponds to the maximum smoothness level [AVD94]. Hence, this parameter can be used to compare the performance of the three smoothing algorithms (SMA, EMA, and Gaussian-based). In simulation results section (section 3.5), we will use Equation (3.11) to show which algorithm is better in terms of smoothness measure.

3.4.3 Constraint on successive power levels

For big solar plants and wind farms, it is necessary that the power generation differences from one time to another is kept as minimum as possible. For example, in La Réunion island (overseas region of France), the grid operator asks energy providers to provide power with a bounded difference in consecutive time slots. One way to define this constraint is to take the derivative of the generated power $P(\cdot)$ as follows:

$$Diff(t) = \left| \frac{\Delta P(t)}{\Delta t} \right| \leqslant \gamma, \tag{3.12}$$

where γ is desired power level difference at time t $\in [1, 2, ..., T]$. By changing the value of γ, we can obtain different battery sizes for smoothing power productions which we deal with it in the next section.

3.4.4 Determining battery size

To determine the battery capacity B (in kWh) needed to absorb fluctuations in power generations of RESs, we consider charging and discharging cases and then we take the maximum of the two cases.

3.4.4.1 Charging capacity

First, we consider a charging case for storing energy in the battery. When actual power of solar and wind $P_{act}(\cdot)$ is greater than smoothed power $P_{sm}(\cdot)$, the

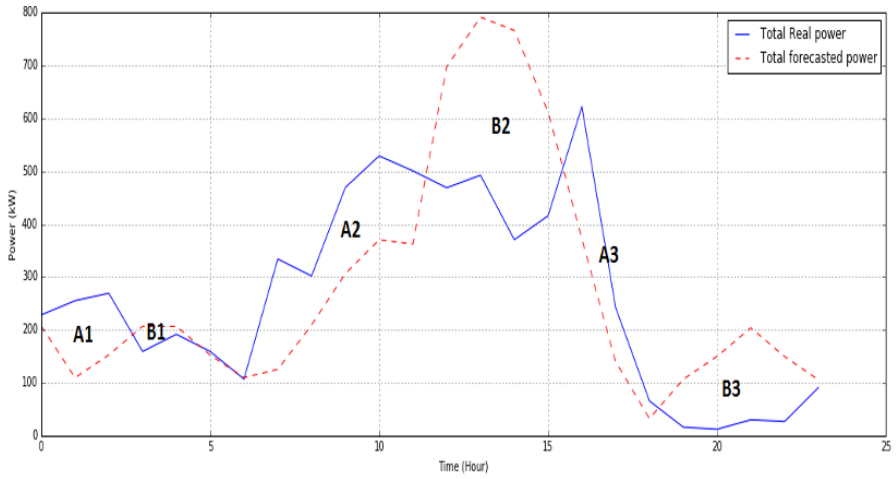


Figure 3.5 – An illustrating example for battery size computation where the As and Bs represent charging and discharging cases, respectively.

capacity is calculated as follows:

$$E_{charge} = \int_1^T \Big[P_{act}(t) - P_{sm}(t)\Big]\mathtt{d}t, \tag{3.13}$$

where $t \in [1, 2, ..., T]$. In Figure 3.5, Equation 3.13 refers to the aggregated areas of the As (i.e., $A1 + A2 + A3$).

3.4.4.2 Discharging capacity

Secondly, the discharge capacity is calculated as follows:

$$E_{dis} = \int_1^T \Big[P_{sm}(t) - P_{act}(t)\Big]\mathtt{d}t, \tag{3.14}$$

provided that smoothed power $P_{sm}(t)$ is greater than actual power $P_{act}(t)$ at time $t \in [1, 2, ..., T]$. Equation 3.14 refers to the aggregated areas of the Bs (i.e., $B1 + B2 + B3$) in Figure 3.5.

3.4.4.3 Final battery capacity

Finally, the rated capacity B of a battery is given as:

$$B = \mathtt{max}\Big(E_{charge}\eta_c, E_{dis}/\eta_d\Big), \tag{3.15}$$

which is the maximum of Equation (3.13) and (3.14) after incorporating η_c and η_d as charging and discharging efficiencies, respectively. Equation (3.15), which is similar to an equation defined in [CGW12], determines the size of a battery considering charging and discharging cycles separately. This results in a bigger

battery size. In reality, the battery alternately charges and discharges multiple times during the considered period which could result in a smaller battery size.

Besides the above case, if we take the difference between the charging and discharging capacities as:

$$B = \left| E_{charge}\eta_c - E_{dis}/\eta_d \right|, \tag{3.16}$$

then, the resulted battery size is smaller than the size determined by Equation (3.15). However, if the charging and discharging capacities are equivalent, we arrive at very small battery size which cannot be sufficient to smooth the power productions. Hence, a compromise between the two methods of battery size determination is important. For instance, a compromise between the two approaches is to look for consecutive charging and discharging cycles to determine battery size [Add+17c]. For this, we setup a recursive function that sums up the consecutive charging/discharging cycles. Then, the battery size is determined as the maximal value of the charging and discharging cases. Mathematically, a recursive function for a charging case is expressed as follows:

$$E_c(t+1) = \begin{cases} E_c(t) + P_{act}(t+1) - P_{sm}(t+1) & \text{if } P_{act}(t) \geqslant P_{sm}(t), \forall t > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Then, the charging capacity is $E_{charge} = max(E_c)$. By the same manner, a function for a discharging case is given below:

$$E_d(t+1) = \begin{cases} E_d(t) + P_{sm}(t+1) - P_{act}(t+1) & \text{if } P_{sm}(t) \geqslant P_{act}(t), \forall t > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

Then, the discharging capacity is $E_{dis} = max(E_d)$. Finally, battery size is determined by taking the maximum of the charging and discharging cases, i.e., $B = \max(E_{charge}\eta_c, E_{dis}/\eta_d)$ which is the same as Equation (3.15).

To make the calculations, we refer to Figure 3.5 as an illustrating example. In the figure, a sample solar and wind power production and their predictions are given (see section 3.5.3 for more description of the data). We consider a lithium-ion battery with η_c and η_d set to 85% which is a common value for lithium-ion batteries according to [CGW12]. Now, using Equation (3.17) for the charging case, we find that $A2$ (646kWh) is the maximum of the As. Then, for the discharging case, $B2$ (1316kWh) is found to be the maximum of the Bs. This leads to battery size of 1316kWh. However, if we compute the battery size using Equation (3.15), we arrive at a size of 2028kWh which is 712kWh bigger than $B2$. Hence, we attained a 35% reduction in battery size by applying our approach. Moreover, if we use Equation (3.16) to compute the battery size, this will result

in a capacity of 670kWh. Apparently, this value is not sufficient for a discharging case at $B2$ which is 1316kWh. The limitations to our approach could be for the cases where discharging cycles are small and charging cycles are big. In these cases, if we consider consecutive charging/discharging cycles to determine the battery sizes, the energy to be stored will be bigger than the determined battery size. The approach in Equation (3.15) that considers charging and discharging cycles separately could be better for these cases.

In this section, we discussed the smoothing algorithms and outlined how to computer battery sizes. The following section provides numerical results on the two parts of this chapter: modeling of DERs and smoothing.

3.5 Simulation results

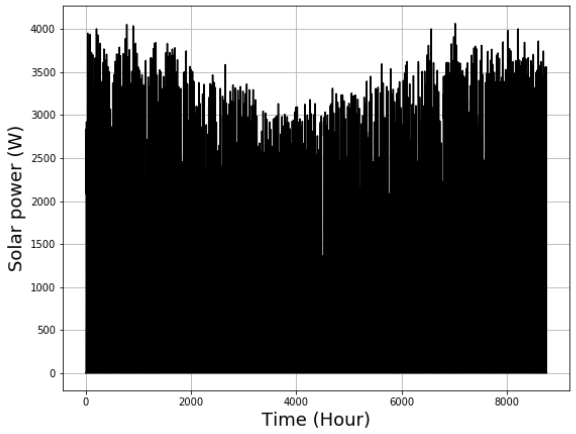
In this section, we provide descriptions of our datasets and discussions on results of different simulation results on both DERs modeling and smoothing parts which are discussed in the above sections.

3.5.1 Description of datasets

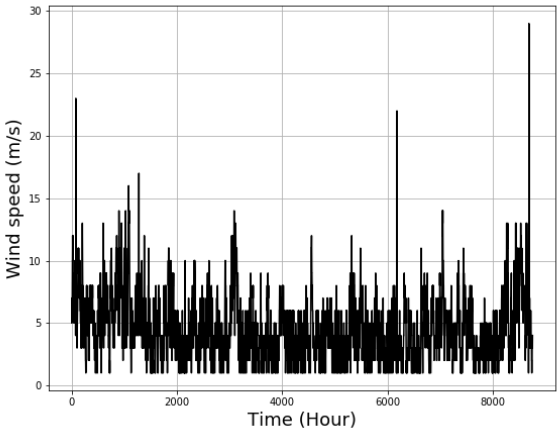
For our experiments, we consider a real datasets of solar and wind power data for an industrial site located in Bourbourg, France. The datasets consist of solar, wind, energy demand of the factory, EPEX spot market prices, and battery parameters.

3.5.1.1 Solar PV data

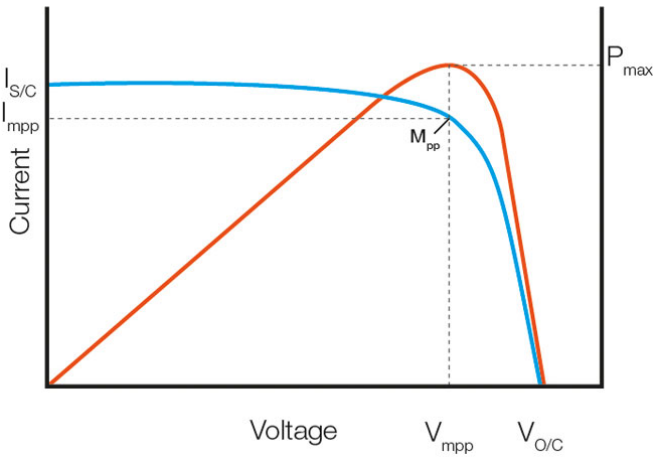
Based on the PVWatts of NREL [PW], we retrieved hourly per unit ($25m^2$) solar PV data as shown in Figure 3.6.1. The PVWatts calculator uses Equation (2.2) (see section 2.2.2.2 in the preceding chapter) as a reference and it also incorporates temperature effect to the equation. For our experiments, We take power production of 100 units of PV panels with the power curve shown in Figure 3.6.3. For forecasting purpose, we use **A**uto **R**egressive **I**ntegrated **M**oving **A**verage (ARIMA) (in the book of Wei [Wei94], chapter 4) model to predict day-ahead solar power production. The ARIMA model has been widely used in time series forecasting applications such as stock [PL05] and next-day electricity prices forecasts [Con+03]. In our simulations, the ARIMA model is trained with solar data of 1 hour interval for 245 days. Then, the prediction is performed for the 246^{th} day. Forecast and real solar powers are depicted in Figure 3.7.1.



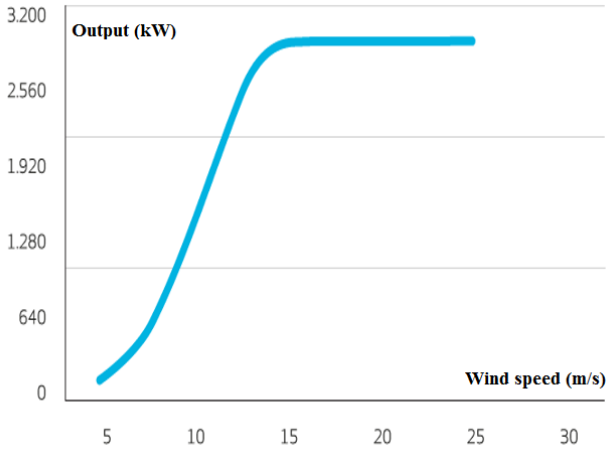
3.6.1: Hourly average solar power data



3.6.2: Hourly average wind speed data



3.6.3: Solar PV power curve

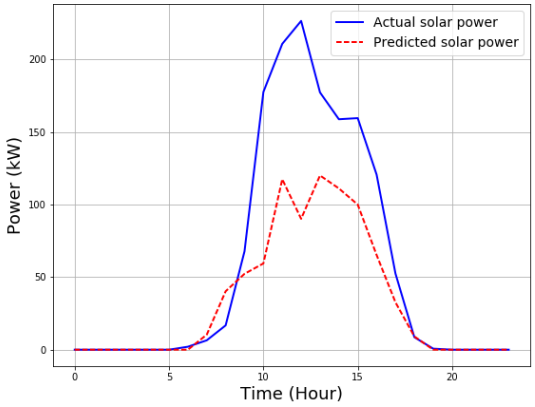


3.6.4: Wind turbine power curve

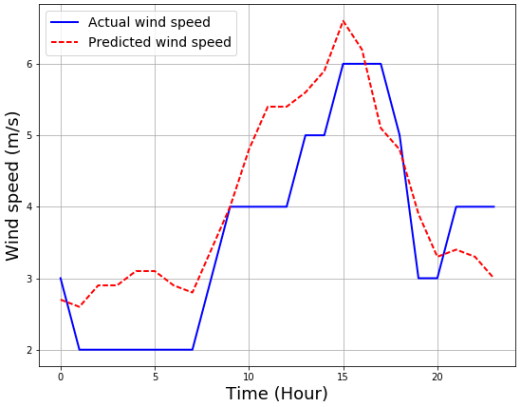
Figure 3.6 – Solar and wind input data for year 2014 and their power curves.

3.5.1.2 Wind speed data

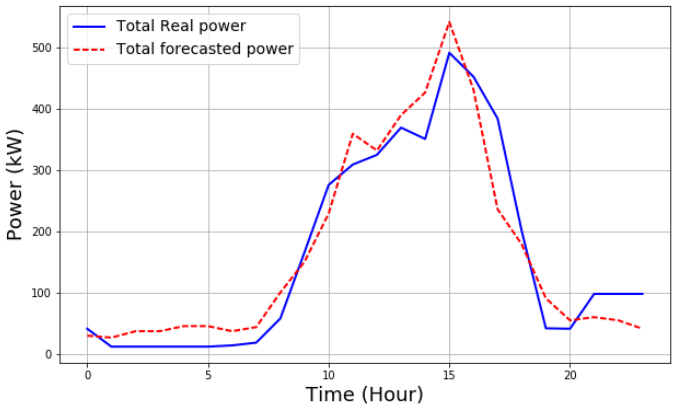
We obtained hourly average wind speed data for the year 2014 from *Weather underground* website [WU]. The hourly wind speed data is shown in Figure 3.6.2. The forecasting process is also performed following the same procedure as that of the solar data. Forecast and real wind speeds are shown in Figure 3.7.2. Using Equation (2.1) in section 2.2.2.1 of the previous chapter, we computed the power extracted from the wind speed for a 3MW wind turbine which has the power curve depicted in Figure 3.6.4. In Figure 3.7.3, the aggregation of actual and predicted solar and wind powers are provided.



3.7.1: Predicted solar power



3.7.2: Predicted wind speed



3.7.3: Actual and predicted total power

Figure 3.7 – Actual and predicted solar and wind data for a period of 24 hours.

3.5.1.3 Battery parameters

A battery can be characterized by its maximum battery capacity B, DoD, charging and discharging rates, efficiency and other parameters. Table 3.1 shows the characteristics of Lead-acid and Lithium-ion batteries. Descriptions of the parameters are provided in section 2.2.2.3 of chapter 2.

3.5.1.4 Energy demand data

For energy demand (load) data, we obtained hourly energy consumption data from METRONLab servers (see Figure 3.8). These data represent a yearly energy demand of the industrial site.

Table 3.1 – Characteristics of two common battery types [Che+09; GFKR15; WLD15; MW15].

Characteristics	Lead-acid	Lithium-ion
Efficiency (%)	75	85
Charge time	8-16h	2-4h
Discharge to charge rate ratio	10	5
Self-discharge per day (%)	0.3	0.1
Depth of Discharge (%)	80	80
Energy density (Wh/L)	80	150
Power density (W/L)	125	450
Lifetime cycle	2000	5000
Expected lifetime (Years)	4	10
Cost ($/kWh)	100 - 200	400 - 600

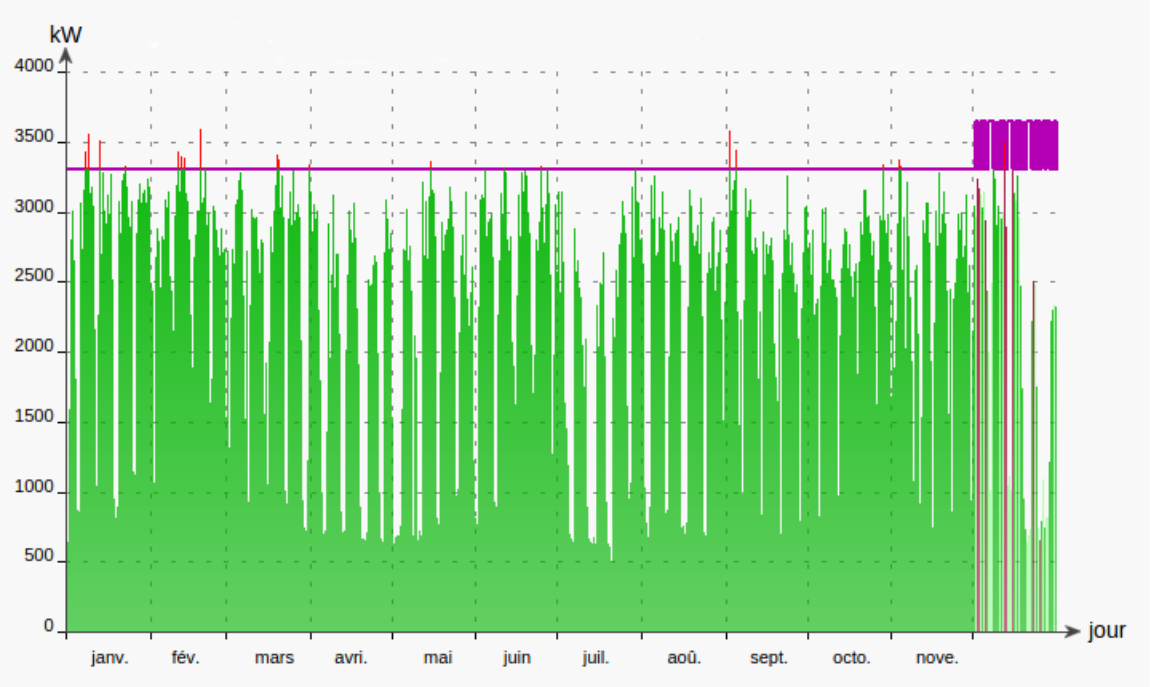


Figure 3.8 – Hourly energy demand of a factory from METRONLab server.

3.5.1.5 Spot market price data

We retrieved hourly market price from EPEX Spot website [EPE17]. These prices were published for the day-ahead spot market in 2014 and they are shown in Figure 3.9. Based on these data, we detail results and discussions on modeling of DERs in the next section.

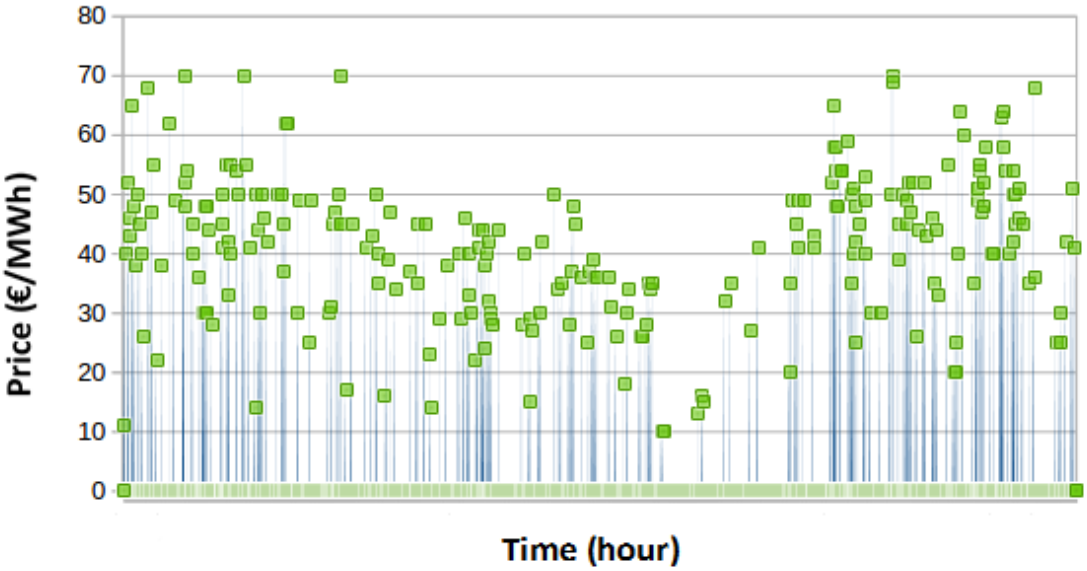


Figure 3.9 – EPEX Day-ahead spot price data for 2014.

3.5.2 Results and discussions on modeling of DERs

In this section, we provide different results on the performance of the cost minimization strategies, namely, *sell surplus energy* (Strategy 1), *store excess energy* (Strategy 2), and *use external energy to charge battery* (Strategy 3). These strategies are described in section 3.3.3 alongside their implementation in Algorithm 1 which relied on the service curve models of DERs. As a performance metric, we refer to Equation (3.6) that computes the net energy procurement cost and we compare the strategies based on the equation. As initial simulation conditions, we set a price of 40€/MWh to buy electricity from a utility grid and batteries can be considered full in beginning of the simulations.

Table 3.2 – Cost comparison of different strategies (taking 40€/MWh for cost of utility grid energy, battery size of 20MWh for the three strategies, and spot market price limit of 5€/MWh for strategy 3).

Strategies	**Net energy procurement cost (in €)**
Grid-only	621,915
Spot-only	586,301
Strategy 1	400,845
Strategy 2	405,815
Strategy 3	396,387

3.5.2.1 Performance of the strategies

First, we compare the strategies against the case of no strategies, i.e., we buy all the energy demand from the utility grid (*Grid-only*) or on the spot market (*Spot-only*). Table 3.2 shows the results of the annual net procurement costs (defined by Equation 3.6) that are calculated by using the strategies with fixed battery size of 20MWh, sport market price limit of 5€/MWh, and utility grid price of 40€/MWh. From the table, we can see that the actual annual cost of the factory is 621,915€ if it buys all the energy from the utility grid. However, if the factory buys all its demands from the sport market, then the cost decreases to 586,301€ per annum. With the specified parameter settings, the three strategies can be comparable with respect to the net energy cost. However, when compared to the *Grid-only* and *Spot-only* strategies, using *Strategy 3* (since it is the minimum of the three) could save upto 36% (from 621,915 down to 396,387€) and 33% (from 586,301 down to 396,387€), respectively. Next, we investigate the effect of battery sizes on the performance of the strategies.

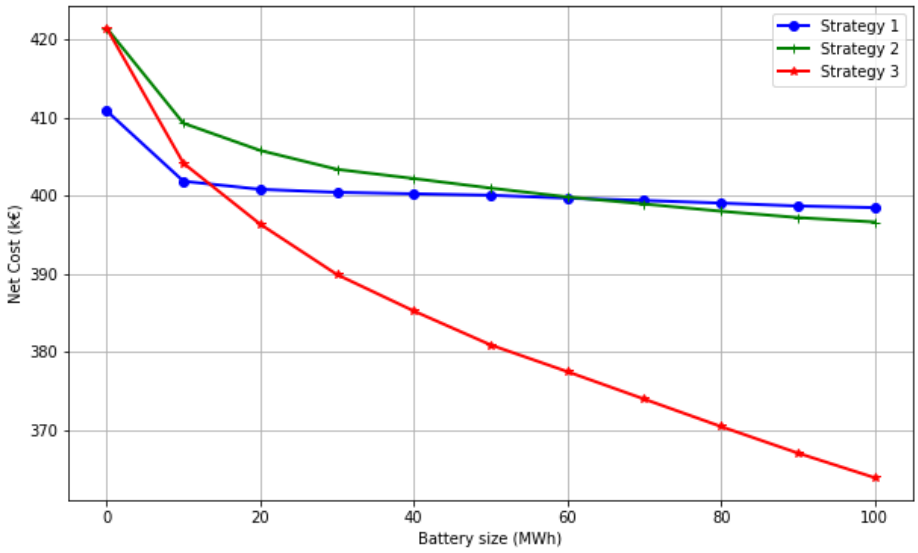


Figure 3.10 – The three strategies with different battery sizes and fixed spot price limit of 5€/MWh for strategy 3.

3.5.2.2 Effect of battery sizes

Now, by varying the battery sizes, we compare the performance of the strategies against each other using the performance metric defined in Equation (3.6). The results of the three strategies are depicted in Figure 3.10. From the figure, we can see that for smaller battery sizes ($B \leqslant 10$MWh), *Strategy 1* is the best among

the three strategies because its net cost is less than the other two. This shows that it is better to sell energy instead of storing it when the battery size is less than 10MWh. However, as the battery size increases ($B > 10$MWh), *Strategy 3* outperforms the others. Hence, for bigger battery sizes, a good strategy is to buy energy from the spot market and then store it. The stored energy could be used later when the prices are high or load-shedding is required.

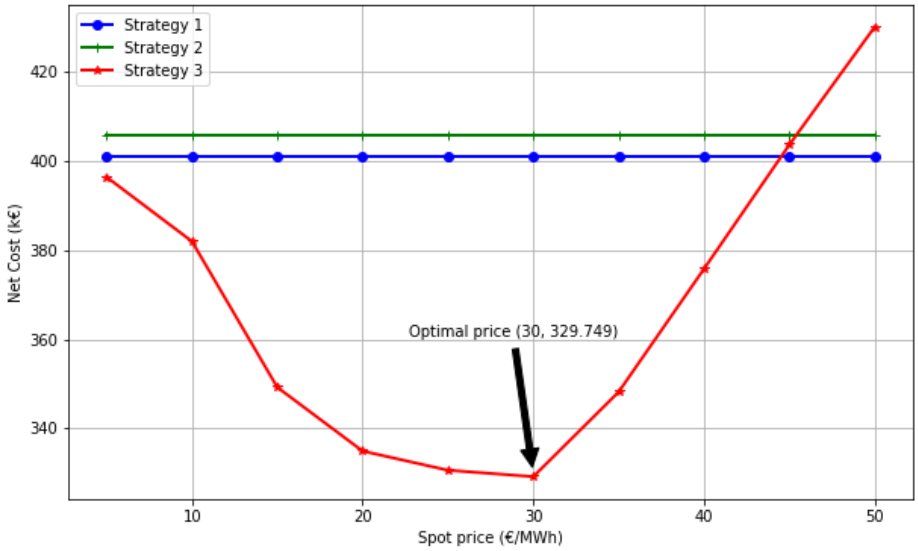


Figure 3.11 – Effect of varying spot price with fixed battery capacity of 20MWh.

