UNIVERSITÀ DELLA CALABRIA

UNIVERSITY OF CALABRIA

Department of Mechanical, Energy and Management Engineering

PhD in

Science and Engineering of Environment, Construction and Energy

CYCLE
XXX

PHD THESIS

CONTROL SYSTEM FOR A NANOGRID FOR HOME APPLICATION: DYNAMIC ANALYSIS AND IMPLEMENTATION ASPECTS USING A BEHAVIOUR TREE

Disciplinary Scientific Sector: ING/IND33

Coordinator:

Prof. (S. Critelli) ____________

Supervisors:

Prof. (D. Menniti) ____________

Eng. (N. Sorrentino) ____________

Eng. (A. Pinnarelli) ____________

Candidate:

(Michele Motta) ____________

The research has been supported by the research project:

“**Sistemi Domotici per il Servizio di Brokeraggio Energetico Cooperativo**”
DISTRETTO DOMUS
Programma Operativo Nazionale Ricerca e Competitività
ASSE I “Sostegno ai mutamenti strutturali”
AZIONI I.3.1 “Distretti ad alta tecnologia”
Ministero dell’Istruzione, dell’Università e della Ricerca (MIUR)

Index

Abstract

The Twenty-First Session of the Conference of the Parties (COP21), which had place in Paris from the 30th of November to the 12th of December in 2015, was concluded with a document that states the risks related to the climate change and all the needed actions aimed to mitigate it. The main goal is to avoid that temperature rise to more than 2°C compared to the pre-industrial temperature. Even if the undertaken commitments will be respected, unfortunately they would not be sufficient to avoid a 2,7°C increase of the mean temperature. The increasing of the mean temperature causes environmental disasters. An index assesses the vulnerability of countries to climate change, and it is known as the Climate Change Vulnerability Index (Fig. 1-1). Paradoxically, the most threatened places are those which have contributed the least to climate change.

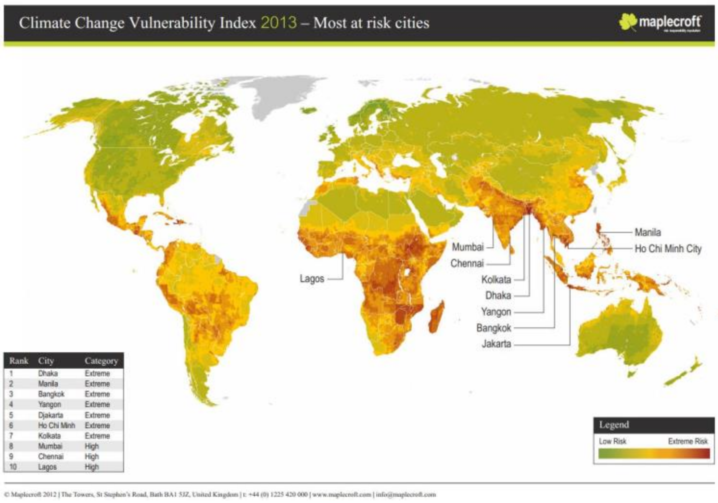


Fig. 1-1 Climate Change Vulnerability Index - 2013

The amount of climatic gas emissions is bounded to the need for energy; this need is transversal and involves all human activities, from transport to air conditioning, from technology to research. A large part of energy production derives from combustion processes within giant legacy power plant. However, the spread of Distributed Energy Resources (DERs) and, in particular of renewable sources, are changing the energy scenario. To facilitate the spread, making them economically attractive, and to overcome technological limits, derived by an electrical infrastructure born to be used with a top-down management, the DERs can be integrated inside a small power system with autonomous decision-making. This small power system is called nanogrid. A nanogrid allows to increase reliability and power quality, reduces the peak load seen from the network, reduces transmission and distribution losses, accelerates the adoption of distributed and renewable generation sources, while reducing emissions and fossil fuel consumption.

In this thesis a control system for a nanogrid has been developed and implemented. The control system wants to ensure flexibility, modularity, reliability, robustness and autonomy. The activity carried out aspires to speed up the adoption of this technology in both academic and industrial fields, to allow a broad use of the nanogrids in the domestic environments. This work wants to be the proof-of-concept needed to attract the attention of industry, which is critical for the development of this new technology.

Riassunto

La ventunesima sessione della Conference of the Parties (COP21), tenutasi a Parigi dal 30 Novembre al 12 Dicembre 2015, si è conclusa con un documento che prende atto dei rischi associati al cambiamento climatico e che valuta le azioni necessarie per mitigarlo. L'obiettivo è di limitare che l'aumento di temperature superi i 2°C rispetto alla temperatura pre-industriale. Purtroppo gli impegni intrapresi, anche qualora venissero rispettati, risultano inadeguati. Provvedimenti che non sarebbero sufficienti ad impedire un incremento della temperatura media di 2,7°C. L'incremento della temperatura è purtroppo causa di disastri ambientali. Esiste un indice per valutare la vulnerabilità dei paesi al cambiamento climatico, noto come Climate Change Vulnerability Index (Fig. 1-2). Paradossalmente i luoghi più a rischio a causa del riscaldamento globale sono quelli che hanno contribuito in misura minore al cambiamento climatico.

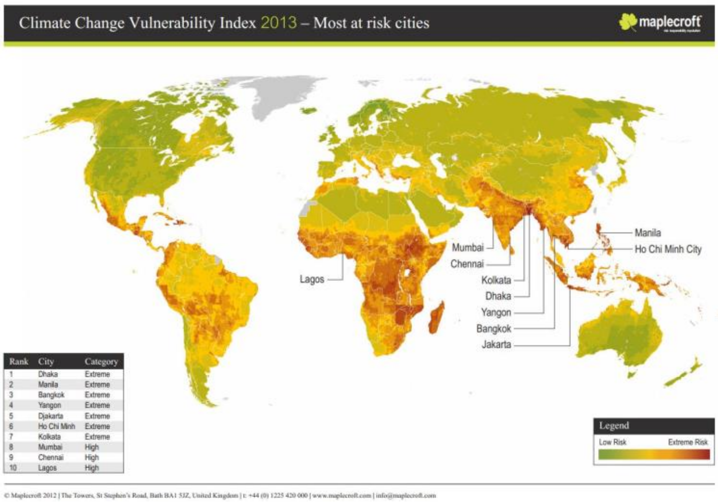


Fig. 1-2 Indice di vulnerabilità del cambiamento climatico - 2013

La quantità di emissioni di gas climalteranti è legata al bisogno di energia; questo bisogno è trasversale e coinvolge tutte le attività umane, dal trasporto alla climatizzazione, dalla tecnologia alla ricerca. La gran parte della produzione deriva da processi di combustione all'interno di grandi impianti termoelettrici. Tuttavia, la diffusione delle risorse di energia distribuita (DERs) e, in particolare, delle risorse rinnovabili stanno cambiando lo scenario energetico. Per favorire la massima diffusione, quindi renderle economicamente attraenti, e per superare i limiti tecnologici, derivati da un'infrastruttura elettrica nata per essere usata secondo una gestione top-down, queste risorse distribuite possono essere integrate all'interno di un piccolo sistema elettrico dotato di capacità decisionale. Questo piccolo sistema elettrico prende il nome di nanogrid. Una nanogrid consente di aumentare qualità e affidabilità dell'alimentazione, ridurre il picco di carico visto dalla rete, riduce le perdite di trasmissione e distribuzione, accelera l'adozione di sorgenti distribuite e rinnovabili, riducendo le emissioni e il consumo di combustibili fossili.

In questa tesi è stato sviluppato e implementato un sistema di controllo per una nanogrid. Il sistema di controllo vuole garantire flessibilità, modularità, affidabilità, robustezza e autonomia. L'attività svolta aspira ad accelerare l'adozione di questa tecnologia sia in ambito accademico che industriale, al fine di consentire un ampio utilizzo delle nanogrid in ambienti domestici. Questo lavoro vuole essere la "proof-of-concept" necessaria per attrarre l'attenzione dell'industria e degli investitori, per consentire lo sviluppo di questa tecnologia.

1 Introduction

For about a century, affordable electrification has been based on economies of scale, with large generating plants producing hundreds or thousands of megawatts; the produced power is sent to distant users through a transmission grid and distributed through the distribution grid. Today, many technological developments make this model more complicated. One of these is the spread of Distributed Energy Resources (DERs). DERs have existed for a long time and have long been technically possible, but the nowadays spread is due to the lower cost in the installation, the increased efficiency, the better use of the cogeneration units and the avoided transmission losses, that make DERs cost-competitive with giant legacy power plants. The news is that we are nearing to a tipping point, beyond which, for many applications, distributed generation will be the cheapest way to provide electricity.

The electrical power system is operated according to a top-down model, where the distribution network should only serve to "distribute" electricity from large legacy power plants (Fig. 1-1). The increasing number of distributed generators connected in the distribution grid may, at different times of the day and in specific areas, result in an inverted energy flow, i.e. the energy from the peripheral distribution grid rises to the transmission one. This phenomenon causes deep concerns for grid operators; indeed, during these events, the operators may not guarantee power quality and security. According to the authors of [1]-[2], adversity problems for the utility operators, related to the new model of electrical network, include scheduling and dispatching of production units, uncertainty in demand and production, and the determination of appropriate levels of reserves.

The spread of DERs entails another reason of concern, that is the production unpredictability. This unpredictability arises from the variability of the primary energy sources, mainly when the production is generated by renewable DERs. Moreover, today the DERs energy production can’t be regulated in real-time by a central system, because the current hardware and software don’t allow to manage the large number of sources spread over an extended area.

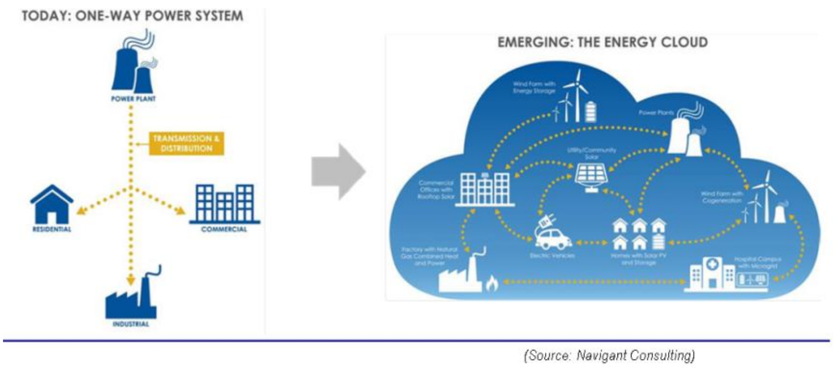


Fig. 1-1 The legacy top-down power system (left) and the future grid (right)

The problem of the electrical grid control is one of the most complex that man has faced and tried to solve. The complexity is related to the extension of the power network, to the tight constraints imposed on power quality, and to the disproportionate number of users who use it in unpredictable ways and times, acting as exogenous disturbances for the control problem. But at the end of the day, what matters is the instantaneous balance at any given instant between the generated power and the consumed power inside the grid. In fact, most of the power plants have a primary control over the generated power that automatically modifies the power delivered within a few seconds to balance production and demand.

The increased power generated by the current DERs implies that the adjustment margins left to the power plants are always lower; this can negatively affect the reliability of the system. Besides, the use of DERs in the regulation within the electrical system doesn’t have a mature power regulatory mechanisms.

Thanks to the electric machines inertia, the response time of the primary control may be about few seconds; without this "delay" caused by the inertia, the control requirements could be violated. When the installed power of DERs increases, the absence of rotating inertia of huge electric machines entails a different dynamic response after a load or generation variation. In fact, the DERs are commonly connected to the power grid through inverters, that don't have a rotating inertia. Thus, even if it's true that DERs can replace legacy power plants in terms of power, they cannot replace them in terms of the physical system yet.

In order to ensure that the reliability levels of the current electrical system are not adversely affected by DERs and that renewable DERs are exploited as much as possible, many technical challenges need to be faced.

One of the interesting solution is to involve the end-user in the power system regulation. The user assumes an active role: it responds to the needs of the network and to the Distributed System Operator (DSO) commands. However, a technological and regulatory evolution is needed. The user involved will not only be the user with DERs, but also who does not produce energy will be involved by adjusting the load according to specific Demand Side Management (DSM) programs, to reduce the uncertainty in demand. The new business models must allow the competitive participation of intermittent sources, by ensuring appropriate investment incentives and ensuring remuneration mechanisms for the users that will assume an active role.

Scientific community and industry are working to move the first steps towards the new electrical infrastructure. Meanwhile, the adjective assigned to the new network is "smart". The new infrastructure will be the smart grid. There are several visions about the concept of smart grid. According to Vision and Strategy of Europe's Electricity Networks of the Future, EU's electricity networks must be [3]:

- Flexible: fulfilling customers' needs whilst responding to the changes and challenges ahead.
- Accessible: granting connection access to all network users, particularly for renewable power sources and high efficiency local generation with zero or low carbon emissions.
- Reliable: assuring and improving security and quality of supply, consistent with the demands of the digital age with resilience to hazards and uncertainties.
- Economic: providing best value through innovation, efficient energy management and 'level playing field' competition and regulation.

Therefore, the definition of Smart Grid in EU's viewpoint is:

"A smart grid is an electricity network that can intelligently integrate the actions of all users connected to it – generators, consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies."

EU envisions the smart grid as an active network:

1. To overcome the limits on the development of distributed generation and storage.
2. To ensure interoperability and security of supply.
3. To provide accessibility for all the users to a liberalized market.
4. To reduce the impact of environmental consequences of electricity production and delivery.
5. To enable demand-side participation.
6. To engage consumers interest.

The definition is very clear, though not exhaustive: this concept often adds other properties and characteristics that the network should have and doesn't highlight the path towards the final goal. But in the definition of [3] the critical points are highlighted and the goals are set. But how? How the network may be "smarter"? And what does smart mean? It is certainly difficult to answer these questions now that this vision starts to meet practical implementations and new technologies. Certainly, the involvement of different scientific sectors is required. The heterogeneous subjects involved in this huge challenge have brought confusion in the definition of smart grid. Even if the idea difficulty moved forward due to the resistances encountered in the fusion of different worlds, that slowed down technological developments, it will probably make possible the evolution toward the future power grid.

The term smart, often misused, indicates a system capable of evolving autonomously in a new state in response to exogenous events or deliberately taking decisions to pursue a goal. The system may show this autonomy if it is provided with:

- A system of sensors for measuring system states;
- A communication infrastructure;
- Control algorithms;
- An actuator system to bring the desired changes.

And this is true also for a simple and trivial example that is probably familiar to many and considered a smart system, i.e. the car cruise control. The cruise control acts on the fuel flow-rate to maintain the speed at a given set-point. All four elements are needed to fulfil the car's smart function. The sensor system and communication infrastructure let know the current condition of the vehicle (speed and acceleration). Control algorithms represent the intelligence of the system; the available controls provide a decision-making mechanism to reach the set target, i.e. the constant speed of the car. But without the system of actuators acting on fuel flow-rate, we could only observe passively; thanks to the actuators, changes in the physical system are possible. These four elements together allow to have "smart" cars, that is cars that can adjust the speed independently by the driver. And indeed, behind any smart or evolved system there are these four combined elements, that exchange information and cooperate to achieve a common goal.

Unfortunately, the complexity of the electrical system is such that the closed loop control concept is difficult to apply directly. The number of measures and the number of actuators is disproportionate; among other things, not all measures are always available and reliable. Actuators are not fully controllable and there's no guarantee that real-time communication is reliable. Lastly, the complexity of the calculations needed to process all this information, plus the latency of the communication infrastructure, would not allow real-time network constraints. Moreover, a central system of this type is not compatible with such a critical application because of the high vulnerability.

In this context, the concept of micro- and nano- grids has developed. The idea is to distribute intelligence on small entities, small grid sections, that are capable to manage autonomously, while they work in harmony with the main grid and other microgrids.

Microgrids have been around since before the “macrogrid” was created, though for the most part they only exist in industrial facilities, large campuses, and off-grid houses. However, difficulty in implementation and lack of standard and off-the-shelf technologies delay the microgrid spread.

1.1 Micro- and Nano- Grids

The concept of microgrid (MG) was first introduced in the technical literature in [4], [5] as a solution for the reliable integration of DERs, including Energy Storage Systems (ESSs) and controllable loads. Such microgrid would be perceived by the main grid as a single element responding to appropriate control signals.

These subsystems allow to aggregate many small users; the DSO (or TSO) can interact with the MG coordinator in order to resolve the technical issues of the DERs in a decentralized manner, reducing the need for a complex and highly branched central coordination, facilitating the transition to a more intelligent grid.

Despite the intense scientific activity on MGs, the practical implementation difficulty, the lack of standards, and the required investments have not allowed the MGs spread outside few experimental testbeds. The large-scale implementation of MGs is certainly technically possible, but it will take some time before it can spread as a de facto standard.

The lengthy payback time before seeing a financial return and the intermittent nature of DERs can be addressed, from a control system point of view, optimising generation and demand; the control system inside a microgrid combines a variety of sources and optimises their use to meet the power demands of small communities, hospitals and university campuses. This power subsystem becomes controllable and flexible; it can connect to or disconnect from the main power grid.

The idea of nanogrid overcomes the resistances encountered in the microgrid spread [6]. A nanogrid (NG) is analogous to a microgrid. The common points regard:

- Both are a power distribution system;
- Both can work islanded or grid-connected;
- Both can operate as either DC, AC or hybrid power structures, although the nanogrid literature clearly favours DC;
- Both have sources, renewable energy sources, and loads.

Although the names give an intuitive idea in what they differ, NG use and its possible application give further differences. The literature tends to enclose a nanogrid inside a single home/building, although there is nothing in the microgrid definition to say that the MG cannot be confined to a single home/building. This distinction should be made for the following reasons [6]:

- Nanogrids play a different role to microgrids in the power hierarchy - by connecting multiple nanogrids a microgrid is formed.
- The potential markets for nanogrids are different to that of microgrids. A nanogrid allows to obtain a power structure at a relatively lower cost than microgrids. This shifts the interest from large/multiple investors to small business owner.
- Given that the nanogrid structure is confined to a single home, the technical objectives, hardware and software, are different. By refining the subject area, the field of research is clarified, allowing to speed up the spreading process.

As usual, the definitions around emerging technologies are a little confused. When no actual standards exist, ambiguity emerges. Thus, the line between microgrids and nanogrids seems to be a bit fuzzy.

Peter Asmus, a researcher who follows the category at Navigant Research, wrote on Forbes about the comparison of microgrids and nanogrids. He suggests that nanogrids rely on solar as their energy source. Microgrids, since they are bigger, tend to incorporate wind and combined heat and power units, using the excess from onsite generation.

A nanogrid allows the development of a modern home network. Nanogrid changes the role of end-users, which from passive users - *consumers* - become also energy producers - *prosumers*. For the end-user, local energy production can become very advantageous in terms of economy, flexibility and continuity in the supply when combined with a nanogrid.

In addition, nanogrids, from the point of view of the DSO, mitigate the randomness of renewable DERs in presence of local storage. On the other hand, the nanogrid must be equipped with an intelligence to receive specific commands from the network operator.

A nanogrid optimizes the operation of distributed generators, storage systems and electric loads, maximizing the use of renewable sources and self-consumption. Moreover, a nanogrid can operate in islanded mode and can tolerate short, medium, and long interruptions, thus ensuring high quality of power supply to critical electrical loads.

The nanogrid can be connected to the power grid, but can also operate in stand-alone mode. When it is designed for working only in stand-alone mode, the system is very similar to the electrical systems present on electric vehicles, both in the automotive and avionics [7].

A local energy management, within a nanogrid, has the following benefits:

- Increases reliability and power quality;
- Reduces the peak load seen from the network and avoids peak energy costs;
- Reduces transmission and distribution losses, thanks to on-site generation and energy storage;
- Provides ancillary service to the network;
- Accelerates the adoption of distributed and renewable generation sources, reducing emissions and fossil fuel consumption.

Versatility of the local energy management are of great interest for the use of nanogrids in different sectors and different applications [8]. Potential markets for nanogrids are:

- Areas where backup power is required (natural disasters - areas subject to hurricanes) or where power quality is important (military installations);
- Rural areas with weak electrical networks;
- Developing countries with limited or no access to electricity;
- Areas where electricity is expensive;
- Areas where renewable energy is a high potential of development.

The building blocks of the NGs are power converters. In a NG different kinds of converters cooperate to reach a given goal. The converters conduct different tasks and shape the small power system; they interface dc and ac sources and loads; but they can also be used to interface the bus of two separated NGs, even if the rated voltage is different. Lastly, the converter can be the interface towards the ac utility grid. When the converter is a bidirectional power connection between the NG and other NGs, MGs or the national grid represents the interface towards external power domains, and it is called power gateway. The gateway allows to purchase power from and sell power to connected power entities, increasing the financial benefit of the distributed generation.

1.2 Nanogrids today…

Decentralization is going to be neither simple nor universal. In some places, decentralization will prevail, with most customers generating much of their own power, typically from solar photovoltaics; in other places the solar energy will be only one of the available sources. Given this heterogeneity in the decentralized production, it is difficult to give a precise definition of NG and what its function is. The idea behind the NG is the ability to work in different contexts autonomously, meanwhile providing everything is needed to interact with a larger system, when it is integrated into it.

Nordman and Christensen give the following definition of NG:

"A nanogrid is a single domain of power -- for voltage, capacity, reliability, administration, and price. Nanogrids include storage internally; local generation operates as a special type of nanogrid. A building-scale microgrid can be as simple as a network of nanogrids, without any central entity."

The nanogrid is conceptually similar to an automobile or aircraft power system, where the isolated grid and all the electronics are powered by batteries and a local alternator moved by the primary engine. Uninterruptible power supplies also perform a similar function in buildings during grid disturbances. However, a NG includes more heterogeneous sources, different kind of loads, an energy storage system and sophisticated controls, while providing a path to vastly extending the value of the available renewable resource. Suddenly, solar power is viable at night and wind power on a calm day. In addition, these islands of power enable the users to send power back to the utility and, of course, provide backup in case of a disruption of the grid [9]. Some features of a NG are: the ability to exchange information with other NGs, a predilection to a DC operation and a predisposition to be used in a building/dwelling. The nominal power may go from a few kW to hundreds of kW according to the application.

The conceptual underpinnings of nanogrids is important, but the entire idea goes nowhere without the supporting technology.

Nordman thinks nanogrids should be universal. This means that they would operate on the exact same communications and voltage standards. But because nanogrids are still mostly conceptual, no organization has attempted to create those standards. Nanogrids can't really scale without them [10].

A NG in the current energy scenario is a matter of certain interest. This scenario is heterogeneous from region to region and it is characterized by the economic development of the region. The electrical energy is a primary asset besides being a nation's development index. The global population have different electrical energy availability. Three main situations can be observed:

- Developed countries, that has a solid and well-established electricity infrastructure.
- Developing countries, where the population has access to electrical energy but with low quality and discontinuity supply.
- Underdeveloped countries, where a large part of the population has no access to electricity but, ironically, has a lot of primary energy.

NG has the potential to become revolutionary in all examined contexts.

In developed countries, the electrical power is guaranteed by a reliable, solid and well-established infrastructure. A NG could allow the integration of distributed resources, managing the energy flows within the cluster according to a specific goal. For example, a typical goal is the reduction in the unit cost of electricity and the reduction in polluting emissions and climatic gas. Moreover, there are usually narrow constraints when a high continuity has to be ensured for some critical loads: the NG ability to work offline increases the reliability of these systems, making them less vulnerable to external attacks. Lastly, the transformation of non-electrical to electrical loads, such as air conditioning and transport, would not allow the current infrastructure to withstand overloads at certain times of the day: a local NG management thanks to DERs production would allow to postpone or to avoid large investments in the increasing capacity of the existing infrastructure.

In developing countries, the NG ensures increased efficiency and continuity of power supply. Thanks to the availability of local sources, even in the event of network malfunctions, the loads stay powered without risk for damaging more sensitive loads thanks to the high power quality provided by the NG.

Finally, in the case of the poorest countries, there is often little access to electricity. The availability of this primary asset could allow the economic development of these populations, sharply increasing the quality of life. The social paradox is that these countries often have a lot of solar energy. The NG could be the solution. Indeed, thanks to its features, it can guarantee a continuous power supply even without an electric grid and thermoelectric generators: this allows access to this primary good without requiring large investments in a power grid. On the other hand, in these places, this investment is unfeasible for the low access to financial resources and the low population density.

In the end, a NG can operate in different contexts and according to different goals. Decision-making autonomy increases the intelligence level in the system. The NG can operate autonomously and indifferently in grid-connected and islanded mode; this means an increasing level of reliability inside the electrical system. All of these elements lead to technological advancement and an improvement in global social conditions.

1.3 …and tomorrow

Despite the small scale of nanogrid solutions, a number of familiar names are already active in these nanogrid markets; among them Bosch, Eaton, Emerson Network Power, Johnson Controls, and NRG Energy. While nanogrids can complement microgrids, there are times and situations when microgrids and nanogrids will be competing for the same customers [11].

Navigant Research forecasts that global nanogrid vendor revenue will grow from $37.8 billion in 2014 to $59.5 billion in 2023 [12]. Besides, Navigant Research found that the sector will be worth $17.5 billion annually by 2024. That represents a big part of the $25 billion value of the remote power systems by that year, according to Navigant. While there are viable nanogrid markets in every region of the world, the Middle East & Africa represents the greatest near-term and long-term market opportunity, with revenue forecast to increase from $14.4 billion in 2014 to $25.2 billion in 2023 (Fig. 1-2). The Asia Pacific region, which accounts for the second greatest share of annual nanogrid vendor revenue, is forecast to climb from $12.8 billion to $20.3 billion over the same timeframe.

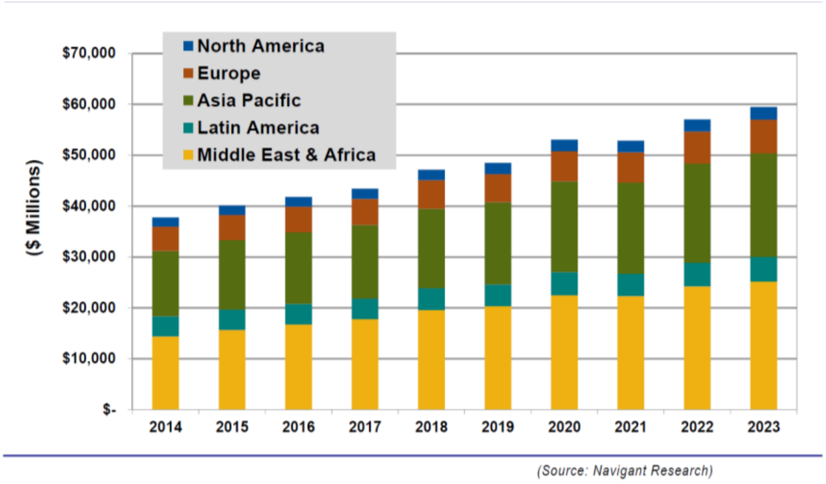


Fig. 1-2 Total Annual Nanogrid Vendor Revenue (Source: [12])

The spread of distributed generation and the rise of microgrids/nanogrids will also be shaped by: advanced control systems; more compact, smarter, and efficient electrical inverters; smart electricity meters and the burgeoning Internet of Things; and the ever-growing ability to extract actionable information from big data. Another improvement could derive from the increasing development of AI and machine learning algorithms. The Internet of Things is a boon for distributed generation because it is giving rise to industries that are producing sensors, microcontrollers, software, and other gear that will be easily and cheaply adaptable for use in future, data-driven grids and microgrids [13].

The new users are perfectly suitable to take part in new forms of management. New forms of management on a local basis, aim to re-organize local energy systems to integrate distributed energy resources, to aggregate consumption and production on a small scale, to realize energy brokerage between local communities and finally to optimally manage resources. The aggregation of users in [14] is called "Power Cloud".

The users will be part of an Integrated Community Energy Systems [15], where the coordinator manages the relationship between stakeholders, utilities and customers to maximize the community benefit [16]. The Energy Cloud in [17] is an example of emerging platforms where advanced technologies and solutions serve new ways to generate and distribute electricity.

The Integrated Community Energy Systems (ICESs) represents locally and collectively organized energy systems and combine the concept of sustainable energy communities, community energy systems, micro-grids community, and peer-to-peer energy. Integrating smart-grid technologies and demand side management increase reliability and efficiency of local energy systems. In this prospective, the Power Cloud (PC) is a management/business model applicable to ICES. The aggregator, called Community Energy Provider (CEP), manages the consumption and the energy generation, organizing sessions of local market and in general applying the Power Cloud management/business model. Obviously, the CEP is also the entity designated to interface market and electrical system operators, to guarantee the correct functionalities of the whole system.

In order to implement this innovative business/management model, innovative physical systems for the generation, transport and use of energy (based on power electronics, loT and ICT in general), such as the nanogrid for Home Application (nGfHA) presented in [14], for the active involvement of final user are essential. The nGfHA is able to transform a user from a simple "consumer", equipped with a storage system, to a "Prosumer", that can generate energy to provide to the PC where it belongs [18].

In [14], it is demonstrated the effectiveness of the nGfHA in the Power Cloud framework obtaining a greater economic savings for the aggregation's users equipped with this technology as a member of the Power Cloud. This type of business model will have future developments even for those types of non-domestic users or users with connection capacity higher than 3 kW. Two frameworks are considered: a stand-alone one, where the single user invests and purchases the nGfHA and the power cloud framework where the investment is provided by the CEP.

1.4 State of the Art

The nanogrid enables zero net-energy consumption and optimal power management for future homes or buildings, integrating multiple renewable energy sources. However, integrating all the sources and loads in a simple, reliable and smart way is still challenging.

The interesting development of small intelligent and autonomous networks is also evident from the enthusiasm of the scientific community (Fig. 1-3) and the attention of the industry [19], [20].

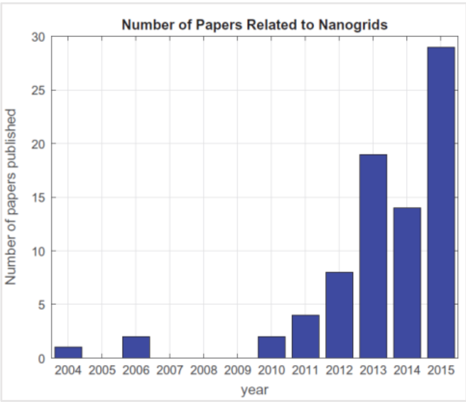


Fig. 1-3 Number of papers related to nanogrids in recent years (Source: [6]*)*

In [17] the idea of nanogrid is discussed and the following definition is provided:

A nanogrid is a single domain for voltage, reliability, and administration. It must have at least one load (sink of power, which could be storage) and at least one gateway to the outside. Electricity storage may or may not be present. Electricity sources are not part of the nanogrid, but often a source will be connected only to a single nanogrid. Interfaces to other power entities are through "gateways". Nanogrids implement power distribution only, not any functional aspects of devices. Components of a nanogrid are a controller, loads, storage (optional), and gateways.

The definition given in [6] is:

A nanogrid is a power distribution system for a single house/ small building, with the ability to connect or disconnect from other power entities via a gateway. It consists of local power production powering local loads, with the option of utilising energy storage and/or a control system.

[11] introduce the concept saying that:

The basic concept behind the nanogrid is simple: small is beautiful.

The importance of NGs lies in simplicity: this provides the reduction of technological development times and the magnitude of the required economic investment. In some cases, nanogrids represent an alternative path forward to increasing resiliency and renewable content into onsite power solutions.

For [11] nanogrids are modular building blocks for energy services that support applications ranging from emergency power for commercial buildings to the provision of basic electricity services for people living in extreme poverty.

In [21], the authors claim that without nanogrids, the smart grid won’t be brought to its full potential. In 2012 they sustained that the next step toward such fulfilment is to build a nanogrid hardware prototype and simulation model.

Ironically, nanogrids are big business. Deployments of nanogrids are already in place. This is because they face less technical and regulatory barriers than their microgrid counterparts.

The control systems applicable to NGs differs in the kind of interaction between the nanogrid and converters controllers. These are illustrated in the following chapter.

A widely used method within the NGs is a kind of distributed control, known as dc-bus signalling (DBS) [22]–[24]. In DBS control the converters communicate with each other through the dc bus voltage signal. As DC bus voltage changes, all power electronics converters - which interfaces with generation and storage units - independently determine both the working point and the operating mode. For the latter, electronic converters operate in constant voltage mode or constant power depending on the measured DC bus voltage level.

Qu et al. propose a DBS technique in [25]; the improvement consists of easy plug and play connection of generation units and electrical loads and in a smooth transition between grid-connected and islanded operating mode.

In [26] an energy management system (EMS) is developed to provide uninterrupted power supply of dc loads of a nanogrid to achieve self-sufficiency in energy by minimizing grid power consumption; a two-stages cascade converter (inverter plus a dual active bridge) interface the dc nanogrid. A bidirectional converter interface for a low-voltage DC nanogrid has been presented. The two-stage converter facilitates power transfer between the 48-V dc nanogrid and ac distribution network. DC bus signal technique allows independent control of a dual-active-bridge converter and the three-phase grid-connected inverter. The dc bus voltage is used to determine the mode of operation. Four voltage levels are defined, and these determine the transitions between the three modes (surplus, deficit, and idle).

Also in [27] is proposed a dc-bus signal strategy for a Smart User Network, that allows the power flows control and energy management. The grid-connected inverter, i.e. the power gateway interface, between the dc small grid and the utility grid is called Power Electronic Interface (PEI). The PEI can provide ancillary services to the utility grid.

A standalone nanogrid with a DC bus signal control is the subject of Schonberger et al. in [28]; the authors include the load shedding functionality in the system control strategy so to reduce the peak generation requirement and protect the system from a complete collapse under overload conditions.

[29] proposes nanogrid control to increase the efficient use of distributed renewable resources. The authors simulate a nanogrid in which the “Nanogrid Control Algorithm” monitors production and consumption, connecting the system to the grid when required, while the “Load Control Algorithm” implements demand side management, which utilises loads that have a thermal output.

In [23] a control structure for converters is presented to allow the implementation of a dc-bus signal technique and provides a procedure for implementing a system control law through independent control of the interface converters of sources and storage.

In [30] the authors present the experimental results obtained from a single-phase laboratory-scale prototype of a DC microgrid for residential applications. It is realized in the Laboratory of Electrical Energy Systems and Renewable Energy of University of Calabria. The power management of the Smart User Network is based on the DC bus signalling control logic to guarantee safe operation both in grid-connected that in stand-alone mode. A technical description of the laboratory-scale prototype components and of the control concepts are reported. Experimental results from the operation of the laboratory-scale prototype demonstrates the validity of the DBS as decentralized control for DC microgrid highlighting all the necessary steps to shift the technology from laboratory-scale to real industry.

In [31] a DC smart microgrid has been realized in a real-life application as a part of pilot site. Some electric and thermal loads, a PV rooftop plant, a building automation system and electrical and thermal storage system have been installed. The pilot site is integrated in the concept of a Smart Community of microgrids that is several areas interconnected from the point of view of energy production and consumption. The local controller implements the DC bus signalling. A distributed control strategy is preferred as the system becomes independent of a central controller. This control strategy maintains the reliability inherent in the structure of the system by using the DC bus itself as the communication link.

In [32] a power management scheme with DC bus voltage signal is proposed to maintain the power balance and stable operation of DC microgrid under any generation or load conditions. The operations of DC microgrid based on modular PV generation system are categorized into four modes. These four operation modes are identified by different DC bus voltage levels, which means the DC bus voltage is used as an information carrier.

Also in [33] a control strategy based on DC bus signal realizes the optimal allocation of power; the bus voltage is controlled by PV converter, battery converter and fuel cell converter unaided and time-sharing. It can minimize the costs of the overall system by giving the first priority to low-cost solar energy.

In [26] optimum power management has been proposed to reduce costs in a hybrid nanogrid in presence of photovoltaic, a diesel generator and batteries. The proposed algorithm is concerned with optimizing power flows in the system to supply the load, maximizing the use of renewable sources, minimizing fuel usage, and increasing battery life. In [34] is proposed an algorithm based on dynamic programming technique to optimize the power flows in the system to meet the load demand, achieve maximum utilization of the solar energy available, minimize fuel consumption and increase battery life cycle, simultaneously. The optimal power management approach to reduce the operating costs for hybrid nanogrid with batteries, photovoltaic and diesel generator.

The NG in [35] consists of an AC bus and a DC bus, interconnected through a tie-converter. All AC loads and distributed energy resources of the community houses are connected to the AC bus while all their DC loads and DERs are connected to the DC bus. The tie-converter exchanges the power between two buses and regulates the voltage in both buses, in the case of off-grid operation. The NG dynamics, with the proposed structure and developed primary and secondary control algorithms are validated through extensive computer simulations in PSCAD/EMTDC.

The concept of the energy control center, which interconnects the dc system to the traditional ac utility grid, is introduced, and the operation function of converter suitable for dc nanogrid application is defined in [36]. The future electric power distribution systems backbone will be also based on power electronic converters taking the role of energy control center, like an "energy router," used to interconnect several subsystems, such as the micro-, mini- and nanogrid. Dong Dong et al. present a two-stage single-phase PWM converter to connect the 380V dc bus of a dc NG to the 230V/50Hz utility grid in [36], [37]; in particular, two phase-legs are used as the full bridge to interface with the grid. The other phase leg is used as a bidirectional SR dc–dc converter to regulate the dc-bus voltage with a small voltage ripple and with a fast dynamic response. Short-circuit protection can be readily attained in either direction by turning OFF the second-stage or full-bridge switches. The black start is simply due to the "buck-type" topology on both sides.

A nanogrid testbed at Virginia Tech's Centre for Power Electronics Systems (CPES) is discussed in [38], [39]. The DC nanogrid testbed is comprised of a 5 kW photovoltaic array, a 3.5 kW vertical axis wind turbine and six 48 V lithium-ion batteries (connected in series) forming a 45 Ah battery bank. The grid, PV and batteries are interfaced with a DC bus (ranging between 360 V and 400 V) via three converters: the nanogrid testbed implements droop control for supply side management. The DC bus operates between 360 V and 400 V and the control allows 2.5% voltage droop from the nominal voltage.

A bidirectional converter is commonly used to transfer energy between the single-phase AC system and the DC system with stable operation in an AC-DC hybrid bus nanogrid. In [40] for long-time term support the single-phase AC-DC converter is designed to avoid using the short lifetime electrolytic capacitor. In this paper, a bidirectional single-phase AC-DC PWM converter using an additional line-level AC storage capacitor as the AC-DC power balancing device is proposed. A condition for zero ripple DC bus voltage is established such that a small DC bus capacitor can be used.

A switched boost inverter (SBI) can supply both dc and ac loads simultaneously from a single dc input. These features make the SBI suitable for dc nanogrid applications. In [41], the SBI is proposed as a power electronic interface in dc nanogrid. This paper also presents a dq synchronous-reference-frame-based controller for SBI, which regulates both dc and ac bus voltages of the nanogrid to their respective reference values under steady state as well as under dynamic load variation in the nanogrid. Unlike the traditional buck-type voltage source inverter (VSI), the SBI can produce an ac output voltage that is either greater or less than the available dc input voltage. Also, the SBI exhibits better electromagnetic interference noise immunity when compared to the VSI, which enables compact design of the power converter.

The local DC distribution system with limited management scope represents an infrastructure typically separate from that operated by the electric utilities through a bidirectional AC-DC converter [42], [43]. From a technology point of view, perhaps the most radical idea behind nanogrids is a clear preference for direct current solutions, whether these systems are connected to the grid or operate as standalone systems. Although dc offers advantages, such as the improved transmission efficiency and the ease of interfacing asynchronous sources, the protection of dc distribution systems against short circuit line fault and ground fault is still under discussion, especially for distribution systems and multisource nanogrids.

The preference for dc current is appealing but challenging from some point of views. One is the protection scheme to adopt to maintain an adequate level of security in the nanogrid. The challenge facing DC micro-grid protection is not only the lack of standards and guidelines, but also the relative lack of practical experience. Almost all of commercially available fuse and circuit breaker designs are optimized for AC current. The natural zero cross condition will aid the zero current condition. Therefore, commercial devices are sub-optimally applied to DC systems.

[44] analyses the feasibility of a DC network for commercial facilities: an alternative power system based on direct current is proposed for low-voltage commercial distribution. The paper studies the feasibility of such a

system by analysing an existing system as a “case study”. Voltage drop and power loss calculations are carried out to choose the most appropriate voltage level for supplying the loads. The authors sustain that in a low-voltage dc system the current interruption is not a problem. Ac breakers can be applied, provided that the ratings are adjusted by proper correction factors that are provided by the manufacturers.

High speed, high power AC to DC converters that interface power sources to the grid and include the functionality of the static switch between source and grid would enhance speed and simplicity. Intelligent system interruption, fault isolation and rapid recovery as opposed to reliance upon external protective devices. The protections are normally an important concern related to DC systems since it is more difficult (therefore more expensive) to extinguish a direct current. However, several authors point out that DC nanogrids are able to disconnect all the elements in the face of a fault, since they have a converter for the connection of each element [45]. In [45] dc circuit breakers are eliminated, proposing smaller grid segmenting contactors, by coordinating the action of these contactors with the action of electronic power converters that are commanded to briefly de-energize the distribution grid, so that the contactors are able to isolate the faulted branch and reconfigure the remaining network. The authors show how a power sequence control scheme can be used to protect and reconfigure a nanogrid dc distribution system in response to short circuit line faults faster than an ac grid can be protected and reconfigured using traditional circuit breakers, and ride-through capability for loads can be provided.

The protection scheme has to account for grounding and ground fault isolation. DC micro-grids need to be floating systems—i.e. there are very high DC and AC impedances between any line and ground. The floating is important for different reasons in the various applications: off-shore wind power systems require ungrounded cables to avoid corrosive effects; navy shipboard systems are floating to allow continuity in supply during a first ground fault; common mode voltage produced by inverters causes a “neutral shift” voltages: in some grid-connected applications with inverters the paths for neutral currents are broken through transformer isolation

In [46] there is a study of the type of passive components based on a two-stage ac-dc converter for dc nanogrid applications. The design of passive components is discussed considering the power density, power quality, EMI effects, and grounding patterns.

Is a DC nanogrid practical in the foreseeable future? Currently most consumer products in homes, retail stores and rolling out of production factories are still AC loads. To retrofit a current house with a DC power system will require either replacing AC loads with the limited DC compatible loads available to purchase or modifying the AC loads to function on DC power. Thus, although the increased efficiency of a DC system would equate to a financial saving in the long run, replacing/retrofitting AC loads will increase initial capital.

The widespread preference of dc NGs compared to ac NGs is mainly due to the fact that most of the domestic electric loads are powered by low-voltage DC power; Goikoetxea et al demonstrated the greater efficiency in feeding these loads using the DGs instead of the converted ac voltage in [47]; the authors analyse the state of the art of domestic DC applications and compares the efficiency of these systems in comparison to conventional AC systems.

The nanogrid should be designed to guarantee a modular structure. Besides, one of the advantages given to NGs is the inclination in the bottom-up approach. This approach means that a larger power system can be developed by interconnecting multiple nanogrids. It’s a sort of hierarchical approach to power distribution: from nanogrid to microgrid to the national grid. This hierarchy takes advantage of the semi-autonomous control structures within the nanogrid to alleviate some of the stresses caused by intermittent power production/consumption. Power can be shared between nanogrids via the network, decreasing the effects of an intermittent power supply and increasing the financial benefit of users. Multiple NGs can reside inside the same power community.

The concept of NG is not different from that of MG. Rather, a feature of the NGs should be the possibility of connecting themselves, within MGs and as well as through the meter connected to the macrogrid [17]. This requires interface standards that can be reliably implemented.

The majority of the nanogrid literature focuses on the control and hardware, with a variety of algorithms and power converter topologies being discussed. The idea of utilising the nanogrid's modular nature to form a network of interconnected nanogrids is also presented within the literature, though in most cases this is still at a conceptual level [6].

Nanogrids have an obvious propensity to interconnect each other within a microgrid. Werth et al. exploit such a propensity in [48]. The interconnected dc nanogrids return an open energy system as an alternative way of exchanging intermittent energy between houses in a local community where each house is equipped with a dc nanogrid, including photovoltaic panels and batteries.

Also Nordman exploits the interconnection of dc nanogrids in [49]; in the paper the interconnection returns a building-wide microgrid with a dc local power distribution, based on layered model of power called Network Power Integration. In order to control the power flow between interconnected nanogrids, Morais et al. present an interlink converter in [50] which joins two dc local grids at 48V to a 380V, respectively.

Although the theory behind nanogrid networks, in general, is still at a high level, exploration has begun and implementation is being pursued. The advantages and future research topics for nanogrid networks, outlined in [6], are:

- Bidirectional power sharing: by sharing power within the nanogrid network, the need to purchase power from an external source (national grid) is reduced. This equates to financial savings for the consumers within the network. Consumption/production peaks can be shaved by the various connected nanogrids, respecting the exchanged power profile with the grid.
- Communication: is what creates an intelligent network. There are multiple layers of communication within a nanogrid network and as with any information sharing, many technical and security based considerations.
- Financial benefits: are a motivating factor for operating a controlled nanogrid and is also an incentive for the interconnection of multiple nanogrids (nanogrids network).
- Reliability in power supply: withstanding to power grid outages is important now our lives rely so heavily on power. The nanogrid network has the ability to island itself from the national grid in the case of a black out.
- Gradual introduction: as nanogrids operate at a single house level is required a small investment. Instead, nanogrid networks can be integrated in to the existing national grid at a manageable rate.
- Grid stability: the interactivity, the flexibility and the communication give to the NGs the ability to respond quickly to commands from the utility grid. This gives the nanogrid network the opportunity to participate in grid stabilization, voltage and frequency control and real-time pricing at a national grid level.

Another critical point, often underestimated, is the degree of interoperability between controller and peripheral units; at a practical level, in order to avoid problems due to non-interoperability nanogrid objects, the objects involved will be purchased by a single manufacturer.

1.5 Thesis Contribution

In this thesis a control system for a nanogrid has been developed and implemented in a real-life application. The activity aspires to boost the technology in both academic and industrial sectors, in order to spread the use of the nanogrids in the domestic environments.