3.5.2.3 Effect of spot market prices

Figure 3.11 shows the effect of different spot market prices and a fixed battery size of 20MWh. The variable spot market prices have likely effect on *Strategy 3* than the others because this strategy buys energy on spot markets to charge the battery during lower price periods. For *Strategy 3*, an optimal point is found at [30€/MWh, 329,749€] which corresponds to a saving of 47%. The other strategies remain unaltered under variable spot market prices.

In Figure 3.11, we considered a fixed battery size to determine the effect of variable sport market prices. To see how different battery sizes affect the performance of *Strategy 3*, we use three battery sizes: 10, 20, and 30MWh. The resulting curves are shown in Figure 3.12. From the figure, we can see that the optimal point of the three curves are obtained when the spot market price is 30€/MWh. However, the cost saving has inverse relationship with the battery sizes: the smaller the battery size the less cost saved. With bigger battery sizes, the cost of having these storage devices could hinder cost savings and it may result in longer payback periods.

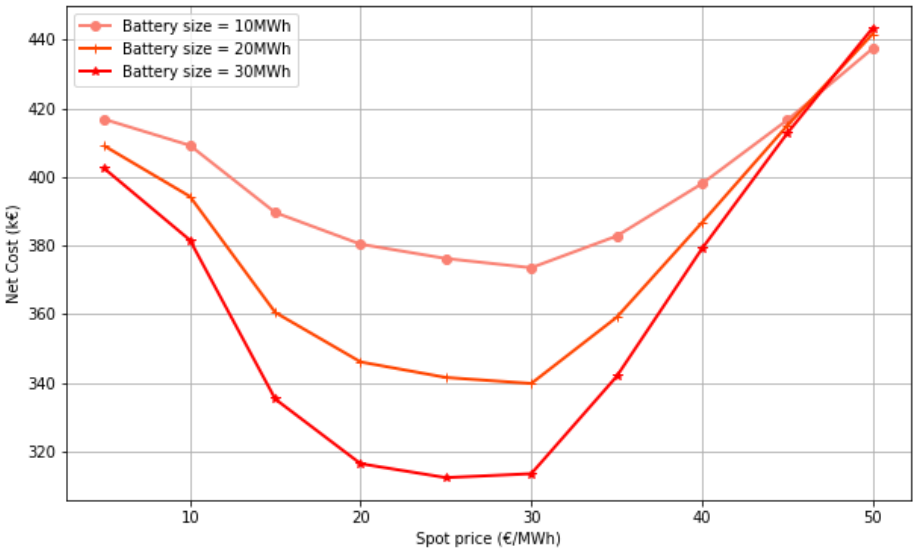


Figure 3.12 – Performance of strategy 3 with spot prices in the range [0,50]€/MWh and battery size in the range [10,30]MWh.

3.5.2.4 Payback period estimation

One of the major hindering factors of microgrid development is the cost of microgrid components such as batteries, wind turbines, solar PV panels, AC/DC (Alternating Current/Direct Current) converters, etc [Bur+13]. To perform the payback period estimation, we consider the costs of the microgrid components (solar, wind, and battery) used in our experiments. Nowadays, the cost of a Lithium-ion battery ranges from $400 to 600 per kWh and that of Lead-acid ranges from $100 to 200 per kWh (see Table 3.1). For solar PV panels, a PV panel that produces 1W cost $4.9 according to [Fel+12]. Concerning the cost of wind turbines, Bolinger and Wiser [BW11] did a study on trends of wind turbine prices for the past decade (from 2000 to 2010). According to the authors, the trend for wind turbine prices declined on an average of 20% from 2002 to 2010. They also pointed out that the price for a wind turbine ranges from $900 to 1,400 per kW and the average cost is $1,100/kW.

We use the following equation to calculate payback periods:

$$Payback = \frac{Cost_{battery} + Cost_{PV} + Cost_{wind}}{Cost_{Grid-only} - \texttt{min}Cost_{stg_i}} \quad \texttt{(in years)}, \tag{3.19}$$

where $srg_i \in [Strategy\ 1, Strategy\ 2, Strategy\ 3]$ and $Cost_{stg_i}$ is computed using Equation 3.6 which is the net energy cost metric.

Taking into account the above device costs, we now calculate payback periods by setting costs of PV panels to $4.9/W, the average cost of $1,100/kW for the

Table 3.3 – Simulation results for payback period estimation by taking 100 PV units, a 3MW wind turbine, and varying sizes of two battery types with costs of \$400/kWh and \$100/kWh for Lithium-ion and Lead-acid batteries, respectively.

Battery		**Solar**	**Wind**	**Payback (in years)**
Type	Size (in MWh)			
Lithium-ion	1	✓	✓	∼23
Lithium-ion	5	✓	✓	∼27
Lithium-ion	10	✓	✓	∼33
Lead-acid	1	✓	✓	∼22
Lead-acid	5	✓	✓	∼22
Lead-acid	10	✓	✓	∼22
Lithium-ion	1	✗	✓	∼15
Lithium-ion	5	✗	✓	∼20
Lithium-ion	10	✗	✓	∼26
Lead-acid	1	✗	✓	∼14
Lead-acid	5	✗	✓	∼14
Lead-acid	10	✗	✓	∼15

wind turbine, \$400/kWh for Lithium-ion, and \$100/kWh for Lead-acid battery. Table 3.3 shows the estimation of payback periods with different battery sizes (for both Lithium-ion or Lead-acid), with/without solar PV panels, and 1 wind turbine. Since the wind turbine generates more power than solar power, we always assume that we have the wind power in the energy mix. Hence, cost of PV panels is \$1.96m (or approx. 1.76m€ at the time of thesis writing) and cost of a 3MW wind turbine is \$3.3m (or approx. 2.95m€). Since the cost of big battery sizes are very high, we use smaller battery sizes ($B \leqslant 10$MWh). The results shown in Table 3.3 use Equation (3.19) to compute the payback periods. When the battery size is 1MWh, *Strategy 1* results in less net cost (408,432€) compared to the other two (refer to Figure 3.10). However, for battery sizes of 5 and 10MWh, *Strategy 3* (taking the optimal market price limit) performs better than the others.

From Table 3.3, we can infer a couple of important points. The first point is that the inclusion of solar PV panels into the energy mix increases the payback periods. For example, considering a 1MWh Lithium-ion battery, the payback period increased from 15 years (without taking account PV panels) to 23 years when there are PV panels. This is because the area where the factory is located has small amount of solar radiation per annum due to its geographical location. Hence, it is better to use a hybrid system of wind and battery in this location. Another important point is that the the payback period depends on both battery type and size. Since the cost of Lead-acid batteries are less than that of Lithium-ion, the payback periods are also less for the Lead-acid batteries. However,

Lead-acid batteries have low energy and power densities and shorter expected lifetime (see Table 3.1). The following sections discusses the second part of this chapter, namely, smoothing of solar and wind power generations.

3.5.3 Results and discussions on smoothing

In this section, we first compare the performance of the smoothing algorithms, viz., **S**imple **M**oving **A**verage (SMA), **E**xponential **M**oving **A**verage (EMA), and Gaussian-based which are presented in section 3.4.1. To measure their performances, we vary their respective smoothing parameters. Then, we determine minimum battery sizes considering two cases: with and without successive power difference constraints. Finally, we determine a power production curve for a day-ahead forecast period based on the real datasets described in section 3.5.1.

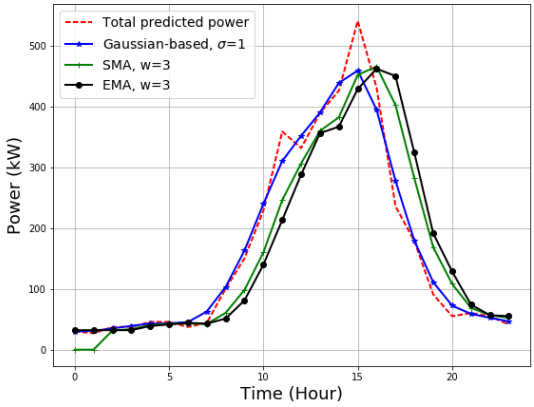
3.5.3.1 Performance of the smoothing algorithms

As for the smoothing parameters, we use the standard deviation (σ) for Gaussian-based and window length (w) for moving average based smoothing algorithms. Hence, we compare the algorithms by changing smoothing parameters and see the effects of increasing or decreasing these parameters. In the following experimentation, we consider the two case: with and without power level constraints.

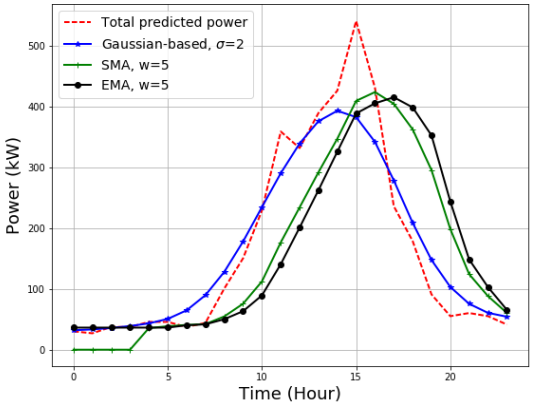
Case 1 - no constraint To start our comparisons, we set σ to 1 for Gaussian-based and w to 3 for moving average methods without imposing constraints on consecutive power levels. Figure 3.13.1 shows smoothing performance of the three algorithms (SMA, EMA, and Gaussian-based) when the smoothing parameters σ and w are 1 and 3, respectively. In the figure, the Gaussian-based smoothed power closely follows the predicted total power of solar and wind. This shows that the smaller the difference between the smoothed and unsmoothed powers, the lesser available power to charge or discharge a battery at any given time. This leads to smaller battery sizes for smoothing.

Using Equation (3.11), we compute the smoothness measures and the results are shown in Table 3.4. From the table, we can see that Gaussian-based algorithm has lower charge/discharge rates than both EMA and SMA. The effect of lower charge/discharge rates is that the smoothness measures will have lower values as the minimum value corresponds to the smoothest curve (see section 3.4.2). In this aspect, the Gaussian-based method performs better than the other two algorithms. Numerically, Gaussian-based is 34% more smoother than EMA and 47% more smoother than EMA.

To investigate the effects of increased smoothing parameters, we refer to Figure 3.13.2 where σ and w set to 2 and 5, respectively. From the figure, we can see that the Gaussian-based approach is more smoother but the difference



3.13.1: Smoothing when σ is 1 and w is 3



3.13.2: Smoothing when σ is 2 and w is 5

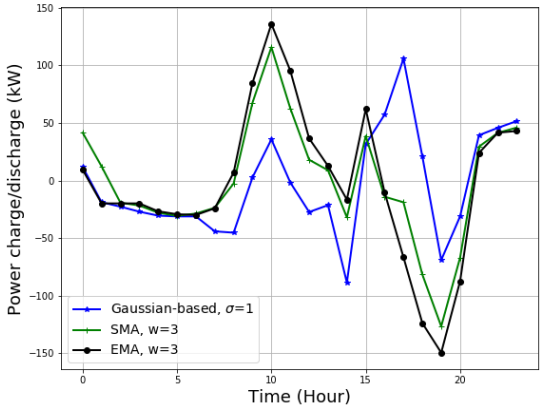
Figure 3.13 – Smoothing performance of SMA, EMA, and Gaussian-based algorithms with different smoothing parameter values.

Table 3.4 – Maximum charge/discharge rates and smoothness measure of the algorithms (when σ is 1 and 2 and w is 3 and 5).

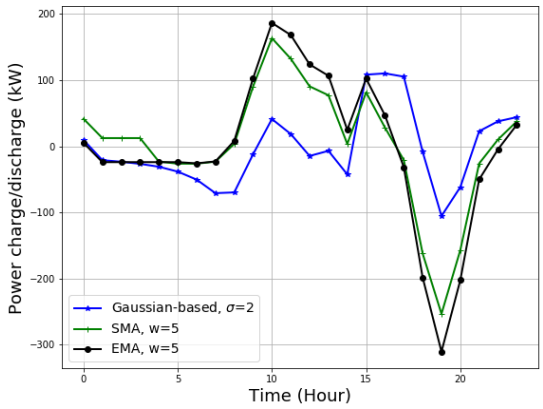
Models	Parameters	Charge rate (kW)	Discharge rate (kW)	Smoothness measure Z
Gaussian-based	$\sigma = 1$	106	89	15980
	$\sigma = 2$	110	105	4217
SMA	$w = 3$	116	127	24189
	$w = 5$	254	163	11163
EMA	$w = 3$	135	149	30204
	$w = 5$	186	310	11444

between the predicted and smoothed powers is getting bigger. The same is true for SMA and EMA. Their curves are shifted right by the size of the averaging window. These bring up a dilemma between increase of smoothness and frequent charging/discharging. When the smoothing parameters are smaller, the result will have less smoothed feature and higher frequency of charging/discharging. On the other hand, as the parameters increase, it is more smooth but the deviation from real values is bigger and this could result in bigger battery sizes. Hence, the choice of the parameters need to align with required applications.

To determine battery sizes, we take the difference between the actual solar and wind powers and the predicted (and smoothed) ones. The positive difference shows a charging case and the negative shows otherwise. Figure 3.14 depicts power charge and discharge rates by varying the smoothing parameters of the three smoothing algorithms. Moreover, maximum charging and discharging rates are also shown in Table 3.5. From Figure 3.14.1 and 3.14.2, we can deduce that when the smoothing parameters increase, the positive and negative amplitudes decrease which give rise to smaller battery sizes. However, the increase in the



3.14.1: Charging and discharging rates when σ is 1 and w is 3



3.14.2: Charging and discharging rates when σ is 2 and w is 5

Figure 3.14 – Charging and discharging rates where positive values represent charging and negative values represent discharging.

Table 3.5 – Battery sizes for different smoothing parameters of the algorithms where $\eta_c = \eta_d = 85\%$.

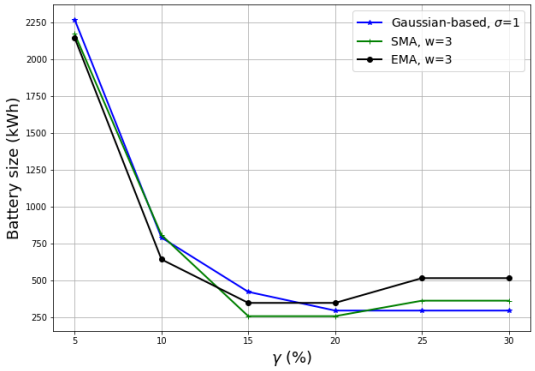
Capacity (kWh)	**Gaussian-based**		**SMA**		**EMA**	
	σ=1	σ=2	w=3	w=5	w=3	w=5
$E_c = E_{charge}\eta_c$	184	276	232	567	316	734
$E_d = E_{dis}/\eta_d$	294	404	361	729	514	938
$B = \texttt{max}(E_c, E_d)$	294	404	361	729	514	938

amplitudes of charging and discharging will result in bigger battery sizes.

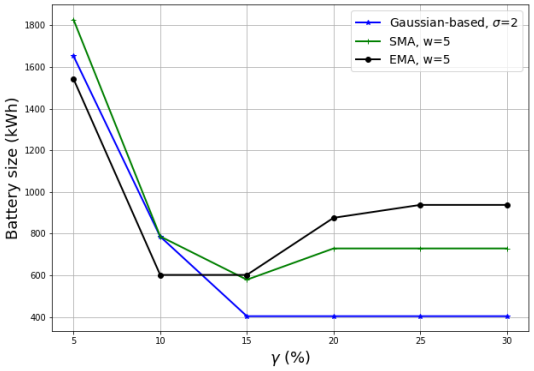
Based on the charge/discharge rates, we then compute battery sizes by applying Equation (3.17) and (3.18) and taking the maximum of the two values. Table 3.5 shows the results of battery size computations when charge and discharge efficiencies are set to 85%. From the table, we can infer that for smaller smoothing parameters, i.e., when σ is 1 and w is 3, the Gaussian-based method achieves 19% (from 361 to 294kWh) smaller battery size than SMA and 43% (from 514 to 294kWh) smaller than EMA. The table also shows the effect of increased smoothing parameters. In Gaussian-based method, the battery size is increased from 294kWh to 404kWh when σ increased from 1 to 2. Moreover, for SMA and EMA, when window w increased from 3 to 5, the battery sizes are also increased from 361 to 729kWh in case of SMA and from 514 to 938kWh in case of EMA. Next, we introduce constraints on power levels between consecutive time slots.

Table 3.6 – Smoothness measures of the algorithms with successive power level constraints of γ% (resulted from Equation (3.12)).

γ (%)	Gaussian-based		SMA		EMA	
	σ=1	σ=2	w=3	w=5	w=3	w=5
5	2037	1614	1686	2349	3247	1433
10	5298	6373	8368	4078	4493	4923
15	8261	4217	13961	15583	20183	7666
20	15037	4217	11674	11163	20183	17266
25	15980	4217	24189	11163	30204	11444
30	15980	4217	24189	11163	30204	11444



3.15.1: Battery sizes when σ is 1 and w is 3

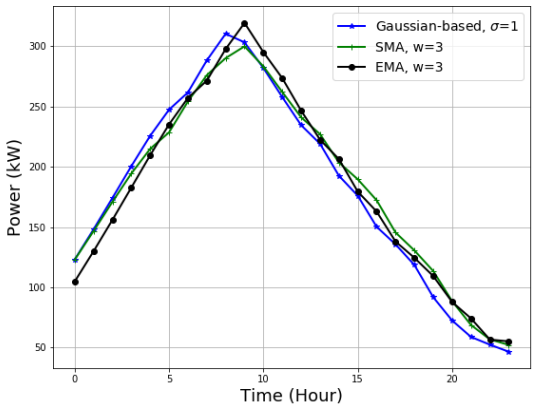


3.15.2: Battery sizes when σ is 2 and w is 5

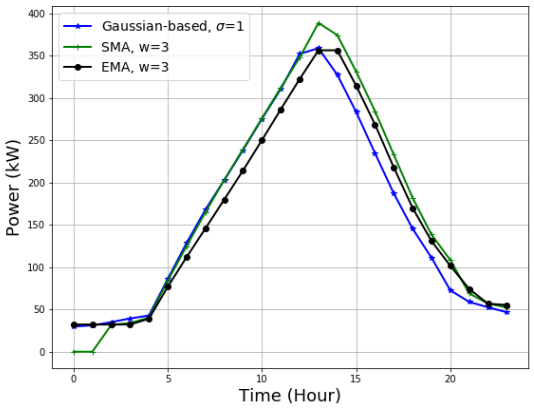
Figure 3.15 – Battery sizing based on successive power level constraints with $\gamma \in [0.05,0.30]$.

Case 2 - power level constraint The above results are obtained without considering power level constraints between two successive time slots. In this case, we now impose the constraints on the power differences and evaluate the performance of the smoothing algorithms in terms of smoothness measure and battery sizes. We use Equation (3.12) to upper bound the successive power differences to some fixed value γ (in %). The value of γ could range from 5 to 30% of the peak power which is found to be 541kW from the datasets. To compute the power differences in power levels in this case, we first smooth the unsmoothed solar and wind data and then apply an averaging mechanism when the difference exceeds P_{diff} = (γ*541kW)/100. We apply recursively the averaging mechanism until the power difference is less or equal to P_{diff}.

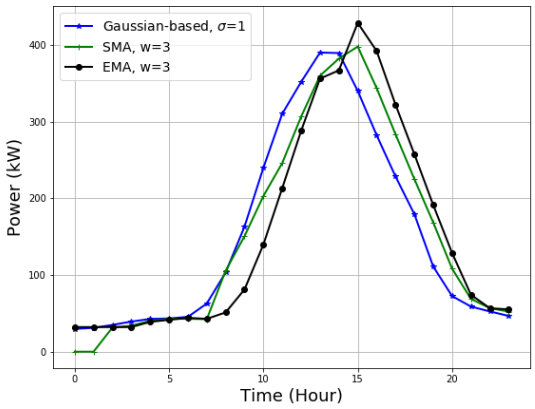
Table 3.6 shows the results of smoothness measures (denoted as Z in Equation (3.11)) of the three algorithms with different smoothing parameters and power level constraints. From the table, we can see that for smaller values of γ (i.e., more



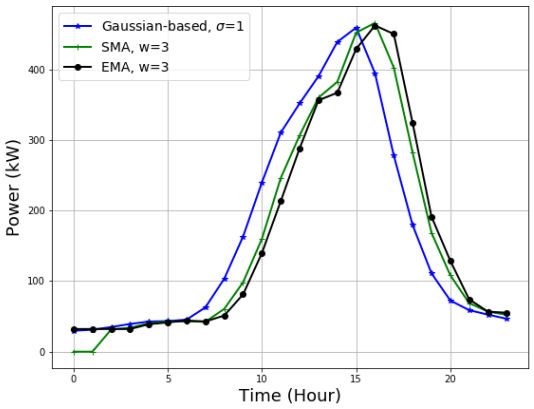
3.16.1: When $\gamma = 5\%$



3.16.2: When $\gamma = 10\%$



3.16.3: When $\gamma = 15\%$

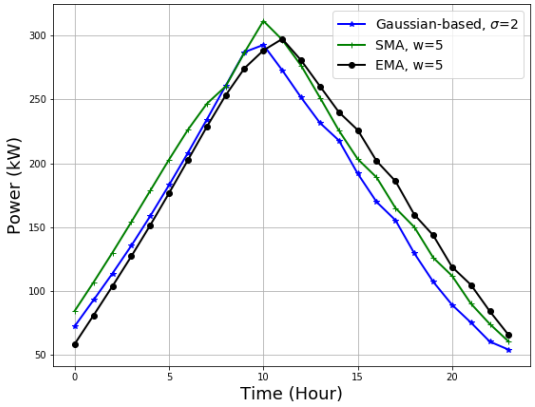


3.16.4: When $\gamma = 25\%$

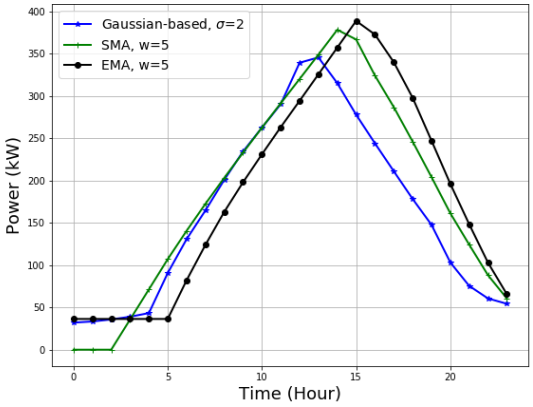
Figure 3.16 – Power curves for a day-ahead forecast considering different of values γ and when σ is 1 and w is 3.

stringent constraint), the value of Z is small when compared to bigger values of γ. This is due to the fact that when γ is small, the averaging mechanism smooths the power data so that the resulted smoothed curve has minimum difference between the successive points. However, as the γ increases, the value of Z also increases until there is no more to smooth. When γ is above 25%, the smoothness measures are the same as Table 3.5 where the constraints are not considered.

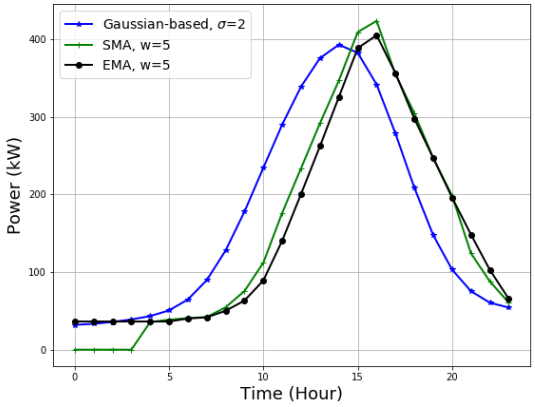
To determine battery sizes in this case, we follow the same procedure as the above case where the battery sizes are computed using Equation (3.17) and (3.18) and then we take the maximum of the two. Figure 3.15 shows the battery sizes for the values of γ ranging from 5 to 30% of the peak power and when the smoothing parameters are $\sigma \in \{1,2\}$ and $w \in \{3,5\}$. In both Figure 3.15.1 and 3.15.2, we



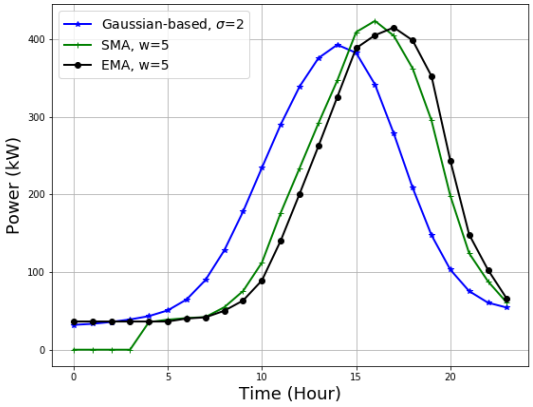
3.17.1: When $\gamma = 5\%$



3.17.2: When $\gamma = 10\%$



3.17.3: When $\gamma = 15\%$



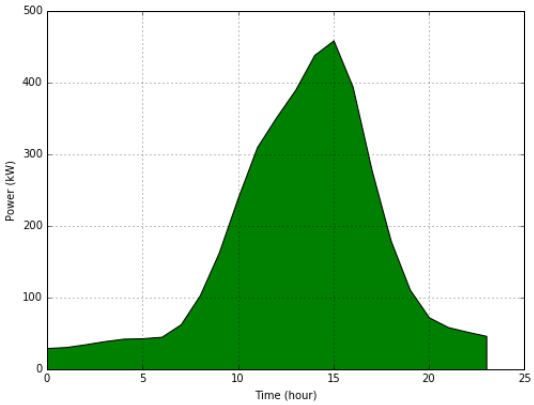
3.17.4: When $\gamma = 25\%$

Figure 3.17 – Power curves for a day-ahead forecast considering different values of γ and when σ is 2 and w is 5.