The control system realizes a nanogrid flexible, modular, reliable, robust, autonomous, efficient and easy-installable. The nanogrid becomes an autonomous entity able to follow some predefined behaviours. The behaviour is the set of actions that the nanogrid applies to optimise the power production, to exploit renewable energy sources, to supply continuously critical loads and to react to any sudden exogenous event that perturbs the physical system.

The sequence of the actions to apply are decided during the decision-making process. The actions are applied to power converters, the nanogrid building blocks. The power converters interface power sources and loads to the common dc bus of the nanogrid. The nanogrid controller controls the nanogrid behaviour according to the specified goal, selecting the operating mode for each converter. The high number of converters and the combinations of all possible operating modes of each converter ensure to put in place advanced control strategies, but at the same time complicate the control algorithm.

Most of the time, when designing a controller, the discrete side of the hybrid system is disregarded and only the continuous side is considered. This simplification is acceptable if the system control requirements are simple or if the practical implementation in a real physical system is not faced. Indeed, architectural design principles are often disregarded and when the implementation stage arrives, a negative choice of the architecture can significantly impede the system development.

The nanogrid system has been modelled like a hybrid dynamical system. The feedback controllers evolve in the continuous state space; the decision-making process evolves in the discrete state space. The careful design of the discrete control allows to the nanogrid controller to become an intelligent entity. This work shows the importance to formally face a complex control problem, designing a hybrid control, i.e. focusing on the discrete and continuous control designing stages.

A significant introduced innovation is the use of a Behaviour Tree (BT) to control the transitions from a state to another in the discrete system. The BT has been chosen because allows to develop intelligent agents, i.e. entities capable to react inside an unstructured environment, much better than other control architectures. The BT, in a more formal way, allows to design and implement the discrete control of a nanogrid. The BT defines the transactions between different kinds of activity, that is the switching between different controllers of the continuous low-level layer. The BT allows to an intelligent agent to reach a given goal without occurring in random, unnecessary or dangerous actions.

The behaviour is implemented in the control algorithm, which follows the simple structure of the tree. The whole behaviour is made by a sequence of sub-behaviours; the design of a sub-behaviour is independent from the overall behaviour, i.e. from the other modules. The sub-BTs can be designed recursively, adding more details (more smartness) time after time. Thanks to the BT the behaviour complexity is given by the composition of a high number of simple algorithms and not by the complexity of the algorithm.

The BT has proved to be a powerful technique. It is efficient, clear and practice. The translation from requirements in the common language to the syntax rules of BT is almost immediate. The behaviour is described in the structure tree, which has a clear graphical representation. The tree structure improves the performance of the algorithms. In this application the BT is executed under hard real-time constraints, with a relatively high frequency, by a microcontroller with modest performance.

Other than a theoretical description, the control problem is faced from a practical implementation. The thesis allows to move from a simulation environment to a laboratory setup toward a mature industrial product. Indeed, all the concerns not encountered during a prototype development have been faced and resolved thanks to the proposed control structure. The thesis wants to drive the reader from the requirements gathering to the final deployment. It is strongly pursued the purpose to present the work from a practical implementation side; indeed, the control algorithms and the software are illustrated in detail.

The hybrid control has proved to be a guide for the programmer during software design, coding and testing. In the work, much attention is reserved to the implementation stage and different prototypes working in real environments have been built. The practical implementation is a critical point, mainly if the project is continually evolving during the development stage; it is even more critical if the final goal of the research activity is to reach a mature system, as near as possible to a commercial system.

The features of the proposed control system are tested and proved in a testbed. The testbed is installed in a building in the campus of the University of Calabria. The building is a student residence of the university. The setup has three different NGs, connected to the same dc bus. The first nanogrid is connected to the utility grid and to a 4.5 kW PV plant. The second nanogrid is connected to a 232V/22 kWh storage system and supplies critical ac loads. The last nanogrid is used to interface the stirling generator.

In the testbed, among the various solution prospected, the attention is focused on a particular distributed strategy, known as dc-bus signal (DBS). The DBS is characterized by a reliable structure, because the communication infrastructure is the same power connection between the sources. This control structure is one of the most promising solution for NG applications, because ensures high reliability, inherent simplicity and low costs. The DBS logic is coded by the BT running inside each nanogrid.

In conclusion, the theoretical and practical approach to the control problem have allowed to develop a system near to the commercial deployment. The nanogrid becomes an actuator and a reactive entity with distributed intelligence inside the smart grid, which gives the opportunity to enable advanced energy management strategies and advanced business models inside the energy community.

In the end, this wants to be the proof-of-concept needed to attract the attention of industry, which is critical to creating this new technology.

In chapter two, the theoretical bases to design a control system for a nanogrid are presented. This literature study has required the contamination from different areas and the exploration of different fields, from power and control system to computer science. First, the common power sharing strategies for nanogrids are illustrated, focusing on dc-bus signal control. Then the most used control architectures are presented; the focus is on the recent technique, known as Behaviour Tree (BT). The control architecture describes how certain activities are performed. The chosen control architecture influences how the system operates in an "unstructured" environment. An unstructured environment is an environment with an unpredictable behaviour in space and time. The control architecture defines the nanogrid. The nanogrid may impact the daily life only if the designer carefully chooses the control architecture and focus on its design.

In chapter three, the proposed control structure for a nanogrid, known as nanoGrid for Home Application (nGfHA) is presented. The nGfHA is a nanogrid with the control boards and all the power electronics enclosed in one device. The only one controller of this device is the nanogrid controller. The proposed control structure, implemented in the nanogrid controller, allows to an already existing home power system to become a nanogrid, allows to the user to became member of a coalition to increase the nanogrid economic benefits, and allows to a remote area to become electrically independent, assuring high levels of power quality and reliability. Thanks to the proposed control system, the presented nanogrid is simple to setup, but it is powerful in its features.

The analytical models for the nGfHA converters and the continuous control, designed using the developed analytical models, are presented. The numerical results of the numerical simulations are provided. In the end of the chapter, two BTs for the nGfHA are presented, because the decision-making process is accurately and clearly designed. This approach allows to give intelligence to the NG controller.

In chapter four, a detailed description about the implemented code in the discussed nanogrid controller is provided. The software design has been focused on developing a modular, easily changeable and reconfigurable software in a bare-metal environment. The software has been used for a number of applications with different energy sources and configurations, to demonstrate the reached flexibility and modularity. These features allow to use the discussed nanogrid in a variety of applications. The nanogrid supervisor has been also designed and implemented and also the supervisor software is illustrated. The nanogrid supervisor allows to the nanogrid to interact with the external world. The specific implementation in the testbed presented in this thesis is illustrated, but the idea is to generalize the structure to embrace other devices. The used IDE allows access to powerful libraries, which have been used to develop a versatile environment, where different kind of objects can interact to pursue a common goal, i.e. the home energy management.

In the last chapter, the testbed, where the proposed nanogrid control system is implemented, is presented. The setup is illustrated and some experimental results have been reported, to demonstrate the effectiveness and the validity of the proposed control structure.

2 Nanogrid Structure and Control

The emerging technologies in renewable and distributed generation can have lower emissions and costs. The microgrid/nanogrid concept gives a solution for integration of a large number of distributed generations without causing disruption in the utility network. The small grids also allow the local control of the distributed generation units and the flexibility to operate autonomously during disturbances in the utility network to increase reliability.

The control and information structure of a nanogrid should be similar to the whole concept of "smart grid"; the dc nanogrid features an information infrastructure, but is envisioned to be communication independent for the basic functioning. This means that the nanogrid system is an autonomous entity able to follow some predefined rules and each converter is able to detect its mode of operation without receiving specific commands from the leading system. Hence, the nanogrid should be a plug-and-play system where the source converters work independently from each other, but are aware of the nanogrid's operating mode.

The nanogrid control gives the ability to coordinate multiple sources and optimises power production and consumption. It is the "brain" of the system and if implemented correctly can increase the efficient operation of the nanogrid. While the community focused on nanogrid has a strong preference for dc power distribution, there isn't a clear predilection for the control structure. The control structure defines how each converter is involved in the power flows management. To manage multiple sources various control structures can be adopted.

A general view to microgrid control structure is provided. However, the nanogrid preferences for renewable energy sources and the lower number of sources make the nanogrid structure slightly different than microgrid. The classical solutions to integrate different kinds of resources inside the nanogrid are presented: for each solution advantages and disadvantages are highlighted. The three main control structures are: centralized, decentralized and distributed [35].

Between the various solution prospected, the attention is focused on a particular distributed strategy, known as dc-bus signal (DBS). The DBS is characterized by a reliable structure, thanks to the communication infrastructure, that coincides with the same power connection between the sources. This control structure is one of the most promising for nanogrid applications, because ensures high reliability, inherent simplicity and low costs.

A nanogrid works in a heterogeneous and unpredictable environment. At any time, solar production may stop or grid connection may fail. Moreover, despite the small size of the nanogrid, when an increasing number of entities is involved in the nanogrid operating, the number of possible operating modes increases as well. Therefore, each intelligent entity in the nanogrid, such as the converters controllers, needs a tool, i.e. a control architecture, that makes simpler the implementation of the behaviour rules. The control architecture should allow the management of the nanogrid behaviour, despite the complexity of the system. The nanogrid modular and flexible behaviour may impact the daily life only if the designer carefully chooses the control architecture and focus on its design.

In general, the features of an intelligent device are codified in the control architecture (CA). The control architecture describes how certain activities are performed. The chosen control architecture influences how the system operates in an "unstructured" environment. An unstructured environment is an environment with an unpredictable behaviour in space and time.

In this chapter, the theoretical bases to design a control system for a nanogrid are presented. This literature study has required the contamination from different areas and the exploration of different fields, from power and control system to computer science. First, the common power sharing strategies for nanogrids are illustrated, focusing on DBS. Then the most used CAs are presented; the focus is on the recent technique, known as Behaviour Tree (BT).

2.1 Energy management systems for micro- and nano- grid

The control structure allows the proper integration of sources inside the small power systems. Regarding the microgrids (MGs), the control structure can take advantage from the experience and qualities of the electrical system. A compromise between fully centralized or fully decentralized control schemes can be achieved through a 3-level hierarchical control [51] [52]. These levels of control differ in response speeds and time intervals, but also in the requirements of the communication infrastructure. Although microgrids do not have the same territorial expansion of the power system, they have a high number of controlled resources and stringent performance requirements; they can benefit from hierarchical control. The hierarchical control provides, therefore, three levels of control. A brief description of each of the three layer is in Table 2-1. These layers have no defined boundaries, rather they are just conceptual subdivisions to simplify the problem, also because there is no standard. For example, in [52] the third level of the hierarchy is considered the EMS of the DC MG, rather than to belong to the host-grid.

Table 2-1 Layers description in a hierarchical control

	Description
Primary control	Dynamic response to avoid system collapse, sharing the power between the available units in real-time.
Secondary Control	This level is responsible for the secure and economic operating of the grid, sharing the loads between the generation units. This task is difficult in isolated grid in presence of variable sources, where the updating of the set-points should be fast enough to follow the sudden change in load demand and not-dispatchable generators. This level works at a lower rate to decouple the dynamic from the first level control, to reduce the bandwidth requirement of the communication infrastructure and to allow the time needed to perform the complex calculation in taking decisions.
Tertiary control	This level is responsible for the coordination of the operating of more MGs, that interact between them and to interact with the host grid for ancillary services (voltage support, frequency regulation, etc…). This level can be considered belonging to the host-grid, rather than the MG.

The MG literature shows that the energy management has this three hierarchical levels [53]:

- distribution network operator (DNO) and market operator (MO);
- microgrid central controller (MGCC);
- local controllers (LCs) associated with each distributed energy resource (DER)/load unit.

The MO is responsible for exchanging information between the microgrid and the electricity market. DNO is a high-level management system that aggregates real-time information and operating commands among multiple microgrids and utility grids. MGCC serves as a gateway between the DNO/MO and LCs within the microgrid. The idea is that microgrid energy management system (EMS) is an information and control center embedded in the MGCC.

The EMS is the control software that can optimally allocate the power output among the distributed generation (DG) units. An EMS monitors, controls, and optimizes the operation of a microgrid to achieve certain operational objectives (e.g., minimize costs). In general, a sophisticated EMS has to operate and coordinate a variety of DGs, DERs, and loads in order to provide high-quality, reliable, sustainable, and environmentally friendly energy in a cost-effective way. The microgrid EMS receives the load and energy resource forecasting data, customer information/preference, policy, and electricity market information to determine the best available controls on power flow, utility power purchases, load dispatch, and DG scheduling [53] (Fig. 2-1).

Energy management in microgrids is typically formulated as a non-linear and offline optimization problem for day-ahead scheduling. Most of the offline approaches assume perfect forecasting of the renewables, of the demands, and of the market, which is difficult to achieve in practice due to the intermittency and variability of renewables, the spatial and temporal uncertainty in controllable loads (e.g., EVs), and the randomness in real-time pricing. Various centralized methods have been proposed to solve it in the literature.

The centralized approaches require high computational capabilities at the MGCC, which is neither efficient nor scalable. Moreover, a centralized EMS requires the MGCC to gather information from DERs (e.g., production costs, constraints, etc.) and loads (e.g., customer preferences, constraints, etc.) as the inputs for optimization. A central approach is based on single entity, which controls the operation of each source and load, as for example in [54], where the authors have been presented a brief review of the existing EMS architectures for microgrids, identifying the main advantages of each approach, and has proposed a centralized EMS architecture for implementation on isolated microgrids in stand-alone mode of operation.

In contrast to centralized control, distributed control in a decentralized EMS constitutes a framework where each component is regulated by one or more local controllers rather than being governed by a central master controller. Every local control monitors and communicates with the other local controllers through the communication network. The local controllers have the intelligence to make operational decisions on their own. But it should be mentioned that a MGCC and its associated EMS still play an important role in even this decentralized framework. For example, an EMS in a decentralized microgrid exchanges energy price information with the DNO and MO and is able to take over the control of the local regulator from the system level in the event of serious contingencies and equipment failure [53].

Another classification of the EMS is possible through a description of the actions in a timing scale. The actions can be divided in long-term energy management, including hourly adjustment and daily programs, and short-term power balancing, including voltage regulation and real-time power sharing. This can produce multi-EMS strategy as in [55], in which according to the different management objectives, the proposed energy supervision system is implemented in two locations: a central EMS of the whole microgrid for the long-term energy management and a local EMS for the short term power balancing.

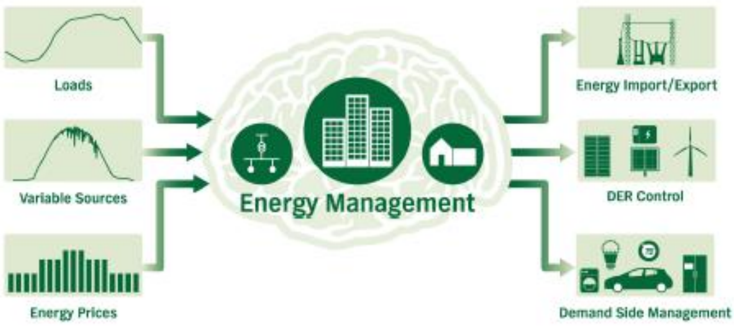


Fig. 2-1 Inputs and outputs of a MG Energy Management System

This view/solution is similar to the primary, secondary and tertiary control of the ac utility grid. The primary control implements the power sharing strategy in real-time, while the secondary control, acting as EMS, ensure the reliable and economical operation of the grid.

In small grid, like nanogrid (NG), the role distinction between EMS and local control could be not always clear as in the previous hierarchical control, also because the NG has a different structure, characterized by renewable energy sources (RESs) and battery energy sources (BESs) and low number of sources compared to MGs. Sometimes the local control, which executes a proper power sharing strategy to avoid NG collapse, provide a very basic EMS: the EMS works in real-time to prevent NG collapse using only the local information. Sometimes the distinction between the local control and the EMS is only algorithmic, i.e. the same processor executes the power sharing strategy and the EMS, while sometimes the two entities are separated and are executed on different machines – the EMS exerts its operation through the control infrastructure.

Probably, the best choose is to maintain the two entities separated, at least logically: the reason is in the different time horizons of the two process. Indeed, the controller works with a sample period at most of a few ms, while the EMS commonly updates its output every 15 minutes for grid-connected applications and of a few minutes for stand-alone installations to face the great variability of loads and sources. Besides, the complexity of the calculation of an advanced EMS could require long time, not compatible with the critical activities of the controller.

The EMS goal coincides with the NG one: the goal of nanogrid control is to optimise supply and consumption patterns to create a more efficient system. This is usually quantified by the savings and the reliable power supply gained from a controlled nanogrid with an advanced strategy compared to a nanogrid with a weak control strategy. Since many devices managed by the EMS reside on the customer side, they require certain level of autonomy and local intelligence that the EMS must be able to provide. A nanogrid should control the important systems inside the home. But most of them still use proprietary protocols and cannot interoperate between them. The EMS must handle the heterogeneity and achieve interoperation. Advanced EMS should interact with load, allowing the actuation of local demand-response, to avoid critical operating mode inside the NG.

The local control has to ensure the power balancing in the grid, i.e. $P_{load} = P_{source}$. This balancing gives two degree of freedom. To each degree corresponds a different kind of management strategy:

- Demand Side Management -DSM: the load control is often discrete, on-off. This regulation is not suitable to ensure a precise power balancing. An alternative to the boolean control is the regulation of the supply voltage; however, this control can't be applied to all kind of loads, because they are designed to work at rated voltage and the power absorbed can't be modulated.
- Source Side Management - SSM: the power produced can be precise regulated to balance the grid. Not all sources can be used to control the balancing in the grid, because not all of them are bidirectional. The source management solution is the most used in the MG/NG control.

Both the supply and demand are extremely dynamic, frequently changing from maximum to minimum consumption/production during a single day [56]. The main control issue in a nanogrid is maintaining the power balance in the system in presence of stochastic sources and loads. Different power sharing strategies can be adopted in the primary control: the most adopted one in MGs is the droop control. However, there is not preference for NGs.

Greater operational reliability is achieved if decentralized control is implemented; in this case, the multiple controllers ensure a fast and robust NG control. In addition, in the case of dc nanogrid is not necessary to set up any communication system, wired or wireless, for coordinating the controllers. To this end, the well-known voltage droop technique enables an easy deployment of decentralized control. More precisely, the voltage droop technique allows power sharing between the generation units, but not their scheduling. This limitation contrasts with the need to allocate dispatch priority to units like renewable sources.

The coordination of low level control is a challenge in a NG for the highly variable nature of sources and loads. The dynamic behaviour of the nanogrid is of high importance, especially due to the fact that nanogrid is envisioned to be an autonomous "plug-and-play" system where its components (sources and loads) made by diverse manufacturers and performing different dynamics have to cooperate and work stable and reliably. Such system features high dependability on the load requirements and availability of the sources, could rapidly change configuration in very short periods of time. Consequently, stability of the nanogrid should not be taken for granted, and must be carefully addressed and analysed, mostly because the reduced extension of the grid and the relatively low number of sources doesn't guarantee enough inertia to allow a slow dynamic response.

2.2 *Classification of control structures for nanogrids*

The structure of the overall control system defines the control topology and an overview is presented in [56]. The control topologies differ for the information and control infrastructure. Each topology has advantages and disadvantages; the pros and cons of each configuration are evaluated according to the following criteria:

- Reliability;
- Modularity;
- Economic viability;
- Efficient power sharing;
- Aptitude to implement a dynamic control law.

The three basic control topologies that can be used in a nanogrid are central, decentralised, and distributed control. These topologies are illustrated in Fig. 2-2.

A. <u>CENTRAL CONTROL</u>

The first control topology is central control, shown in Fig. 2-2. A fast central controller controls each source in real-time by measuring parameters throughout the system, performing control calculations, then adjusting the output of each source to balance the load. Since all control action is performed in the central controller, a high-bandwidth communications link is required to convey the control information with minimum delay. In this scheme all the control layers are executed in the same processor. The separation between the control layers could be algorithmic or, in opposition, could be only the primary control which acts like a basic EMS. To avoid delay in the actuation fast processors must coexist with fast algorithm; so, it’s hard to envision that this scheme could be used to apply complex and advanced (long-term) EMS strategies.

Advantage:

- A supply-side control law can be implemented easily, given that the central controller is aware of each node in the system;
- Ability to dynamically change the control law, for example to alter the utilisation priority of the generators such that the most economic mix of generation supplies the load.

Disadvantages:

- Reliability of the system is degraded – single-point of failure;
- Real-time control of a large multi-input/multi-output nonlinear system is not an easy task and it requires high computing power. For these reasons, central control alone is not commonly used for the control of power systems;
- High costs to guarantee reliability with redundancies and backup systems. The costs increase as distances between generators and central unit rise.

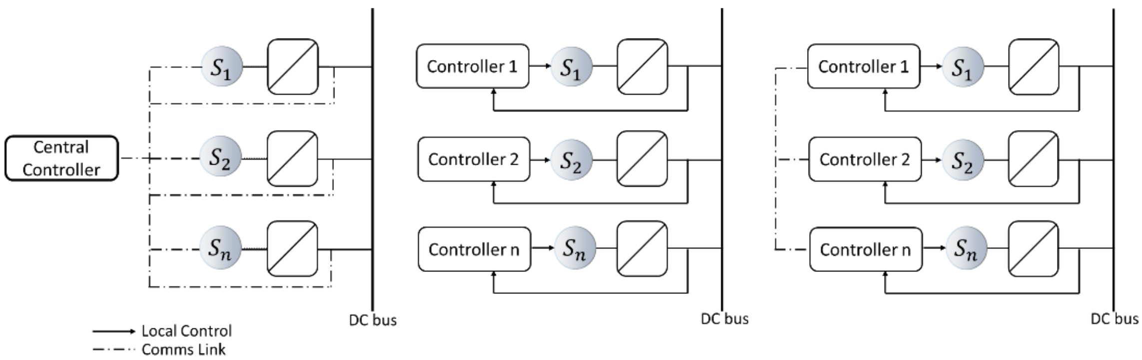


Fig. 2-2 Nanogrid Control topologies: central control (left);decentralised control (central); distributed control (right)

B. DECENTRALISED CONTROL

Decentralised control is fast and maintains the reliability inherent in the electrical structure of the system. This control strategy is widely used to control the instantaneous power sharing between parallel-connected sources in both ac and dc systems.

In dc systems, decentralised power sharing between parallel-connected sources is typically implemented using voltage droop control. Without voltage droop, the current supplied by each source to the system is governed by the interconnection impedance between the source itself and the load. Designing each source with a voltage droop characteristic allows the current supplied by each source to be controlled, even in presence of interconnection impedance.

Decentralised control provides only a primary control to implement a power sharing strategy. Indeed, the EMS is the same static power sharing implemented by the primary control, so it's not possible to change dynamically the strategy to face new operating modes or to perform dynamic optimization.

Advantages:

- Fast and reliable;
- No dependency on a single controller and communications link.

Disadvantages:

- Each source node is unaware of the other source nodes in the system;
- Energy production from RESs can't be maximized if all converter obeys to the same rule;
- Static, no on-line optimization possible.