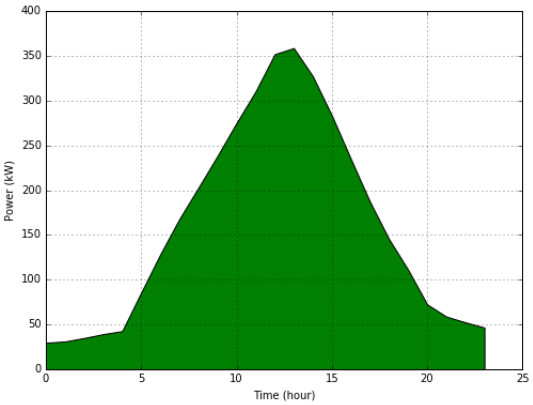
can see that for smaller values of γ (for $\gamma < 10\%$), the battery sizes are bigger. However, as the constraint is relaxed (for $\gamma \geqslant 10\%$), the battery sizes are reduced. Here also, when $\gamma \geqslant 25\%$, the battery sizes are the same as the case of without constraints.

3.5.3.2 The final power production curve

The final output of our simulations is to determine a power production curve for a day-ahead forecast period. The power production curve enables one to know precise power production of solar and wind and helps one to decide buying energy from the utility grid or the spot market if energy demand is higher than energy supply. To attain this goal, we did multiple simulations considering the three



3.18.1: Power curve when σ is 1 and battery size is 294kWh



3.18.2: Power curve when σ is 1, γ is 10%, and battery size is 834kWh

Figure 3.18 – Gaussian-based power production curves for day-ahead forecast considering the two cases.

smoothing algorithms: Gaussian-based, SMA, and EMA. In Figure 3.16 and 3.17, the power production curves are shown for the smoothing algorithms considering power level constraints and different smoothing parameters. From these figures, we can see that the peak powers for the smoothed curves are less when the constraints are strict, i.e., smaller values of γ. For example, fixing σ to 1, w to 3 and γ to 5%, the peak power is 310kW for Gaussian-based, 300kW for SMA, and 320kW for EMA. However, when γ is increased to 10%, the peak powers are also increased by 14%, 22% and 10% for Gaussian-based, SMA and EMA, respectively. This shows that by relaxing the constraints, we could have more peak power and it requires smaller battery sizes to smooth the power productions of the **R**enewable **E**nergy **S**ources (RESs).

Among the smoothing algorithms, the Gaussian-based algorithm performed better than the other two in terms of battery sizes and smoothness measures in both cases. Figure 3.18 shows the power curves for a day-ahead period considering the two cases. These curves are determined by Gaussian-based algorithm with battery size of 294kWh in no-constraint case and 836kWh in the other case. In the above simulations, we considered smoothing and battery sizing for only one day-ahead. Our approaches can be extended easily to handle multiple days.

3.6 Summary

This chapter discussed two of our works on the supply side of the industrial microgrid. In section 3.3, we adopted service curves of Network Calculus theories to model **D**istributed **E**nergy **R**esources (DERs) such as solar, wind and battery.

We relied on the service curve concepts for modeling the capacity of DERs. In this context, the service curves model the minimum amount of energy that a DER can offer. After modeling the resources, we set up three strategies for testing the benefits of our approach in terms of minimizing energy procurement costs. The strategies are *sell surplus energy (Strategy 1)*, *store excess energy (Strategy 2)*, and *use external energy to charge battery (Strategy 3)*. Based on real datasets, we compared the performance of different strategies. Our results show that we could gain an energy cost saving upto 47% which can be very interesting to large industries.

Section 3.4 detailed the Gaussian-based smoothing algorithm and compared its performances by bench-marking two classical approaches, namely, **S**imple **M**oving **A**verage (SMA) and **E**xponential **M**oving **A**verage (EMA) to smooth out power fluctuations of renewable energy sources. In our simulations, we used real datasets of energy demand, spot market prices, solar and wind for an industrial site located in France. To show the performances of the algorithms, we considered two cases: with and without power level constraints. In both cases, the Gaussian-based algorithm performed better than SMA and EMA when smoothness measure is taken to account. Furthermore, the proposed method required up to 19% less battery size than the other two for the purpose of smoothing a day-ahead forecast. Finally, we determined a power production curve for a day-ahead forecast period using the Gaussian-based algorithm as it performed better than SMA and EMA.

The above two paragraphs summed up our works on the energy supply side. In the following chapter, we move to the demand side of the industrial microgrid and we present our works on Demand Response and related concepts.

Chapter 4

Industrial Demand Side Management

Contents

4.1 Introduction

This chapter details our works in the demand side of industrial microgrid energy management. In the industrial context, the energy demand side consists of

different industrial processes to produce a variety of end products. According to [IEA16], the industrial processes are one of the major energy consumers that are accounted for 42.5% of global electricity consumption in 2015. Due to increased energy costs and environmental concerns such as elevated carbon footprint, manufacturing sectors are investigating different options to address these challenges. The potential to reduce energy costs can be gained through implementing **D**emand **S**ide **M**anagement (DSM) approaches such as **E**nergy **E**fficiency (EE) and **D**emand **R**esponse (DR). EE refers to achieving the same output with less energy consumption by adopting a more efficient technology or production processes or enforcing policies to commonly accepted methods to reduce energy losses. However, EE is not the only approach to reduce energy consumption in manufacturing sectors. There are also DR mechanisms which include peak-shaving and load shifting for moving loads from peak to off-peak hours to reduce energy cost.

DR programs throttle energy demands of different loads such as industrial, commercial and residential for adjusting demands to available productions. For example, with DR, a manufacturing plant is requested to stop energy consumption by a legacy energy provider (e.g., Électricité de France (EDF) in France). In so doing, the plant can get paid for not consuming. This helps the energy provider to smooth consumption peaks instead of buying energy at the highest price on spot markets. Hence, the more a consumer has DR potential the more stable the grid performance will be and more benefits for both parties (for detailed description on DR types and benefits, refer to section 2.3 of chapter 2). According to a study in [SAM13] on top 20 USA industries, the DR potential in those industries could reach up to 12GW in available load flexibility.

To harness the potential of DR in industrial context, a crucial step could be merging both production process modeling and scheduling which were separately tackled in the literature. For production process modeling, a substantial amount of research is devoted to analyze performance metrics such as throughput analysis and job completion time using queuing theory models. These models capture the dynamics of production processes better than simulation approaches. In the production scheduling and cost minimization aspects, the objective is to adapt production according to available power and production rate constraints. This chapter provides analytical and simulation results for both problems beginning with a literature review in section 4.2. Then, in section 4.3, we present our works on modeling of a production line using models from queuing theory which were discussed in section 2.5 of chapter 2. We use the queuing model to characterize temporal evolution of a production line. After that, section 4.4 details a DR scheduling algorithm that adapts power consumption of a production line to available power and production rate constraints. In section 4.6, we provide our experiments with **Open A**utomated **D**emand **R**esponse (OpenADR) protocol. Finally, we summarize this chapter in section 4.7.

4.2 Related Work

This section provides a literature review on the two aspects, namely, modeling of production lines and DR scheduling under available power and production rate constraints.

4.2.1 Production line modeling

Modeling and analysis of manufacturing systems are important factors to support design and operational decision makings. According to Papadopoulos and Heavey [PH96], modeling techniques such as generative and evaluative models are used to address manufacturing design and operational problems. Generative models provide optimal solutions to a modeling problem. However, they are restrictive in terms of their structural assumptions that are only transparent to the model designer. Unlike generative models, evaluative models are more flexible, but they don't guarantee optimal solutions. Nevertheless, they provide valuable performance measures of the problem under consideration. In this thesis, we consider queuing theory-based models which are evaluative models. Referring to the book of Curry and Feldman [CF10], the queuing models estimate system's performance measures from cumulative number of arrivals and departures of jobs or parts to/from workstations.

Queuing theory-based models for manufacturing systems have been gained much attention since a couple of decades ago with initial works by Buzacott and Shanthikumar [BS92], Askin and Standridge [AS93], and Govil and Fu [GF99]. Comparing with simulation, queuing theory based analytical modeling is much faster in estimating manufacturing system performance and provides more insights for performance improvement. However, existing queuing models focus on simple extensions of the classical queuing theory and fail to question its applicability to the complicated manufacturing systems. To address this issue, different works on the extensions of queuing theory have been given in the literature. For instance, Manitz [Man08] used a queuing model for performance analysis in assembly manufacturing lines in which work-pieces from two or more input workstations have to be merged to form a new one for further processing downstream. The author set up a mathematical model to find an optimal schedule in a single workstation scheduling problem. The output of the model decides when to turn ON or OFF the workstation based on energy prices.

Another recent work on the performance analysis of manufacturing systems using queuing models is described in [Jia+16]. The authors focused only on a transient state of the manufacturing system to study performance metrics such as production rate, workstation starvation and blockage. For throughput analysis of flow lines (can be named as production lines or serial lines), there is an extensive body of research that considered reliable and unreliable workstations, see research

works in [BS93], [DPH07] and [Alt12]. These works mostly aimed at performance analysis of asynchronous transfer lines and other manufacturing types discussed in section 2.2.3 of chapter 2. However, models on a **S**ynchronous **P**roduction **L**ine (SPL) are less addressed. We see a gap in the literature on these manufacturing types and we rely on a queuing model for performance analysis of both transient and steady states. In section 4.3, we formulate the arrival and departure processes at workstations of an SPL system based on the queuing model described in section 2.5 of chapter 2.

4.2.2 Scheduling in production lines

On the scheduling side, the most important production scheduling problems in manufacturing systems include minimizing makespan [MDC04], improving production efficiency, reducing power consumption and energy costs [Shr+14]. In the literature, different optimization approaches and mathematical models have been used in the production scheduling problems. Most of these methods consist of algorithms such as linear programming, integer programming, dynamic programming, genetic algorithms, etc. Among these techniques, few handle specific objectives and few tackle specific problem instances that takes into account necessary computational times.

Fang et al. [Fan+11] proposed a general multi-objective mixed integer linear programming for flow shop scheduling problem that considers peak power load, energy consumption, and associated carbon footprint in addition to makespan. They considered the operation speed of two workstations in the flow shop as an independent variable that can be changed to affect the peak load and energy consumption. However, it is not necessary true that all workstations are regulated by their processing speeds as some workstations have only ON and OFF modes (e.g., hydraulic presses, grinders, choppers, etc) according to [SAM13].

Focusing on scheduling during off-peaks times, Luo et al. [Luo+13] used ant colony optimization to solve a hybrid flow shop scheduling considering production and EE. They proposed a multi-objective optimization solution for minimizing makespan and energy consumption power with **T**ime **O**f **U**se (TOU) prices. They assumed fixed electricity price during some time of a day. However, changes of electricity prices in short periods should also be considered as it is the case with a real-time pricing DR mechanism such as **R**eal-**T**ime **P**ricing (RTP). Similar works are also discussed in [MSP13], [Shr+14] and [THL13].

All the research works mentioned above used heuristic algorithms to solve cost minimization problems by relying on variable electricity prices. That is, they focused on price-based DR programs discussed in section 2.3.1.1 of chapter 2. However, an incentive-based DR (refer to section 2.3.1.2 of chapter 2) was not considered. In this type of DR, power consumption is upper bounded with the request of a legacy energy provider. For instance, in France, EDF asks its

customers to shave some percentage of their loads in response to changes in electricity prices due to environmental conditions. Then, in case of industrial customers, they stop some industrial processes to respect the upper bound on their power consumption. Hence, finding an scheduling algorithm to adapt the power consumption of the industrial processes to the available power has a paramount significance. In section 4.4, we present our DR scheduling algorithm which adapts power consumption of industrial processes to available power and production rate constraints. Illustrating examples for an SPL system are also provided.

In the following section, we detail our works on the modeling of an SPL system using a queuing theory-based model.

4.3 Modeling a Synchronous Production Line

In this section, we first define a virtual cell concept which is used to model the property of an SPL system. Secondly, we provide a theoretical framework based on queuing theory to analyze temporal evolution of the system. For formal description of the SPL system, refer to section 2.2.3.1 of chapter 2.

4.3.1 Virtual cell

Virtual cell is a logical concept that allows us to decompose conveyor of the SPL system into several consecutive areas of the same size and shape. Each part/task is placed in the cell center when it arrives to the system. The distance between two cell centers is computed according to conveyor speed, V_c, and fixed inter-arrival time, T_a, of tasks at the conveyor. Let VC_i denote the virtual cell at position i, and $\text{dist}(VC_i, VC_{i+1})$ denote the distance between VC_i and VC_{i+1} cell centers. Then, $\text{dist}(VC_i, VC_{i+1})$ is given by the following formula:

$$\text{dist}(VC_i, VC_{i+1}) = V_c * T_a. \tag{4.1}$$

According to Equation (4.1), the time the conveyor needs to move tasks between two consecutive cells is T_a. In order to establish the synchronization (processing at the same time) between workstations (machines hereafter), their positions must coincide with cell centres. It means that the position of any machine from the 1st virtual cell is a multiple of cell size. More formally, for machine M_Q, there exists a strictly positive integer l such that:

$$\text{dist}(VC_1, M_Q) = l * \text{dist}(VC_i, VC_{i+1}). \tag{4.2}$$

Since the positions of machines are fixed with respect to the conveyor, the task inter-arrival time must be carefully selected to satisfy Equation (4.2). Thus, for

$l \in \mathbb{N}^*$ and merging both Equation (4.1) and (4.2), the value of T_a is given by:

$$T_a = \frac{dist(\mathrm{VC}_1, M_Q)}{l * V_c}. \tag{4.3}$$

4.3.2 Queuing theory-based model for the SPL

Based on the virtual cell concept discussed above, the SPL system can be constructed as shown in Figure 4.1. We assume that the system contains M machines labeled as M_1, M_2, ..., M_M located on virtual cells at K_1, K_2, ..., K_M, respectively with $K_0 = VC_1 \leqslant K_1 \leqslant K_2 \leqslant ... \leqslant K_M = VC_k \leqslant K_{M+1} = \infty$. Machine M_i has a constant processing time C_i, $0 < C_i < \infty$, to process a task. Note that processing times of two machines M_i and M_j, $i \neq j$, can be the same ($C_i = C_j$) or different ($C_i \neq C_j$). For mathematical convenience, we assign C_0 = 0 as an initial condition for fictitious machine M_0. The inter-arrival time of tasks to the conveyor is noted as T_a. Due to the idle time of the conveyor when the machines process tasks, T_a is modified according to the maximum processing time among the machines.

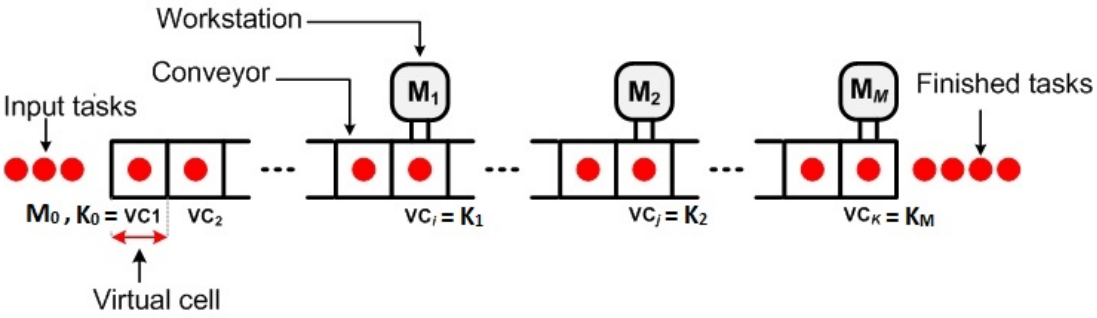


Figure 4.1 – General configuration of a synchronous production line

We decompose the SPL system into M sub-systems (since there are M machines). We assume that the sub-systems are reliable, i.e., there is no failure. Each sub-system i is composed of machine M_i and a set of virtual cells of size $K_i - K_{i-1} - 1$ that are just after the predecessor machine M_{i-1}. We represent the sub-system i as $\mathrm{D/D/1/}(K_i - K_{i-1})$ queuing system (for descriptions of the queue model, refer to section 2.5 of chapter 2). Indeed, each queue has a single server (which is the machine itself), a buffer composed of a set of virtual cells with capacity $K_i - K_{i-1}$ and a **F**irst **I**n **F**irst **O**ut (FIFO) service discipline. The inter-arrival and service times of each queue are deterministic and they can evolve based on synchronization between the queues. Tasks entering the queue are blocked when there is no free virtual cell downstream. Moreover, if the processing time of machine M_i is greater than that of M_{i-1}, machine M_{i-1} suspends any activity (i.e., it is blocked) and the tasks waits until a departure occurs from machine M_i. After this, simultaneous departures take place in both machines. In

general case, the machine with maximum service time blocks all other machines and all departures happen at the same time. This is why the SPL system can be modeled as a network of $D/D/1/(K_i - K_{i-1})$ queues with **B**locking **A**fter **S**ervice (BAS) mechanism.

Before defining temporal behaviors of the machines, we first need to ensure steady state conditions. For this purpose, we provide the following lemma.

Lemma 4.1 (Steady state conditions)**.**
Once the SPL system starts working, it will always reach a steady state.

Proof. To be in a steady state, the inter-arrival time must be greater than the maximum processing time among the machines. Injecting tasks on the conveyor are stopped when at least one machine is processing. It means that the initial deterministic inter-arrival time T_a is modified to take into account the blockage time. The blocking time takes the maximum processing time among the machines and it can be changed as the first task moves from the first machine to a machine having the maximum processing time (any machine downstream). This phase represents a transient state. After that, the system is in steady state. Consequently, the modified inter-arrival time T_a^* is given below:

$$T_a^* = T_a + \max_{1 \leqslant j \leqslant M} C_j > T_a \tag{4.4}$$

■

Now, our objective is to characterize and predict the behavior of each machine in temporal domain. In other words, we need to know the exact state of the system at any time t. We don't focus on expressing the performance of the system in average values. Hence, we don't rely on the well-known results of queuing network theories with BAS mechanism. To attain our goal, we define a temporal behavior of any machine M_Q as a function of utilization, $U_{M_Q}(t)$, for any time t. Before providing the utilization function, we define arrival instants of tasks to the machines. This helps us to characterize arrival and departure processes at every machine. The utilization function is based on the two processes.

4.3.2.1 Arrival instants

Let us denote arrival instant of task n to the virtual cell corresponding to M_Q as $\text{Inst}(n, M_Q)$. To compute the arrival instant $\text{Inst}(n, M_Q)$, we consider the SPL configuration example shown in Figure 4.2 where four machines having different processing times are linked by the same conveyor. The arrival instants of tasks to the virtual cells (machines included) are given in Table 4.1. From the table, we note that the arrival instants of tasks to machine 1, $\text{Inst}(n, M_1)$, are divided into four levels (highlighted by colored cases) based on increments on T_a (it is due to synchronizations between machines). The 1^{st} level represents

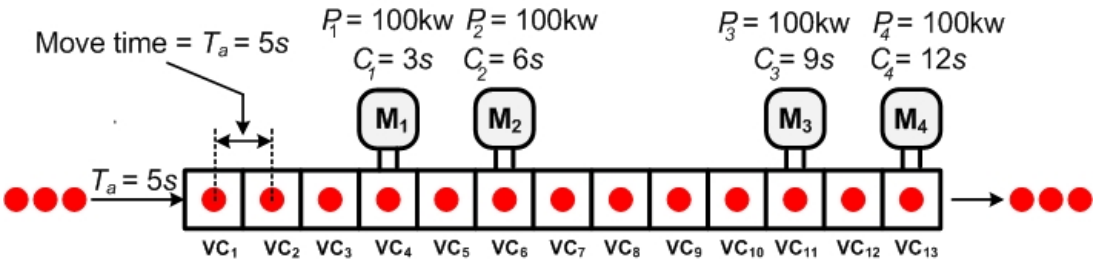


Figure 4.2 – An SPL system with 13 virtual cells and 4 machines.

Table 4.1 – Instantaneous arrival times of tasks on the virtual cells.

Tasks	VC_1	VC_2	VC_3	M_1	VC_5	M_2	VC_7	VC_8	VC_9	VC_{10}	M_3	VC_{12}	M_4	VC_{14}
1	0	5	10	15	23	31	42	53	64	75	86	100	114	131
2	5	10	15	23	31	42	53	64	75	86	100	114	131	148
3	10	15	23	31	42	53	64	75	86	100	114	131	148	165
4	15	23	31	42	53	64	75	86	100	114	131	148	165	182
5	23	31	42	53	64	75	86	100	114	131	148	165	182	199
6	31	42	53	64	75	86	100	114	131	148	165	182	199	216
7	42	53	64	75	86	100	114	131	148	165	182	199	216	233
8	53	64	75	86	100	114	131	148	165	182	199	216	233	250
9	64	75	86	100	114	131	148	165	182	199	216	233	250	267
10	75	86	100	114	131	148	165	182	199	216	233	250	267	284
11	86	100	114	131	148	165	182	199	216	233	250	267	284	301
12	100	114	131	148	165	182	199	216	233	250	267	284	301	318
13	114	131	148	165	182	199	216	233	250	267	284	301	318	335
14	131	148	165	182	199	216	233	250	267	284	301	318	335	352

the time between receiving the 1st task by machines 1 and 2. In this level, the increment value is $T_a + C_1 = 8s$ and concerns tasks 1, 2 and 3. We can see that $\text{Inst}(2, M_1) = \text{Inst}(1, M_1) + (T_a + C_1)$ and $\text{Inst}(3, M_1) = \text{Inst}(2, M_1) + (T_a + C_1)$. The 2nd level represents the time between receiving the 1st task by machines 2 and 3. The increment value of this level is $T_a + \max_{1 \leqslant j \leqslant 2} C_j = T_a + C_2 = 11s$, and the concerned tasks are from 4 to 8. The 3rd level represents the time between receiving the 1st task by machines 3 and 4, and the increment value is $T_a + \max_{1 \leqslant j \leqslant 3} C_j = T_a + C_3 = 14s$. The last level begins when the 1st task is received by machine 4 until the last task is processed. The increment value becomes $T_a + \max_{1 \leqslant j \leqslant 4} C_j = T_a + C_4 = 17s$. In the same way, we remark that the arrival instants of tasks to machines 2, 3 and 4 have three, two and one levels, respectively. In general case, the arrival instant levels of tasks to a machine depend on the maximum processing times of other machines and on the order of successors/predecessors. To determine the existence of new levels, we give the following lemma.

Lemma 4.2 (Level change)**.**
The arrival instant of tasks to machine M_i will have a new level if and only if

there exits a machine M_j (downstream of M_i) having a greater processing time.

Proof. The proof contains two aspects. First, let us consider machine M_j which is downstream of M_i with $C_j > C_i$. Before the 1st task arrives at machine M_j, the maximum increment value is $T_a + \max_{1 \leqslant l < j} C_l$. When it arrives at the machine, the new increment value will be $T_a + C_j > T_a + \max_{1 \leqslant l < j} C_l$. Hence, we have a new level. Secondly, let us consider that machine M_i has new level. It means that we have a new increment value $T_a + \alpha$ with $\alpha > T_a$. This shows that there is machine M_k downstream with $C_k = \alpha$. ■

Based on Lemma 4.2, if a task arrives in a machine with maximum processing time among all the machines, we won't have a new level. If the first machine is the one with the maximum processing time, all the machine will have only one level. Now, we give a lemma to show the arrival instants of tasks at beginning of each level. For mathematical simplicity, we suppose that the levels of machine M_Q are numbered from Q. That is, the first level of machine M_Q is numbered as Q, second level as $Q + 1$, and so on. The following lemma formalizes task arrival instants to the machines.

Lemma 4.3 (Beginning of level change due to task arrival)**.**
A level $m = Q, Q + 1, ..., M$ in machine M_Q begins with an arrival instant $t_0 + \sum_{i=0}^{m-1}(K_{i+1} - K_i)(T_a + \max_{0 \leqslant j \leqslant i} C_j)$.

Proof. The 1st level of each machine begins when the 1st task arrives at the machine. The corresponding arrival instants can be given by:

$$\begin{cases} \text{Inst}(1, M_1) = t_0 + (K_1 - K_0)(T_a + \max\limits_{0 \leqslant j \leqslant 0} C_j) \\ \text{Inst}(1, M_2) = \text{Inst}(1, M_1) + (K_2 - K_1)(T_a + \max\limits_{0 \leqslant j \leqslant 1} C_j) \\ \vdots \\ \text{Inst}(1, M_M) = \text{Inst}(1, M_{M-1}) + (K_M - K_{M-1})(T_a + \max\limits_{0 \leqslant j \leqslant M-1} C_j) \end{cases} \tag{4.5}$$

By replacing the recursive equations, we generalize as $\text{Inst}(1, M_Q) = t_0 + \sum_{i=0}^{Q-1}(K_{i+1} - K_i)(T_a + \max_{0 \leqslant j \leqslant i} C_j)$, where $Q = 1, 2, ..., M$. Now, let us consider level $m > 1$ of machine M_Q. Based on Lemma 4.2, there exists a machine $M_{Q'}$ where $Q' > Q$ having greater processing time. When task number 1 arrives at this machine, a new level begins on all upstream machines. It is easy to see that $M_{Q'}$ is the same with M_m due to synchronization (see diagonal values in Table 4.1). We recall that $m \geqslant Q$. Let tasks number 1 and n arrive simultaneously at machines M_Q and M_m, respectively. It implies that $\text{Inst}(n, M_Q) = \text{Inst}(1, M_m) = t_0 + \sum_{i=0}^{m-1}(K_{i+1} - K_i)(T_a + \max_{0 \leqslant j \leqslant i} C_j)$ ■

To generalize the arrival instants of tasks for every machine, we provide the following theorem.