C. DISTRIBUTED CONTROL

Distributed control exhibits characteristics present in both central and decentralised control topologies, having an external communications link but no central controller. The control law is implemented by embedding the portion of the control law related to the operation of each source in the source's local controller. Each source controller communicates with the other nodes that affect its operation in order to determine its own mode of operation. Although controlling a power system in a distributed fashion is possible, this control strategy is not widely used since it cannot match the performance and control flexibility.

Advantages:

- Reliability of the system is improved since the system is independent from the central controller.

Disadvantages:

- The system is still dependent on an external communications link for correct operation;
- Since the control law is embedded in each source, it cannot be readily altered to permit dynamic operation;
- Complex design and problematic plug-and-play.

D. HYBRID POWER STRUCTURES

The basic control topologies can be combined to form hybrid control strategies that exhibit the advantages present in their constituent control strategies. Two hybrid approach are possible:

- Hybrid central control topology uses local controllers for instantaneous power balancing and a central controller like coordinator;

- Hybrid distributed control is based on an exchange of information between local controllers carried over power lines.

In [56] hybrid topologies control systems are presented for NGs.

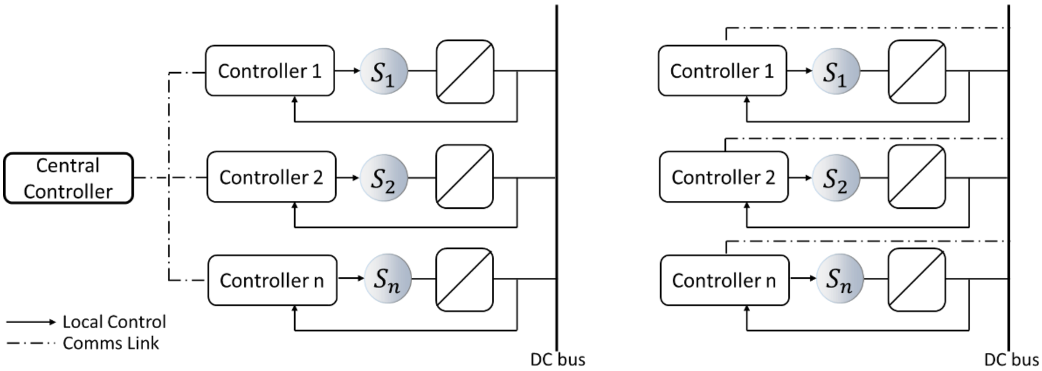


Fig. 2-3 Nanogrid hybrid control topologies: hybrid central control (left); hybrid distributed control (right)

HYBRID CENTRAL CONTROL

Hybrid central control, a combination of central and decentralised control, is a hierarchical topology as shown in Fig. 2-3. Decentralised control is used for instantaneous power sharing at the source level, relieving the control burden on the central controller. The task of the central controller becomes one of central coordination rather than real-time control of each node. In this scheme, the distinction between low level control and EMS is clear. The primary control that implements a power sharing strategy is executed in the local control, while the central control performs the EMS role.

The local decentralised control can be a control such as droop control, which performs instantaneous power sharing. Coordination and management of the system is accomplished using a supervisory control and data acquisition system and an automatic generation control program running on a central computer. The central control system updates the set-point of each generator, while control of each generator's power output is performed locally.

To improve the reliability of the control strategy, redundant controllers and communications links must be included at an extra cost. Nevertheless, hybrid central control generally offers the best compromise between performance and reliability.

Advantages:

- The presence of a central controller allows a dynamic supply-side control law;
- If the NG can operate for the basic functioning without the central control the system is also reliable.

Disadvantages:

- The system remains dependent on the central controller and communications link for correct operation.

The EMS has to be informed each time a new source is joined to the grid, so the info about the new source can be used as input in the EMS algorithm. The central controller of such control topology should holds the following functionality:

- Acquires primary data;
- Scheduling of operation and set points for interface converters;

- Defines the price function, based on difference between predicting load patterns and generation patterns (weather forecast is used for RES);
- Controls loads operations – shifting, shedding, scheduling;

In [54] the CEMS architecture consists of a central controller provided with the relevant information of every distributed energy resource within the microgrid and the microgrid itself (e.g., cost functions, technical characteristics/limitations, network parameters and mode of operation), as well as the information from forecasting systems (e.g., local load, wind speed, solar radiation), in order to determine an appropriate unit commitment and dispatch of the resources according to the selected objective.

HYBRID DISTRIBUTED CONTROL

Hybrid distributed control is a control strategy in which the control function is distributed among the source controllers as with distributed control. However, the communication between controllers takes place over the system power bus rather than an external communications link. Thus each source is effectively controlled using local variables as with decentralised control. Hybrid distributed control combines the advantages of both distributed and decentralised control strategies.

Each source performs the local strategy. However, the local strategy takes into account the information transmitted through the power lines. The separation layers in the control structure could exists locally, even if only algorithmic.

Advantages:

- The system has the same reliability advantages as decentralised control since the system is independent of a central controller and external communications link.
- The lack of an external communications link also reduces the implementation cost of this control strategy.
- Simple and reliable

Disadvantages:

- A supply-side control law implemented using this strategy is static for all operating conditions since it cannot be readily altered.

Although hybrid distributed control is not as widely used in practice as hybrid central control, it is employed in certain niche applications. One of the applications in which this hybrid control strategy is adopted is industrial control schemes that rely on a power line carrier (PLC) system to provide communications between control nodes. However, additional control hardware is required at each node to allow for the modulation and demodulation of the control signals.

Another application for which hybrid distributed control has been proposed is for implementing a demand-side management scheme in a dc network [28]. The system, comprising controlled rectifiers feeding a dc bus, uses inverters to supply ac loads from the dc bus. The rectifiers are controlled with an active droop characteristic such that the bus voltage falls when there is a power shortage or system overload. Each inverter is then controlled based on the level of the dc bus. A voltage threshold is assigned to each inverter, and if the dc bus decreases below this threshold, the inverter shuts down. This action restores the power balance in the system, allowing the high-priority loads to enjoy an uninterrupted supply of power.

The only major drawback with this scheme is that because it relies on voltage levels to convey system information, the presence of transmission line impedance has the potential to corrupt the information conveyed between sources.

In this scheme the functionality of the intelligent grid is dispersed between its elements. Each intelligent source or load provides its own schedule of operation; proposed distributed system still have central node, which holds functionality of SCADA system, which monitors and stores parameters and provides information to the grid, like weather forecast.

2.2.1 Dc Bus Signal (DBS)

DC bus signalling (DBS) is the term used to describe the mechanism by which the source interface converters in a nanogrid are controlled to implement a supply-side management strategy using hybrid distributed control.

The dc bus voltage is the variable used to inform the devices, which operate inside the NG, about the overall state of the grid: the more voltage is high, the more the local production of energy is high and vice versa. This simple claim allows to build up the rules to implement the DBS logic. Worth noting, that the information coming from the dc bus is not used “analogically”, which is what it happens in droop control. Indeed, the information is not used like controlled variable, but is used like information on the NG operating mode. The voltage value is used to activate the operating mode of the converters. Moreover, this variable can be modified by the same converters; when a converter modifies its value, it changes the status of the grid and the other converters react consequently to this change. The operative range of the voltage is obviously limited by the physic of the components between $V_{\max_{\mathrm{abs}}}$ and $V_{\min_{\mathrm{abs}}}$. Within this range the operating thresholds are defined; each threshold is separated by a certain ΔV.

Each converter works in a definite operating mode inside an assigned voltage threshold. The converters not only respond to the level of the dc bus, but they also change the level of the dc bus when their maximum power output is exceeded, thus automatically controlling other converters in the system. The NG is so auto-regulated.

The behaviour of each converter is defined by the status of the source, that is its capability to supply or absorb energy, other than the dc bus level.

The interface converter exhibits three modes of operation: off, constant voltage, and constant power. To each threshold is settled at least one converter that works in Voltage Control (VC). When the voltage is within the i-th threshold, the i-th converter works in VC. As a consequence, the converter forces the voltage to the reference value of the i-th threshold. The converter acting as voltage source enables the operating mode of the others converter in the NG. The VC mode works until the source capability to supply or absorb power is not saturated: if this condition occurs the converter switch to constant power operation. As a consequence, the voltage level on the dc bus changes. These voltage level changes in turn affect the operation of other source/storage interface converters with different voltage thresholds.

The mechanism of DBS works correctly only if each source comes online instantly when the bus voltage triggers its voltage threshold, before it triggers a new threshold. For storage nodes, this requirement doesn’t present ideally problems, even if the real dynamic of the converters will be essential to ensure the proper functioning of the logic; however, for nodes such as fuel cells and backup generators, a finite start-up time or ramp up time prevents the source from immediately supplying the sudden power demand. Consequently, the bus voltage will tend to collapse due to the power shortfall. As the bus voltage decreases, sources with a lower priority that can respond instantly will come online or loads will be shed. In this condition, a long-term start-up sources should be combined with buffer sources.

It should be noted that a droop characteristic may be added to the converter in constant voltage mode to allow it to share power with any other storage nodes that are present in the system. The droop characteristic would avoid a high number of threshold or, anyway, different thresholds for the same kind of sources. When operating in constant voltage mode, the storage interface converter may limit the power it draws from the system to prevent the storage device from charging at a dangerously high rate. This mode of operation is known as constant power limit.

The order of the thresholds is established a-priori and can't be casual. The order is important to implement an effective power sharing strategy.

With central control, the power supplied by each source to the network is set by the central controller. The central controller, aware of each node in the system, dispatches the power to be supplied by each node. With DBS however, the power supplied by each source is not so precisely controlled.

When multiple sources are assigned to the same voltage threshold, they may not come online at exactly the same instant. This phenomenon is the result of transmission line impedance causing the dc bus voltage to propagate unequally through the system. Sources assigned to the same threshold experience different voltages if they are connected to different points of the transmission network.

This indeterminate behaviour may be insignificant when the interconnection impedance between a group of sources assigned to the same voltage threshold is small. However, with a large interconnection impedance, such as in the case of two sources with the same voltage threshold being connected to the two extremes of a network, the behaviour may be significant. Droop could be used in a bid to minimise this effect and to maintain power sharing between sources of the same priority at a predetermined ratio. Even with the inclusion of droop, the transmission line impedance still impacts on the power sharing between sources.

To allow power dispatch based on a dynamic market, a more advanced control system, such as a hybrid central control system, is needed. The control system must change the utilisation priority of the generators in the system according to generation costs.

Although DBS does not permit a dynamic control law, this type of control law is unnecessary in a system that is primarily based on uncontrollable renewable sources. Optimal economical operation is achieved by maximising use of the renewable sources. The system is sized such that the renewable sources supply the average load demand; the controllable sources, the storage and backup generation, are only brought online to maintain the power balance in the system. With such an arrangement, a fixed control law is all that is necessary to manage the sources because the utilisation priority of the sources remains fixed.

The control law implemented using DBS is static, determined by the voltage threshold to which each source is assigned. In a free-market system, a dynamic control law is needed since the utilisation priority of each source changes as generation prices fluctuate [56].

In this work some of the DBS limits are overcome: the problem to dynamically change the operating mode has been solved thanks to a flexible algorithm combined with the coordination of a supervisor controller.

2.3 Hybrid systems, control architectures and behaviour trees

A nanogrid works in a heterogeneous and unpredictable environment. At any time, solar production may stop or grid connection may fail. Moreover, despite the small size of the nanogrid when an increasing number of entities is involved in the nanogrid operating, the number of possible operating modes increases as well. Therefore, each intelligent entity in the nanogrid, such as the converters controllers, needs a tool, i.e. a control architecture, that makes simpler the implementation of the behaviour rules. The control architecture should allow the management of the nanogrid behaviour, despite the complexity of the system. The nanogrid modular and flexible behaviour may impact the daily life only if the designer carefully chooses the control architecture and focus on its design.

To have an idea of the importance of the control architecture an example is given. The idea of autonomous vehicles is one of the most complex idea of our time, but despite the technological progress, from sensors to machine learning algorithms, the vehicles are far to be used in the daily life. The vehicles are already intelligent and able turn left or right, speed up or down following the street; the difficulties and uncertainties are related to a non-deterministic reaction in the face of an unpredictable event. A once-for-all-programming can't be used due to the unpredictability and the variability of this kind of environment; on the other hand, a manual and iterative upgrade makes the project useless. Not by chance, nowadays the scientific community works to realize the efficient operation of intelligent devices in an unstructured environment. The autonomous vehicles will spread only when the right control architecture will allow to operate with reliability and safe inside the unstructured environment of the street.

In general, the features of an intelligent device are codified in the control architecture (CA). The control architecture describes how certain activities are performed. The chosen control architecture influences how the system operates in an "unstructured" environment. An unstructured environment is an environment with an unpredictable behaviour in space and time.

Among the features that a control architecture should show, there are [58]:

- Hierarchical organization;
- Continual closed-loop execution: in unstructured environments the action can't be planned offline and the open loop control is not applicable;
- Reusable code: code reusability is essential in complex long-terms projects. The ability to reuse code allows to build different projects using the same code;
- Modular design: ability to build big project, starting from small independent software modules;
- Human readable: the reduction of developing and debugging costs need a legible structure. Structure should be kept legible even for big systems;
- Expressive: CA must be sufficiently expressive to encode a large variety of tasks;
- Suitable for analysis: critical applications need deep and detailed analysis of qualitative and quantitative qualities, such as:
    - Safety: avoid damaging and irreversible behaviours;
    - Robustness: great operating domain inside the state space;
    - Efficiency: time to complete a task;
    - Reliability: probability to succeed;
    - Composability: conservation of features when activities are composed;
    - Suitable for automatic synthesis: autonomous synthesis of the actions order to complete an activity by using task-planning or machine learning techniques.

There are many control systems' architectures that are currently used; each of them has its own specific benefits and drawbacks. Now, starting from the classification in [58], it's possible to highlight advantages and disadvantages of the most important used control architectures (Table 2-2).

The most common architecture is the Finite State Machine (FSM). It represents the mathematical model where the system may be in a finite number of states. A FSM moves from a state to another reacting to some events; this change from a state to another is called "transition". The FSM is widespread in numerous sectors and is aimed to implement and coordinate the devices' behaviour (or behaviours). The behaviour is described in the graphical model. This important technique is described by the graphic model that makes the algorithm readily comprehensible; and furthermore, since it requires the execution of one state at a time, the FSM appears to be computationally feasible and efficient. The current state decides if making the transition or not, according to specific conditions (for the current state).

This kind of control architecture is widespread because it has two important benefits:

- Intuitive structure: readily comprehensible for a human operator;
- Easy to implement: FSMs can be implemented in sequential programming languages.

Anyway, FSMs start to show their limits when the activities are complex and when the number of state increases; for *N* states there will be *NxN* possible transitions. Given that *N* could be very high, then the number of transitions becomes hard to manage. Furthermore, the amends and corrections in one state implicates modifications in the whole model of the FSM: the modifications become very lengthy, time-consuming, error-prone and very costly.

In FSMs may be identified the following drawbacks:

- Maintainability: adding or removing states requires to re-evaluate all the transition and the internal state of the FSM. This implies that the FSMs are widely sensitive to the errors of human design;
- Scalability: FSMs with many states and transitions are difficult to be modified, both by computers and programmers;
- Reusability: the transitions depends on internal variable, thus the use in different project of the same code used for an activity is impractical.

An evolution of the FSMs is the Hierarchical FSM (HFSMs). HFSMs try to solve some of the drawbacks of the FSMs. In a HFSMs, one state may involve one or more states: in this case it is called superstate. A transition from a superstate to another is called generalized transition, which prevents the recurrence of redundant transition.

Another architecture is the subsumption architecture. To implement an activity different controllers are executed in parallel. Each controller produces both the commands for the actuation stage and a binary value. The binary value is a boolean that indicates if the i-th controller is active or inactive. The controllers are ordered by a priority order, previously determined. Thus, a controller with a higher priority is able to subsume a lower level one.

The Sequential Behavior Composition increases the domain of a controller using many other controllers, where the asymptotical static equilibrium is the objective of each controller or it can be found in the region of attraction of the following controller. Given that each controller moves the state towards its own local objective, the state goes through the domains of the controllers. This process is repeated until the state achieves the final controller's domain, that leads the state to the final objective.

The need to get a specific objective, reactive to external stimuli from the environment has led to the Teleo-Reactive (TR) programs. A TR program consists of a set of rules conditions-actions, ordered by a certain priority. The condition-action rules are constantly scanned as long as a condition will be verified; that's when the corresponding action will be performed. The actions, in a TR program, are usually durative; a durative action continues indefinitely in time and is executed as long as its corresponding condition remains the condition with the highest priority among the ones that it holds. Thus, the conditions must be evaluated continuously so that the action associated with the current highest priority condition that holds, is always the one being executed.

A Decision Tree (DT) makes possible to achieve an objective, to make a decision, with a list of nested "if-then". The algorithm has a tree structure; it is a clear and readily comprehensible graphical representation. The leaf-nodes represent decisions, conclusions or actions, while the no leaf-nodes represent the conditions that need to be assessed.

Almost all the control architectures mentioned above don't show the properties needed to develop a nanogrid with the desired features. For example, let's consider the finite state machine (FSM), the most used control architecture, that models a robot control, whose task is to look for, grasp a ball and move it inside a bin. (Fig. 2-4). Even for this easy task the FSM became extremely complex and with a lot of transitions [58]. Though the FSM has an intuitive structure, a complex and interesting project could fail if a FSM is used. In each state, the controller has to check for all possible events, reading all the n-inputs to the system. This is a trivial task for a system with a few inputs, but it becomes very hard to code in a more complex system given all the conditions that need to be checked in each state. The designer probably forgets to check some transitions, the programmer gets lost in the if-then-else conditions and the code becomes early unmanageable. Even if the programmer writes the best code ever, when the control logic is tested and in one state of the system appears a not considered event, the update does not include only the indicted state, but also all the states that have a connection (transition) with it.

Table 2-2 Most used control architectures

Control Architecture	***Advantages***	***Disadvantages***
Finite State Machine	- Intuitive structure - Easy to implement	- Maintainability - Scalability - Reusability
Hierarchical Finite State Machine	- Modularity - Behavioural inheritance from the super-state	- Maintainability - Non-intuitive hierarchical
Subsumption architecture	- Easy to develop - Modularity - Hierarchical	- Scalability - Maintainability
Sequential Behavior Composition	- Executive analysis - Modularity - Robustness	- Maintainability - Reusable - Humans readable
Teleo-Reactive programs	- Reactive execution - Intuitive structure	- Maintainability - Trouble management
Decision Tree	- Modularity - Hierarchical Structure - Intuitive structure	- Repetitive structure - Maintainability

The FSM is used in a great variety of applications, as well as in the videogame industry. This drawback has led the videogame developers, who are currently huge users of FSMs, started to look for new CAs. In videogames autonomous entities (Non-Playable-Characters) coexist with characters governed by players; these entities react to a lot of exogenous events. Thus, these NPCs show a kind of autonomous deliberation that are desirable in complex control systems. One of the developed architecture inside this industry is the behaviour tree (BT) [59].

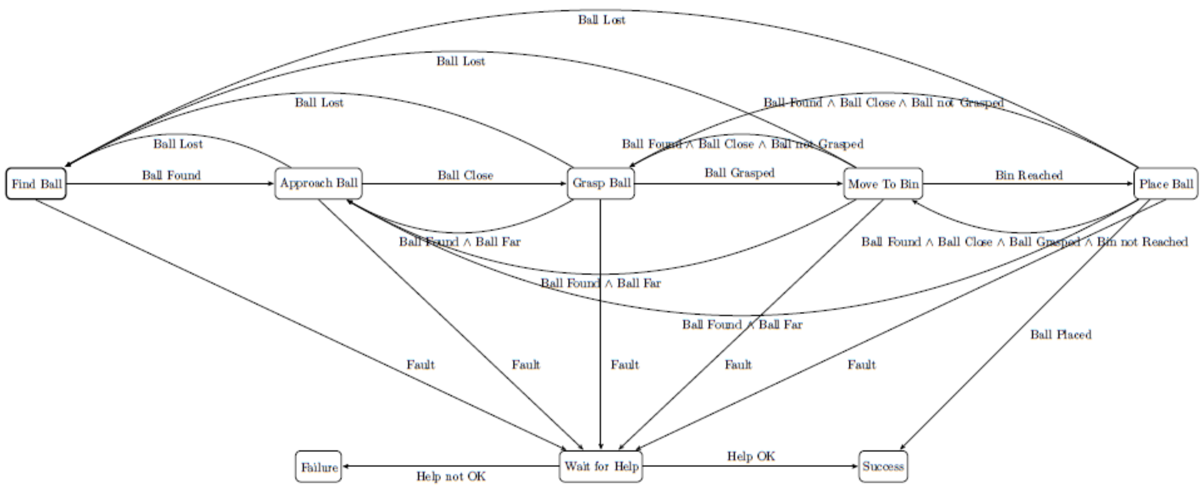


Fig. 2-4 An example of FSM (Source: [58]*)*

The nature of BTs is to code modular and reactive behaviours. Modularity is the capacity to divide and combine single behaviours. Reactivity is the ability to react to exogenous events. With respect to the last example, by using the BT in place of the FSM, the desired behaviour is described in modules in Fig. 2-5. Each module includes a sub-behaviour. The whole behaviour is a sequence of sub-behaviours; the design of a sub-behaviour is independent from the overall behaviour, i.e. from the other modules.

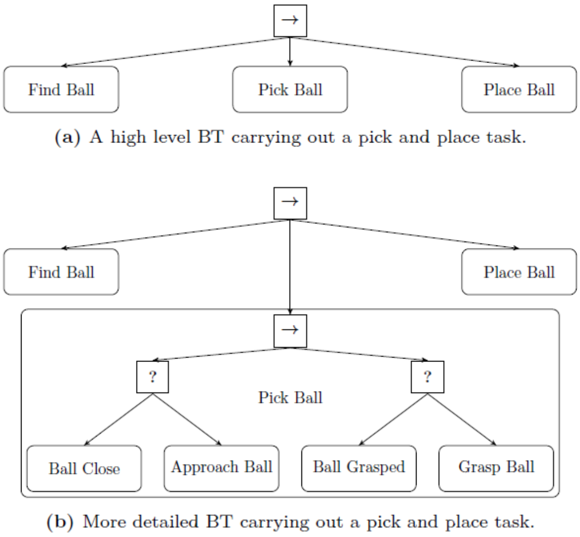


Fig. 2-5 Behaviour tree for the same task of the FSM in Fig. 2-4 (Source: [58]*)*

This area of research on BT was focused for a long time by AI and videogames programmers; the videogames are virtual and for this reason their nature doesn't include problems of robustness connected to the interaction with the external world.

It could be useful to deepen the videogames industry to see if and how this knowledge could be employed in applications where systems interact with unstructured and unpredictable environments, i.e. if the BT can be applied inside a physical real-world. In this work, the BT is investigated as a useful tool to implement the discrete controller of a NG; the BT becomes the making-decision core of the NG controller.

2.3.1 Behaviour Tree

The control of converters inside a NG is an area of active research. To date, academic and industrial efforts focused on how the converter has to perform an activity in the right way, that is the design of feedback controllers, instead of caring about the execution of the right task, that is the design of how to select the right controller, i.e. the control architecture. However, when more features are requested to the nanogrid, the problem of how and when move from a task to another and between the corresponding controller is becoming increasingly difficult. All the actions should have a logic coordination, in order to avoid random, unnecessary and dangerous actions.