Theorem 1 (Arrival instant of tasks)**.**
The instantaneous arrival time of task n in machine M_Q is given by:

$$\begin{cases} \text{Inst}(n, M_Q) = t_0 + \sum_{i=0}^{m-1} (K_{i+1} - K_i)(T_a + \max_{0 \leqslant j \leqslant i} C_j) + \\ \qquad\qquad (n - (K_m - K_Q) - 1)(T_a + \max_{1 \leqslant j \leqslant m} C_j), \\ \text{with } K_m - K_Q + 1 \leqslant n < K_{m+1} - K_Q + 1, \\ \text{and } m = Q, Q+1, ..., M \end{cases} \tag{4.6}$$

Proof. It can be easily proved by generalizing Equation (4.5) to handle n arrivals to all machines. ■

4.3.2.2 Arrival processes at machines

Now, under steady state conditions and based on the arrival instant function, the arrival processes to every machine in the SPL system can be given as:

$$A_{M_Q}(t) = \begin{cases} 0 & \text{if } t_0 \leqslant t < \text{Inst}(1, M_Q), \\ n & \text{if } \text{Inst}(n, M_Q) \leqslant t < \text{Inst}(n+1, M_Q). \end{cases} \tag{4.7}$$

Then, by applying Theorem 1 to Equation 4.7, we can compute generalized arrival processes as follows:

$$A_{M_Q}(t) = \begin{cases} 0 \text{ if } t_0 \leqslant t < \text{Inst}(1, M_Q), \\ \left\lfloor \frac{t - (t_0 + \sum_{i=0}^{m-1} (K_{i+1} - K_i)(T_a + \max_{0 \leqslant j \leqslant i} C_j) - (K_m - K_Q)(T_a + \max_{1 \leqslant j \leqslant m} C_j))}{T_a + \max_{1 \leqslant j \leqslant m} C_j} \right\rfloor + 1 \\ \text{if } \text{Inst}(n, M_Q) \leqslant t < \text{Inst}(n+1, M_Q). \end{cases} \tag{4.8}$$

4.3.2.3 Departure processes at machines

Under the same assumptions as the arrival processes, the departure processes from the machines in the SPL system can be given as:

$$D_{M_Q}(t) = \begin{cases} 0 & \text{if } t_0 \leqslant t < \text{Inst}(1, M_Q) + \max_{1 \leqslant j \leqslant Q} C_j, \\ n & \text{if } \text{Inst}(n, M_Q) + \max_{1 \leqslant j \leqslant m} C_j \leqslant t < \text{Inst}(n+1, M_Q) + \max_{1 \leqslant j \leqslant m} C_j. \end{cases} \tag{4.9}$$

By applying Theorem 1 to Equation 4.9, we generalize the departure processes as follows:

$$D_{M_Q}(t) = \begin{cases} 0 \text{ if } t_0 \leqslant t < \text{Inst}(1, M_Q) + \max_{1 \leqslant j \leqslant Q} C_j, \\ \left\lfloor \frac{t - (t_0 + \sum_{i=0}^{m-1} (K_{i+1} - K_i)(T_a + \max_{0 \leqslant j \leqslant i} C_j) - (K_m - K_Q)(T_a + \max_{1 \leqslant j \leqslant m} C_j)) + \max_{1 \leqslant j \leqslant m} C_j}{T_a + \max_{1 \leqslant j \leqslant m} C_j} \right\rfloor + 1 \\ \text{if } \text{Inst}(n, M_Q) \leqslant t < \text{Inst}(n+1, M_Q). \end{cases} \tag{4.10}$$

4.3.2.4 Utilization function

The utilization function includes two states: *working* and *blocking (or idle)* states. The working state is more important because it is the state in which the machine consumes power. This defines an effective utilization function $\hat{U}_{M_Q}(\cdot)$ of machine M_Q. The function is given by:

$$\hat{U}_{M_Q}(t) = U_{M_Q}(t) * g(t), \tag{4.11}$$

where $g(\cdot)$ is a filter function to separate the two states, so that the working state takes C_Q time units. This filter function can be expressed as:

$$g(t) = \begin{cases} 1 & \text{if } U_{M_Q}(t) - U_{M_Q}(t - C_Q) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

The utilization function $U_{M_Q}(t)$ indicates whether a task exists under machine M_Q or not at time t. This can be expressed in queuing theory as the difference between arrival and departure processes to/from the corresponding machines. Based on the arrival and departure processes (Equation (4.8) and (4.10)), we can now express the utilization function $U_{M_Q}(\cdot)$ as follows:

$$U_{M_Q}(t) = A_{M_Q}(t) - D_{M_Q}(t). \tag{4.13}$$

4.3.2.5 Power consumption and utilization function

In the above section, we determined the effective utilization function that shows the temporal evolution of the machines. Based on Equation. (4.11), we can determine power consumption $P_{M_Q}(t)$ of machine M_Q at time t as:

$$P_{M_Q}(t) = \hat{U}_{M_Q}(t) * p_Q, \tag{4.14}$$

where p_Q is power consumption of machine M_Q in *working* state. When a machine is idle (i.e., blocked after processing), it consumes a small amount of power. We assume that this amount is negligible in this work. Hence, the total power consumption of the SPL system is expressed as:

$$P_M(t) = \sum_{Q=1}^{M} P_{M_Q}(t). \tag{4.15}$$

Equation (4.15) gives system's total power consumption at any time t under steady state conditions. It describes the temporal power consumption evolution of the system. In section 4.5, we provide analytical and simulation results of queuing theory-based temporal characteristics of the SPL system.

The next section presents our DR scheduling algorithm that adjusts the power consumption of the SPL system to satisfy DR and production rate constraints.

4.4 Scheduling in the SPL system

The second part of this chapter describes our DR scheduling algorithm which is used to adapt power consumption of all machines to the available power (or DR threshold). In section 4.4.1, we highlight the scheduling problem in the SPL system. Then, section 4.4.2 presents the scheduling algorithm using examples. DR optimality conditions are discussed in section 4.4.3. We also illustrate existence and feasibility conditions of schedules. Finally, we provide a necessary and sufficient conditions to accept DR requests positively in section 4.4.4. At first, we provide the following definitions.

Definition 4.1 (Production rate)**.**
In manufacturing systems, production rate is the number of items that can be produced during a given period of time, i.e., items per unit time.

Definition 4.2 (Power consumption block)**.**
Power consumption block of a machine is defined by its height which represents the amount of power required to process a task on the machine and by its width which represents the processing time of a task.

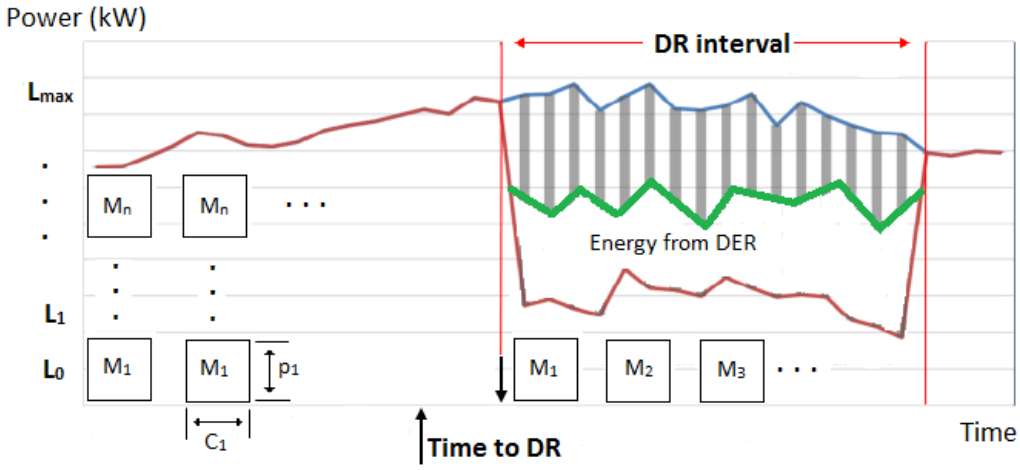


Figure 4.3 – Power consumption blocks and additional energy from DERs.

We refer to Figure 4.3 for better understanding of definition 4.2. From the figure, we can see that machine M_1 has a power consumption block of C_1*p_1 where C_1 is its processing time and p_1 is its power consumption when it processes a task. Under normal conditions, the available power is enough to accept all the power consumption blocks. However, during DR interval, the available power may satisfy only one consumption block at a time. The existence of local **D**istributed **E**nergy **R**esources (DERs) can increase the total available power so that one or more consumption blocks can be added to maximize the production rate (see

definition 4.1). However, the intermittent nature of DERs requires energy storage devices (e.g., battery) to fill voids created by forecasting errors. For this purpose, determining battery size and smoothing out of fluctuations were investigated in chapter 3, but we do not consider it in this chapter. Based on the above definition, we now define production mode of a manufacturing system.

Definition 4.3 (Production mode)**.**
Production mode of a manufacturing system is defined by the available power levels which defines discrete values to match power consumption blocks of machines.

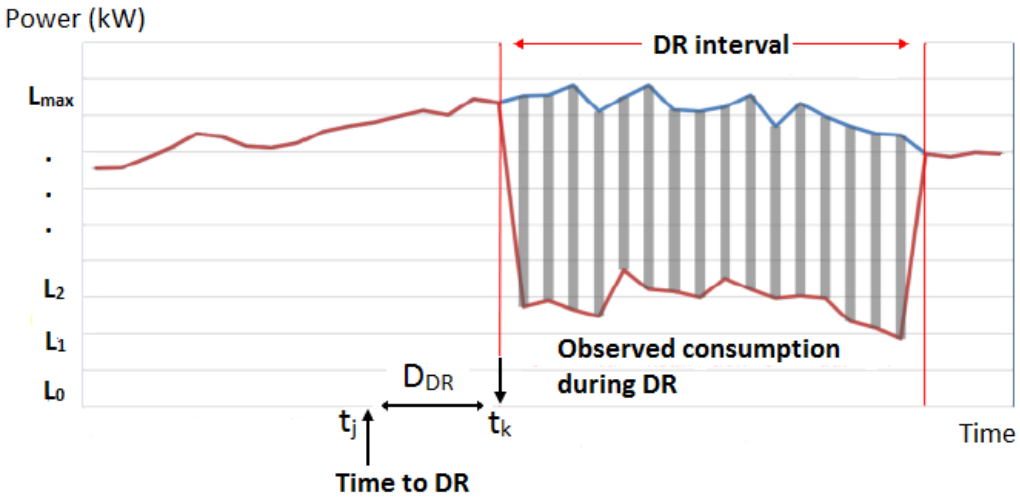


Figure 4.4 – DR mechanism and available power levels.

To clarify the above definition, we refer to Figure 4.4. In the figure, possible power levels are marked by L_0, L_1, ..., L_{max}. L_0 corresponds to a case where there is no available power. At this power level, the system is in power outage (or in complete shutdown). However, at L_{max}, power requirement of all machines can be satisfied and this provides the maximum production rate in the system. Based on these information, we can construct status and power consumption vectors that consist of the status (ON/OFF) of machines and their corresponding power consumption. Let x_i (1 for ON and 0 for OFF) denotes the status of machine M_i and p_i denotes power consumption of M_i when it is processing a task. Hence, a status vector for a production line with M machines is represented as $X = [x_1, x_2, ..., x_M]$ and the corresponding power consumption vector is $p = [p_1, p_2, ..., p_M]$. Then, total power consumption is given by:

$$\mathrm{P} = X^T * p. \tag{4.16}$$

Figure 4.4 also shows power consumption modes under maximum available power and observed power consumption during DR interval. Under normal

conditions, every machine can get its share of the available power. To enter the DR power consumption mode, a DR event (with reduction percentage and duration) should be communicated at some time (*Time to DR* in the figure) before the actual mode change happens. Assume that the DR event is emitted from a legacy energy provider at t_j and the actual load shaving starts at t_k. We represent the difference between these times (t_k - t_j) by D_{DR} (deadline for DR). The length of D_{DR} can vary from 10 minutes to 1 hour. In most of the cases, DR event is communicated 1 hour earlier than the start of DR. In some cases, a DR event can be issued in short time (*emergency* DR) to correct forecast errors or to handle some unanticipated events such as partial electricity grid failure. In such situations, the availability of local DERs such as wind, solar, etc could help the manufacturing plants to accept the DR call positively.

The following section states the scheduling problem of the SPL system under available power and production rate constraints.

4.4.1 Scheduling problem

A *schedule* in the SPL system is a finite sequence of actions spaced by a deterministic (not necessarily constant) inter-arrival time and performed at the conveyor input. An action consists of inserting a new task (denoted by symbol I) or not inserting ($\overline{I}$). Consequently, a schedule can be considered as a word over alphabet $\Sigma = \{I, \overline{I}\}$. Each schedule word has a *prefix*, *sub-word* and *suffix*. The *prefix* is used to resume actions in transient phases, while the *sub-word* represents steady phases, and the *suffix* word is used to flush the conveyor (i.e., to empty the system). Note that the prefix word necessarily begins with I and has length K, the sub-word is a periodic sequence of symbols, and the suffix word takes $\overline{I}^K$ value where K is the number of virtual cells in the system. For instance, for the SPL shown in Figure 4.2, a schedule is $I^{13}I^n\overline{I}^{13}$ where I^{13} is the prefix word formed by thirteen consecutive task insertions (in a transient phase), I^n is the sub-word representing the insertion of n consecutive tasks in the steady state, and finally $\overline{I}^{13}$ is the suffix word to purge the conveyor formed by not inserting tasks for 13 times. It is clear that this schedule provides a maximum production rate, and results in higher total power consumption because all the machines work simultaneously. However, the schedule $I(\overline{I}^{12}I)^n\overline{I}^{13}$ ensures that at each time only one task is finished before inserting a new task and at maximum only one machine is working and this results in reduced power consumption. In the following lemma, we show how to compare two schedule words based on a production rate.

Lemma 4.4 (A schedule word with higher production rate)**.**
For two schedule words of the same length, the schedule having more number of occurrence of I produces maximum number of finished tasks.

Proof. Let w_1 and w_2 are two schedule words of the same length, i.e., $||w_1|| = ||w_2||$. Let n_1 and n_2 (with $n_1 > n_2$) are number of occurrence of symbol I within w_1 and w_2, respectively. Based on schedule word definition, both words ends with the same suffix $\overline{I}^K$ where K is number of virtual cells in the system. Consequently, there are no tasks on the conveyor when the schedule ends. All the inserted tasks are processed in both schedules. Since $n_1 > n_2$, schedule word w_1 produces more finished tasks than w_2. ■

Now, we formally define the scheduling algorithm based on the construction of a graph of activities with power consumption constraints.

4.4.2 DFSM schedule coding

The working principle of SPL can be regarded as a **D**eterministic **F**inite **S**tate **M**achine (DFSM) [Sak09] where states represent whether there is a task or not in each virtual cell and transitions from one state to another represent symbols of schedules. We distinguish left and right successors according to the symbols, i.e., we go to the left for symbol I and go to the right for symbol $\overline{I}$. We write a state with binary values of size K, the number of virtual cells. More precisely, a state is expressed as $(b_1b_2...b_K)_2$ where b_i can be 1 if there is a task in VC_i or 0 otherwise. The state $(00...0)_2$ is both start and end state. The left successor of $(b_1b_2...b_K)_2$ corresponds to shifting right by one bit and adding $(2^{K-1})_2$. The new state will have $(b_1b_2...b_K)_2 \gg 1 \oplus (2^{K-1})_2$ value. The right successor of $(b_1b_2...b_K)_2$ corresponds to shifting right by one bit. Then, the new state will have $(b_1b_2...b_K)_2 \gg 1$ value. The duration of each transition in the DFSM takes the deterministic inter-arrival T_a time units. The time spent in each state depends on maximum processing time among working machines. Figure 4.5 illustrates an example of DFSM of an SPL system with three machines. We consider that each machine corresponds to a virtual cell, i.e., there are three virtual cells in the SPL.

Based on the above description, the DFSM is defined as a quintuple $(\Sigma, V, v_0, \delta, F)$, where:

- $\Sigma = \{I, \overline{I}\}$ is the input alphabet;
- V is the set of states with cardinality $\mathcal{O}(2^K)$,
- $v_o = (00...0)_2$ is the initial state,
- δ is the state transition function, δ: $V \times \Sigma \rightarrow V$. According to the input symbol: $\delta(I, (b_1b_2...b_K)_2)$ shifts right the state value by one bit and adds $2^{K-1}{}_2$, and $\delta(\overline{I}, (b_1b_2...b_K)_2)$ shifts right by one bit,
- F is the set of final states. It has only one state, v_0.

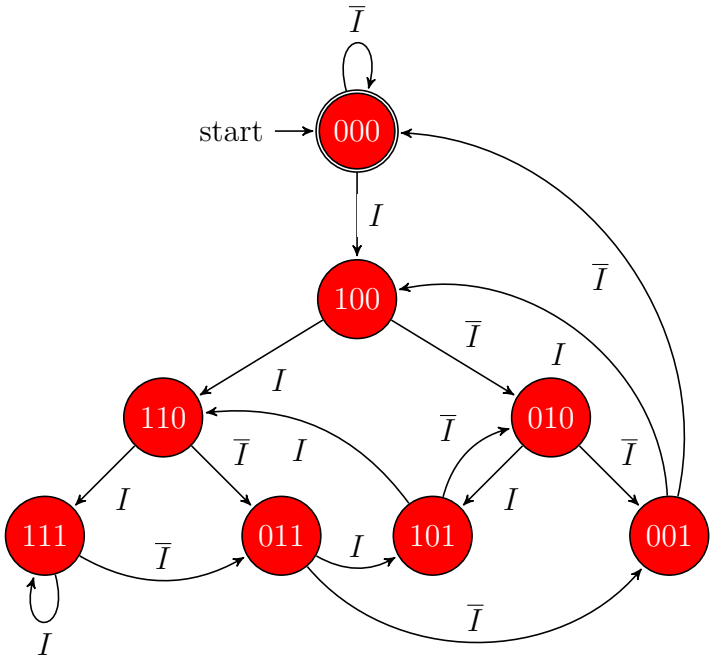


Figure 4.5 – A DFSM for an SPL system with three machines

A default schedule in the SPL is to insert a task every T_a time unit. Therefore, the corresponding schedule word is given by $I^K I^n \overline{I}^K$, where n is the number of consecutive tasks inserted during the steady state. Regarding the DFSM, the prefix of the default schedule (I^K) is to go from the first state $(00...0)_2$ to $(11...1)_2$ taking the left successors during transitions. The sub-word (I^n) is to stay in the state $(11...1)_2$ by looping for n times. The suffix ($\overline{I}^K$) is to go back to end state $(00...0)_2$.

4.4.3 Optimal schedule under DR

When a given schedule transits to state $(11...1)_2$, the SPL system consumes maximum power. The following lemma confirms this SPL behavior.

Lemma 4.5 (State of peak power consumption)**.**
The peak power consumption is attained when the system is in state $(11...1)_2$.

Proof. The total power consumption at state $(11...1)_2$ is $\sum_{i=1}^{M} p_i$ which is derived from Equation (4.16). Since it is the state where all machines are working, it is obvious that the power consumption is peak. ■

Since the default schedule is looping in state $(11...1)_2$, the power consumption is peak from the above lemma. Assume that a DR signal is emitted at t_r from start of the SPL demanding to reduce power consumption in D_{DR} time unit for duration of Γ by an amount of Δ (refer to Figure 4.6). During the DR interval,

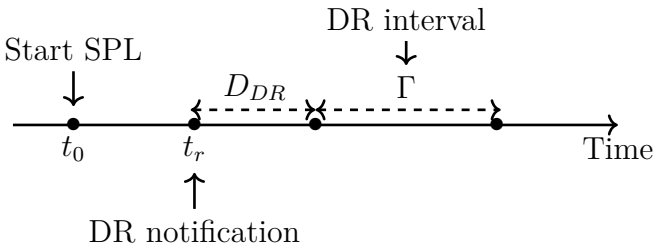


Figure 4.6 – DR timelines

the default schedule is not attainable due to its peak power limitation, i.e., the available power is not enough for all machines to work at the same time. Now, we need to find a new schedule reducing the peak power consumption in the DR interval by amount of Δ as formulated below:

$$\begin{cases} P(t) \leqslant \max\{P(t')\} - \Delta, \\ \text{with } t_0 \leqslant t' < t_r + D_{DR} \text{ and } t_r + D_{DR} \leqslant t \leqslant t_r + D_{DR} + \Gamma \end{cases} \tag{4.17}$$

Under the DR constraints, the scheduling algorithm consists of three phases. The first phase determines if a feasible schedule exits or not. Then, the second phase computes the optimal schedule satisfying DR constraints. Finally, in the third phase, we search for the shortest path to go from the previous schedule to the new optimal one. The procedures in the three phases are described in the following section.

4.4.3.1 Phase 1 - Existence of feasible schedule

In phase one, we would like to know if a feasible schedule exists or not. The following lemma considers if a schedule exits or not under DR constraint.

Lemma 4.6 (Existence of feasible schedule)**.**
A feasible schedule exists if the power consumption of each machine is less than the available power.

Proof. A simple schedule is when there is only one task in the SPL system at a time. Consequently, only one machine is processing in the system at maximum. Since the power consumption of each machine is below DR threshold, the total consumption using this schedule is less than the available power. Hence, the schedule word exists and it is given by $I(\overline{I}^{K-1}I)^{n}\overline{I}^{K}$. ■

Based on Lemma 4.6, a DR constraint cannot be less than a lower bound given by the schedule in which only one machine is processing a task at a time. This schedule $(I(\overline{I}^{K_M-1}I)^{n}\overline{I}^{K})$ corresponds to go from state $(00...0)_2$ as 1st step to the most right side state $(00...01)_2$ until state $(10...0)_2$ and go back to state $(00...01)_2$ to form a cycle. For example, let us see the feasible schedule of an SPL

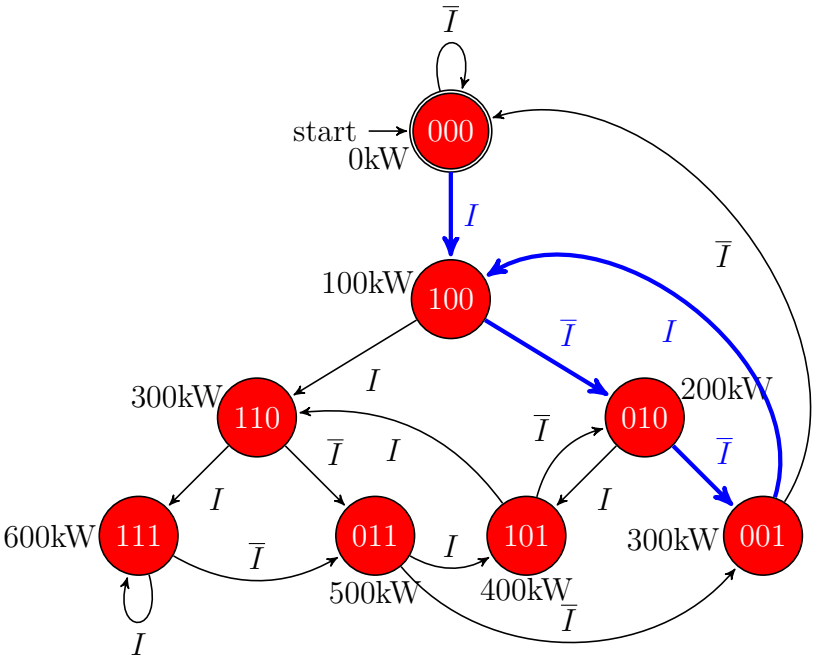


Figure 4.7 – Feasible schedule under DR constraint.

system with three machines in three consecutive virtual cells. We assume that the power consumption of machines 1, 2 and 3 are 100kW, 200kW and 300kW, respectively. Furthermore, we set Δ to 200kW. Hence, the new available power (or DR threshold) is 400kW according to Equation (4.17). The feasible schedule is to go from state $(000)_2$ to $(100)_2$, then we take right until state $(001)_2$. After that, we go back to state $(100)_2$. The transitions (thick lines) in Figure 4.7 represent a feasible schedule. We can see that the peak power consumption in the SPL is 300kW which is less than the DR threshold of 400kW.

4.4.3.2 Phase 2 - Finding optimal schedule

In phase 2, the objective is to find, if possible, an optimal schedule better than the feasible schedule that satisfies DR power constraint. This schedule must give the maximum production rate under the constraint. In other words, the number of occurrence of I in the schedule should be higher. A simple algorithm consists of exploring the DFSM by taking left successors to increase number of Is verifying the DR constraint at each state. This algorithm can be seen as a branch and bound ($B\&B$) [Cla99] algorithm where the entire candidate solutions are enumerated and a search is optimized based on upper and lower bounds. This algorithm tries to find a prefix word going from initial state to the depth (K) of DFSM and then it finds the shortest cycle that satisfies DR power constraints to form a sub-word.

In order to find the optimal prefix, we apply Algorithm 2 to the binary tree

Algorithm 2 Prefix searching in DFSM

Require: $(\Sigma, V, v_0, \delta, F)$, a DFSM
  Pow_dr, the available power under DR
  $x.state \leftarrow v_0$
  $x.visited\_left \leftarrow false$
  $x.visited\_right \leftarrow false$
  $Prefix \leftarrow \epsilon$
  Sub-word $\leftarrow \epsilon$
  while $|Prefix| < K$ **do**
    if $x.visited\_left == true$ & $x.visited\_right == true$ **then**
      $x \leftarrow parent[x]$
      $Prefix \leftarrow Prefix - Last\_symbol(Prefix)$
    else
      if $x.visited\_left == false$ **then**
        $y \leftarrow left[x]$
        $x.visited\_left \leftarrow true$
        if $y.power \leqslant Pow\_dr$ **then**
          $Prefix \leftarrow Prefix.I$
        else
          $y \leftarrow x$
      else
        $y \leftarrow right[x]$
        $x.visited\_right \leftarrow true$
        if $y.power \leqslant Pow\_dr$ **then**
          $Prefix \leftarrow Prefix.\overline{I}$
        else
          $y \leftarrow x$
    $x \leftarrow y$

resulted by removing cycles from the DFSM. All states having impaired value or the last bit fixed to one, $(1)_2$, become leaf nodes. This algorithm defines a preorder traversal: while a leaf node is not visited and under meeting the DR constraint, we recursively do a preorder traversal of the left sub-tree, followed by a recursive preorder traversal of the right sub-tree.

Figure 4.8 illustrates the optimal prefix (transitions in thick lines) after executing Algorithm 2 on the binary tree of the DFSM shown in Figure 4.7. We

Algorithm 3 Sub-word searching in DFSM

Require: $(\Sigma, V, v_0, \delta, F)$, a DFSM, and the last state found by the prefix searching algorithm, v_{prefix}

```
Pow_dr, the available power under DR
x.state ← v_prefix
x.visited_left ← false
x.visited_right ← false
cycle ← false
Sub-word ← ϵ
while cycle == false do
    if x.visited_left == true & x.visited_right == true then
        x ← parent[x]
        Sub-word ← Sub-word − Last_symbol
    else
        if x.visited_left == false then
            y ← left[x]
            x.visited_left ← true
            if y.power ⩽ Pow_dr then
                Prefix ← Sub-word.I
                if y.state == v_prefix then
                    cycle ← true
            else
                y ← x
        else
            y ← right[x]
            x.visited_right ← true
            if y.power ⩽ Pow_dr then
                Prefix ← Sub-word.Ī
                if y.state == v_prefix then
                    cycle ← true
            else
                y ← x
    x ← y
```

can see that the peak power consumption of this prefix in the SPL is 400kW which is equal to the DR threshold.

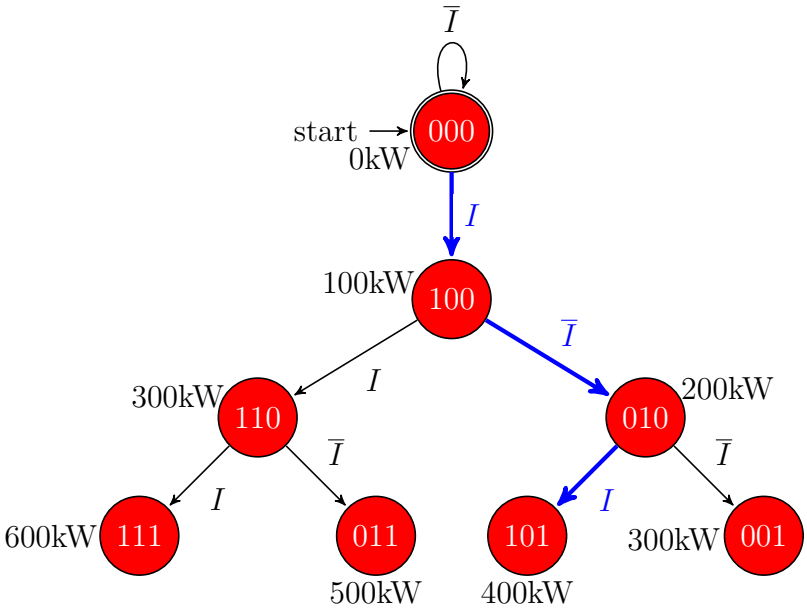


Figure 4.8 – Prefix searching example.

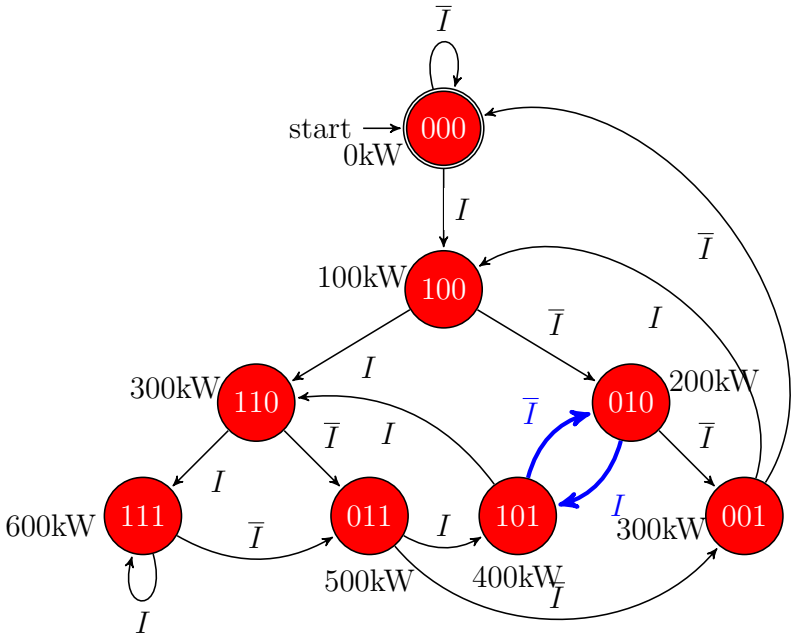


Figure 4.9 – Optimal sub-word searching example.

In order to find the sub-word of the schedule, we search a cycle going from the last state (at depth K) obtained by prefix searching and that eventually returns to the state. It is easy to see that the best method to continue producing at higher rate is the same as that used for prefix searching, i.e., while the starting state is not visited twice and under meeting DR power constraint, we recursively do a preorder traversal of the left successors, followed by a recursive preorder

traversal of the right successors. Algorithm 3 details such a principle.

Figure 4.9 illustrates the optimal sub-word (transitions in thick lines) after executing Algorithms 3 on the DFSM shown in Figure 4.7. We can see that the peak power consumption of this sub-word in the SPL system is 400kW which is equal to the DR threshold.

The performance of Algorithms 2 and 3 depend on the number of virtual cells K of the system. When K increases, the total number of states increases exponentially to 2^K states. Hence, the worst time complexity to search in the DFSM is $\mathcal{O}(2^K)$. Since the number of cells on the conveyor is limited, this algorithm could be of great help to find optimal solutions. To reduce this *curse of dimensionality*, we rely on a more efficient algorithm to search for a near-optimal solution based on linear programming (LP) formulations. To compute the prefix word, we find theoretically the best leaf state in the binary tree having a path from the root state satisfying the DR power consumption on each intermediate state. Then, we use the reverse path to set the prefix. The LP problem for the leaf state can be formulated as follows:

$$\max \sum_{i=1}^{K} x_i \tag{4.18}$$

$$\text{s.t: } A * x \leqslant P_{avail} \tag{4.19}$$

$$x_i \in \{0,1\}, i = 1,2,...,K-1 \text{ and } x_K = 1$$

where:

- x_i indicates whether a task exists or not in virtual cell i. For the leaf states, the last bit is set to one, i.e., $x_K = 1$,
- x is the vector of x_i,
- A is an upper triangular matrix representing power of intermediate states in the binary tree:

$$A = \begin{pmatrix} p_1 & p_2 & \cdots & \cdots & p_K \\ & p_1 & \cdots & \cdots & p_{K-1} \\ & & \ddots & \cdots & \vdots \\ & 0 & & \ddots & \cdots \\ & & & & p_1 \end{pmatrix} \tag{4.20}$$

- P_{avail} is the available power.

The objective function (Equation (4.18)) is to choose machines so as to maximize production rate and Equation (4.19) poses constraint on total power consumption in the path from root to leaf states to respect the available power constraint.

By applying the LP equations to the binary tree of the DFSM shown in Figure 4.7, we find that the state $(101)_2$ satisfies both DR threshold and maximum number of 1s in the vector x. The reverse path is $(000)_2 \rightarrow (100)_2 \rightarrow (010)_2 \rightarrow (101)_2$, and so the prefix is $I\overline{I}I$ which corresponds to the one found using Algorithm 2. To find a cycle starting and ending at the state obtained by the LP equations above, we rely on the well known problem of finding minimum cycle in weighted directed graphs [Kar78]. We replace cost metric by power consumption metric. We define the power consumption of an arc as the maximum between its ending states, and a path's power consumption as the maximum power consumption among its arcs. The mathematical formulation of the problem can be given as:

$$\begin{aligned} \underset{z(e)}{\text{maximize}} \quad & \{p(e)z(e), e \in E\} \\ \text{subject to} \quad & \sum_{e \in V^+(v)} z(e) - \sum_{e \in V^-(v)} z(e) = 0, \ \ \forall v \in V \\ & \sum_{e \in E} z(e) = 1 \\ & p(e) \leqslant p_{avail} \\ & z(e) \geqslant 0 \end{aligned} \tag{4.21}$$

where $V^+(v)$ is the set of arcs having extremities in v, $V^-(v)$ is the set of arcs originated in v and $z(e)$ is the variable associated to each arc e in the DFSM. By applying our LP equations for cycle searching in the DFSM (shown in Figure 4.7), we find a loop between $(101)_2$ and $(101)_2$.

4.4.3.3 Phase 3 - Path to optimal schedule

In phase three, we would like to find a shortest path from current state to the state obtained in phase 2 (transitions in red dotted lines in Figure 4.10). For this purpose, we use the well-known Dijkstra's algorithm [Dij59].

Dijkstra's algorithm finds a shortest path starting from an *initial* node to a *target* node (or all other nodes) on a graph. The algorithm assigns initial distance values and try to improve them step by step. In our case, we would like to find the shortest path between two states on the DFSM graph (see Figure 4.10), i.e., the short path between the optimal state $(101)_2$ found in phase 2 and the initial state which is $(111)_2$. The following steps are used to find the shortest path starting from initial state $(111)_2$:

(1) Assign some values for every state on the DFSM graph: set zero for the initial state $(111)_2$ and infinity for all other states,

(2) Set the initial state as current. Mark all other state unvisited. Construct an *unvisited set* for representing all unvisited states within the DFSM,

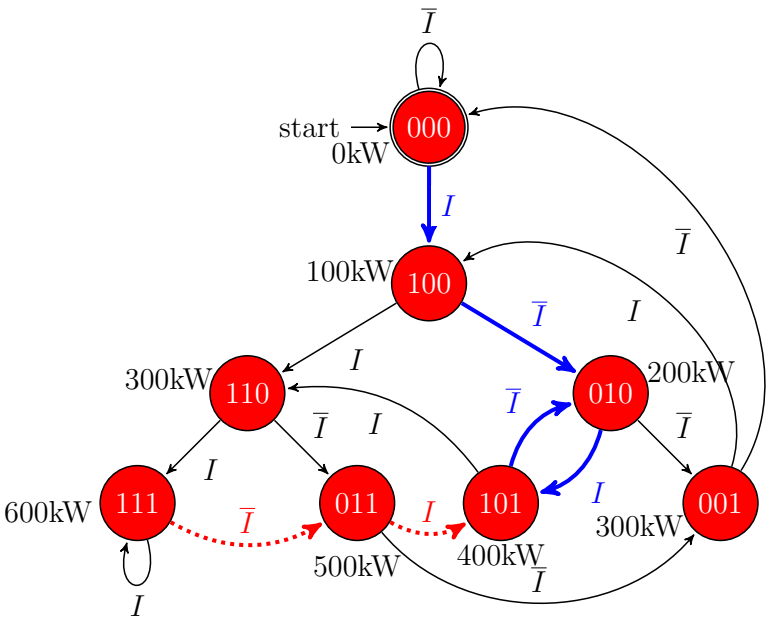


Figure 4.10 – Shortest path to the optimal schedule (phase 3) marked in red dotted lines.

(3) For the current state, consider all of its neighbors and calculate their *tentative* distances. Here, we assign a distance of value 1 for a neighbor state. For example, the distance beween states $(111)_2$ and $(011)_2$ is 1 as can be seen from Figure 4.10,

(4) After examining all the neighbors of the current state, mark the current state as visited and remove it from the *unvisited set*,

(5) If the destination state has been marked visited, then stop. The algorithm has finished, else

(6) Select the unvisited state that is marked with the smallest distance, i.e., state $(011)_2$ and set it as the new *current node*, and go back to step (3).

By following the above steps, state $(101)_2$ can be reached with a distance value of 2 which is the shortest path through state $(011)_2$. This path is marked by dotted lines in Figure 4.10.

The total duration the shortest path takes from current state to the state obtained in phase 2 determines if it is possible to accept the DR request or not. The next section analyses acceptance conditions of a DR request.

4.4.4 DR acceptance conditions

We now give a necessary and sufficient condition to be able to accept a DR request positively or not.

Lemma 4.7 (Condition to accept DR)**.**
A necessary and sufficient condition to be able to positively answer to a DR request is that the time to switch from any state A to another state B satisfying the maximum power imposed by DR should be less than or equal to duration of D_{DR} (*deadline to DR*, refer to Figure 4.4). This time is equal to the duration of each visited states in the DFSM graph described in Figure 4.10 plus the time of the state transitions (T_a) for each transition.

Proof. Under normal conditions, the system is at the state $(11...1)_2$ with duration corresponding to the maximum processing time among the machines. Then, starting from this state, we go to right successor to the next state $(11..10)_2$ which takes T_a and its processing time coincides with the working machines. We sum both the maximum processing time and transition time until we arrive at the required state B. If this sum is less than duration of D_{DR}, it is obvious that the DR requested can be accepted. ■

From phase 3 above, the transition path required for DR in the DFSM graph of Figure 4.10 leads to a state maximizing the production rate by searching for a shortest path to the state. As an example, let us consider the case of a production line with three machines represented by a DFSM in Figure 4.7. During normal production periods, the system is at state $(111)_2$ (i.e., current state) and the system's total power consumption is 600kW. If DR asks for a maximum power consumption of 400kW, which corresponds to state $(101)_2$, the production line system will follow, according to Algorithm 2, the path $(111)_2$ with state transition $\bar{I}$ of duration T_a reaching state $(011)_2$ where M_1 will not be working. Then after a state transition I of duration T_a, we reach at state $(101)_2$ where M_2 will not be working. The state is still valid for the power threshold of 400kW. A production cycle satisfying maximum power 400kW can indeed start after DR, i.e., the cycle is $(101)_2 \rightarrow (010)_2 \rightarrow (101)_2, \ldots$. Notice that it is the production cycle that maximizes the production rate. The total duration for answering DR to reach this production cycle is then $\max(C_1, C_2, C_3) + T_a + \max(C_2, C_3) + T_a$ that should be less than or equal to duration of D_{DR} from Lemma 4.7, where C_i is processing time of machine M_i, $i = 1, 2, 3$.

In the above sections, we discussed the two works: modeling of the SPL system using a queuing model and DR scheduling under constraints of available power and production rate. In the following section, we provide numerical results.

4.5 Analytical and Simulation Results

In this section, we present our analytical and simulation results of both modeling of SPL and DR scheduling under available power constraint. As a study model,

we consider the example shown in Figure 4.2 which has 13 virtual cells and 4 machines. For processing times (C_i) of the machines, we take values of 3s, 6s, 9s and 12s (s = seconds) for machine 1, 2 , 3 and 4, respectively. Since there is no restrictions on these values, any other values can be taken. We assume that each machine consumes 100kW when it processes a task and 0kW otherwise. We also assume that the SPL system starts working at time $t = 0$. We commence with results on SPL modeling in the following section.

4.5.1 Results on SPL modeling

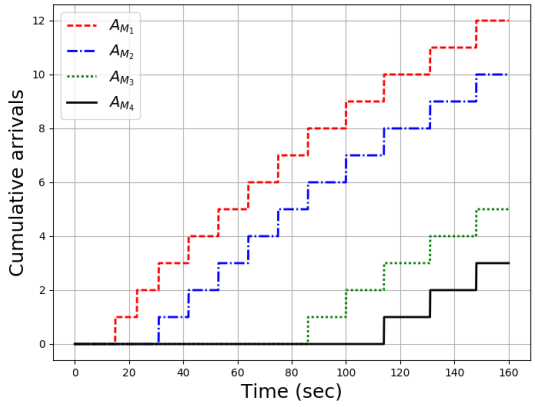
As discussed in section 4.3, we relied on queuing theories to model the SPL system with deterministic inter-arrival and service times. The $D/D/1$ queuing model was used to characterize arrivals and departures of parts/tasks in the SPL system. This section presents analytical and simulation results of the system.

4.5.1.1 Analytical results

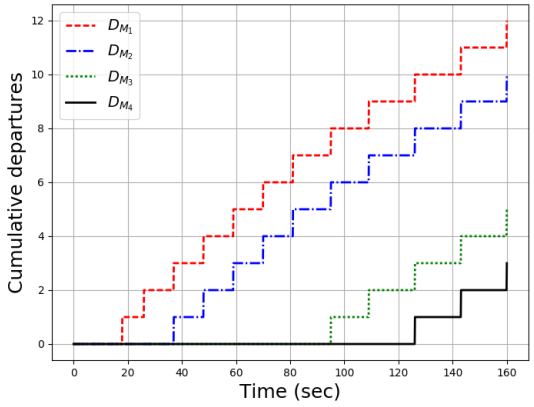
To analyze the SPL system, we used python programming language to implement Equation (4.8) and (4.10) which represent the arrivals and departures processes at the machines, respectively. Based on these processes, the utilization and effective utilization functions (Equation (4.13) and (4.11)) are also implemented in python. Moreover, analytical power consumption function $P_{M_i}(\cdot)$, for machine M_i, $i = 1$, 2, ..., M, is also computed by using Equation (4.14). Then, we compare it with a real power consumption of a typical production process of a factory taken from METRONLab server.

Figure 4.11.1 shows cumulative arrivals of tasks to the 4 machines. From the figure, we can see that the arrival of the 1st task to machine M_1 at time $t = 15$s initializes the SPL system in a transient phase. As soon as the 1st task arrives at machine M_2, the two machines (M_1 and M_2) synchronize themselves. This continues until the 1st task reaches machine M_4 in which all the machines are synchronized to the one with the maximum processing time (i.e., C_4). Furthermore, the arrival of the 1st task at M_4 at $t = 114$s signals the end of the transient phase. Afterwards, the SPL system is in a steady state. The cumulative number of departed tasks from the machines are shown in Figure 4.11.2. In the steady state, the departures from all of the machines occur at the same time.

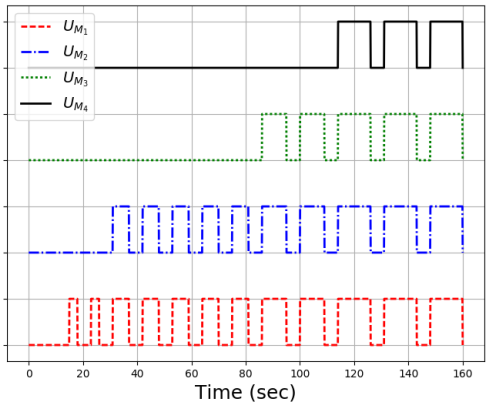
In Figure 4.11.3, the difference between the cumulative number of arrivals and departures (i.e., the utilization) of all the machines are shown. For clearer view, we added some constant value to each $U_{M_i}(\cdot)$ so that the curves are moved up the y-axis. However, the values of $U_{M_i}(\cdot)$ is either 1 or 0 because only one task is processed at any time because each machine is considered as a single server. Since the utilization functions are derived from the arrival and departure processes, the figure can also show both the transient and steady states. However, the utilization function cannot differentiate between the two modes of a machine (*working* or



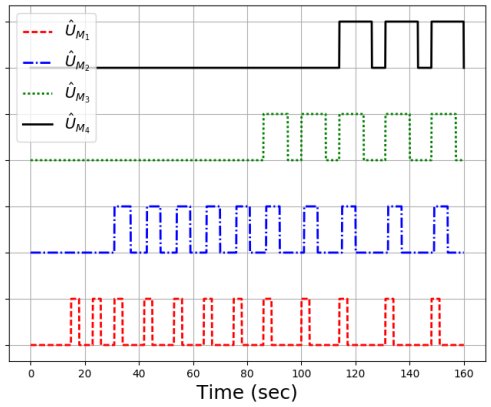
4.11.1: Arrival processes using Equation (4.8)



4.11.2: Departure processes using Equation (4.10)



4.11.3: Utilization using Equation (4.13)



4.11.4: Effective utilization using Equation (4.11)

Figure 4.11 – Arrival and departure processes, utilization and effective utilization of the four machines with processing time of 3s, 6s, 9s and 12s for machine 1, 2, 3 and 4, respectively.

idle) and it cannot show us the power consuming mode of the machines. Hence, we use the effective utilization functions as depicted in Figure 4.11.4. The figure shows that the machine start processing tasks at the same time but don't finish at the same as their processing times are different. The *idle* modes are effectively suppressed and only the *working* modes are prevailed. This is important because we can multiply the effective utilization function of a machine by its power consumption to get a power consumption curve. For example, Figure 4.12 shows such a power consumption curve for machine M_2 when it consumes 300kW while processing a task. This value is taken from a real production process power consumption trace shown in Figure 4.13. As can be seen from Figure 4.12 and 4.13, there is a small difference between the approximated power curve (i.e., based

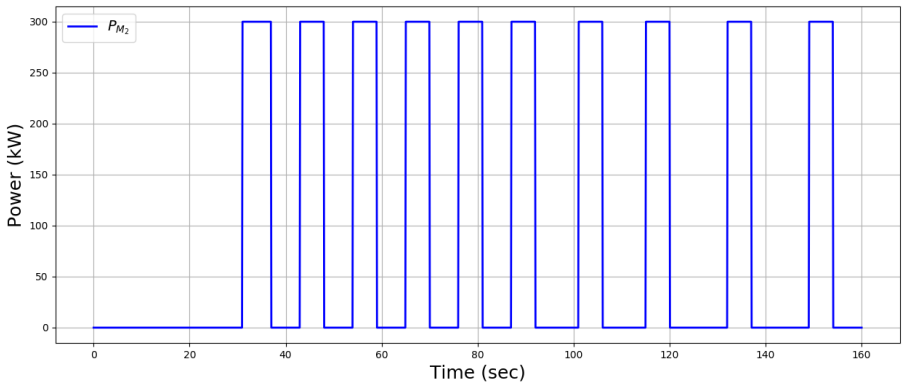


Figure 4.12 – Analytical power consumption of machine M_2 using Equation (4.14) which is derived from the effective utilization function.

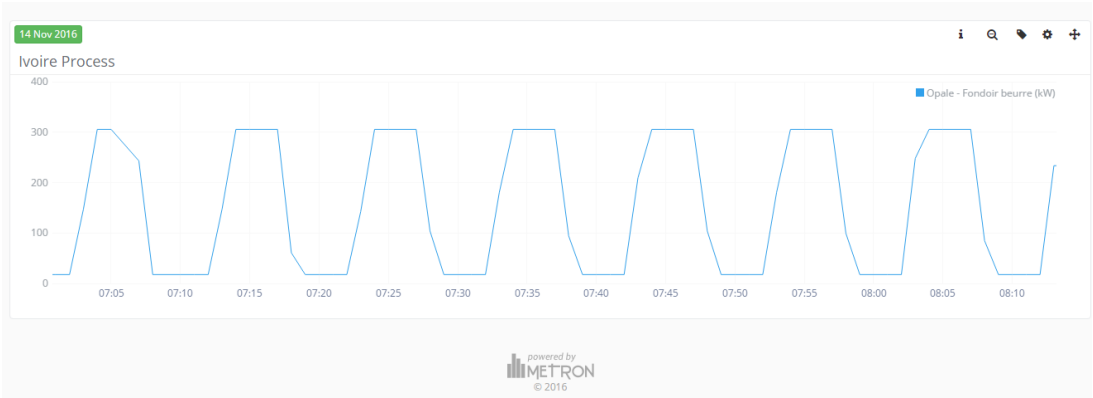


Figure 4.13 – A real power consumption of a typical production process from METRONLab server.

on the queuing model) and the real power curve. This happens when a machine changes its mode from *idle* to *working* and vice-versa. The difference can be corrected by adding a positive slope when the mode changes from *idle* to *working* and a negative slope when the mode changes from *working* to *idle*. The slopes can be inferred from the real power consumption curve.

4.5.1.2 Simulation results

To verify our analytical works, we use an **O**bjective **M**odular **Net**work **T**estbed in C++ (OMNET++) [OMN] simulator which is a modular, component-based C++ simulation library and framework, primarily used for building network simulators. It has been widely used as a modeling tool in applications such as communication networks, distributed or parallel systems, etc [Var01]. In our work, we modified source codes of OMNET++ so that it fits to our objective of modeling the SPL system using a queuing library in OMNET++. In the

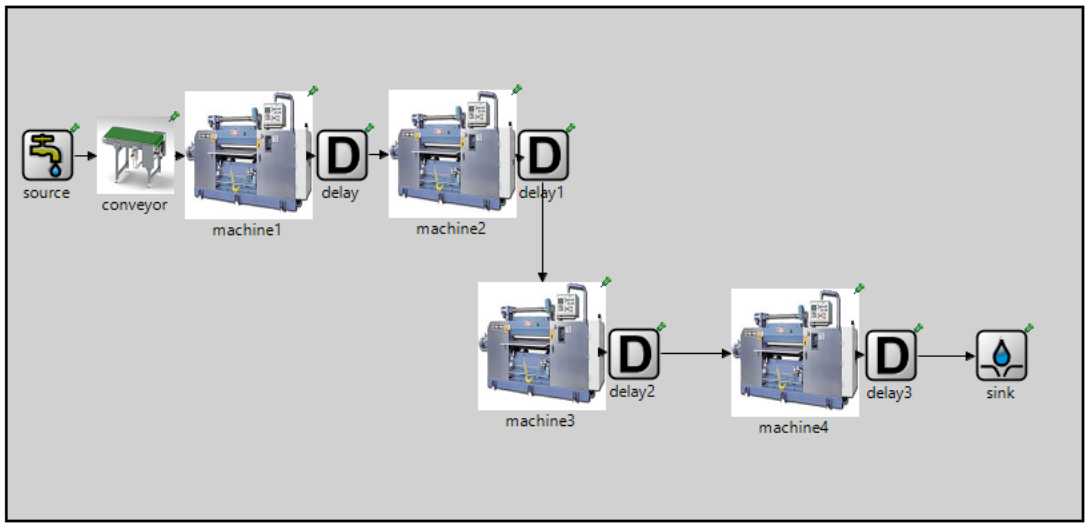


Figure 4.14 – OMNET++ network description of the SPL system with 4 machines.

queuing library, OMNET++ contains modules such as source, sink, delay, queue, etc to describe a specific topology that users want to simulate. Descriptions of the modules and use cases are provided in the website of [OMN]. However, these modules could not satisfy our needs to simulate the SPL system in the OMNET++ simulator. Hence, we created *conveyor* and *machine* modules in OMNET++ by modifying the C++ source codes of the queuing library.

As shown in Figure 4.14, we use 5 different OMNET++ module types to describe the SPL system. The modules are given below:

- The *source* module is used to generate the tasks labeled from 1 to n,
- The *machine* module is used to represent a machine. The source codes of a *queue* module in queuing library is modified for this purpose,
- The *conveyor* module is synchronizing the machines by realizing the tasks at a specific time,
- The *delay* module is used to keep (or delay) a task until a downstream machine is available.
- The *sink* module can be viewed as a storage for finished tasks.

After creating the necessary modules, a **NE**twork **D**escription (NED) file is created by drag and drop of the modules. The final NED graph is shown in Figure 4.14. The next step is to run the OMNET++ program so that live task transfers between the machines can be viewed. Figure 4.15 shows when the SPL system is running in OMNET++. After the source module releases tasks (marked

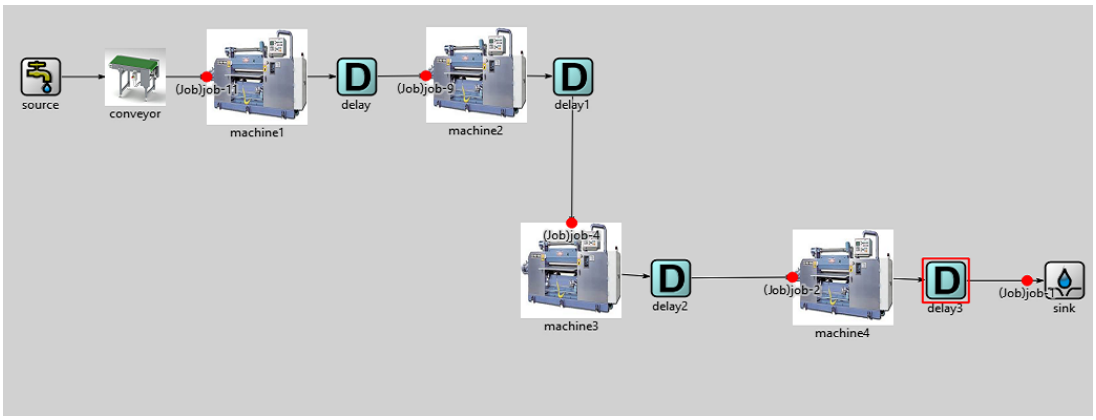


Figure 4.15 – The OMNET++-based SPL system in action.