The nanogrid exhibits different operating modes. Each operating mode has a dynamic behaviour, described by a system of differential equations. The dynamic, and thus the associated system of equations, changes when the operating mode changes. While the dynamic has a continuous domain, the jump from state to state is described in a discrete state space. These two mixed domains are represented by a Hybrid Dynamical System (HDS).

Hybrid systems arise in embedded control when digital controllers, computers, and subsystems modelled as finite-state machines are coupled with controllers and plants modelled by partial or ordinary differential equations or difference equations. Thus, such systems arise whenever one mixes logical decision making with the generation of continuous-valued control laws [60].

A hybrid dynamical system is an indexed collection of ordinary differential equations (ODEs) along with a map for "jumping" among them, switching from a dynamical system to another. This jump occurs whenever the state variables satisfy certain conditions, also said "guard conditions". Formally, a hybrid dynamical system (HDS) is a system $H = (Q, \Sigma, A, G)$, where [60]:

- $\boldsymbol{Q}$ is the countable set of index or discrete states.
- $\boldsymbol{\Sigma} = \{\Sigma_q\}_{q \epsilon Q}$ is the collection of systems of ODEs. Each system Σ_q has vector fields $f_q : X_q \to R^{d_q}$, representing the continuous dynamics, evolving on the continuous state spaces $X_q \subset R^{d_q}$, $d_q \in \mathcal{Z}^+$.
- $\boldsymbol{A} = \{A_q\}_{q \epsilon Q}$, $A_q \subset X_q$ for each $q \in Q$, is the collection of autonomous jump sets.
- $\boldsymbol{G} = \{G_q\}_{q \in Q}$, where $G_q : A_q \to S$ are the autonomous jump transition maps and represent the discrete dynamics.

Thus, $S = \bigcup_{q \epsilon Q} X_q \times \{q\}$ is the hybrid state space of H. $A = \bigcup_{q \epsilon Q} A_q \times \{q\}$ is the autonomous jump set. $G : A \to S$ is the autonomous jump transition map.

While the discrete state of a hybrid systems is q, the state x evolves according to time-driven dynamics which are usually described by some differential equation such as $\dot{x} = f(x)$ with a given initial condition x_0. Discrete state transitions can occur when a guard condition is satisfied.

The system for |Q|=1 and $A = \emptyset$ is described by all differential equations. With |Q|=N and each $f_q = 0$ the system is represented by the automata. The state space may change. This is useful in modelling changes in dynamical description based on autonomous (or controlled) events which change it.

The case of HDS with |Q| finite is a coupling of finite automata and differential equations. The most common modelling framework for hybrid systems is provided by hybrid automata. There, each node is a constituent ODE. Each edge represents a possible transition between constituent systems, labelled by the condition for which the transition is "enabled", and the update of the continuous state.

The behaviour of a HDS can be modified if the ability to make decisions, that is control an HDS, is provided. The control chooses among sets of possible actions at various times. Specifically:

(i) Possibility of continuous controls for each ODE;
(ii) Ability to make decisions at the autonomous jump points;
(iii) Additional controlled jumps, where one may decide to jump or not and have a choice of destination when the state satisfies certain constraints.

Formally, a controlled hybrid dynamical system (CHDS) is a system, namely $H_c = (Q, \Sigma, A, G, C, F)$, where the arguments are the same of HDS, except as follows:

- $\boldsymbol{\Sigma} = \{\Sigma_q\}_{q \epsilon Q}$ is the collection of controlled ODEs. Each system Σ_q has vector fields $f_q : X_q \times U_q \to R^{d_q}$, representing the controlled continuous dynamics on the continuous control spaces $U_q \subset R^{m_q}$, $m_q \in \mathcal{Z}^+$.
- $\boldsymbol{G} = \{G_q\}_{q \in Q}$, where $G_q : A_q \times V_q \to S$ are the autonomous jump transition maps, modulated by the discrete decision sets V_q.
- $\boldsymbol{C} = \{C_q\}_{q \in Q}$, $C_q \subset X_q$ is the collection of controlled jump sets.
- $\boldsymbol{F} = \{F_q\}_{q \in Q}$, where $F_q : C_q \to 2^S$ is the collection of set-valued controlled jump destination maps.

Mode transitions in hybrid automata may occur due to endogenous or exogenous events. An exogenous event, as the term suggests, is an event originating from the outside world to force a discrete state. An endogenous event occurs when a time-driven state variable enters a particular set. If the endogenous event is associated with the violation of an invariant condition, then it becomes the event responsible for forcing a discrete state transition. If it simply enables a guard condition, then an exogenous event is subsequently required to cause a transition related to this guard condition [61].

In videogames autonomous entities move in something like a continuous space, while react to player actions choosing inside a set of predefined actions. The combined system could be modelled as a HDSs. The control architecture of automated competitors is often designed utilizing FSM; this happens for almost all the embedded systems. However, with the growing complexity of the behaviours, the number of discrete states of the HDS grows, and becomes more and more difficult maintain and improve the implementation. Thus, even if the HDS are modular when describing a continuous dynamic, the discrete dynamic is explicitly coded in the transition functions, which make the HDS less modular than one could envision at first glance. For example, if a discrete state is removed from the HDS, all the transitions from and to the removed state have to be redesigned. Thus, the absence of modularity strongly reduces the scalability and reusability.

Indeed, the programmers look for new AI algorithms to make the game more and more interactive, that is give more behaviour to characters, i.e. high number of states and possible transitions. In 2005 a new control architecture, the BTs, appeared in the videogame industry to model the behaviour of NPCs characters [59], [62].

The behaviour trees (BTs) are a recent technique to create behavioural algorithms; they are graphical representation tools used initially in non-critical applications, where a rigorous mathematical formulation can be avoided. However, when BTs are involved in more complex applications, as real robot, there is the need of mathematical foundation and rigour analytical. This problem has been faced in some papers, as for example in [63], where the formulation is given in terms of accuracy and compactness. Accuracy means to be inversely proportional to the degree of misinterpretation that a certain statement can be subjected to; compactness means to be inversely proportional to the amount of definitions required to fully specify an idea.

BTs are a particular type of hybrid system (HDS) [60], where the transitions of the HDS are implicitly codified inside a tree structure, instead of being explicitly indicated on the transition map. In [64] the relation between HDSs and BTs is analysed; the authors have proved that whatever HDS can be written, or more accurately, translated in BT.

An interesting evolution of BTs is their potentiality of representing Controlled Hybrid Dynamical Systems (CHDSs). In [63] advantages and disadvantages of the techniques used to model HDSs are analysed: the authors concluded that with BTs the ability to be in a certain state is lost, but is achieved modularity. A BT is modular if its control-flow nodes and the subsets are laid out in such a way that the represented goal succeeds or fails in finite time.

BT is a way to code a HDS which replaces the transition functions that are explicitly listed in the HDS with transitions implicitly given by the tree structure. A given BT could be inserted as a substructure everywhere inside any other BT and it is formally valid, moreover the transitions between new and old parts of the tree will be, by implication, the result of the combined structure of the tree. In the same way every part of a BT could be removed without this leading to invalid transitions between states. Anyway, as Petri nets give an alternative to the FSMs that emphasize the concurrency, the BTs give an alternative that emphasize the modularity.

The behaviour tree has found out with some difficulties an unequivocal formalism. The lack of documentations, formalism and a rapid adoption of BT produced an inconsistency of names, structures and definitions in papers and tutorials [65]–[68]. Improvement that BT provides compared to the FSMs are:

- Maintainability: transitions are defined by the structure and not by the conditions inside the states. In this way nodes of the tree are independently designed, then, adding or removing new nodes (or subtrees) in a BT doesn't require further changes;
- Scalability: when a BT has a lot of nodes, they can be compressed in smaller subtrees retaining its readability of the graphic model;
- Reusability: because of the independency between the nodes, the subtrees are independent too. This allows the reusability of the nodes or the subtrees in other different projects;
- Goal-Oriented: even if nodes are independent they are related thank to the tree structure. That allows to the designer to build specific subtrees for a given objective, without lost model flexibility;
- Parallelization: inside a BT could be specified nodes that make the child nodes parallelized, without lost the control of the model that is running.

Finally, the tree structure improves the performance of the algorithms when the automatic synthesis in terms of learning or planning techniques are adopted. The BTs generalize many of the illustrated CAs, are good for analysis (system properties are analysed and designed in a modular way), are good for planning and learning [69]. Obviously, BTs present some disadvantages, are less mature than other CAs; they require a new approach when designing a task. For structured environments, the classical choice of FSMs work just as well [70]–[72].

2.3.1.1 Classic formulation

Formally a BT is an acyclic graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with $|\mathcal{V}|$ vertices and $|\mathcal{E}|$ edges. Each vertex has a parent and may have one or more children. The vertices that doesn't have children are called "leaves", while the only one node without parents is known as "root".

Each node, except the root, belongs to one of the following categories:

- Flow-control nodes:
  - Selector nodes;
  - Sequence nodes;
  - Parallel nodes;
  - Decorator nodes.
- Leaf nodes (also known as execution nodes):
  - Action nodes;
  - Condition nodes.

Fig. 2-6 shows a minimalist example of a BT.

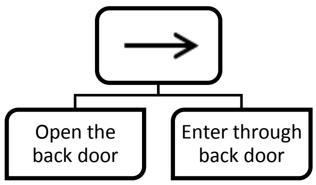


Fig. 2-6 Minimal example of BT with the goal to enter in a room

The root periodically, with a frequency *ftick*, emits an enabling signal called *tick*, that spreads between the branches on the basis of the tree structure. The *tick* stops to spread when it reaches a "leaf": at this point, the "leaf" may implement an action or take a decision; inside a "leaf" are performed calculations, often taking into account internal states and data acquired by sensors. The type of data returned from a leaf is one of the values in Table 2-3. The execution of the BT is equivalent to the execution of the function:

$$tick(child(i))$$

The actions alter the system configuration and result in one of the following possible values: *Succeed*, *Failed*, or *Running*.

Conditions don't alter the system configuration, resulting in one of the following possible values: *Succeed* or *Failed*.

Table 2-3 Possible returned states from the nodes of the BT

Name	***Value***	***Description***
Start	*0*	*Node not yet reached by the tick*
Succeed	*1*	*Action-node: executed correctly;* *Condition node: positive outcome*
Running	*2*	*Action-node: performing an action that takes more time than the tick period*
Failed	*3*	*Action node: not executed correctly;* *Condition node: positive outcome*

The meaning of each possible state returned by a node is explained below:

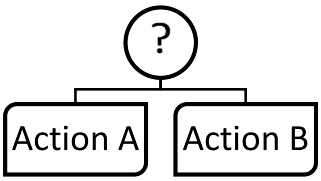
- Start: node not yet reached by the tick;
- Succeed: condition-node that has verified a condition with positive outcome or action-node that has concluded an action with positive outcome;
- Running: action-node that is performing an action that takes more time than the tick period;
- Failed: condition-node that verifies a condition with negative outcome or action-node that concludes an action with a negative outcome;

The value resulting from the "leaf" is propagated back to the branch to which it belongs; on the basis of the tree structure the algorithm proceeds in its execution, eventually triggering other "leaves". Each ticked leaf node returns back its status, until one of these states reaches the "root". Nodes that aren't reached by the tick are reported as "not-ticked". Then the BT waits before generating a new tick, in order to maintain a constant *ftick*.

It is important to recall that the *ftick* is not related to the low-level controller frequency: the two layers work asynchronously.

The BT is concluded when the "root" returns a state. The value returned by the "root" is the outcome of the BT execution. The control-flow nodes are now described.

- **Selector Node**: The selector nodes are used when more actions represent alternative ways for reaching a similar goal. The selector node will try to invoke each of its children, from left to right, and will return Success when one of its children nodes gives Success. It will return Running when the first of its child nodes returns Running and it will return Failure only when all the child nodes have failed.



The selector node allows to increase the robustness, i.e. to increase the size of the attraction region, and to affect the efficiency of a BT. The selector node selects an action from a set of admissible actions. Each action is different from the others in the sense that each action follows a different path, but all of them allow to reach the given goal, i.e. the successful region (Fig. 2-7). The action is enclosed in the leaf node and the priority of each action is assigned positioning the leaf nodes from left to right.

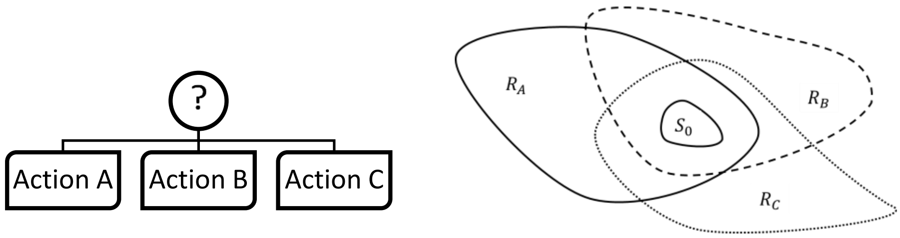


Fig. 2-7 The selector node select one of the action to reach a goal: example of a selector node (left), correspondent state space (right)

Another possible use of the selector node is to direct the system state towards the successful region S_A. If *ActionA* fails, *ActionB* drives the system in the convergence region R_A, as shown in Fig. 2-8. Indeed, *ActionB* has a success region S_B that intersects the convergence region of the first action R_A. BT's success occurs when the system reaches the S_A region. If the BT has to handle different situations, it is not need to implement a most complex version of the task, but rather it is possible to combine another BT that can handle the situation and move the system to a part of the state space that the original BT can handle. The selector node, used in such a way, is called implicit sequence.

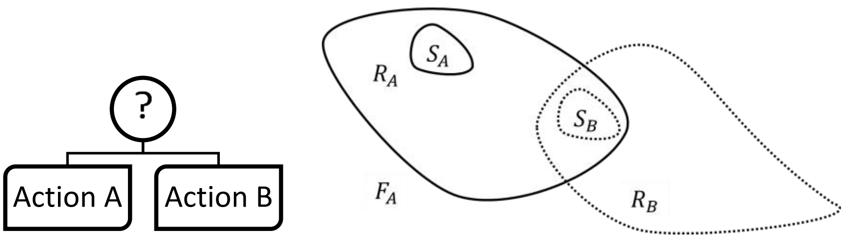
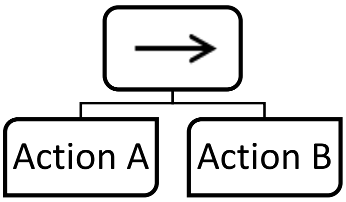


Fig. 2-8 The selector node can be drive the operating point in the state space in the successful region SA (Action A) using the Action B

- **Sequence Node**: sequence nodes are used when some actions need to be performed in sequence and when the positive outcome is essential for the execution of following actions. A sequence returns back immediately with a failure or running status if one of its children returns back failure or running. Children nodes are ticked from left to right.



The sequence node is useful to drive the system in a safety region, i.e. to avoid that the system never enters in a particular state space region, also called obstacle region. The obstacle region in Fig. 2-9 is F_A. *ActionA* must ensure that the system never enters in the Failure region F_A. The R_A region moves away from a system failure in F_A. The width of the S_A region allows to the *ActionB* to operate safely in any state sufficiently distant from F_A, without worrying about the possible failure of the system.

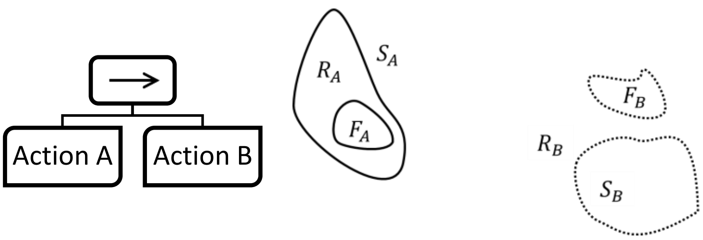
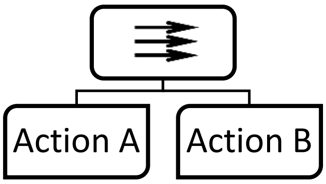
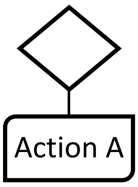


Fig. 2-9 The safe operation of the system is ensured by ActionA

- **Parallel node**: parallel node ticks (invokes) all its children simultaneously. If M of the N children return back Success, then the parallel node will do the same. If more than N-M nodes return Failure, then the parallel node will return failure. If no condition is verified the node returns Running.



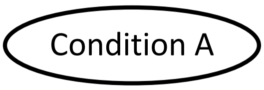
- **Decorator node:** decorator nodes are used to give special behaviour to a sub-tree. For example, it can be used to reverse the outcome of a child, to mask a negative outcome or to execute its sub-BT only one time. The decorator node is a control flow node with a single child that manipulates the return status of its child according to a user-defined rule and also selectively ticks the child according to some predefined rules.



The leaf nodes are now described.

- **Action node**: an action node performs an action and gives back Success if the action is completed, while it gives back Failure if it cannot be completed or Running if the completion requires time;

- **Condition node**: a condition node determines if a given condition has been verified, then, Success/Failure are often interpreted as true/false. These nodes are a subset of the action node, but are classified differently and they have a different graphic symbol in order to improve the BT readability, to remember that they never return Running and don't change any variable/state inside the system.



Some BT implementations don't include the Running State. Instead they wait for the Failure or the Success of the action. These BTs are indicated as non-reactive, given that they don't allow actions except for that one is ongoing, then they can't be responsive to the changes. This is a significant limitation of the non-reactive BTs.

2.3.1.2 BTs functional formulation

In [73], the authors give the following functional formulation. This formulation allows to formalize the BT; a clear formal representation allows an analytical approach in analysing, designing, testing and debugging the BT. Besides, this formalism allows to analyse how the properties are preserved on the modular composition of the BTs.

In this formulation, the tick is replaced by a call to a recursive function that include the returned state, the system dynamic and the system state.

A BT is a 3-tuples described by:

$$T_i = \{f_i, r_i, \Delta t\}$$

where $i \in \mathbb{N}$ is the index tree, $fi : \mathbb{R}n \rightarrow \mathbb{R}n$ is the right side of an ordinary differential equation (ODE), Δt is the time-step, $ri : \mathbb{R}n \rightarrow \{R, S, F\}$ is the returned state that could be: Running (R), Success (S) o Failure (F).

The state space, inside which the BT is defined, can be divided in a Running/Activation (Ri), Success(Si) and Failure (Fi) region:

$$Ri = \{x: ri(x) = R\}$$
$$Si = \{x: ri(x) = S\}$$
$$Fi = \{x: ri(x) = F\}$$

The execution of a BT T_i is an ordinary differential equation:

$$x_{k+t}(t_{k+1}) = f_i(x_k(t_k))$$

$$t_{k+1} = t_k + \Delta t$$

Several BTs that control different subsystems evolve in different state spaces. They can be combined in one single BT. In this case all the system will evolve in a bigger state space, that is the cartesian product of the smaller state spaces.

The composition of many BTs in one single structure can be achieved via operators.

Two BTs are joined in a more complex BT, using a sequence operator, such that the new BT T_0 is:

$$T_0 = Sequence(T_1, T_2)$$

Then r_0, f_0 are defined as:

$$If\ x_k \in S_1$$

$$r_0(x_k) = r_2(x_k)$$

$$f_0(x_k) = f_2(x_k)$$

$$else$$

$$r_0(x_k) = r_1(x_k)$$

$$f_0(x_k) = f_1(x_k)$$

T_1 e T_2 are called children of T_0. It should be noted that when the new BT is in execution, T_0 continues to perform T_1 until T_1 returns Running or Failure. The second child will be performed when T_1 has returned Success. T_0 returns Success if all the children have returned Success.

Two or more BTs can be composed in one more complex BT by using a selector operator, for which:

$$T_0 = Selector(T_1, T_2)$$

Then r_0, f_0 are defined as

$$
\begin{aligned}
&If\ x_k \in F_1 \\
&\quad r_0(x_k) = r_2(x_k) \\
&\quad f_0(x_k) = f_2(x_k) \\
&else \\
&\quad r_0(x_k) = r_1(x_k) \\
&\quad f_0(x_k) = f_1(x_k)
\end{aligned}
$$

It should be noted that when the new BT T_0 is executed, the first child T_1 continues to be invoked as long as it continues to return Running or Success. The second child is called only when the first one returns Failure. T_0 returns Failure only when all the child nodes have tried, but have failed.

Lastly, the parallel operator allows to create a new BT on the basis of two different trees. When more trees are composed in parallel, the division of the state (x_1, x_2)so that $f_1(x) = (f_{11}(x), f_{12}(x))$ implies that $f_{12}(x) = 0$ and $f_2(x) = (f_{21}(x), f_{22}(x))$ implies that $f_{21}(x) = 0$, that is the two BTs control different parts of the state space. The parallel operator returns a new BT T_0, that is:

$$T_0 = Parallel(T_1, T_2)$$

Let's suppose that $x = (x_1, x_2)$ is the division of the space state, then $f_0(x) = (f_{11}(x), f_{22}(x))$ and r_0 is defined as:

$$
\begin{aligned}
&If\ M = 1 \\
&\quad r_0(x) = S\ If\ r_1(x) = S\ \vee\ r_2(x) = S \\
&\quad r_0(x) = F\ If\ r_1(x) = F\ \wedge\ r_2(x) = F \\
&\quad r_0(x) = R \\
&else\ if\ M = 2 \\
&\quad r_0(x) = S\ If\ r_1(x) = S\ \wedge\ r_2(x) = S \\
&\quad r_0(x) = F\ If\ r_1(x) = F\ \vee\ r_2(x) = F \\
&\quad r_0(x) = R\ else
\end{aligned}
$$

In [73], [74] the authors define safety, time efficiency and robustness for the BT. The robustness is indicative of regions of attraction depth. It has nothing to do with disturbance rejection, or other forms of robustness.

A more formal definition of robustness is now derived. First of all, the given FTS definition is:

A BT is finite time successful (FTS) with region of attraction R', if for all starting points $x(0) \in R' \subset R$, there is a time τ, and a time $\tau'(x(0))$ such thta $\tau'(x) \leq \tau$ for all starting points, and $x(t) \in R'$ for all $t \in [0, \tau')$ and $x(t) \in S$ for all $t \geq \tau'$.

Time efficiency corresponds to reaching another part of the space in time, i.e.:

A BT is time efficient if satisfies the FTS definition with a small τ. Besides, the BT is robust if satisfies the FTS definition with a large region of attraction R'.

Safety corresponds to avoiding parts of the space, i.e. region where the system could become dangerous:

A BT is safe if with respect to the obstacle region $O \subset \mathbb{R}^n$, and the initialization region $I \subset R$, if for all starting points $x(0) \in I,\ x(t) \notin O$, for all $t \geq 0$.

Robustness corresponds to achieving both of the above from a large set of initial positions

2.3.1.3 BT and FSM

BTs modularize the FSMs in the HDS. Modularity is important during design and testing, but it allows the reutilisation of complex activities with switching structures.

A FSM has a corresponding BT, in fact a FSM may have three different transitions that correspond to the three possible results of a BT – Success, Running, Failure. The FSM, executed with a period equals to the tick period, has a structure similar to that of the BT (Fig. 2-10).