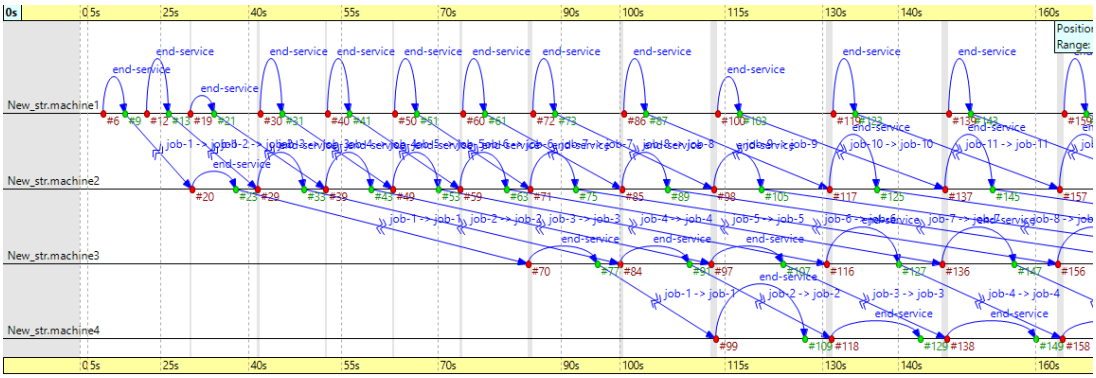


Figure 4.16 – OMNET++ time line graph for the SPL system.

by red dot), the tasks pass under each machine until the sink module collects them. From the OMNET++ simulator, we can also see the arrival and departure times of the tasks to and from the machine as depicted in Figure 4.16. From the figure, we notice that the system synchronizes when the 1st task arrives at machine M_4. This is in accordance with the one that we have achieved using the analytical effective utilization function shown in Figure 4.11.4.

4.5.2 Result on DR scheduling

In section 4.4, we provided details of the our DR scheduling algorithm that adapts power consumption of the SPL system to available power and production rate constraints. We used different illustrating examples and described the p participles on how the algorithm works. In this section, we use the exemplary SPL system configuration shown in Figure 4.2 with 4 machines and 13 virtual cells to compute prefixes and sub-words of schedules. As describe in the above section, processing times (C_i) are set to values 3s, 6s, 9s and 12s for machine 1, 2, 3 and 4, respectively. Now, we assume that each machine consumes 100kW when it processes a task (but this value can be any amount). Furthermore, the

Table 4.2 – Prefix and sub-word when DR threshold is 100kW.

State	Symbol	max. power conso. (kW)	maxC_i (sec)
1111111111111	.	400	12
0111111111111	$\overline{I}$	400	12
0011111111111	$\overline{I}$	400	12
0001111111111	$\overline{I}$	400	12
0000111111111	$\overline{I}$	300	12
0000011111111	$\overline{I}$	300	12
0000001111111	$\overline{I}$	200	12
0000000111111	$\overline{I}$	200	12
0000000011111	$\overline{I}$	200	12
0000000001111	$\overline{I}$	200	12
0000000000111	$\overline{I}$	200	12
0000000000011	$\overline{I}$	100	12
1000000000001	I	100	12
1100000000000	I	0	0
0110000000000	$\overline{I}$	0	0
0011000000000	$\overline{I}$	100	3
0001100000000	$\overline{I}$	100	3
0000110000000	$\overline{I}$	100	6
0000011000000	$\overline{I}$	100	6
0000001100000	$\overline{I}$	0	0
0000000110000	$\overline{I}$	0	0
0000000011000	$\overline{I}$	0	0
0000000001100	$\overline{I}$	100	9
0000000000110	$\overline{I}$	100	9

available power levels for DR (i.e., DR thresholds) are considered to be 400, 300, 200, 100, and 50kW. These values are used to find the schedule words in the next section. We also investigate the monetary gain of accepting DR request positively in section 4.5.2.2.

4.5.2.1 Finding schedule words

Based on Lemma 4.6, when the available power is 50kW, there is no feasible schedule as it is not enough even for a single machine. The next threshold is when the available power is 100kW in which a simple feasible schedule exists. Table 4.2 shows the resulting symbols of a schedule that satisfies the DR threshold of 100kW. In the table, each row of the 1st column has length of 13 bits which correspond to the virtual cells (1 if there is a task in a virtual cell or 0 otherwise).

Table 4.3 – Prefix and sub-word when DR threshold is 300kW.

State	Symbol	max. power conso. (kW)	maxC_i (sec)
1111111111111	.	400	12
0111111111111	$\overline{I}$	400	12
0011111111111	$\overline{I}$	400	12
1001111111111	I	400	12
0100111111111	$\overline{I}$	300	12
0010011111111	$\overline{I}$	300	12
1001001111111	I	300	12
1100100111111	I	200	12
1110010011111	I	300	12
1111001001111	I	300	12
1111100100111	I	300	12
1111110010011	I	300	12
1111111001001	I	300	12
1111111100100	I	300	9
0111111110010	$\overline{I}$	200	6
0011111111001	$\overline{I}$	300	12
1001111111100	I	300	9
0100111111110	$\overline{I}$	200	9

Machine 1, 2, 3 and 4 are on virtual cell 4, 6, 11 and 13, respectively (see Figure 4.2). Hence, the values of these virtual cells are required to compute the system's total power consumption (3$^{\text{rd}}$ column) and the maximum processing times among the machines (4$^{\text{th}}$ column). The 2$^{\text{nd}}$ column represent an action of inserting (I) or not ($\overline{I}$) a task at the conveyor input.

From Table 4.2, we can easily find the prefix and the sub-word of the schedule when the available power is 100kW. The prefix (path to optimal state) is $\overline{I}^8$, i.e., we arrive at this state by not inserting tasks for 8 times or taking right successors 8 times from the initial state $(11...1)_2$. Then, the sub-word is $(\overline{I}I^2\overline{I}^{10})$ (gray colored rows) which repeats until the end of the DR interval. During this interval, the peak power consumption is 100kW, i.e., only one machine is working at a time. In the sub-word, the number of Is determines the production rate per sub-word's total duration. The duration can be computed from the 4$^{\text{th}}$ column of the table by summing up the processing times (12 + 12 + ... + 9 = 60s) at each state in the DR interval and the transition times (12*5 = 60s). Hence, the production rate in the sub-word is 2 items per 120 seconds (1 item per minute). However, under normal conditions, the production rate is 13 items per 216 seconds (3.6 items per minute).

To see the effect of less strict DR constraint than the above case, we consider another case where the available power is 300kW. Table 4.3 shows results under the given DR constraint. In this case, the length of prefix ($\overline{I}^2 I \overline{I}^2$) is shorter than the prefix of the above case. It means that we arrived at the optimal state with few transitions. The sub-word is found to be ($\overline{I} I^8 \overline{I}^2 I \overline{I}$). Referring to the table, the number of Is in the sub-word is 9 and their total duration is 165s (12*9 + 9*3 + 6 + 5*12). The resulting production rate becomes 9 items per 2.75 minutes (approx. 3 items per minute). In this case, the number of Is in sub-word is higher than that of the above case and is in accordance with Lemma 4.4 that compare two schedule words based on the number of Is for higher production rate.

4.5.2.2 Monetary gains of accepting DR

Based on our discussion in section 2.3.1.2 of chapter 2, incentive-based DR programs pay customers for not consuming power (or consuming less) in DR intervals. Hence, customers can benefit from monetary gain for accepting the DR positively. The monetary gain can be formulated as follows:

$$G_{money} = \left(N_{norm} - N_{DR}\right) * Pr_{unit} * In_{pct} \tag{4.22}$$

where N_{norm} is number of items produced during normal periods, N_{DR} is number of items produced during DR periods, Pr_{unit} is unit price of an item and In_{pct} is percentage of incentive for the items that are not produced due to DR.

Now, we use Equation (4.22) to calculate a monetary gain for positively answering a DR request. Assume that a factory could receive 20% as an incentive for not consuming power to produce items using the default schedule ((I^{13})^n, n >0). In the default schedule, the factory produces 13 items. As a first case, we consider that the available power is 100kW and the corresponding number of produced items are 2. That is, only two items are produced instead of 13 items and the factory gets paid for the remaining 11 items that are not produced. Taking 50€ per item, the monetary gain is found to be G_{money} = (13 - 2)*0.2*50 = 110€.

Considering an available power of 300kW where the number of produced items are 9, the monetary gain could be G_{money} = (13 - 9)*0.2*50 = 40€. This shows that the less the factory consumes the more money paid for not consuming power and vice-versa. If the factory owns some **R**enewable **E**nergy **S**ources (RESs) in its local premises, this will help it benefit more from the DR mechanism. Assume that the RESs generate power in order of 100kW. That means, the total available power is 400kW that is enough for all the machines. In this way, the number of items produced is 13 which is the same with the default schedule. Besides producing at full capacity, the factory can save 40€. In conclusion, the benefits should be interesting for the factory to participate in DR programs.

In the above sections, we discussed analytical and simulation results of our works on modeling of an SPL system and DR scheduling. In the next section, we details our experiments with the DR protocol: OpenADR.

4.6 Experimentation with OpenADR

Based on the description of OpenADR in section 2.3.4 of chapter 2, the objective of OpenADR is to facilitate information exchange between electricity service providers, aggregators and energy consumers. In this section, we provide the experiments that are done in METRON for the purpose of integrating OpenADR to METRONLab platform and testing OpenADR on a real testbed. Setup procedures of the testbed is given below.

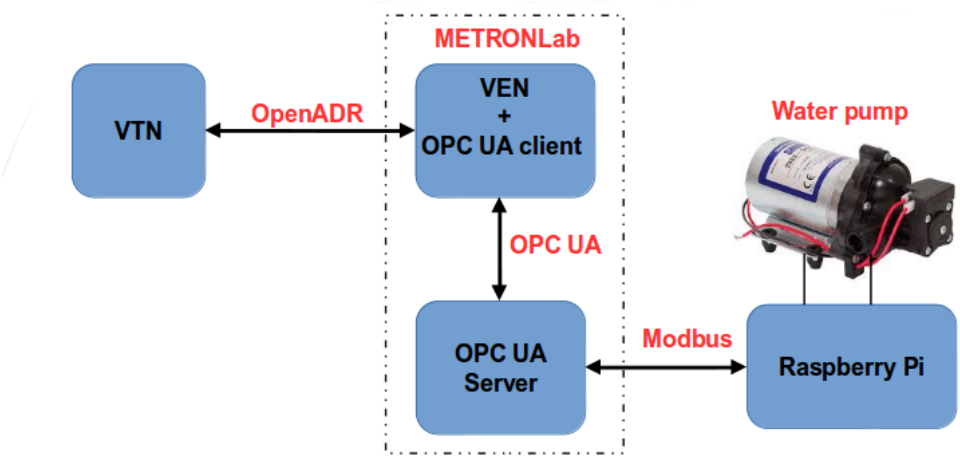


Figure 4.17 – Testbed for implementing OpenADR.

4.6.1 Testbed setup

Figure 4.17 shows our testbed that is used to implement the OpenADR protocol. The testbed contains two parts: hardware and software parts.

4.6.1.1 Hardware

The hardware part of the testbed consists of:

- *Personal computer (PC)*: a PC with Ubuntu 16.04 is used as a **V**irtual **T**op **N**ode (VTN). It emits DR requests as per users requirements,
- *METRONLab*: the METRONLab device hosts the **V**irtual **E**nd **N**ode (VEN). The VEN processes DR requests of the VTN,

- *Water pump*: the pump is used as a controllable load, i.e., it can be turned ON or OFF based on DR requests,

- *Raspberry Pi*: the raspberry pi (Model B) board receives a command from the METRONLab and controls the pump through a relay. A relay module can be interfaced to the pi board by soldering the module on the General Purpose Input/Output (GPIO) pins.

4.6.1.2 Software

In the software part, the following software are used in the experiment:

- *OpenADR source code*: VTN and VEN source codes are available in EnerNOC's GitHub [Ene]. Since the source code of EnerNOC only provide the *EiEvent* service, major modifications have been done to implement the rest 7 services of OpenADR described in section 2.3.4 of chapter 2.

- ***X**ML **M**essaging and **P**resence **P**rotocol (XMPP) server*: as an XMPP server, we used an OpenFire [OFR] server which is instant messaging and group chat server. OpenFire accounts (JID - Jabber ID) are created for both VTN and VEN nodes,

- *Modbus TCP (Transmission Control Protocol)*: Modbus [Mod04] is a de-facto standard communication protocol for connecting industrial electronic devices. It is a software originally developed in Schneider Electric. Modbus TCP is used for communications over TCP/IP (Internet Protocol) networks,

- ***O**pen **P**latform **C**ommunications **U**nified **A**rchitecture (OPC UA)*: OPC UA is a machine to machine communication protocol developed for indutrial automation. For more decription of the OPC UA protocol, refer to [LM06]. As shown in Figure 4.17, we included codes of the VEN node together with OPC UA client in METRONLab device.

- *Grails/Groovy*: For web application framework in VTN, we used Grails/-Groovy [GRL] web framework which is used in the OpenADR source code. We extended the Grails/Groovy source code so that a user can view the status of a VEN as a form of *live feed* (see Figure 4.18).

After configuring both the hardware and software parts, a simple web page can be displayed as shown in Figure 4.18. In the right hand side of the figure, a portion is dedicated for showing incoming messages and the sending VENs. The tabs at the center display information including real-time status, available VENs, DR event creation and emitting to VENs, and market context (i.e., available DR programs such as TOU, RTP, etc).

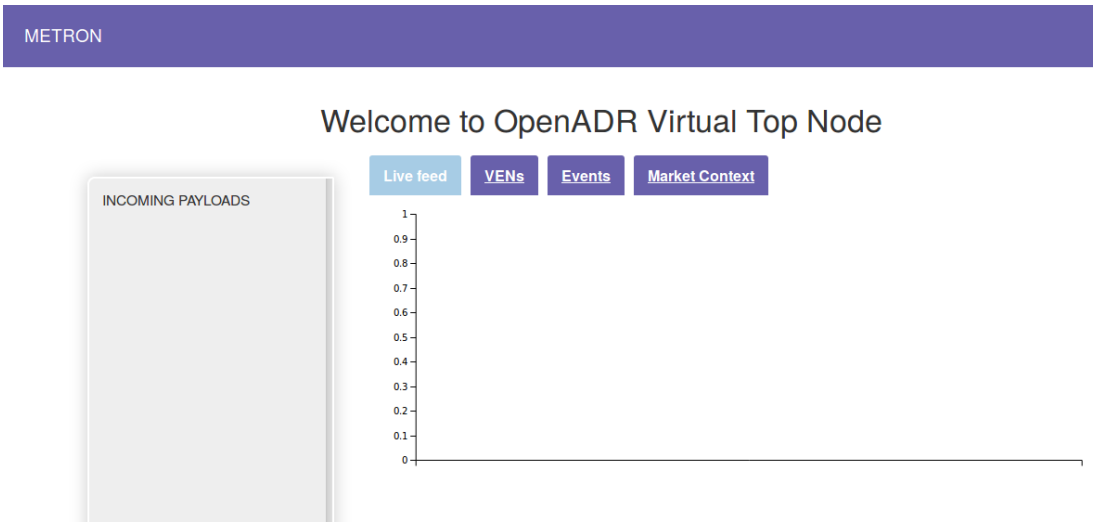


Figure 4.18 – Web interface of the VTN.

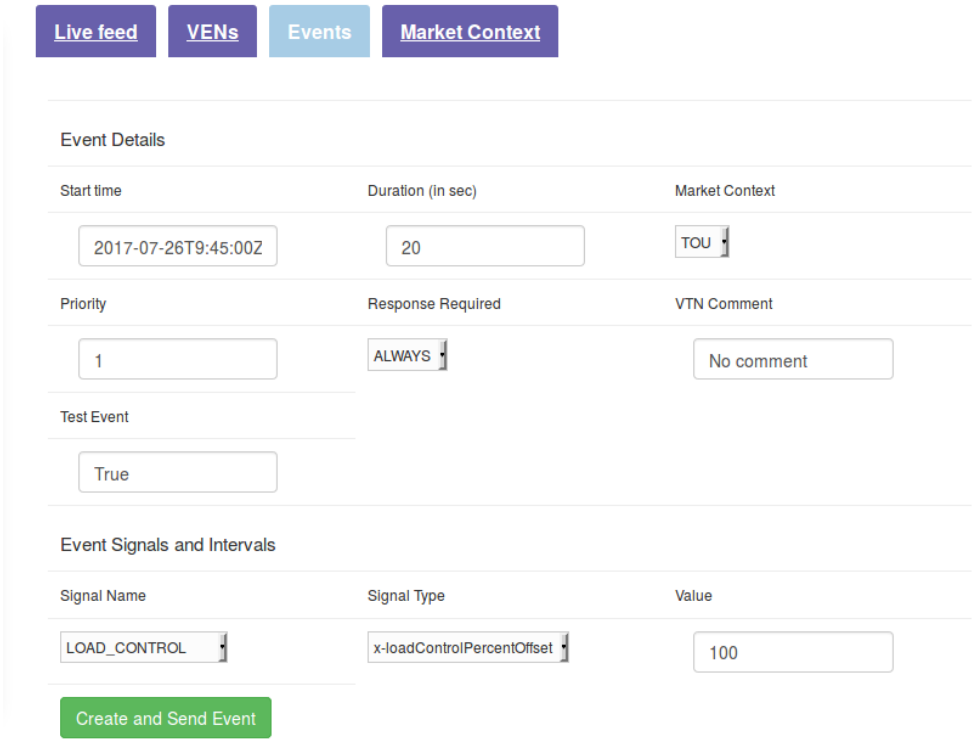


Figure 4.19 – Parameters of a DR event.

4.6.2 Description of a DR event

Referring to Figure 4.19, important DR event parameters include:

- *Start time* is a time in near future when an actual load shed starts. According to OpenADR specification in [All13], the *time* variable is represented by *ZULU* (same as UTC (Coordinated Universal Time)) time format,

- *Duration* is the length of the DR interval (Γ in Figure 4.6),
- *Market Context* is a DR program (e.g., TOU, RTP, etc),
- *Priority* is set to 1 if the event has higher priority than the other events which were emitted and that did not start,
- *Response Required* is set to *ALWAYS* if a response is required for a DR event that a VTN sent to a VEN,
- *Signal Name* is a name of an event signal (e.g., *ELECTRICITY_PRICE*, *LOAD_DISPATCH*, *LOAD_CONTROL*, etc),
- *Signal Type* is an element of a signal name (e.g., *LOAD_CONTROL* has signal types of *x-LoadControlPercentOffset*, *x-LoadControlSetpoint*, etc). [All13] provides the full list of signal names and types,
- *Value* is a corresponding value to the chosen signal name and type.

Figure 4.20 – Real-time status of the pump during normal periods.

4.6.3 Communicating DR events between VTN and VEN

For description of service types to establish initial communication between VTN and VEN, see section 2.3.4.2 of chapter 2. During normal periods, the status of the water pump is ON (or 1) as shown in Figure 4.20. At some time, if the VTN wants to send a DR event to the VEN, it uses *oadrDistributeEvent* payload of *EiEvent* service. The *oadrDistributeEvent* payload specifies all the

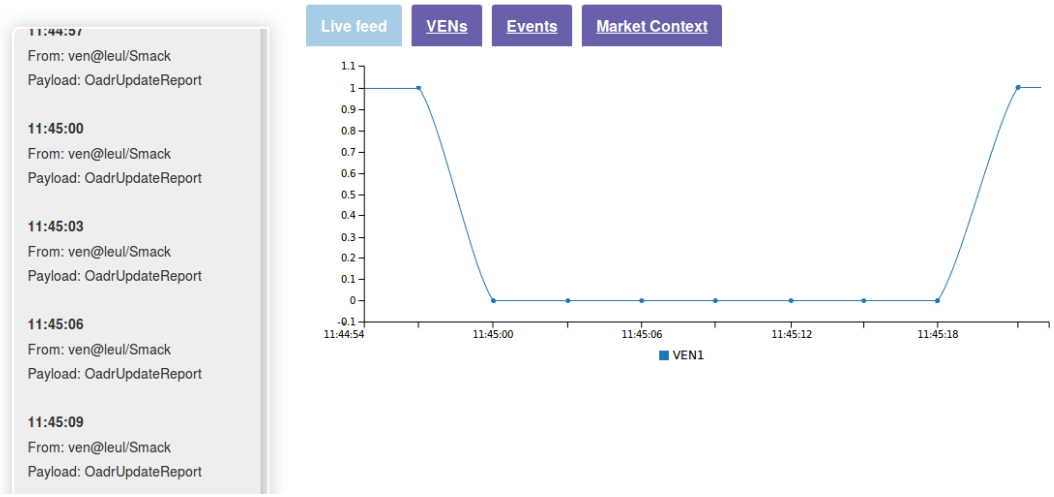


Figure 4.21 – Real-time status of the pump during DR periods.

event parameters described in the above section. After the VEN receives the event, it replies with *oadrCreatedEvent* payload of *EiEvent* service. As shown in Figure 4.21, the pump has changed its status from ON to OFF (or 0) for the specified duration (20 seconds) to respect the DR constraint.

4.7 Summary

This chapter presented two of our works on the demand side of the industrial microgrid together with experiments of a **D**emand **R**esponse (DR) protocol. In section 4.3, we detailed modeling of a **S**ynchronous **P**roduction **L**ine (SPL) based on $D/D/1$ queuing model. The SPL system was represented by a tandem of $D/D/1$ queues. It was used to characterize temporal properties such as arrival and departure processes, utilization and effective utilization of the SPL system. Analytical and simulation results were also given for a specific SPL configuration.

Then, section 4.4 discussed our DR scheduling algorithm that adapts power consumption of the SPL system to constraints of production rate and available power. The algorithm was coded using **D**eterministic **F**inite **S**tate **M**achine (DFSM) in which state transitions happen by inserting a task or not. Finding an optimal schedule under DR constraint has followed three phases: *existence of a schedule*, *finding optimal schedule* and *shortest path to optimal schedule*. We showed different results of the scheduling algorithm using illustrative examples.

Finally, we provided our experiments with a DR protocol called **Open** **A**utomated **D**emand **R**esponse (OpenADR) in section 4.6. For this, we setup a testbed that contains devices such as water pump, raspberry pi and METRONLab.

The following chapter summarizes the works done in thesis by mentioning important points. It also highlights perspectives of the thesis.

Part III

Conclusions and Perspectives

Chapter 5

Conclusions and Perspectives

Contents

5.1 Conclusions

In this thesis, we have addressed energy management challenges in the context of industrial microgrid. The challenges can be categorized into the two sides of microgrid energy management:

- In the supply side:
  - We proposed a mathematical framework that is used to model power generations of **D**istributed **E**nergy **R**esources (DERs) so that the power sources can be seamlessly integrated into the microgrid. Moreover, we also proposed cost minimization strategies that enable us choose from an energy mix (solar, wind, battery, utility grid) which reduces total energy costs,
  - To reduce the inherent power fluctuation behavior of **R**enewable **E**nergy **S**ources (RESs), an approach of smoothing their power production was also proposed. Its performance comparisons have been done against other smoothing algorithms.
- In the demand side:
  - A queuing theory-based mathematical model was proposed that provides temporal characterization of a **S**ynchronous **P**roduction **L**ine (SPL) such as arrival and departure processes,

– We proposed a **D**emand **R**esponse (DR) scheduling algorithm which addresses how an SPL system adapts its power consumption to constraints such as available power and production rate.

Figure 5.1 summarizes the works in this thesis with corresponding publications. The following sections provide more insights to the works in both supply and demand sides of microgrid energy management.

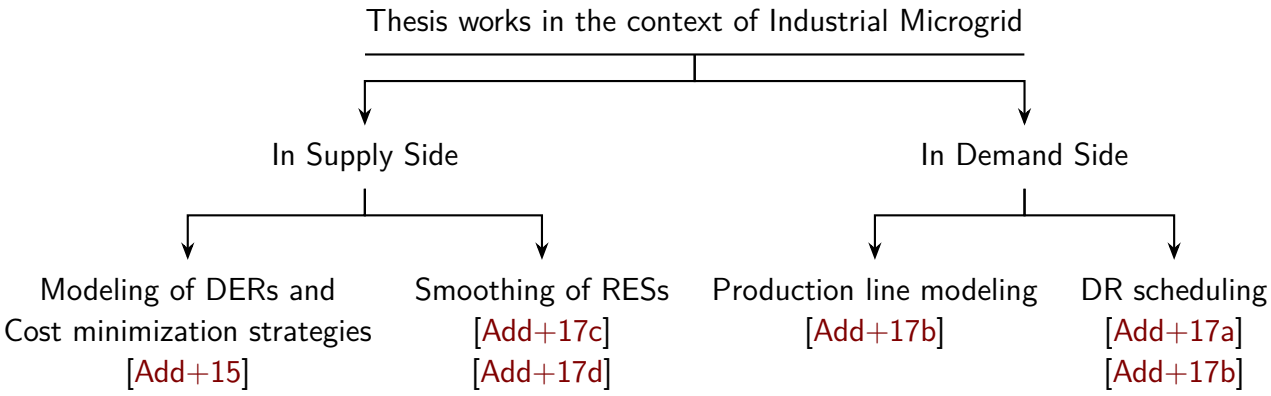


Figure 5.1 – Summary of our works in this thesis.

5.1.1 Modeling of DERs and cost minimization strategies

We discussed modeling of DERs in the supply side of the industrial microgrid in section 3.3 of chapter 3. The following questions can be inferred from the section:

- What is a minimum power produced by DERs? How can we model this value using a mathematical framework?
- When energy demand is greater than energy supply, can we buy energy from spot markets, utility grid, or discharge from **B**attery **E**nergy **S**torage **S**ystems (BESSs)? What happens if supply is greater than demand?

A service curve concept of **N**etwork **C**alculus (NC) was proposed for responding to the above questions. We used service curves to model the minimum power the DERs can provide. To obtain total power production, we aggregate the service curves of the DERs. After that, a balance should be reached between energy demand and supply. For cases of imbalance (i.e., when supply is greater than demand or vice-versa), we proposed different strategies that minimize energy procurement costs. The strategies were *sell surplus energy*, *store excess energy* and *use external energy to charge battery*. To compare the performance of the strategies, we used real power consumption data of a factory located in France. Our results showed that interesting saving can be made by adopting the strategies.

5.1.2 Smoothing of RESs

In section 3.4 of chapter 3, we focused on smoothing of RESs to reduce power fluctuations in conjunction with BESSs. The following questions were raised:

- How to smooth power production curves of RESs and how to measure the smoothness of the curves?
- What battery size is required so that a production curve for a day-ahead forecast period can be guaranteed?
- From regulatory aspect, how much power differences between any two successive time slots are tolerated?

To deal with the above questions, we proposed a Gaussian-based smoothing algorithm. We compared its performances by bench-marking classical approaches such as **S**imple **M**oving **A**verage (SMA) and **E**xponential **M**oving **A**verage (EMA) for the purpose of smoothing out power fluctuations of RESs. The performances of the algorithms were showed based on real datasets and considered two cases: with and without power level constraints. In both cases, the Gaussian-based algorithm performed better than both SMA and EMA when smoothness measures and battery sizes are taken to account.

5.1.3 A queue theory-based model of a production line

We discussed modeling of an SPL system using a queuing model in section 4.3 of chapter 4.4. It could give an interesting solution to the following inquiries:

- How to model temporal behaviors of an SPL system such as arrival and departure processes to/from a machine? Which model to use?
- Relationship between effective utilization functions and real power consumption of a machine?

Based on the description of an SPL system in section 2.2.3 of chapter 2, we modeled each machine in SPL as $D/D/1$ queue where each queue has a single server (which is the machine itself) with **F**irst **I**n **F**irst **O**ut (FIFO) service disciple, deterministic inter-arrival and service times. Hence, the SPL system is considered as a tandem of $D/D/1$ queues that can be synchronized by a conveyor. After setting up the system, we defined temporal behaviors of the machines in the form of arrival and departure processes and utilization functions. Then, the effective utilization functions can be used predict the behavior of a machine in temporal domain and it can help us know the status of a machine (either *working* or *idle*).

To validate mathematical computations, we showed analytical and simulation results in section 4.5.1 of chapter 4. For analytical results, we used python

programming language to implement different equations. For simulation results, **O**bjective **M**odular **Ne**twork **T**estbed in C++ (OMNET++) simulation software was used to simulate the SPL system with 13 virtual cells and 4 machines.

5.1.4 DR scheduling in a production line

In section 4.4 of chapter 4.4, we presented our works on a DR scheduling algorithm which can respond to the following questions:

- How can we effectively schedule the SPL system considering available power and production rate constraints?
- What are the conditions for existence of feasible schedules?
- What are the conditions for accepting a DR request positively?
- If we accept a DR request positively, how much could we gain in terms of monetary values?

To respond to the above questions, we proposed a novel scheduling algorithm that adapts the power consumption of the SPL system to available power under DR. The algorithms was coded in **D**eterministic **F**inite **S**tate **M**achine (DFSM). The term *deterministic* refers to a deterministic action of either inserting a task or not at a conveyor input in the SPL system. Through multiple illustrating examples, we not only studied feasibility conditions for existence of schedules but also showed DR acceptance conditions. Besides these conditions, we setup a procedure on how to find optimal schedules under DR constraints. We have also investigated the monetary gain of accepting DR requests and showed analytical results of a specific SPL system configuration with 4 machines.

Alongside our DR scheduling algorithm, we also presented a use case of **Open** **A**utomated **D**emand **R**esponse (OpenADR) protocol which is a DR protocol for communicating DR signals. In OpenADR, a **V**irtual **T**op **N**ode (VTN) (e.g., utility grid) sends a DR requests to **V**irtual **E**nd **N**ode (VEN) (e.g., electricity customers) due to surge in total power consumption of the VTN's premises. A testbed is setup in METRON for implementing OpenADR and making it an integral part of METRONLab platform.

5.2 Perspectives

We are sure that this thesis has opened up a lot of interesting questions and further improvements can be investigated. As usual, we divide a non-exhaustive list of perspectives into supply and demand sides as follows:

- In supply side:

  - Since the service curve models of DERs are offline approach, extending it to an online approach is conceivable. In the online approach, the system can adapt to real-time supply and demand,
  - Regarding the smoothing algorithms (Gaussian-based, SMA and EMA), we considered fixed smoothing parameters. An interesting extension to this aspect could be finding optimal smoothing parameters given battery size and *allowable* range of power fluctuations,
  - Multiple days-ahead smoothing and energy losses due to conversion could be interesting to incorporate into the smoothing algorithms.

- In demand side:

  - Our queuing theory-based model can be extended to model other manufacturing systems described in section 2.2.3 of chapter 2. Furthermore, another queuing model could also be provisioned based on working principles of the manufacturing systems,
  - Our DFSM-based DR scheduling algorithm can be enhanced by further studies and possibly extended to handle different manufacturing systems,
  - Another extension to our DR scheduling algorithm could be to add some probabilities in the DFSM graph. This can handle a case where the machines are unreliable and could fail with a given probability. We have started working on it for while but cannot add to thesis manuscript due to time limitations.
  - As given in our work [Add+17a], a mixed-criticality approach (refer to [Ves07] and [BD13]) is an interesting aspect to investigate how mixed-criticality scheduling principles can be used to address DR scheduling in production lines. Hence, it requires further study into the problem formulation.

Collaborative works are also on the horizon that expand the theories and practices of research developed during this thesis to other application areas:

- To further expand the scope of our DR scheduling algorithm, we are investigating other domains where the algorithm can be applied. Application of the algorithms to energy efficiency in data centers seems interesting because data centers are one of immense energy consumers,

- After having fruitful discussions with *Dr Rafik Zitouni* (lecturer at Engineering School of Paris - ECE Paris), applications of energy management in DERs powered Road Side Unit (RSU) of Vehicular Networks seems promising.

List of symbols

$\mathbb{Z}$	Integers numbers: $\ldots, -2, -1, 0, 1, 2, \ldots$
$\mathbb{N}$	Natural numbers: $0, 1, 2, \ldots$
$\mathbb{R}$	Real numbers
$\lceil x \rceil$	Ceiling of x
$\lfloor x \rfloor$	Floor of x
$\|x\|$	Absolute value of x ($\|x\| = x$ if $x \geqslant 0$ else -x)
$\|\|x\|\|$	Cardinality of x (i.e., number of elements in set x)
$x \otimes y$	Convolution of functions x and y
$[x, y]$	Interval of real values: $\{a \in \mathbb{R} \| x \leqslant a \leqslant y\}$
$[x, y)$	Half-open interval of real values: $\{a \in \mathbb{R} \| x \leqslant a < y\}$
max	Maximum
min	Minimum
W	Watt (unit of power derived from joule per second)
Wh	Watt-hour (unit of energy derived from joule)
$S_i(t)$	Service curve of energy resource i at time t
α_i	Service curve parameter of energy resource i (in kWh)
β_i	Service curve parameter of energy resource i (in kW)
N	Total number of **R**enewable **E**nergy **S**ources (RESs)
$Pr(t)$	Selling or buying energy price at time t (in €/MWh)
$E_{sell}(t)$	Sold energy volume at time t (in MWh)
$E_{buy}(t)$	Bought energy volume at time t (in MWh)
$D(t)$	Energy demand at time t
$b(t)$	Battery status (stored energy) at time t
η_c	Charge efficiency of a battery
η_d	Discharge efficiency of a battery
B	Rated battery capacity (or maximum capacity)
B_{min}	Minimum battery level in terms of stored energy
$P^c_{bat}(t)$	Charge rate of a battery at time t
$P^d_{bat}(t)$	Discharge rate of a battery at time t
E_{charge}	Charged energy volume to a battery
E_{dis}	Discharged energy volume from a battery
$P_{pv}(t)$	Solar PV power at time t
$P_{wind}(t)$	Wind power at time t
$P_{act}(t)$	Actual power of solar and wind at time t
$P_{sm}(t)$	Smoothed power of solar and wind at time t
Z	Smoothness measure of solar and wind powers
γ	Desired power level difference between successive time slots
T_a	Inter-arrival time of parts/tasks on a conveyor system

n	Sequence number of parts/tasks
$\text{Inst}(n, S_i)$	Arrival instant of part n at machine/station i
C_i	Processing time of machine i
VC_i	Virtual cell at position i
$\text{dist}(VC_i, VC_{i+1})$	Distance between VC_i and VC_{i+1} virtual cell centers
K	Number of virtual cells in a production line
K_i	Position of machine M_i on a conveyor
M	Number of machines in a production line
p_i	Power consumption of machine i when processing a task
$A_{S_i}(t)$	Total number of arrivals at time t in station i
$D_{S_i}(t)$	Total number of departures at time t in station i
$N_{S_i}(t)$	Effective number of parts at time t in station i
$U_{S_i}(t)$	Utilization of station i at time t
$\hat{U}_{S_i}(t)$	Effective utilization of station i at time t
$P_i(t)$	Power consumption of machine i at time t
L_j	Available power at level j
Δ	Amount of energy to be reduced in **D**emand **R**esponse (DR) interval
D_{DR}	Time between DR notification and start of DR
Γ	Duration of DR (or DR interval)

Glossaries

Acronyms

ARIMA

***A**uto **R**egressive **I**ntegrated **M**oving **A**verage*. 60, 129,
—
Glossary: ARIMA

BAS

***B**locking **A**fter **S**ervice*. 17, 83, 129,
—
Glossary: BAS

BESSs

***B**attery **E**nergy **S**torage **S**ystems*. 14–16, 42, 47, 118, 119, 129,
—
Glossary: BESSs

CPP

***C**ritical **P**rice **P**eaking*. 21, 23, 129,
—
Glossary: CPP

DERs

***D**istributed **E**nergy **R**esources*. vii, 3, 7, 8, 10–12, 25, 29, 30, 35, 42, 45–48, 50, 53, 60, 63, 64, 75, 76, 88–90, 117, 118, 121, 129, 131,
—
Glossary: DERs

DFSM

***D**eterministic **F**inite **S**tate **M**achine*. viii, 91, 92, 94–96, 98, 99, 101, 114, 120, 121, 129,
—
Glossary: DFSM

DoD

***D**epth of **D**ischarge*. 15, 16, 50, 62, 129, 132,
—
Glossary: DoD

DR

***D**emand **R**esponse*. vii, viii, xiii, 4, 6–8, 10, 16, 19–21, 23–28, 42, 78–81, 87–90, 92–101, 106–114, 118, 120, 121, 124, 129, 130, 132, 133,
—
Glossary: DR

DSM

***D**emand **S**ide **M**anagement*. vii, 6, 19, 78, 130,
— Glossary: DSM

EE

***E**nergy **E**fficiency*. vii, 6, 19, 20, 78, 80, 130,
— Glossary: EE

EMA

***E**xponential **M**oving **A**verage*. xiv, 47, 48, 55–57, 69–72, 75, 76, 119, 121, 130,
— Glossary: EMA

ESSs

***E**nergy **S**torage **S**ystems*. vii, 5–8, 10–12, 14, 47, 130,
— Glossary: ESSs

FIFO

***F**irst **I**n **F**irst **O**ut*. 38, 42, 82, 119, 130,
— Glossary: FIFO

IBR

***I**nclining **B**lock **R**ate*. xiii, 21, 22, 130,
— Glossary: IBR

IID

***I**ndependent and **I**dentically **D**istributed*. 37–39, 131,
— Glossary: IID

MPP

***M**aximum **P**ower **P**oint*. 13, 131,
— Glossary: MPP

MPPT

***M**aximum **P**ower **P**oint **T**racking*. 14, 131,
— Glossary: MPPT

NC

***N**etwork **C**alculus*. vii, 7, 8, 10, 29, 30, 36, 42, 46, 47, 50, 118, 131,
— Glossary: NC

NED

NE*twork* ***D****escription*. 105, 131,
— Glossary: NED

OMNET++

O*bjective* ***M****odular* ***Ne****twork* ***T****estbed in C++*. viii, 7, 104–106, 120, 131,
— Glossary: OMNET++

OPC UA

O*pen* ***P****latform* ***C****ommunications* ***U****nified* ***A****rchitecture*. 111, 131,
— Glossary: OPC UA

OpenADR

Open ***A****utomated* ***D****emand* ***R****esponse*. viii, 8, 20, 25–29, 42, 78, 110–112, 114, 120, 132, 133,
— Glossary: OpenADR

PCC

P*oint of* ***C****ommon* ***C****oupling*. 11, 132,
— Glossary: PCC

PV

P*hotovoltaics*. xvii, 5, 10–14, 46, 47, 49, 53–55, 60, 67, 68, 131, 132,
— Glossary: PV

RESs

R*enewable* ***E****nergy* ***S****ources*. 4–6, 8, 14, 25, 46, 47, 53, 57, 75, 109, 117, 119, 123, 129, 132,
— Glossary: RESs

RTP

R*eal-****T****ime* ***P****ricing*. xiii, 21–23, 80, 111, 113, 132,
— Glossary: RTP

SMA

S*imple* ***M****oving* ***A****verage*. xiv, 47, 48, 55–57, 69–72, 75, 76, 119, 121, 132,
— Glossary: SMA

SoC

S*tate of* ***C****harge*. 15, 129, 132,
— Glossary: SoC

SoH

***S**tate of **H**ealth*. 15, 132,
— Glossary: SoH

SPL

***S**ynchronous **P**roduction **L**ine*. 8, 17, 80–83, 86–88, 90–94, 96, 98, 101, 102, 104–106, 110, 114, 117–120, 129, 133,
— Glossary: SPL

TOU

***T**ime **O**f **U**se*. xiii, 21–23, 80, 111, 113, 133,
— Glossary: TOU

VEN

***V**irtual **E**nd **N**ode*. 26–29, 110, 111, 113, 114, 120, 133,
— Glossary: VEN

VTN

***V**irtual **T**op **N**ode*. 26–29, 110, 111, 113, 120, 133,
— Glossary: VTN

XMPP

***X**ML **M**essaging and **P**resence **P**rotocol*. 27, 29, 111, 133,
— Glossary: XMPP

Glossary

ARIMA

Auto **R**egressive **I**ntegrated **M**oving **A**verage (ARIMA) models are applied to time series data in order to better understand the data or perform forecasting of future data points. 125

BAS

In queuing systems, the **B**locking **A**fter **S**ervice (BAS) mechanism allows a job to be blocked upon service completion at a node until a downstream node is available to process the job. 125

BESSs

Battery **E**nergy **S**torage **S**ystems (BESSs) are electrochemical storage technologies. The most common BESSs are lead-acid and lithium-ion batteries. 125

CPP

Critical **P**rice **P**eaking (CPP) is a price-based **D**emand **R**esponse (DR) mechanism in which prices for electricity during a specified time period is substantially raised due to utilities' anticipation of high wholesale market prices or power system emergency conditions. 125

DERs

Distributed **E**nergy **R**esources (DERs) are not limited to generations of electricity by **R**enewable **E**nergy **S**ources (RESs) but also include a device to store energy. 125

DFSM

A **D**eterministic **F**inite **S**tate **M**achine (DFSM) is a deterministic version of FSM (Finite State Machine) which is a mathematical model of computation. In DFSM, state change is deterministic, i.e., transitions are only occurred by few known conditions. For example, In our **S**ynchronous **P**roduction **L**ine (SPL) model, transitions happen by inserting a task or not at a conveyor. 125

DoD

Depth of **D**ischarge (DoD) describes how deeply the battery is discharged. DoD uses percentage points (100% = empty; 0% = full). The DoD is the complement of **S**tate of **C**harge (SoC): as one increases, the other decreases. 125

DR

DR refers to changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized. 125

DSM

The **D**emand **S**ide **M**anagement (DSM) encompasses systematic utility and government activities designed to change the amount and/or timing of the customer's use of electricity for the collective benefit of the society, the utility and its customers. 126

EE

Energy **E**fficiency (EE) involves technology measures that produce the same or better levels of energy services (e.g., light, space conditioning, motor drive power, etc.) using less energy. The technologies that comprise efficiency measures are generally long-lasting and save energy across all times when the end-use equipment is in operation. Depending on the timing of equipment use, EE measures can also produce significant reductions in peak demand. 126

EMA

An **E**xponential **M**oving **A**verage (EMA), also known as an exponentially weighted moving average (EWMA), is a type of infinite impulse response filter that applies weighting factors which decrease exponentially. That is, it gives more weights to recent values in a sliding window. 126

ESSs

Energy **S**torage **S**ystems (ESSs) capture of energy generated at one time for use at a later time. Common ESSs include mechanical, thermodynamic, electromagnetic, and electrochemical storage systems. 126

FIFO

First **I**n **F**irst **O**ut (FIFO) is a service discipline where customers are served according to their order of arrival to a system (e.g., queuing system). 126

IBR

Inclining **B**lock **R**ate (IBR) divides the electricity price into blocks. The 1^{st} block of electricity is at the lowest price. As the customer purchases more electricity during a month, the electricity bought will eventually fall in the 2^{nd} block which is more expensive than price of the 1^{st} block. 126

IID

Random variables are said to be **I**ndependent and **I**dentically **D**istributed (IID) if each random variable has the same probability distribution as the others and all are mutually independent. 126

MPP

The **M**aximum **P**ower **P**oint (MPP) is a point on a power curve of a **P**hotovoltaics (PV) system that has the highest value of the product of its corresponding voltage and current, or the highest power output. 126

MPPT

Maximum **P**ower **P**oint **T**racking (MPPT) is a technique used commonly with solar PV system to maximize power extraction by tracking the sun. The MPPT finds the MPP and keeps it there. 126

NC

Network **C**alculus (NC) a set of mathematical results which give insights into man-made systems such as concurrent programs, digital circuits and communication networks. It gives a theoretical framework for analysing performance guarantees (e.g., delay and backlog) in computer networks. We used the concepts of service curves in NC to model power production of DERs. 126

NED

NEtwork **D**escription (NED) is the topology description language of **O**bjective **M**odular **Ne**twork **T**estbed in C++ (OMNET++). NED lets users declare simple modules, and connect and assemble them into compound modules. 127

OMNET++

OMNET++ is a modular, component-based C++ simulation library and framework, primarily for building network simulators. It is a discrete event simulator. 127

OPC UA

Open **P**latform **C**ommunications **U**nified **A**rchitecture (OPC UA) is a platform independent service-oriented architecture that allows communication of machine to machine for industrial automation. 127

OpenADR

Open Automated Demand Response (OpenADR) is a communications data model designed to facilitate sending and receiving DR signals from a utility or independent system operator to electric customers. 127

PCC

The **Point of Common Coupling (PCC)** is a point in the electrical system where multiple electrical sources or loads may be connected. For instance, the PCC circuit breaker connect or disconnect a microgrid to/from a utility grid. 127

PV

PV is a term that refers to the conversion of sunlight into electricity using semiconducting materials that exhibit photovoltaic effects. 127

RESs

RESs are naturally replenished energy resources such as solar, wind, waves, etc. 127

RTP

Real-Time Pricing (RTP) gives consumers information about the actual cost of electricity at any given time (every 15 minutes or one hour). It lets consumers adjust their electricity usage accordingly, for example, scheduling usage during periods of low demand to pay cheaper rates. 127

SMA

A **Simple Moving Average (SMA)** is an arithmetic moving average calculated by adding data points inside a sliding window and then dividing this total by the window's length. 127

SoC

SoC describes how full a battery is. The units of SoC are percentage points (0% = empty; 100% = full). The SoC is the complement of DoD: as one increases, the other decreases. 127

SoH

State of Health (SoH) is a *measurement* that reflects the general condition of a battery and its ability to deliver the specified performance compared with initial conditions of a battery. It takes into account such factors as charge acceptance, internal resistance, voltage and self-discharge. 128

SPL

The SPL is a manufacturing system where transfer of parts/tasks are synchronous in which stations/machines start working simultaneously. 128

TOU

Time **O**f **U**se (TOU) tariffs apply different prices for electricity at different times of the day. Mostly, time is divided into *peak* and *off-peak*. During *off-peak* periods electricity prices will be cheaper than *peak* periods. 128

VEN

In OpenADR, **V**irtual **E**nd **N**ode (VEN) acts as a client and it is used to interact with the **V**irtual **T**op **N**ode (VTN). 128

VTN

In OpenADR, VTN acts as a server and it is used to interact with the resources enrolled in DR programs. 128

XMPP

XML **M**essaging and **P**resence **P**rotocol (XMPP) is a communications protocol for message-oriented middleware based on XML (Extensible Markup Language). It enables the near-real-time exchange of structured yet extensible data between any two or more network entities. 128

Author's publication list

Journal papers

EnergyProcedia'17 Alemayehu Addisu, Laurent George, Pierre Courbin, and Vincent Sciandra. "Smoothing of renewable energy generation using Gaussian-based method with power constraints". In: *Energy Procedia* 134 (2017), pages 171–180. DOI: 10.1016/j.egypro.2017.09.555

IJMERR'17 Alemayehu Addisu, Pierre Courbin, Vincent Sciandra, and Laurent George. "Gaussian-Based Smoothing of Wind and Solar Power Productions Using Batteries". In: *International Journal of Mechanical Engineering and Robotics Research* 6.2 (2017), pages 154–159. DOI: 10.18178/ijmerr.6.2.154-159

Conference papers

SmartGridComm'17 Alemayehu Addisu, Hakim Badis, Laurent George, and Pierre Courbin. "An efficient production scheduling based on queuing theory in systems with synchronous part transfer during a demand response event". In: *IEEE International Conference on Smart Grid Communications (SmartGridComm 2017)* 8 (2017), pages 558–564

CSCI'15 Alemayehu Addisu, Laurent George, Pierre Courbin, and Vincent Sciandra. "Modelling of Distributed Energy Resources in Industrial Context Using Service Curves of Network Calculus". In: *Computational Science and Computational Intelligence (CSCI), 2015 International Conference on*. IEEE. 2015, pages 330–335. DOI: 10.1109/CSCI.2015.81

Workshop paper

RTSOPS'17 Alemayehu Addisu, Laurent George, Hakim Badis, and Pierre Courbin. "A Mixed Criticality Approach for Industrial Smart Energy Management and Demand Response". In: *International Real-Time Scheduling Open Problems Seminar (RTSOPS 2017)* 8 (2017), pages 7–8

Bibliography

[AVD94] Kenneth J Adams and Donald R Van Deventer. "Fitting yield curves and forward rate curves with maximum smoothness". In: *The Journal of Fixed Income* 4.1 (1994), pages 52–62.
(Cited on page 57).

[Add+17a] Alemayehu Addisu, Laurent George, Hakim Badis, and Pierre Courbin. "A Mixed Criticality Approach for Industrial Smart Energy Management and Demand Response". In: *International Real-Time Scheduling Open Problems Seminar (RTSOPS 2017)* 8 (2017), pages 7–8.
(Cited on pages 118, 121, 135).

[Add+17b] Alemayehu Addisu, Hakim Badis, Laurent George, and Pierre Courbin. "An efficient production scheduling based on queuing theory in systems with synchronous part transfer during a demand response event". In: *IEEE International Conference on Smart Grid Communications (SmartGridComm 2017)* 8 (2017), pages 558–564.
(Cited on pages 118, 135).

[Add+17c] Alemayehu Addisu, Pierre Courbin, Vincent Sciandra, and Laurent George. "Gaussian-Based Smoothing of Wind and Solar Power Productions Using Batteries". In: *International Journal of Mechanical Engineering and Robotics Research* 6.2 (2017), pages 154–159. doi: 10.18178/ijmerr.6.2.154-159.
(Cited on pages 48, 54, 59, 118, 135).

[Add+15] Alemayehu Addisu, Laurent George, Pierre Courbin, and Vincent Sciandra. "Modelling of Distributed Energy Resources in Industrial Context Using Service Curves of Network Calculus". In: *Computational Science and Computational Intelligence (CSCI), 2015 International Conference on*. IEEE. 2015, pages 330–335. doi: 10.1109/CSCI.2015.81.
(Cited on pages 51, 118, 135).

[Add+17d] Alemayehu Addisu, Laurent George, Pierre Courbin, and Vincent Sciandra. "Smoothing of renewable energy generation using Gaussian-based method with power constraints". In: *Energy Procedia* 134 (2017), pages 171–180. doi: 10.1016/j.egypro.2017.09.555.

(Cited on pages 56, 118, 135).

[AZLB12] Hussein Al-Zubaidy, Jorg Liebeherr, and Almut Burchard. "A network calculus approach for the analysis of multi-hop fading channels". In: *arXiv preprint arXiv:1207.6630* (2012).

(Cited on page 36).

[AMS14] MJE Alam, Kashem M Muttaqi, and Danny Sutanto. "A novel approach for ramp-rate control of solar PV using energy storage to mitigate output fluctuations caused by cloud passing". In: *Energy Conversion, IEEE Transactions on* 29.2 (2014), pages 507–518. doi: 10.1109/TEC.2014.2304951.

(Cited on pages 55, 56).

[Ale+02] E Alessio, A Carbone, G Castelli, and V Frappietro. "Second-order moving average and scaling of stochastic time series". In: *The European Physical Journal B-Condensed Matter and Complex Systems* 27.2 (2002), pages 197–200. doi: 10.1140/epjb/e20020150.

(Cited on page 55).

[All13] OpenADR Alliance. "OpenADR 2.0 profile specification, B profile". In: *document 20120912–1* (2013).

(Cited on pages 25, 27, 112, 113).

[Alt12] Tayfur Altiok. *Performance analysis of manufacturing systems*. Springer Science & Business Media, 2012. isbn: 9781461219248.

(Cited on page 80).

[Alt+11] Matthias Altmann et al. "Effect of smart metering on electricity prices". In: *Study for the European Parliament* 130 (2011).

[AS93] Ronald G Askin and Charles R Standridge. *Modeling and analysis of manufacturing systems*. John Wiley & Sons Inc, 1993. isbn: 978-0-471-51418-3.

(Cited on page 79).

[BW11] Mark Bolinger and Ryan Wiser. *Understanding trends in wind turbine prices over the past decade*. Report LBNL-5119E. Lawrence Berkeley National Laboratory, Oct. 2011.
(Cited on page 67).

[Bor08] Severin Borenstein. "Equity effects of increasing-block electricity pricing". In: *Center for the Study of Energy Markets* (2008).
(Cited on page 21).

[BN02] Lars Burgstahler and Martin Neubauer. "New modifications of the exponential moving average algorithm for bandwidth estimation". In: *Proc. of the 15th ITC Specialist Seminar*. 2002.
(Cited on page 55).

[BD13] Alan Burns and Robert Davis. "Mixed criticality systems-a review". In: *Department of Computer Science, University of York, Tech. Rep* (2013).
(Cited on page 121).

[Bur+13] Michael T Burr et al. *Minnesota microgrids: barriers, opportunities, and pathways toward energy assurance*. Report CRA No. D06090. Microgrid Institute, Sept. 2013.
(Cited on page 67).