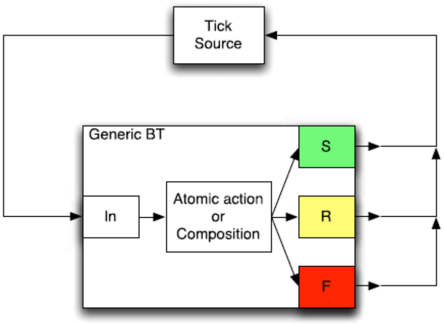


Fig. 2-10 Equivalent FSM of the BT (Source: [73]*)*

Some observation will follow. From BTs is possible design FSMs (and then HDSs). The corresponding FSM of a BT has a more complex graphical representation, and this complexity will grow exponentially as the BT complexity grows. A further consideration is about the modularity. The automata of the example could be complex, but this complexity is enclosed inside the outer frame, with one simple input transition and three outputs (the BT states), exactly what the corresponding BT represents. If the designer wants to add a behaviour, he has to go into the substance of the automata implementation and has to know what each transition inside the FSM represents. This doesn't happen in BT.

Executing a sub-BT corresponds to a call to a function within a code. After the function is executed, the program flow is left to the code line after the called function. How to behave after the call is not decided by the same function, but the code line following the call to the function will decide the behaviour. This is very different from the FSMs standard, where the decision of what to do is taken within the same state, something similar to what was happening in the legacy GOTO statement. With this frame of reference, it can be deduced that the function implementation does not depend on how the result will be used and the user does not have to know how the function has been implemented; this allows to use the same function in different programs (re-usability), but above all allows a program to be modular.

Lastly, another small point in favour of BT is the graphical representation. The FSM has an arrow for any possible transition, but the current conditions for transfers do not have a graphical representation. While for BT a condition (success or failure) occurs directly from the tree structure.

It should also be noted that there is no BT superiority over FSMs from a purely theoretical point of view. In fact, in [73] the equivalence between the two representations of a HDS has been verified. Using a IT language, BT and FSM are equivalent, as several programming languages are equivalent in the sense of Turing completeness but differ in modularity, readability, and code reusability.

3 Proposed Hybrid Control for Nanogrid

A nanogrid (NG) works in a heterogeneous and unpredictable environment. The control structure of the NG allows to coordinate multiple sources and to optimise power production and consumption; it establishes how the system behaves in response to an event. One or more controllers react to the event, selecting the operating mode of each converter.

The proposed control structure has been developed for a NG with the control boards and all the power electronics enclosed in one device. The only one controller of this device is the NG controller. The proposed control structure implemented in the NG controller allows to an already existing home power system to become a NG, allows to the user to became member of a coalition to increase the NG economic benefits, and allows to a remote area to become electrically independent, assuring high levels of power quality and reliability. Thanks to the NG controller the realized NG is simple to setup, but it is powerful in its features.

This work shows the importance to formally faces a complex control problem. The NG system is modelled considering the discrete and continuous behaviour. In this way, the design focuses on development of a hybrid control, i.e. a discrete and continuous control. This approach allows to give intelligence to the NG controller, because the decision-making is accurately and clearly designed. A significant introduced innovation is the use of a Behaviour Tree (BT) to control the transitions from a state to another in the discrete system.

The BT, in a more formal way, is used to design and implement the discrete control of a NG, modelled like a hybrid dynamical system (HDS). The BT defines the transactions between different kinds of activity, that is the switching between different controllers of the continuous low-level layer. This allows to the intelligent agent to reach a given goal without occurring in random, unnecessary or dangerous actions.

The BT has proved to be a powerful technique. It is efficient, clear and practice. The translation from requirements in the common language to the syntax rules of BT, with a little of experience, is almost immediate. The preparation/planning is an important phase in the BT design; a scrupulous eye can find some correlations between the overall system and the BT structure. This can help during the design stage to develop more robust structure. The behaviour is described in the structure tree, which has a clear graphical representation. In the end, the BT and the formal analysis of the control problem have allowed to develop a system near to the commercial deployment.

In this chapter the nanogrid for Home Application (nGfHA) idea and the hardware structure are briefly presented. Then, the proposed control structure for nGfHA follows; the control structure allows to implement the behaviour of the NG and to manage the power flows inside the NG and from/to external power systems. The control structure allows the interaction with a number of entities to reach a common goal. The analytical models for the NG converters and the continuous control, designed using the developed analytical models, are presented. Some numerical results are provided. In the end of the chapter, two BTs for the NG are presented.

3.1 nanoGrid for Home Applications (nGfHA)

The discussed NG, also known as nano Grid for Home Applications (nGfHA), is the idea of NG presented in [14]. The NG has a dc bus, where different types of sources as renewable energy sources (RESs) and battery energy systems (BESs) are connected. An inverter connected to the dc bus is responsible for the supply of uninterruptible loads. The electrical structure is in Fig. 3-1. The peculiarity of this NG is that it is controlled and managed by only a device, a custom device developed ad hoc, as the one shown in Fig. 3-2.

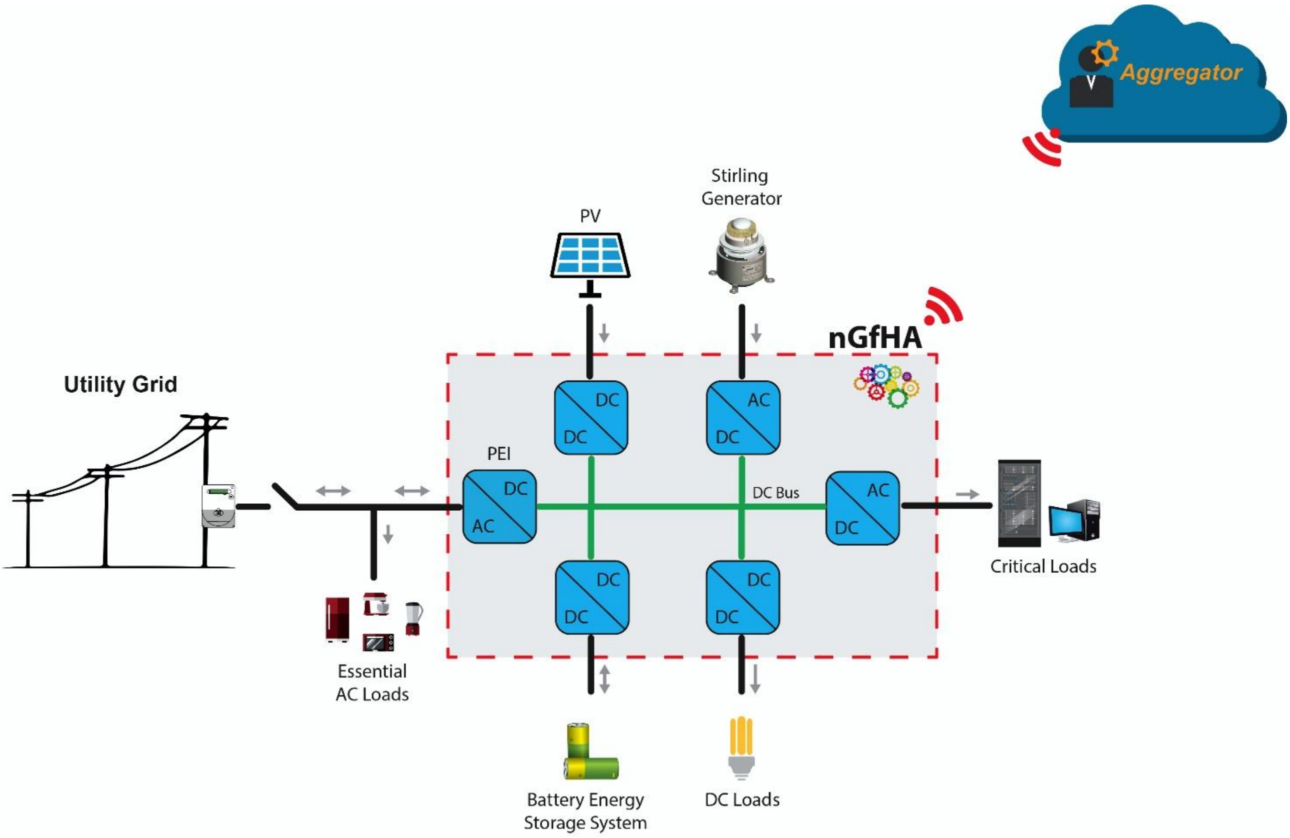


Fig. 3-1 A graphical representation of the nGfHA

A nGfHA is designed to work both in grid-connected and islanded mode. In grid-connected mode, the nGfHA is connected to public distribution in ac, via a gateway power converter, called power electronic interface (PEI), which controls the power exchanged between the grid and the NG in a bidirectional way. When a point of delivery (PoD) is present and the operating conditions of the grid doesn't present disturbances, the nGfHA can exchange power with the ac utility grid; thus, more NGs can interact between them and can exchange energy through the distributor grid. Thanks to this capability, when the generated power in a NG exceeds the local demand, the NG allows to give the surplus to main grid or to give it to other NGs in a coordinated way. The user becomes a "prosumer".

The nGfHA is an important dowel for the smart grid implementation: actually, the nGfHA is one of the actuators of the smart grid. Indeed, if more entities like NGs are aggregated and simultaneously coordinated and if the sources and loads can actuate remote commands, thanks to the NG controller, the burden of the grid coordinator is reduced. The interaction between NGs can allow the implementation of new models for the intelligent management of the energy in aggregated form [14].

The electrical loads in the nGfHA are divided in essential and critical electrical loads. The essential loads can be interrupted only for a short period, but they carry out important tasks, therefore the supply is restored readily. The critical loads can't be interrupted, because the interruption can cause damages and costs. Other kind of loads directly connected to the grid, between the PoD and the main switch of the nanogrid, are not supplied when the nGfHA works in stand-alone.

A nGfHA can works in islanded mode if a fault on the ac grid cause a momentary service disruption. The supply of the essential loads can be restored through the PEI converter, which switches its control to become a grid forming unit. The loads need to tolerate a short supply discontinuity, if a more complicate control technique is not applied to produce a seamless transition.

A nGfHA can be used in islanded mode if a public electrical infrastructure is not available. This mode is particularly useful for houses in rural village or in developing countries, where the electrical distribution is not mature or not even started. In this context, the property of interconnections between NGs is useful to realize a dc MG to increase quality and reliability in supply and economic benefits.

Technically, different kind of dc sources can be connected to the dc bus through third party converters. However, the idea is to develop a power converters structure that can be easily used to build a nanogrid. The device has a number of inputs to connect additional sources, usually enough to satisfy the distributed resources available in a single home. To overcome this limit more nGfHAs can be interconnected through a common dc bus. Thanks to the control structure developed the device recognises the interconnection and automatically works in conjunction with the other nGfHAs to realize a wider grid.

Thus, if the inputs are not enough another nGfHA can be added and connected to the same dc bus. This solution allows to avoid compatibility problems in terms of EMI and of dynamical stability with third party converters. The device allows to manage the NG more efficiently and with more flexibility, mainly when a sudden event leads to a critical operating mode. For example, to increase the continuous operation of the NG and to exploit the pv production as long as possible it is better to modulate the PV production in an islanded NG during a surplus condition, rather than disconnect the PV plant, or worse, shut down the NG.

The device used to setup a nGfHA has the converters, the controllers and all the control signals all enclosed in a single hardware structure. The device has a control board that allows to adapt measurements and signals to different powers. This structure has allowed the implementation of prototypes, for different applications, with rated power between 1 to 7 kW. The power limit is not introduced by the control board, but by the installed power switches, which are independent from the control board [75].

The device can manage every kind of source commonly used in distributed generation. Besides, there is an important feature in the developed system, in both hardware and software structures: the aptitude to easily add new sources and new behaviours, according to the needs.

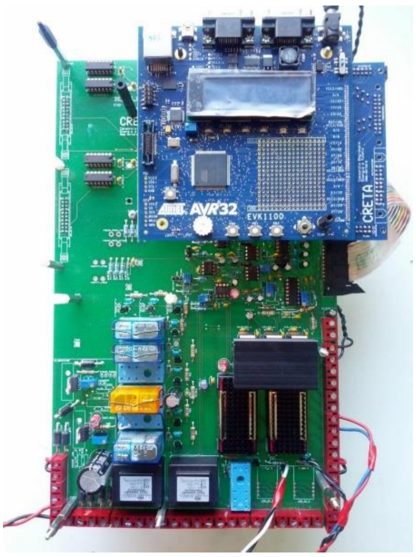


Fig. 3-2 The control board used to build a nGfHA

The goal of this research is the development of a control structure that allows to a compact and modular hardware-device (Fig. 3-2) to control and manage a nGfHA. The requirements for the control structure are: flexibility, modularity and reliability. Thanks to the developed control structure the nGfHA is easily integrated in an already existing home power system. The device can be installed simply as a standard PV inverter; it is simple to setup, but it is powerful in its features. It allows to the user to became member of a coalition to increase the NG economic benefits. It allows to a remote area to become electrically independent, assuring high levels of power quality and reliability.

3.2 Control Structure for nGfHA

An intelligent entity exhibits the behaviour implemented in the control architecture. The control architecture allows to control a discrete system with a finite number of states. However, the real-systems are not simple: usually, the discrete system is coupled with other subsystems physically modelled with partial or ordinary differential equations. Such systems are called hybrid systems and they arise whenever one mixes logical decision making with the generation of continuous-valued control laws.

In almost all the engineering activities some simplifications are accepted in order to model a process. Under some hypotheses the simpler model is still a valid description of the real system, even if some information is disregarded. In this simplification, most of the time, the discrete side of the hybrid controller is disregarded and only the continuous side is considered. The simple model is used to analyse the system and to design a controller, which ensures compliance with the requirements. Thus, the design stage of the discrete control is often disregarded or left to the programmer. Other times, an unsuitable control architecture is chosen. These reasons can significantly impede the system development. The simplification is acceptable only if the system control requirements are simple or if the practical implementation in a real physical system is not faced. When one of this two conditions is not verified, it will be needed a hybrid controller.

The core of the nGfHA control structure is a hybrid control. The adoption of a rigorous approach was born of the difficulties encountered disregarding the overall control problem and focusing only on the continuous system.

3.2.1 Hybrid control for the nGfHA

A schematic of the proposed control structure for the nGfHA is in Fig. 3-3. The grey corner encloses all the components of the previous Fig. 3-2. The first requirement for the NG control is to ensure a robust dynamic response to sudden load or production changes, through a real-time power sharing strategy. This task is entrusted to the hybrid control of Fig. 3-3. With an analogy to computer science, the hybrid controller has two layers: the continuous layer interacts directly with the converters of the nGfHA, while the upper layer executes the discrete controller. Both first and second layer control make the nanogrid system a Controlled Hybrid Dynamical System (CHDS).

The NG sources and loads are interfaced with the dc bus through power converters. The power converters are the hardware actuator that adapt the currents and voltages to feed or to absorb power from the dc bus, according to the specific source or load.

Each power converter has a control-loop, in order to accomplish the given task and to follow the given reference value. For each possible NG state, each power converter has a defined control-loop with a given goal, such as controlling the current or the voltage. The set of all the control loops for one state is the continuous control for the NG. The continuous control is strictly related to the hardware and is responsible for NG dynamic and stability. An important step in the development of this control structure is the feedback controllers design; the controllers have to guarantee dynamic stability and the desired dynamic performance inside the NG. This is not an easy requirement because several converters are involved in the control of the system; the risk for interactions between controllers is real. A design requirement is the dynamic decoupling among the same kind of converters, designing the possible interacting controllers with different time constants.

The continuous control adopts a power sharing strategy, in order to ensure the power balancing in the system. The power sharing strategy is decided by the discrete control, allocating the kind of feedback controllers for the involved converters. The continuous control is executed in the first control layer.

The discrete control chooses among sets of possible actions at various times. Specifically, the control selects the continuous control for each state, represented by its own ODE, decides what to do at the autonomous jump points and decides autonomously to jump or not to another state when the state satisfies certain constraints, selecting the destination state. The discrete control updates the reference values and the goal for the continuous control. The discrete layer establishes, for each converter, the kind of low-level control and the reference value: it decides the operating mode of each converter, i.e. the operating mode of the NG. This layer is responsible for the energy power flows in the NG. The discrete control, acting like a real-time EMS, is executed in the second layer.

If the desired features for the nanogrid controller require a rich behaviour, the development of this layer could become really complex and hard to manage. Moreover, the algorithm that implements the behaviour has to be light because it has to be executed with a small period, under a hard real-time requirement. This requirement is needed because the discrete controller must reconfigure all the converters in case of a sudden event, as a disconnection from the utility grid. A fast reconfiguration is needed to ensure increased power quality and reliability. While in an extended grid, with a high number of dispatchable sources and high inertia, the disconnection of one source is not a big issue, in the NG is a difficult event to manage. The algorithm has to be robust, efficient and light. The second layer is more a real-time energy management strategy than a long-term one, because it takes decisions only with the current data available locally.

It’s important to emphasize that the upper layer, i.e. the discrete layer, has no idea of which kind of hardware is governing; this is a powerful feature in the presented structure. The abstraction realized between the hardware and the second layer allows to change hardware, to try different kinds of low-level control without updates for the second layer. The only thing the discrete control cares about are the source types, because it has to prioritize them to reach a given goal and to optimize the power flows inside the NG.

In the implemented solution the first and second layers are executed in the same controller, i.e. the nanogrid controller. This is possible thanks to the developed hardware structure, which allows to positioning all the converter close to each other. The central controller executes real-time software. Blocking activities are not allowed and the safe operating of the NG is determined by the only one processor. This is an interesting solution because a central controller allows to implement more dynamic behaviours. The vulnerability is the only drawback, because if the central control fails, then the NG shuts down. However, this issue could be partially solved if the software is, as much as possible, bug-free. Again, the second layer has to be designed with a particular robust structure.

Moreover, the limited processing power and the limited number of inputs and outputs restricts the number of controllable converters. This is not a big drawback considering that a NG is a small grid at building level, where a small number of sources is usually available.

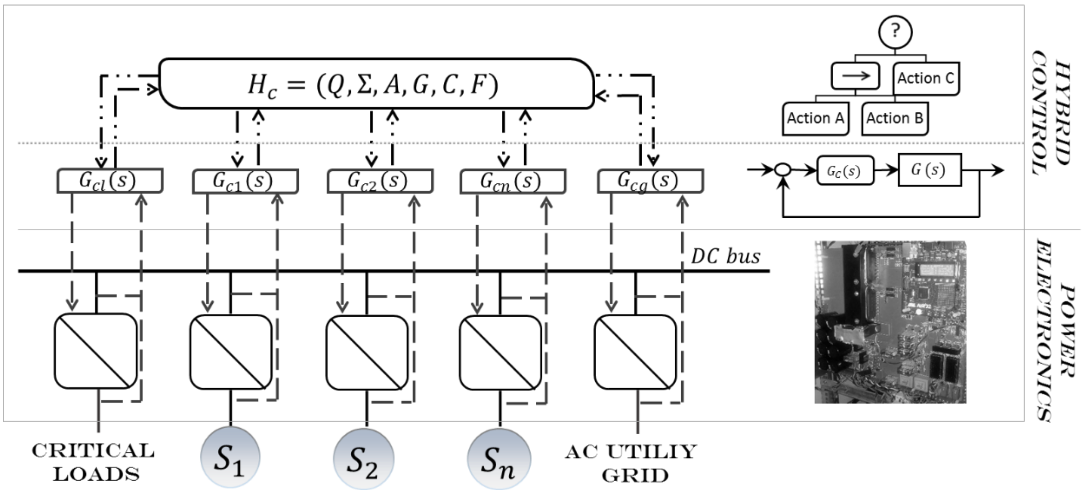


Fig. 3-3 Proposed control structure for the nGfHA control system

The structure in Fig. 3-3 guarantees the NG basic operations; it may work grid-connected or stand-alone and manage efficiently all the available resources. However, the interaction with external environment and with multiple entities is limited, because the real-time task of the discrete control is not compatible with the elaboration of a great amount of data and with the unavoidable communication delays. This is a huge limitation if a more advanced strategy is desired. To overcome this limitation an additional processor is used to gather all the data from the involved entities and to elaborate advanced energy management strategies. The request to apply these strategies is sent to the NG controller, which uses the additional information in the decision-making process. The NG controller actualise the request if it is physically admissible in the particular operating condition.

The additional layer in the control structure not only strictly manage the communication between the NG controller and the external world, but it is able to manage other peripherals. Peripherals or entities important in the energy management of the building, even if don't belong directly to the NG, such as for example the home automation system. This solution allows to integrate heterogeneous equipment without interact with the real-time actions of the NG controller. This additional processor is the nanogrid supervisor, as shown in Fig. 3-4.

The additional controller can be the intermediary between the aggregator and the i-th NG, when the NG is inside an integrated community energy system (ICES). The aggregator in order to maximize the benefits for the coalition assigns to each nanogrid an exchange profile. The NG becomes an actuator, an entity with distributed intelligence inside the smart grid, which gives the opportunity to enable advanced energy management in the system and advanced business models inside the community.

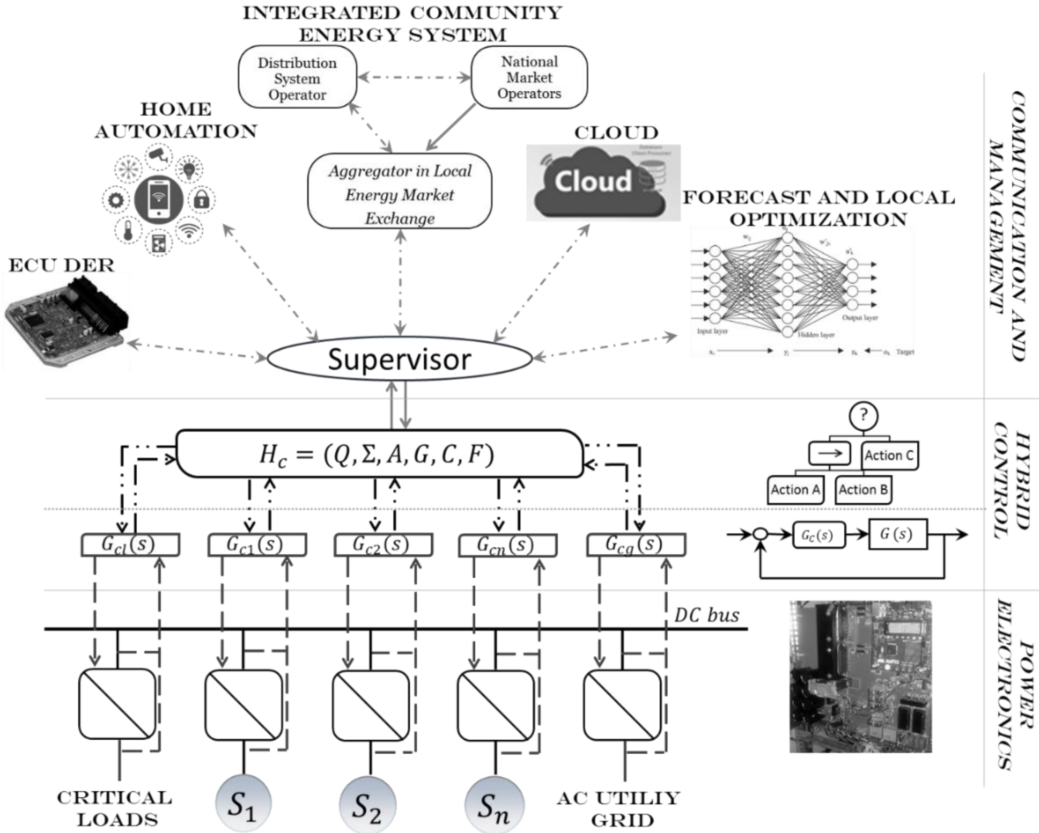


Fig. 3-4 Nanogrid supervisor role inside the proposed control structure

In the energy management other hardware and software modules can be involved. The supervisor can interact with the control logic of the available DERs to observe the status, to monitor the current production and eventually, if admissible, to regulate the power production according to the actual specific needs.