[BS92] John A Buzacott and J George Shanthikumar. "Design of manufacturing systems using queueing models". In: *Queueing systems* 12.1-2 (1992), pages 135–213. doi: 10.1007/BF01158638.
(Cited on pages 42, 79).

[BS93] John A Buzacott and J George Shanthikumar. *Stochastic models of manufacturing systems*. Volume 4. Prentice Hall Englewood Cliffs, NJ, 1993. isbn: 978-0138475673.
(Cited on page 80).

[Car+13] C Carrillo, AF Obando Montaño, J Cidrás, and E Díaz-Dorado. "Review of power curve modelling for wind turbines". In: *Renewable and Sustainable Energy Reviews* 21 (2013), pages 572–581. doi: 10.1109/TEC.2003.816593.
(Cited on page 12).

[CS02] Luis Castaner and Santiago Silvestre. *Modelling photovoltaic systems using PSpice*. John Wiley and Sons, 2002. ISBN: 9780470855539. (Cited on page 46).

[Cha00] Cheng-Shang Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag London, 2000. ISBN: 978-1-4471-1147-4. DOI: 10.1007/978-1-4471-0459-9. (Cited on page 29).

[Che+09] Haisheng Chen et al. "Progress in electrical energy storage system: A critical review". In: *Progress in Natural Science* 19.3 (2009), pages 291–312. ISSN: 10020071. DOI: 10.1016/j.pnsc.2008.07.014. (Cited on pages 14, 63).

[CGW12] SX Chen, Hoay Beng Gooi, and MingQiang Wang. "Sizing of energy storage for microgrids". In: *IEEE Transactions on Smart Grid* 3.1 (2012), pages 142–151. DOI: 10.1109/TSG.2011.2160745. (Cited on pages 15, 58, 59).

[Cla99] Jens Clausen. "Branch and bound algorithms-principles and examples". In: *Department of Computer Science, University of Copenhagen* (1999), pages 1–30. (Cited on page 94).

[CN09] CM Colson and MH Nehrir. "A review of challenges to real-time power management of microgrids". In: *Power & Energy Society General Meeting, 2009. PES'09. IEEE*. IEEE. 2009, pages 1–8. DOI: 10.1109/PES.2009.5275343. (Cited on page 3).

[Con+03] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. "ARIMA models to predict next-day electricity prices". In: *IEEE transactions on power systems* 18.3 (2003), pages 1014–1020. DOI: 10.1109/TPWRS.2002.804943. (Cited on page 60).

[Cru91] Rene L Cruz. "A calculus for network delay. I. Network elements in isolation". In: *IEEE Transactions on information theory* 37.1 (1991), pages 114–131. DOI: 10.1109/18.61109. (Cited on page 29).

[CF10] Guy L Curry and Richard M Feldman. *Manufacturing systems modeling and analysis*. Springer Science & Business Media, 2010. isbn: 9783642166181.

(Cited on pages 40, 41, 79).

[DOE11] DOE. *DOE Microgrid Workshop*. Technical report. Office of Electricity Delivery and Energy Reliability Smart Grid R&D Program, Aug. 2011.

(Cited on page 10).

[DG92] Yves Dallery and Stanley B Gershwin. "Manufacturing flow line systems: a review of models and analytical results". In: *Queueing systems* 12.1-2 (1992), pages 3–94. doi: 10.1007/BF01158636.

(Cited on page 17).

[Den+15] Ruilong Deng, Zaiyue Yang, Mo-Yuen Chow, and Jiming Chen. "A survey on demand response in smart grids: Mathematical models and approaches". In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), pages 570–582. doi: 10.1109/TII.2015.2414719.

(Cited on pages 20, 21, 22, 24).

[Den+10] Paul Denholm, Erik Ela, Brendan Kirby, and Michael Milligan. "The role of energy storage with renewable electricity generation". In: (2010).

(Cited on page 14).

[DPH07] Alexandros C Diamantidis, Chrissoleon T Papadopoulos, and Cathal Heavey. "Approximate analysis of serial flow lines with multiple parallel-machine stations". In: *IIE Transactions* 39.4 (2007), pages 361–375. doi: 10.1080/07408170600838423.

(Cited on page 80).

[Dij59] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pages 269–271. doi: 10.1007/BF01386390.

(Cited on page 99).

[EPE17] Spot EPEX. *Day-ahead auction*. 2017. url: http://www.epexspot.com/en/market-data/dayaheadauction.

(Cited on pages 6, 19, 51, 63).

[EPR] EPRI. *Electric Power Research Institute*. URL: https://www.epri.com/#/.

(Cited on page 29).

[Ell+12] Abraham Ellis, David Schoenwald, Jon Hawkins, Steve Willard, and Brian Arellano. "PV output smoothing with energy storage". In: *Photovoltaic Specialists Conference (PVSC), 2012 38th IEEE*. IEEE. 2012, pages 001523–001528. DOI: 10.1109/PVSC.2012.6317885.

(Cited on page 47).

[Ene] EnerNOC. *EnerNOC's GitHub*. URL: https://github.com/EnerNOC.

(Cited on pages 29, 111).

[EBL08] Sandra Eriksson, Hans Bernhoff, and Mats Leijon. "Evaluation of different turbine concepts for wind power". In: *Renewable and Sustainable Energy Reviews* 12.5 (2008), pages 1419–1434. DOI: 10.1016/j.rser.2006.05.017.

(Cited on page 12).

[Erl09] Agner Krarup Erlang. "The theory of probabilities and telephone conversations". In: *Nyt Tidsskrift for Matematik B* 20.33-39 (1909), page 16.

(Cited on page 36).

[EC07] Trishan Esram and Patrick L Chapman. "Comparison of photovoltaic array maximum power point tracking techniques". In: *IEEE Transactions on energy conversion* 22.2 (2007), pages 439–449. DOI: 10.1109/TEC.2006.874230.

(Cited on page 13).

[Fan+11] Kan Fang, Nelson Uhan, Fu Zhao, and John W Sutherland. "A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction". In: *Journal of Manufacturing Systems* 30.4 (2011), pages 234–240. DOI: 10.1016/j.jmsy.2011.08.004.

(Cited on page 80).

[Fel+12] David Feldman et al. *Photovoltaic (PV) pricing trends: historical, recent, and near-term projections*. Report LBNL-6019E. Lawrence Berkeley National Laboratory, Nov. 2012.
(Cited on page 67).

[Flo+13] Christodoulos A Floudas et al. *Handbook of test problems in local and global optimization*. Volume 33. Springer Science & Business Media, 2013. isbn: 978-1-4757-3040-1.
(Cited on page 48).

[FFG06] Fabrice Frances, Christian Fraboul, and Jérôme Grieu. "Using network calculus to optimize the AFDX network". In: (2006).
(Cited on page 36).

[FCD96] Yannick Frein, Christian Commault, and Yves Dallery. "Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers". In: *IIE transactions* 28.7 (1996), pages 545–554. doi: 10.1080/15458830.1996.11770699.
(Cited on page 18).

[GRL] GRL. *Grails: a powerful Groovy-based web application framework*. url: https://grails.org/.
(Cited on page 111).

[GDR11] Jean-Philippe Georges, Thierry Divoux, and Éric Rondeau. "Network calculus: application to switched real-time networking". In: *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*. 2011, pages 399–407. isbn: 978-1-936968-09-1.
(Cited on page 36).

[GW07] Stanley B Gershwin and Loren M Werner. "An approximate analytical method for evaluating the performance of closed-loop flow systems with unreliable machines and finite buffers". In: *International Journal of Production Research* 45.14 (2007), pages 3085–3111. doi: 10.1080/00207540500385980.
(Cited on page 18).

[GFKR15] Yashar Ghiassi-Farrokhfal, Srinivasan Keshav, and Catherine Rosenberg. "Toward a realistic performance analysis of storage systems in smart grids". In: *IEEE Transactions on Smart Grid* 6.1 (2015), pages 402–410. ISSN: 1949-3061. DOI: 10.1109/TSG.2014.2330832.

(Cited on pages 15, 63).

[GF99] Manish K Govil and Michael C Fu. "Queueing theory in manufacturing: A survey". In: *Journal of manufacturing systems* 18.3 (1999), page 214.

(Cited on pages 16, 42, 79).

[Haa13] Barry Haaser. *Enabling The Standard for Automated Demand Response*. 2013. URL: http://www.openadr.org/assets/docs/openadr%20plma%20seminar_final.pdf.

(Cited on pages 26, 27, 28).

[Hay13] Ernie Hayden. "Introduction to microgrids". In: *Securicon Report* (2013), pages 1–13.

(Cited on page 4).

[HGB10] Thomas D Hund, Sigifredo Gonzalez, and Keith Barrett. "Grid-tied PV system energy smoothing". In: *Photovoltaic Specialists Conference (PVSC), 2010 35th IEEE*. IEEE. 2010, pages 002762–002766. DOI: 10.1109/PVSC.2010.5616799.

(Cited on page 47).

[IEA16] IEA. *Key World Energy Statistics, International Energy Agency,* 2016. URL: https://www.iea.org/publications/freepublications/publication/KeyWorld2016.pdf.

(Cited on pages 6, 78).

[Jia+16] Zhiyang Jia, Liang Zhang, Jorge Arinez, and Guoxian Xiao. "Performance Analysis of Assembly Systems With Bernoulli Machines and Finite Buffers During Transients". In: *IEEE Transactions on Automation Science and Engineering* 13.2 (2016), pages 1018–1032. DOI: 10.1109/TASE.2015.2442521.

(Cited on pages 42, 79).

[JL11] Libin Jiang and Steven Low. "Real-time demand response with uncertain renewable energy in smart grid". In: *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. IEEE. 2011, pages 1334–1341. DOI: 10.1109/Allerton.2011.6120322.

(Cited on pages 24, 25).

[JL08] Yuming Jiang and Yong Liu. *Stochastic network calculus*. Volume 1. Springer, 2008. ISBN: 978-1-84800-127-5.

(Cited on page 30).

[Joh+13] Jay Johnson et al. "PV output smoothing using a battery and natural gas engine-generator". In: *Photovoltaic Specialists Conference (PVSC), 2013 IEEE 39th*. IEEE. 2013, pages 1811–1816. DOI: 10.1109/PVSC.2013.6744494.

(Cited on page 47).

[Kak+09] Naoto Kakimoto, Hiroyuki Satoh, Satoshi Takayama, and Kouichi Nakamura. "Ramp-rate control of photovoltaic generator with electric double-layer capacitor". In: *IEEE Transactions on Energy Conversion* 24.2 (2009), pages 465–473. DOI: 10.1109/TEC.2008.2001580.

(Cited on page 47).

[Kar78] Richard M Karp. "A characterization of the minimum cycle mean in a digraph". In: *Discrete mathematics* 23.3 (1978), pages 309–311. DOI: 10.1016/0012-365X(78)90011-0.

(Cited on page 99).

[Kat+12] David Kathan et al. *Assessment of Demand Response and Advanced Metering*. Staff Report. Federal Energy Regulatory Commission, Dec. 2012.

(Cited on page 20).

[LBN] LBNL. *Demand Response Research Center*. URL: https://drrc.lbl.gov.

(Cited on page 25).

[LP04] Robert H Lasseter and Paolo Paigi. "Microgrid: a conceptual solution". In: *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*. Volume 6. IEEE. 2004, pages 4285–4290. doi: 10.1109/PESC.2004.1354758.
(Cited on page 10).

[LB98] Jean-Yves Le Boudec. "Application of network calculus to guaranteed service networks". In: *IEEE Transactions on Information theory* 44.3 (1998), pages 1087–1096. doi: 10.1109/18.669170.
(Cited on page 36).

[LBT01] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Volume 2050. Springer Science & Business Media, 2001. isbn: 9783540453185.
(Cited on pages 29, 30, 32, 33).

[LBT12] Jean-Yves Le Boudec and Dan-Cristian Tomozei. "A demand-response calculus with perfect batteries". In: *International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*. Springer. 2012, pages 273–287. doi: 10.1007/978-3-642-28540-0_23.
(Cited on page 47).

[LM06] Stefan-Helmut Leitner and Wolfgang Mahnke. "OPC UA–service-oriented architecture for industrial applications". In: *ABB Corporate Research Center* (2006).
(Cited on page 111).

[Li04] Jingshan Li. "Modeling and analysis of manufacturing systems with parallel lines". In: *IEEE transactions on automatic control* 49.10 (2004), pages 1824–1832. doi: 10.1109/TAC.2004.835584.
(Cited on page 18).

[LAR05] Jingshan Li, Jeffrey M Alden, and John R Rabaey. "Approximating feeder line reliability statistics with partial data collection in assembly systems". In: *Computers & Industrial Engineering* 48.2 (2005), pages 181–203. doi: 10.1016/j.cie.2005.01.006.
(Cited on page 18).

[LH05] Jingshan Li and Ningjian Huang. "Modelling and analysis of a multiple product manufacturing system with split and merge". In: *International Journal of Production Research* 43.19 (2005), pages 4049–4066. DOI: 10.1080/00207540500031899.

(Cited on page 18).

[Li+09] Jingshan Li, Dennis E. Blumenfeld, Ningjian Huang, and Jeffrey M. Alden. "Throughput analysis of production systems: recent advances and future topics". In: *International Journal of Production Research* 47.14 (2009), pages 3823–3851. DOI: 10.1080/00207540701829752.

(Cited on pages 16, 17, 42).

[LCL11] Na Li, Lijun Chen, and Steven H Low. "Optimal demand response based on utility maximization in power networks". In: *Power and Energy Society General Meeting, 2011 IEEE*. IEEE. 2011, pages 1–8. DOI: 10.1109/PES.2011.6039082.

(Cited on page 24).

[Li+11] Xiangjun Li, Yong Li, Xiaojuan Han, and Dong Hui. "Application of fuzzy wavelet transform to smooth wind/PV hybrid power system output with battery energy storage system". In: *Energy Procedia* 12 (2011), pages 994–1001. DOI: 10.1016/j.egypro.2011.10.130.

(Cited on page 47).

[LZ14] Hao Liang and Weihua Zhuang. "Stochastic modeling and optimization in a microgrid: A survey". In: *Energies* 7.4 (2014), pages 2027–2050. DOI: 10.3390/en7042027.

(Cited on page 12).

[Lit61] John DC Little. "A proof for the queuing formula: L= λ W". In: *Operations research* 9.3 (1961), pages 383–387.

(Cited on page 38).

[Luo+13] Hao Luo, Bing Du, George Q Huang, Huaping Chen, and Xiaolin Li. "Hybrid flow shop scheduling considering machine electricity consumption cost". In: *International Journal of Production Economics* 146.2 (2013), pages 423–439. DOI: 10.1016/j.ijpe.2013.01.028.

(Cited on page 80).

[Mac08] David JC MacKay. *Sustainable Energy–without the hot air*. UIT Cambridge, 2008. ISBN: 0954452933.
(Cited on page 14).

[Man08] Michael Manitz. "Queueing-model based analysis of assembly lines with finite buffers and general service times". In: *Computers & Operations Research* 35.8 (2008), pages 2520–2536. DOI: 10.1016/j.cor.2006.12.016.
(Cited on pages 42, 79).

[MW15] Schuyler Matteson and Eric Williams. "Residual learning rates in lead-acid batteries: Effects on emerging technologies". In: *Energy Policy* 85 (2015), pages 71–79. DOI: 10.1016/j.enpol.2015.05.014.
(Cited on page 63).

[MDC04] Sharif Melouk, Purushothaman Damodaran, and Ping-Yu Chang. "Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing". In: *International journal of production economics* 87.2 (2004), pages 141–147. DOI: 10.1016/S0925-5273(03)00092-6.
(Cited on page 80).

[Mer13] Konrad Mertens. *Photovoltaics: Fundamentals, Technology and Practice*. John Wiley & Sons, 2013. ISBN: 9781118703373.
(Cited on page 13).

[Mod04] IDA Modbus. "Modbus application protocol specification v1. 1a". In: *North Grafton, Massachusetts (www. modbus. org/specs. php)* (2004).
(Cited on page 111).

[MRLG10] Amir-Hamed Mohsenian-Rad and Alberto Leon-Garcia. "Optimal residential load control with price prediction in real-time electricity pricing environments". In: *IEEE transactions on Smart Grid* 1.2 (2010), pages 120–133. DOI: 10.1109/TSG.2010.2055903.
(Cited on page 24).

[MSP13] Joon-Yung Moon, Kitae Shin, and Jinwoo Park. "Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency". In: *The International Journal of Advanced Manufacturing Technology* 68.1-4 (2013), pages 523–535. DOI: 10.1007/s00170-013-4749-8.
(Cited on page 80).

[OAS] OASIS. *Organization of Structured Information Standards*. URL: https://www.oasis-open.org/.
(Cited on page 25).

[OFR] OFR. *Openfire Server from Ignite Realtime*. URL: https://www.igniterealtime.org/projects/openfire/.
(Cited on page 111).

[OMN] OMNET++. *Objective Modular Network Testbed in C++*. URL: https://omnetpp.org/.
(Cited on pages 104, 105).

[Ope] OpenADR. *OpenADR Alliance*. URL: http://www.openadr.org/.
(Cited on page 25).

[PW] PW. *PVWatts Calculator*. URL: http://pvwatts.nrel.gov/pvwatts.php.
(Cited on page 60).

[PL05] Ping-Feng Pai and Chih-Sheng Lin. "A hybrid ARIMA and support vector machines model in stock price forecasting". In: *Omega* 33.6 (2005), pages 497–505. DOI: 10.1016/j.omega.2004.07.024.
(Cited on page 60).

[PH96] HT Papadopoulos and Cathal Heavey. "Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines". In: *European journal of operational Research* 92.1 (1996), pages 1–27. DOI: 10.1016/0377-2217(95)00378-9.
(Cited on pages 42, 79).

[Par+15] Sina Parhizi, Hossein Lotfi, Amin Khodaei, and Shay Bahramirad. "State of the art in research on microgrids: A review". In: *IEEE Access* 3 (2015), pages 890–925. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2015.2443119.

(Cited on pages 4, 10).

[Pie+09] Mary Ann Piette, Ghatikar Girish, Kiliccote Sila, Palensky Peter, and McParland Charles. "Open automated demand response communications specification (Version 1.0)". In: *Lawrence Berkeley National Laboratory* (2009).

(Cited on page 25).

[Qdr06] Q Qdr. "Benefits of demand response in electricity markets and recommendations for achieving them". In: *US department of energy* (2006).

(Cited on pages 20, 21, 22, 23).

[RR11] Magdi Ragheb and Adam M Ragheb. *Wind turbines theory-the betz equation and optimal rotor tip speed ratio*. Edited by Rupp Carriveau. It is chapter 2 of the book "Fundamental and Advanced Topics in Wind Power". INTECH Open Access Publisher, 2011. ISBN: 978-953-307-508-2.

(Cited on page 12).

[RY07] Md Habibur Rahman and Susumu Yamashiro. "Novel distributed power generating system of PV-ECaSS using solar energy estimation". In: *IEEE Transactions on Energy Conversion* 22.2 (2007), pages 358–367. DOI: 10.1109/TEC.2006.870832.

(Cited on pages 13, 14).

[RT88] Saifur Rahman and Kwa-sur Tam. "A feasibility study of photovoltaic-fuel cell hybrid energy system". In: *IEEE transactions on energy conversion* 3.1 (1988), pages 50–55. DOI: 10.1109/60.4199.

(Cited on page 47).

[Riv05] Charles River. *Primer on demand side management: with an Emphasison Price-Responsive Programs*. Report CRA No. D06090. World Bank, Feb. 2005.

(Cited on pages 19, 20, 21).

[RLK06] S Mansoor Roomi, IM Lakshmi, and V Abhai Kumar. "A recursive gaussian weighted filter for impulse noise removal". In: *GVIP Journal* 6.3 (2006), pages 33–37.
(Cited on page 56).

[Sak09] Jacques Sakarovitch. *Elements of automata theory*. Cambridge University Press, 2009. ISBN: 0521844258 9780521844253.
(Cited on page 91).

[Sam+10] Pedram Samadi, Amir-Hamed Mohsenian-Rad, Robert Schober, Vincent WS Wong, and Juri Jatskevich. "Optimal real-time pricing algorithm based on utility maximization for smart grid". In: *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE. 2010, pages 415–420. DOI: 10.1109/SMARTGRID.2010.5622077.
(Cited on page 24).

[Shr+14] Fadi Shrouf, Joaquin Ordieres-Meré, Alvaro García-Sánchez, and Miguel Ortega-Mier. "Optimizing the production scheduling of a single machine to minimize total energy consumption costs". In: *Journal of Cleaner Production* 67 (2014), pages 197–207. DOI: 10.1016/j.jclepro.2013.12.024.
(Cited on page 80).

[SPK03] JG Slootweg, H Polinder, and WL Kling. "Representing wind turbine electrical generating systems in fundamental frequency simulations". In: *IEEE Transactions on energy conversion* 18.4 (2003), pages 516–524. ISSN: 0885-8969. DOI: 10.1109/TEC.2003.816593.
(Cited on page 12).

[SAM13] Michael Starke, Nasr Alkadi, and Ookie Ma. "Assessment of Industrial Load for Demand Response across US Regions of the Western Interconnect". In: *Oak Ridge National Laboratory* (2013).
(Cited on pages 78, 80).

[SW12] Wencong Su and Jianhui Wang. "Energy management systems in microgrid operations". In: *The Electricity Journal* 25.8 (2012), pages 45–60. DOI: 10.1016/j.tej.2012.09.010.
(Cited on page 3).

[Szt12] János Sztrik. "Basic queueing theory". In: *University of Debrecen, Faculty of Informatics* 193 (2012).
(Cited on pages 37, 39, 40).

[THL13] Yuan-yuan TAN, Ying-lei HUANG, and Shi-xin LIU. "Two-stage mathematical programming approach for steelmaking process scheduling under variable electricity price". In: *Journal of Iron and Steel Research, International* 20.7 (2013), pages 1–8. DOI: 10.1016/S1006-706X(13)60118-1.
(Cited on page 80).

[TKF89] K-S Tam, Prem Kumar, and Mark Foreman. "Enhancing the utilization of photovoltaic power generation by superconductive magnetic energy storage". In: *IEEE Transactions on Energy Conversion* 4.3 (1989), pages 314–321. DOI: 10.1109/60.43230.
(Cited on page 47).

[Tes+11] Samson G Tesfahunegn, Øystein Ulleberg, Preben JS Vie, and Tore M Undeland. "PV fluctuation balancing using hydrogen storage–A smoothing method for integration of PV generation into the utility grid". In: *Energy Procedia* 12 (2011), pages 1015–1022. DOI: 10.1016/j.egypro.2011.10.133.
(Cited on page 47).

[VBK16] Amaury Van Bemten and Wolfgang Kellerer. "Network Calculus: A Comprehensive Guide". In: *Technical University of Munich* (2016).
(Cited on pages 32, 35, 36).

[Var01] András Varga. "Discrete event simulation system". In: *Proc. of the European Simulation Multiconference (ESM'2001)*. 2001.
(Cited on page 104).

[Ves07] Steve Vestal. "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance". In: *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*. IEEE. 2007, pages 239–243. DOI: 10.1109/RTSS.2007.47.
(Cited on page 121).

[Ves] Vestas. *V90-3.0 MW wind turbine of Vestas*. URL: https://www.vestas.com/en/products/turbines/v90%203_0_mw#!.
(Cited on page 13).

[VEZ14] Christopher Villarreal, David Erickson, and Marzia Zafar. *Microgrids: A regulatory perspective*. Technical report. California Public Utilities Commission Policy & Planning Division, Apr. 2014.

(Cited on page 4).

[WU] WU. *Weather Underground*. URL: https://www.wunderground.com/.

(Cited on page 61).

[WLL11] Kai Wang, Steven Low, and Chuang Lin. "How stochastic network calculus concepts help green the power grid". In: *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE. 2011, pages 55–60. DOI: 10.1109/SmartGridComm.2011.6102385.

(Cited on page 46).

[Wan+12] Kai Wang, Florin Ciucu, Chuang Lin, and Steven H Low. "A stochastic power network calculus for integrating renewable energy sources into the power grid". In: *IEEE Journal on Selected Areas in Communications* 30.6 (2012), pages 1037–1048. DOI: 10.1109/JSAC.2012.120703.

(Cited on pages 35, 46).

[Wei94] William Wu-Shyong Wei. *Time series analysis*. Addison-Wesley publ Reading, 1994. ISBN: 9780691042893.

(Cited on page 60).

[Wei02] Eric W Weisstein. "Gaussian function". In: (2002).

(Cited on page 56).

[WLD15] David L Wood, Jianlin Li, and Claus Daniel. "Prospects for reducing the processing cost of lithium ion batteries". In: *Journal of Power Sources* 275 (2015), pages 234–242. DOI: 10.1016/j.jpowsour.2014.11.019.

(Cited on page 63).

[WJM12] Kui Wu, Yuming Jiang, and Dimitri Marinakis. "A stochastic calculus for network systems with renewable energy sources". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE. 2012, pages 109–114. DOI: 10.1109/INFCOMW.2012.6193470.

(Cited on pages 46, 47).

[YK05] Dan York and Martin Kushler. "Exploring the Relationship Between Demand Response and Energy Efficiency: A Review of Experience and Discussion of Key Issues". In: American Council for an Energy-Efficient Economy. 2005.

(Cited on page 19).

[ZL17] Arthouros Zervos and Christine Lins. *Renewables 2017 global status report*. report REN21 2017. Renewable Energy Policy Network for the 21st Century, 2017.

(Cited on pages 4, 5).

[ZG16] Qi Zhang and Ignacio E Grossmann. "Planning and scheduling for industrial demand side management: advances and challenges". In: *Alternative Energy Sources and Technologies*. Springer, 2016, pages 383–414. DOI: 10.1007/978-3-319-28752-2_14.

(Cited on pages 20, 22).

[Zha+16] Xiao Zhang, Gabriela Hug, J Zico Kolter, and Iiro Harjunkoski. "Model predictive control of industrial loads and energy storage for demand response". In: *Power and Energy Society General Meeting (PESGM), 2016*. IEEE. 2016, pages 1–5. DOI: 10.1109/PESGM.2016.7741228.

(Cited on page 16).

[ZYF07] Wei Zhou, Hongxing Yang, and Zhaohong Fang. "A novel model for photovoltaic array performance prediction". In: *Applied energy* 84.12 (2007), pages 1187–1198. DOI: 10.1016/j.apenergy.2007.04.006.

(Cited on page 46).

[Zuk13] Moshe Zukerman. "Introduction to queueing theory and stochastic teletraffic models". In: *arXiv preprint arXiv:1307.2968* (2013).

(Cited on pages 39, 40).