The home automation system can be used to save energy and money, but mainly it helps the NG controller to deal with critical situations, when for example the local low production is not enough to supply all the loads. If the less critical loads are disconnected the temporary deficit can be efficiently managed without the need to shut down all the power system.

The NG has now access to remote services, such as weather, supply and demand forecasting and energy price. The forecast can be obtained locally if the algorithms run in the local processor. However, historical data and training of the algorithm made this solution harder to implement. Regardless of how the forecast is obtained, an offline optimization algorithm uses all this information to enable long-period strategies. The result of the optimization is actuated, if the solution is admissible, by the nanogrid controller.

The software inside the supervisor manages and coordinates the entities inside a heterogeneous environment. A specific implementation of this layer in a real project will be presented in the last chapter. Now, starting from the communication capability of this processor and from the presented control structure, the NG becomes a modular building block, that can be replicated to enhance a high degree of cooperation among more NGs connected together.

3.2.2 Interconnection of nGfHAs with hybrid control

The control structure of each NG allows to a number of NGs to work together in a coordinated way. The presented structure is easily scalable to allow energy exchanges among more NGs. Some possible configurations are now described.

The first basic kind of interactions that could happen is through the ac utility grid (Fig. 3-5). The grid inverter of each NG controls the power exchanged with the grid. The value of the power to exchange is decided by a common supervisor, i.e. the aggregator, which tries to optimize the global management of all the NGs in the community. The nanogrid supervisor is the interface between the NG and the aggregator. The supervisor takes in charge the aggregator request: the request, by means of the supervisor, becomes a new input for the nanogrid controller, which will elaborate it and, if physically admissible, will apply through a new command for the PEI connected to the ac utility grid. Thus, the power between NGs can be effectively exchanged through the already existing ac utility grid infrastructure, even if they are far from each other.

The basic example, the simplest one, of this kind of interaction is the scenario in which one NG is a NG with only a PV plant and the grid connected inverter, without local loads. The second NG is a NG with only a storage and a grid connected inverter. The aggregator could command the second NG to absorb energy from the grid when the first NG feed-in energy into the utility grid: the energy absorbed by the second NG is the same energy supplied from the first NG. If the power of the first NG is higher than the second NG capability, more NGs with storage can be used.

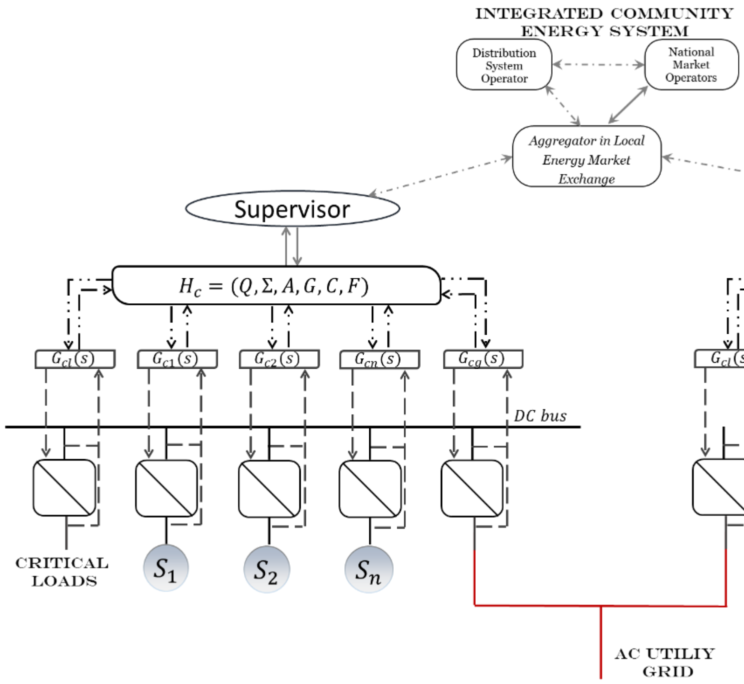


Fig. 3-5 Two nGfHAs cooperating through the ac utility grid operating inside a ICES

If the two NGs are not too far, the interaction can take place also through a direct connection between the two dc buses (Fig. 3-6). The power flow between the NGs is indirectly controlled by the action of the hybrid controller. The power flows through the dc connection, instead of through the ac utility grid, reducing the number of conversion stages, thus increasing the efficiency. The NG controllers have to be coordinated to put in place an effective power sharing strategy. The coordination is possible only if NGs exchanged information. The costs and complexity associate to a communication infrastructure are high, mainly due to the fast dynamic of the converters. However, the NGs can share information through the dc bus. If the dc bus voltage is the used information, the realized control is the dc bus signal control (DBS).

DBS allows to build a simple, reliable and low-cost nanogrid; there is no need for external supervisor and no additional converter to manage the power flow between the NGs. Thus, this solution could be an effective solution for the installation in remote area, also in absence of an ac grid, where only low investments are possible.

However, the number of the NGs connected to the common dc bus could be limited by the number of thresholds adopted for the DBS. Moreover, the trigger of one threshold can be negatively affected by the dynamic behaviour of the controllers. The interactions among converter controllers may produce oscillations and instability, if the controllers are not properly designed.

In this setup the nanogrid supervisor is optional, because it is not directly involved in the power flows between NGs. Obviously, the independence from the supervisor increases the reliability of the microgrid; on the other hand, the supervisor allows advanced and dynamic strategies, mainly if long-term energy management strategies are adopted. For example, it can be used to carry out the results of an offline optimization problem.

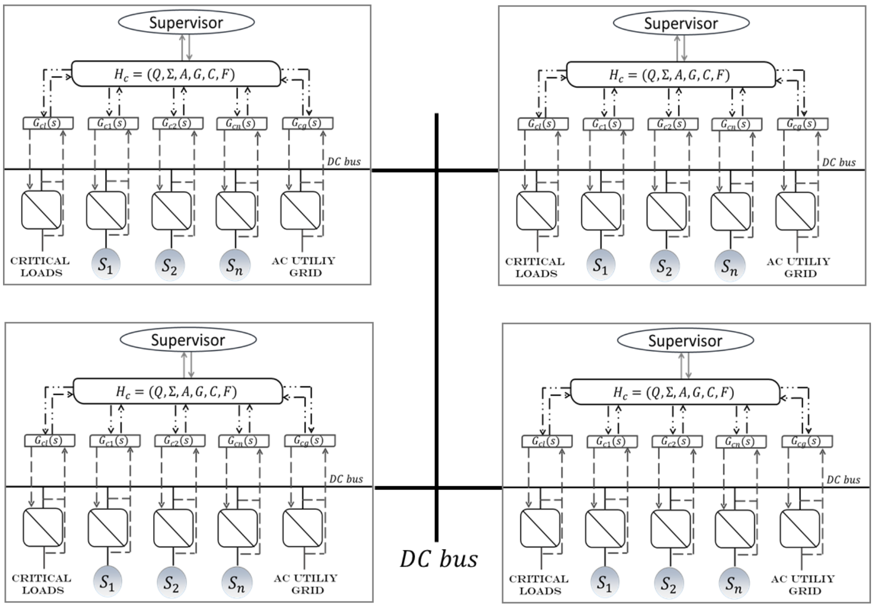


Fig. 3-6 Interconnected NGs through a common dc bus

To enhance the control capability of interconnected NGs, the setup in Fig. 3-7 can be adopted. Each NG is decoupled from the others and executes its own strategy. Each NG has a dcdc converter, which is controlled by the microgrid central control (MGCC). The MGCC controls one converter in voltage mode, in order to stabilize the common bus balancing the power flows in the MG; this converter could be connected to a common, optional storage of the coalition. The master converter acts either as load or as source depending on the sum of all other currents of the units. The other converters are current controlled and the reference current is decided by the MGCC. For the NG the external additional converter is another source or loads; this allows to the single NG to continue to operate autonomously. However, for the critical operations a high reliable communication infrastructure is needed. Thus, the costs are high, also due to the numerous additional components and to the redundancy and backup systems. The reliability of the single NG is not negatively affected by the communication link, because each NG can continue to operate autonomously, while the reliability of the MG is related to the MGCC and the communication infrastructure.

The dcdc converter for the power flow management between the common dc bus and the i-th bus of the i-th NG has to be a bidirectional converter. Moreover, the input and output voltage could be different and could change during the operations of the system. Probably, the best converter suited for this application is the dual active bridge converter, also for his high density power, high power capability and an isolation stage provided by the high-frequency transformer.

Another solution is to embed the gateway dcdc converter, used to interface the NG with the common dc bus, inside the NG, controlled by the same NG controller. The converter can be treated as the other converters and the commands from the MGCC are realized by the NG controller. The advantage is a more compact structure, a reduced components number thanks to the embedded control board already available, and the possibility to left the dynamic stability study to the NG designer.

The dcdc converter interfaced with the dc bus of the coalition is controlled by the discrete layer, that receives inputs from the supervisor. The MGCC has to realize a strategy to avoid collapse of the common dc bus. The only one converter controlling the dc bus, acting as master converter, is a valid solution also in this setup, but other strategies like DBS or droop control can be effectively implemented. However, this setup reduces the number of inputs of each NG, so a reduced number of sources can be connected to the device.

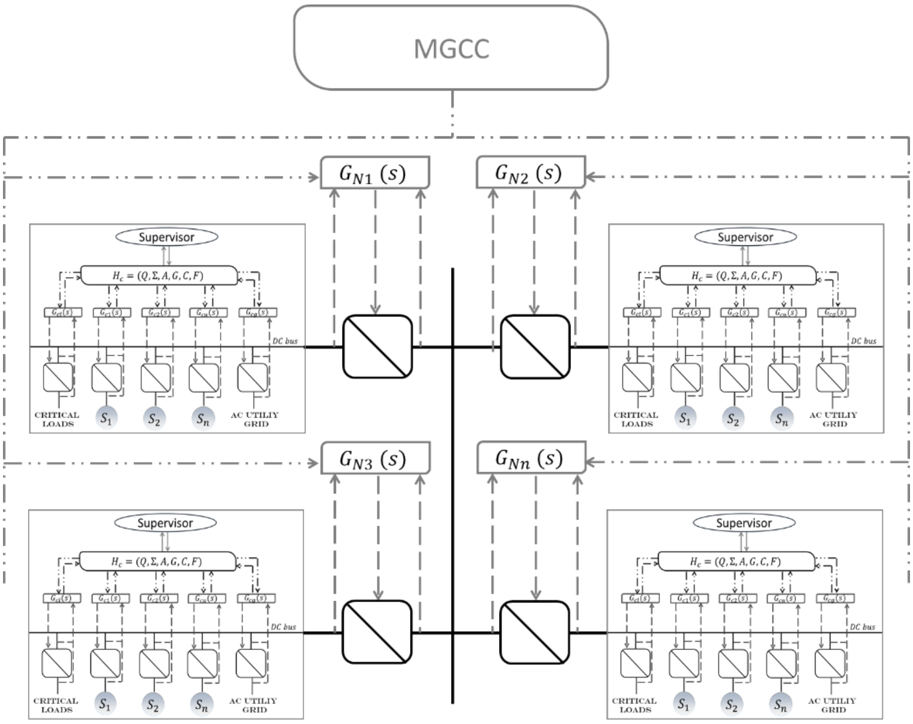


Fig. 3-7 Interconnected NGs coordinated by a MG supervisor controlling external power converters

3.3 Converters Models and Continuous Control Layer

There are a variety of technologies used with nanogrids, but the subject that dominates the nanogrid literature is converter topologies. Converters are responsible, within the nanogrid, for manipulating voltages to meet the requirements of a specific task. This is typically interfacing the nanogrid's sources with the systems bus, the national grid and interfacing the nanogrid's loads with the bus.

The conversion system has two different stages: the hardware stage and the control one. The hardware has reactive components, which define the filter stage useful to remove high frequency harmonics, and switching components such as diodes, metal-oxide-semiconductor field-effect transistors (MOSFETs) and insulated-gate bipolar transistors (IGBTs). The control stage produces a pulse width modulation (PWM), which is the output of the control stage and the input for the hardware stage. By rapidly switching the converter between states using PWM, the desired output voltage can be obtained. Moreover, by measuring the output (or input) voltage (or current) of the converter, the PWM can be altered to ensure the output voltage remains stable even when the input voltage varies. This property is essential when implementing nanogrid control strategies. Other than perform the output voltage control, the converter can perform other tasks within the nanogrid, such as maximum power point tracking (MPPT) of a renewable source or charge a battery bank.

The control algorithm and the modulation technique is specific for each converter. The algorithm is implemented in a digital processor and the digital control system has to be designed for the specific converter based on the requirements. For each task of the converter a different control has to be carried out. A lot of modern software allow accurate numeric simulation in time-domain; with these programs, however, simulation results are not suitable to obtain direct information about the converter modes. Obtaining a model necessary for control purposes cannot be done by using these software packages, at least not directly. Certainly, it is possible to identify a power electronic converter based on the input-output variables evolution file obtained by simulation and to elaborate a frequency-domain model (black-box model). But, since the quasi-totality of power electronic converters is naturally a nonlinear or linear time-varying system and any linear input-output model depends on the operating point, the information acquired has validity only for the specific operating point used to linearize the converter. Besides, it's useful to have a non-linear analytical model to use for designing purposes and to simulate the converters with equation systems that describe their behaviour. To that end, it's needed to produce a model for the physical system.

Modelling a phenomenon is the process of capturing into an approximate, but sufficiently exhaustive representation, the most significant features of an application.

There are two main modelling approaches: one that uses black-box models, based on the observation of the response to some known input signals, producing input-output characteristics, and one based on the information about the system to be modelled.

The power electronic converter modelling should rely in using the "information" approach. This means that model representations will be made using the available physical knowledge, i.e. the physical structure, about the considered converter. In general, physical knowledge about system results in mathematical description of mass and energy conservation laws. Thus, energy accumulation variations within the system are described by the so-called state variables. In the particular case of power converters, information is embodied in Kirchhoff's laws of the converter circuit, Ohm's laws for the various loads and, finally, in the states of various solid-state switches.

There are various modelling methodologies. The choice of which one is suitable relies firstly on the intended use. Regarding other possible uses of power electronic device models, one can identify roughly three main classes: models used for control purposes, models dedicated to simulation purposes and models employed for sizing and dynamical analysis. An analytical model based on knowledge of the circuit's physical behaviour is needed for control purposes. Other criteria in the choice of the model are the following: the required dynamic or steady-state accuracy; whether internal, input or output variables should explicitly appear in the model; an acceptable level of complexity; the domain of definition.

These requirements often are antagonistic, necessitating an optimal choice. For example, the accuracy of plant replication increases with model complexity. To simplify the modelling process, it's suitable to consider some hypotheses, which eliminate some information, but ensuring a simpler model. Some of the simplifying hypotheses are:

1. Switches are considered 'perfect' in the sense that they behave as a zero-value resistance during conduction and as an infinite value resistance when the switch is turned off. Also, the switching time is infinitely short;
2. Generators are considered ideal;
3. Passive elements are considered linear and invariant – with no connection with external and internal variables;

Each modelling technique has some characteristics, which make them interesting for a specific application. A satisfying overview on modelling technique used for converters are in [76]; the most common are:

- Switched model describes the electrical equations for each circuit configuration. The switched model is the product of such an analysis and it is particularly suitable for designing nonlinear control laws, such as variable-structure or hysteresis control.

- A sampled-data model is a model that provides information about the system state in a periodic manner.
- Average models replicate an average behaviour of the system state. It is computed on a time window of width T, which is sufficiently small in relation to the system dynamics.

Almost all converters have a non-linear dynamic behaviour. In order to perform a modal analysis or to build linear control laws a linear model is needed. The linear model could be obtained linearizing a non-linear model around a certain operating point [77]. These linearized models are valid only for slight variations around the considered operating point: the reliability of the model is ensured only for small input perturbations around the linearized point. The linearized models are called small-signal models. Conversely, the initial models, valid on the entire definition range, are called large-signal models by the power electronics community.

The large signal model has validity inside all the defined domain, i.e. for all the plausible operating conditions of the device. The small signal is derived by the large model, forcing the change in time of state variables to zero. This condition is the stationary solution of a given operating condition.

An interesting feature in modelling, desired for some particular applications, is to represent behaviour of converters containing AC stages. In this regard, an interesting modelling technique is the generalized state-space average (GSSA); it can handle averages of higher-order harmonics, thus able to represent ac variables, that is important in any power electronics application containing AC power stage. This technique has been chosen to model the converters in the nanogrid for its general approach.

The controllers are SISO controllers, even if the overall system has more inputs and outputs. However, with SISO controllers each converter operates its own task: the important design requirement is to take the dynamic decoupled between controllers. This is possible by designing the controllers that could interact with different time constants. Now, some control loops used in the various prototypes are illustrated. The design has been done using the analytical models produced for each converter.

The control system for the converters has been designed using classical control techniques, such as interactive loop shaping with Bode and root-locus plots on nonlinear converter models that include switching effects using methods such as AC frequency sweeps and system identification.

3.3.1 GSSA modelling technique

Starting with the non-linear time-variant equations it's possible "to average" the behaviour of the real physical system, preserving only the low-frequency behaviour. The building idea behind the GSSA is to describe the generic state variable $x(t)$ according to Fourier series, that is:

$$x(t) = \sum_{k=-\infty}^{+\infty} x_k(t)\, e^{jkwt}$$

Where w is the fundamental pulsation and $x_k(t)$ is the coefficient of the k-th harmonic. From the series definition, the coefficient is the average on a period T=2π/ω:

$$x_k(t) = \int_{t-T}^{t} x(\tau) e^{-jkw\tau} d\tau$$

The function is integrated on a time window T moving across the time axis. The time-variant result of the expression corresponds to the sliding average. A difference with the arithmetic average is that the average change in time. However, if the signal is periodic and stationary the two averages could coincide.

The coefficient results from the sliding averaging operation follows the notation:

$$x_k(t) = \langle x \rangle_k(t)$$

The equations with the variables expressed by the sliding average shown two fundamentals properties; these properties are demonstrated in [76]. The final model is the result of the application of these properties and simplifications. The first property allows to rewrite the averaged variables like:

$$\frac{d}{dt}\langle x\rangle_k(t) = \left\langle\frac{d}{dt}x\right\rangle_k(t) - jkw\langle x\rangle_k(t)$$

The second property allows to rewrite the sliding average product of two quantities of the k-th order. The result is that the product is the summation of k-terms according to the following equation:

$$\langle x\, y\rangle_k(t) = \sum_i \langle x\rangle_{k-i}(t)\langle y\rangle_i(t)$$

Thus, for example, the product between two variables with k equals to 0 is:

$$\langle x\, y\rangle_0 = \langle x\rangle_0\langle y\rangle_0 + 2(\langle x_{1R}\rangle\langle y_{1R}\rangle + \langle x_{1I}\rangle\langle y_{1I}\rangle\,)$$

The product of the first order between two variables can be written according to:

$$\langle x\, y\rangle_{1R} = \langle x\rangle_0\langle y\rangle_{1R} + \langle x\rangle_{1R}\langle y\rangle_0$$

$$\langle x\, y\rangle_{1I} = \langle x\rangle_0\langle y\rangle_{1I} + \langle x\rangle_{1I}\langle y\rangle_0$$

Applying the rules for each state variable, where each variable has a given number of harmonics, the final model is obtained.

The time-domain variable can be reconstructed by using the information provided by the model in terms of averaged variables. Let's consider an AC variable, not necessarily sinusoidal, but whose information of interest is contained in its fundamental term. The variable is time-variant and can be approximated by its first-order sliding harmonic:

$$y(t) \approx\ \langle y\rangle_1 e^{jwt} + \langle y\rangle_{-1}e^{-jwt}$$

giving after some algebra

$$y(t) \approx\ 2[Re(\langle y\rangle_1)\cos wt - Im(\langle y\rangle_1)\sin wt]$$

For simplicity let us put $x_1 = Re(\langle y\rangle_1)$ and $x_2 = Im(\langle y\rangle_1)$. The equation will be rewritten under the form:

$$y(t) \approx\ 2(x_1\cos wt - x_2\sin wt)$$

The equation shows a way in which the real waveform $y(t)$ can be obtained based on the real and the imaginary parts of its first-order sliding harmonic, $\langle y\rangle_1(t)$.

The classical averaged model is a particular case of the GSSA, in which only the zero harmonic is taken into account, but it is not suitable to represent ac variables inside a converter. Moreover, considering a higher number of harmonics allows to produce a more accurate model, because one can take more harmonics for variables that have an average value, but their value is not time-invariant in stationary condition.

To demonstrate the steps needed to produce a model, two of the most important converters adopted in the nanogrid are modelled according to the GSSA technique. These models have been used during the preliminary stage for numeric simulations, for the design process of the low level control of each converter and for the tuning stage during the experimental tests to obtain the desired dynamic performance.

When designing a power converter, the simulation should consider to perform the following tasks:

- Designing a feedback controller for the specific goal;

- Optimizing RLC components concurrently with the controller design;
- Estimating the steady-state and dynamic characteristics of the semiconductor switches;
- Analysing dynamic performance and power quality;
- Preparing the digital controller for the implementation on an embedded microprocessor or an FPGA.

3.3.2 Storage system converter

The interface converter for the storage system controls the power flow between the storage system and the dc bus. The adopted converter is the half-bridge converter (Fig. 3-8). The half bridge converter is a buck/boost converter with bidirectional power flow. The structure is very simple and it can be obtained easily using one leg of an integrated power module to save development time. The converter has a LC filter in the output stage to cut off the high frequency harmonics. The values of the filter depend on the maximum current, the maximum ripple and the switching frequency.

An analytical state space average model for the buck/boost is given in [78]. However, a GSSA model is now presented; this model is preferred because the input and output capacitor voltages appear in the model, other than the ac component of the state variables. The model is produced for the specific application because the converter has to be capable to control the input or the output voltage, according to the specific task required.

The inductor current and the two voltages on the input and output capacitors are the selected state variables. The three state variables allow to write three equations for the converter:

$$\begin{cases} (V_{Cin} + V_{rcin})u_1 = V_{rds1}u_1 + V_L + V_{rl} + V_{Cout} + V_{rcout} + V_{rds2}u_2 \\ I_{in} = i_{Cin} + i_L u_1 \\ i_L = i_{Cout} + I_o \end{cases}$$

Where u_1 is the switching function for S1 and u_2 is the denier of u_1:

$$u1 = \begin{cases} 1 & 0 < t < DT_s \\ 0 & DT_s < t < T_s \end{cases}$$

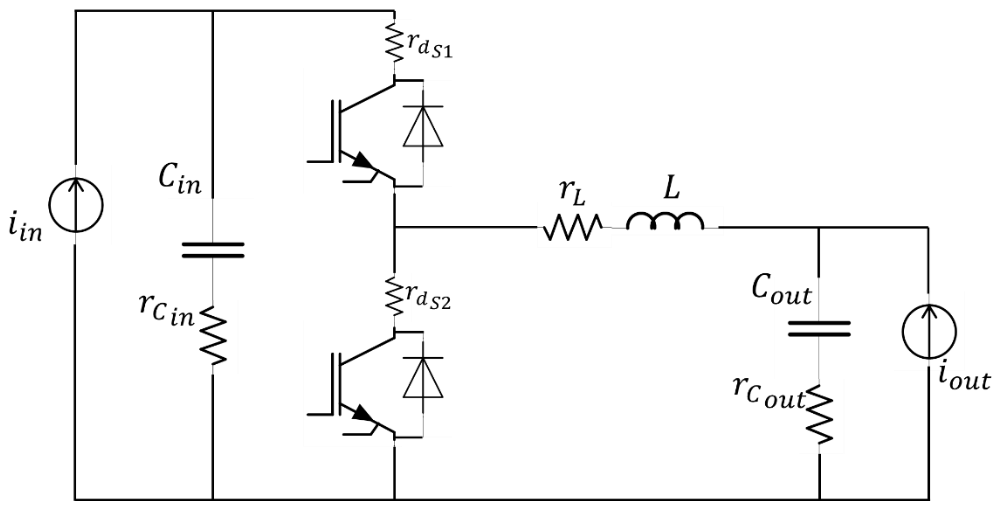


Fig. 3-8 Half bridge converter

To simplify the model $V_{rds2}u_2$ can be neglected and the equations can be rewritten as:

$$\begin{cases} L\dfrac{di_L}{dt} = v_{Cin}\,u_1 + r_{cin}\,i_{cin} - r_l i_L - v_{Cout} - r_{cout}\,i_{cout} \\ C_{in}\dfrac{dv_{Cin}}{dt} = I_{in} - i_L u_1 \\ C_{out}\dfrac{dv_{Cout}}{dt} = i_L - I_o \end{cases}$$

$$\begin{cases} L\dfrac{di_L}{dt} = v_{Cin}\, u_1 - r_l i_L - v_{Cout} + r_{cin}\,(I_{in} - i_L) - r_{cout}\,(i_L - i_o) \\ C_{in}\dfrac{dv_{Cin}}{dt} = I_{in} - i_L u_1 \\ C_{out}\dfrac{dv_{Cout}}{dt} = i_L - I_o \end{cases}$$

$$\begin{cases} L\dfrac{di_L}{dt} = v_{Cin}\, u_1 + (-r_l - r_{cout} - r_{cin})i_L - v_{Cout} + r_{cin}\, I_{in} + r_{cout}\, I_o \\ C_{in}\dfrac{dv_{Cin}}{dt} = I_{in} - i_L u_1 \\ C_{out}\dfrac{dv_{Cout}}{dt} = i_L - I_o \end{cases}$$

The variables number is related to the coefficient order of Fourier series. The first harmonic is considered only for the current inductor i_L; the zero harmonic is considered for the other two state variables. Thus, five equations are taken out: i_L is described by one equation with the continuous value and by other two equations that are the real and imaginary components.

The first equation for current is the equation with the zero harmonic:

$$L\frac{d\langle i_{L0}\rangle}{dt} = L\,\langle\frac{di_{L0}}{dt}\rangle = \langle v_{Cin0}\, u_0\rangle + (-r_l - r_{cout} - r_{cin})\langle i_{L0}\rangle - \langle v_{Cout0}\rangle + r_{cin}\,\langle I_{in0}\rangle + r_{cout}\, I_o$$

$$L\,\langle\frac{di_{L0}}{dt}\rangle = \langle v_{Cin0}\rangle\langle\, u_0\rangle + (-r_l - r_{cout} - r_{cin})\langle i_{L0}\rangle - \langle v_{Cout0}\rangle + r_{cin}\,\langle I_{in0}\rangle + r_{cout}\, I_o$$

The other two equations describe the first harmonic current in terms of real and imaginary components:

$$L\frac{d\langle i_L\rangle_1}{dt} = L\,\langle\frac{di_L}{dt}\rangle_1 - jw\langle i_L\rangle_1 = \langle v_{Cin}\, u\,\rangle_1 - R\langle i_L\rangle_1$$

$$\begin{cases} \langle\dfrac{di_L}{dt}\rangle_{1R} = jwj\langle i_L\rangle_{1I} + \dfrac{1}{L}[\langle v_{Cin}\rangle_0\langle\, u\,\rangle_{1R} - R\langle i_L\rangle_{1R}] \\ \langle\dfrac{di_L}{dt}\rangle_{1I} = jw\langle i_L\rangle_{1R} + \dfrac{1}{L}[\langle v_{Cin}\rangle_0\langle\, u\,\rangle_{1I} - Rj\langle i_L\rangle_{1I}] \end{cases}$$

The other equation describes the voltage on the input capacitor, so the original equation becomes:

$$\langle\frac{dv_{Cin}}{dt}\rangle_0 = \frac{1}{C_{in}}\,[\langle I_{in}\rangle_0 - \langle i_L u_1\rangle_0]$$

$$\langle\frac{dv_{Cin}}{dt}\rangle_0 = \frac{1}{C_{in}}\,[\langle I_{in}\rangle_0 - [[\langle i_L\rangle_0\langle u_1\rangle_0 + 2(\langle i_L\rangle_{1R}\langle u_1\rangle_{1R} + \langle i_L\rangle_{1I}\langle u_1\rangle_{1I})\,]]$$

The last equation is the continuous value for the output voltage:

$$\langle\frac{dv_{Cout}}{dt}\rangle_0 = \frac{1}{C_{out}}\,[\langle i_L\rangle_0 - I_o]$$

To complete the model, the switching function has to be expressed in terms of sliding average. So the continuous value and the first harmonic of the switching function are expressed as:

$$\langle u_1\rangle_0 = d$$

$$\langle u_1 \rangle_1 = \frac{j}{2\pi}\left(e^{-j2\pi d} - 1\right) = \frac{\sin(2\pi d)}{2\pi} + j\left(\frac{cos(2\pi d) - 1}{2\pi}\right)$$

In the end, considering all the equations, the model is described by:

$$\begin{cases} \langle \frac{di_{L0}}{dt} \rangle = \frac{1}{L}\left[\langle v_{Cin0} \rangle d - R\langle i_{L0} \rangle - \langle v_{Cout0} \rangle + r_{cin}\, I_{in} + r_{cout}\, I_o\right] \\ \langle \frac{di_L}{dt} \rangle_{1R} = -w\langle i_L \rangle_{1I} + \frac{1}{L}\left[\langle v_{Cin} \rangle_0 \frac{\sin(2\pi d)}{2\pi} - R\langle i_L \rangle_{1R}\right] \\ \langle \frac{di_L}{dt} \rangle_{1I} = w\langle i_L \rangle_{1R} + \frac{1}{L}\left[\langle v_{Cin} \rangle_0 \left(\frac{cos(2\pi d) - 1}{2\pi}\right) - R\langle i_L \rangle_{1I}\right] \\ \langle \frac{dv_{Cin}}{dt} \rangle_0 = \frac{1}{C_{in}}\left[\langle I_{in} \rangle_0 - \langle i_L \rangle_0 d - 2\langle i_L \rangle_{1R} \frac{\sin(2\pi d)}{2\pi} - 2\,\langle i_L \rangle_{1I}\left(\frac{cos(2\pi d) - 1}{2\pi}\right)\right] \\ \langle \frac{dv_{Cout}}{dt} \rangle_0 = \frac{1}{C_{out}}\;\left[\langle i_L \rangle_0 - I_o\right] \end{cases}$$

The inductor current in the time-domain is the sum of the all sliding values and is expressed by:

$$i_L(t) = \langle i_{L0} \rangle + 2\langle i_L \rangle_{1R} \cos(wt) - 2\langle i_L \rangle_{1I} \sin(wt)$$

The model has been used to design and tune the feedback controller of the converter. The converter may control the output voltage in master mode or the current in current control mode. The input voltage during the last part of the recharge can be also controlled, according to constant voltage charging, if there is not a BMS to determine the charging current (Fig. 3-9).

The designed control loop has an internal fast current loop that regulates the battery current to the reference value provided by one of the external loops. The dc bus voltage control and the power control are the two possible external loops; the dynamic is governed by $G_{CV}(s)$ and $G_{CP}(s)$ respectively. These loops generate a reference current for the internal loop. However, the current value has to be within certain limits, according to the physical system. There are two physical limits, i.e. two maximum admissible currents to consider: the limit current of the energy storage system and of the power devices. While the latter is static and is selected during the initial configuration, the first is dynamic and is related to the actual condition of the energy storage system. This current limit is the result of the activity of a BMS, which generates a maximum charging and maximum discharging current. The BMS gives the maximum currents that the storage can feed or absorb, according to the temperature, the SOC and other parameters considered in the BMS control logic. However, if the BMS is not available, such as in lead acid batteries, the charging current is calculated inside an external loop; in this condition the charge follows a first phase in constant current mode and the last phase in constant voltage mode: in this way, the charging current is gradually reduced until the storage is full.

With the solution in Fig. 3-9 the physical constraints of the battery are respected. For example, let's suppose the storage converter is the master and is absorbing power from the dc bus to maintain the reference value for the dc voltage. If the SOC of the battery rises to 100% the battery can't continue in this operating condition. To safeguard the battery, the control system has to stop the charging as soon as the BMS set the charging current to zero. This is directly done in the control system in Fig. 3-9.

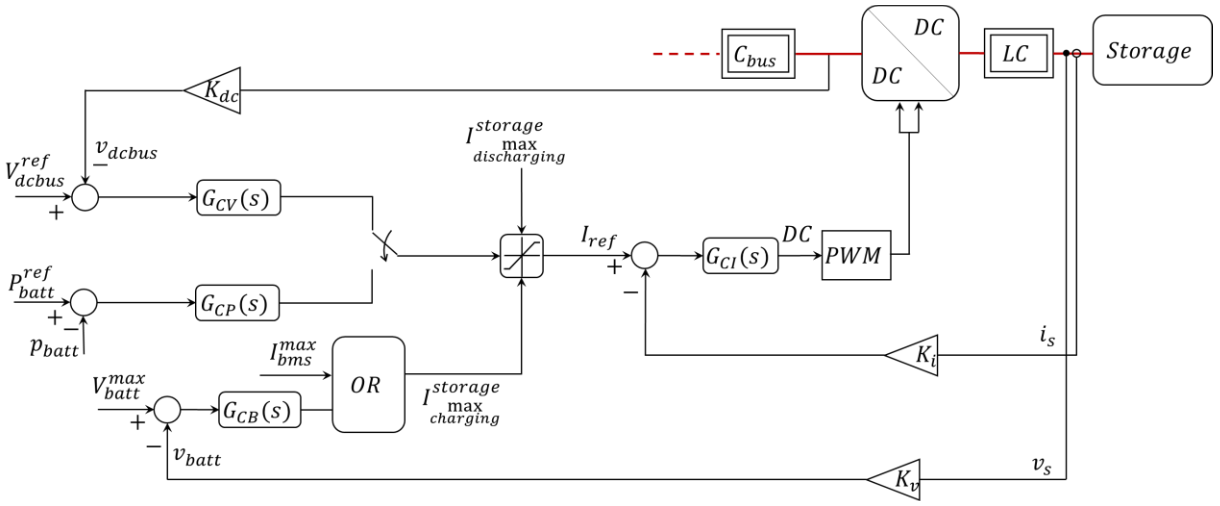


Fig. 3-9 Control loops of the half-bridge converter for the energy storage system

The equations of the developed analytical model have been implemented in Simulink. Moreover, to validate the model, the converter has been simulated using the components available in the Simulink library using standard circuit components. The converter parameters are in Table 3-1.

Table 3-1 Parameters of the simulated half-bridge converter

Component	Value
DC voltage source	200 V
rsource	0.05 Ohm
Cout	25e-6 F
rCout	1e -3 Ohm
L	1e-3 H
Rl	0.2 Ohm
Cbus	1e-3 F
rCbus	1e-3 Ohm
Rload	1000 Ohm

Both the simulations, the analytical model and the physical model simulations, suppose the following:

- The goal for the converter is to control the dc bus using a dc source;
- The dc bus reference value at time t=2s jumps from 400 to 420 V;
- A dc bus load jumps from 0 to 2 A at t=4 s.

The control system for the converter has been designed using classical control techniques such as interactive loop shaping with Bode and root-locus plots.

The simulated converter using the standard circuit components is in Fig. 3-10.

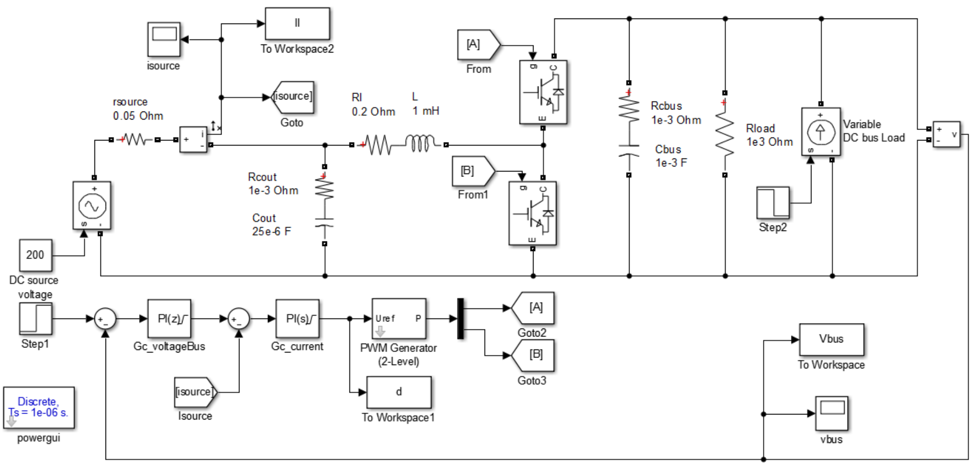


Fig. 3-10 Simulated physical half-bridge converter using standard components

The simulation is run with a discrete time-step of 1 us. The switching frequency is equal to 15 kHz. The goal of the controller is to regulate the DC bus voltage to 400 V. The system needs to reach a stationary operating point after the start of the simulation. When the stationary operating point is reached (t=1s) the reference value of the controller is changed with a step input from 400 to 420 V. When t=2 s the current of the dc bus load is increased from 0 to 2 A.

Fig. 3-11 and Fig. 3-12 show the dynamic behaviour of the dc bus controller, but also the comparison between the analytical model and the physical simulated converter. Especially, the Fig. 3-12 shows how the current of the physical simulated converter, as expected, has a richer harmonic content than the analytical model, where only the continuous component and the first harmonic are considered.

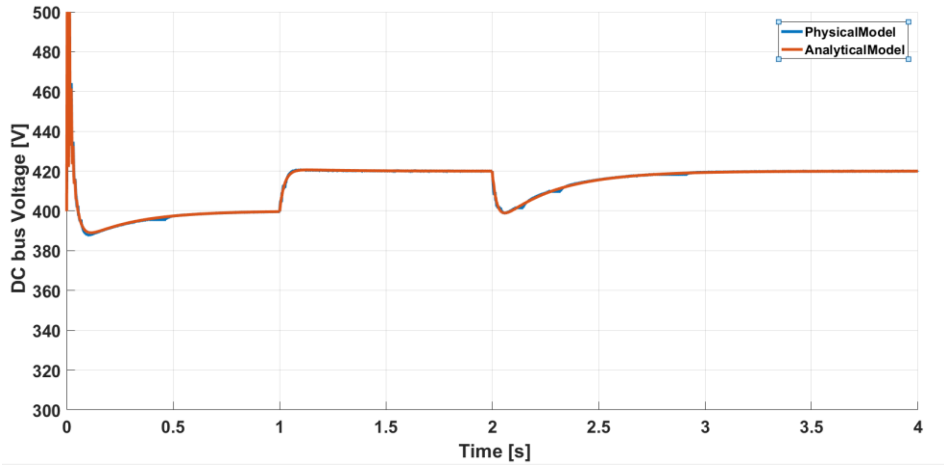


Fig. 3-11 Dc bus voltage of the simulated systems: physical simulated converter (blue) and analytical model (orange)

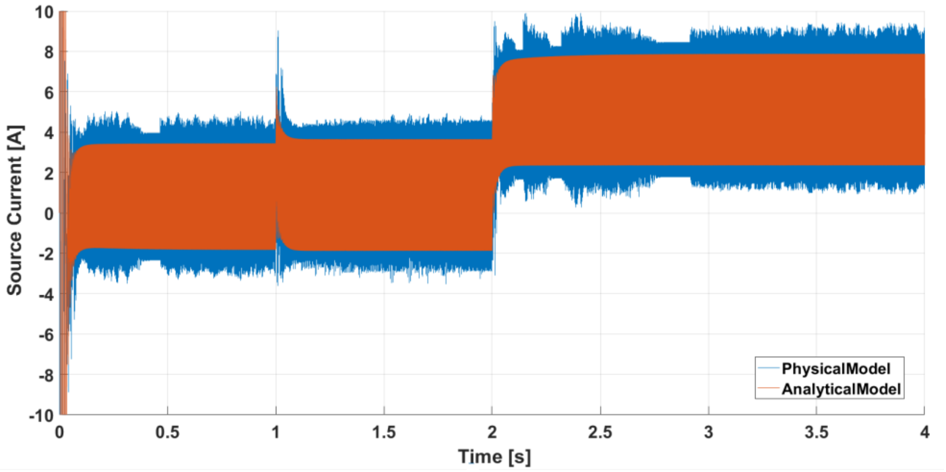


Fig. 3-12 Storage current of the simulated systems: physical simulated converter (blue) and analytical model (orange)

3.3.3 Grid connected inverter

In a dc NG the connection with the ac utility grid is provided by a dc-ac converter. This converter is important in grid-connected applications of the NG because is the interface with the external power domain.

The inverter is connected to the utility grid through a filter, which is important for the dynamic behaviour and the current ripple produced by the inverter. In [76] the model for a full-bridge grid-connected inverter with a L filter is presented. The model has been obtained with the GSSA model technique. With reference to Fig. 3-13, the state variables chosen are the current in the filter i_L and the dc bus voltage v_0. i_S is a current of a generic source on the dc bus; the grid voltage is indicated with e and u is the switching function. Two equations describe the system in Fig. 3-13 are:

$$\begin{cases} L\dfrac{di_L}{dt} = e - v_0 u \\ C\dfrac{dv_0}{dt} = i_L u - i_S \end{cases}$$

The converter dynamic behaviour is controlled by modifying either the amplitude of the input signal β or its phase (measured with respect to the grid voltage phase) in order to vary the balance between active and reactive AC power.

The GSSA technique consider a finite number of harmonics for each equation. The current in the inductor is an ac current, therefore the first-order sliding harmonic is considered. The voltage on dc side is a continuous value and the zero-order sliding harmonic is used. Obviously, greater is the order of the harmonics considered, the better is the accuracy of the model. The result equations are:

$$\begin{cases} L\left\langle\dfrac{di_L}{dt}\right\rangle_1 = \langle e\rangle_1 - \langle v_0 u\rangle_1 \\ C\left\langle\dfrac{dv_0}{dt}\right\rangle_0 = \langle i_L u\rangle_0 - \langle i_S\rangle_0 \end{cases}$$

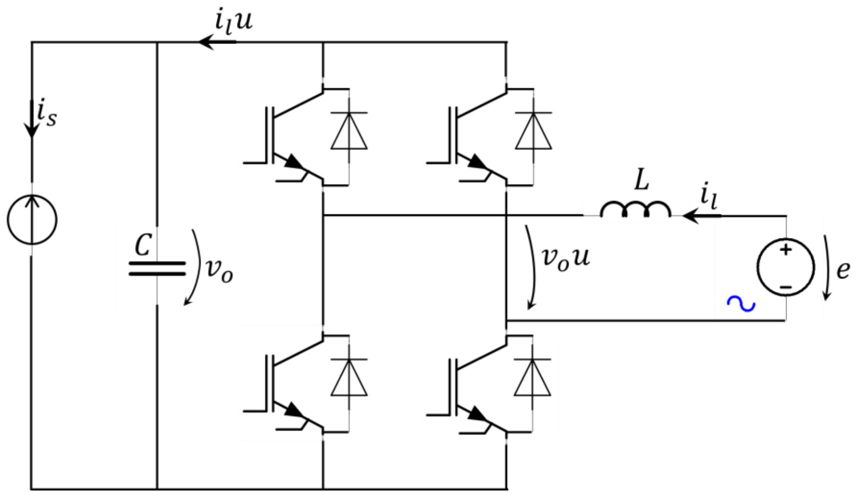


Fig. 3-13 Grid-connected full-bridge converter

It's simple to associate the real and imaginary components of the equations derived from the GSSA model inside a dq frame [76]. The resulted model, which represents the model of the grid-connected inverter, is:

$$\begin{cases} \dfrac{di_d}{dt} = wi_q + \dfrac{E}{L} - \dfrac{\langle v_0 \rangle_0}{L}\beta_d \\ \dfrac{di_q}{dt} = -wi_d - \dfrac{\langle v_0 \rangle_0}{L}\beta_q \\ \dfrac{d\langle v_0 \rangle_0}{dt} = \dfrac{1}{2C}(i_d\beta_d + i_q\beta_q) - \dfrac{1}{C}\langle i_S \rangle_0 \end{cases}$$

The model presented in [76] has the big advantage, over other models, to take into account the dc voltage and the ac current in the same model, without performing simplifications. This allows to design controllers that control both the state variables and to simulate the ac and dc sides of the inverter without simplifications.

The grid-connected inverter has two operating modes. The first mode is the DC bus voltage control. The bidirectional inverter exchanges power with the utility grid to control the dc bus voltage, maintaining the voltage to a reference value. The other mode is the power control, in which the inverter controls the active power exchanged with the utility grid (Fig. 3-14).

The two modes share an inner current loop. The current loop has the goal to produce a sinusoidal current between the filter and the ac grid equals to the sinusoidal reference current i_{ref}. The actual current is compared with the reference current and this value is the input of the $G_{CI}(s)$ controller. The output of this controller is added to the feed-forward value of the ac voltage. The output is the voltage waveform of the inverter needed to obtain the desired reference current. The $G_{CI}(s)$ is a PR-controller, where the resonance frequency has the same value of the utility grid frequency.

The different control mode is selected with the insertion of the external control loop, realizing a cascade control. The $G_{CV}(s)$ controls the voltage on the dc bus. This mode is selected when the inverter is the master control inside the NG. The controller is a PI. The time constant of this controller is 10 times higher, to avoid interactions with the internal current loop.

Also the power controller produces a current reference value for the inner loop. The difference is the feedback value; the power compared with the reference value is calculated inside the power block module. In designing the $G_{CP}(s)$ controller the delay introduced by the power block module has to be taken in account, because this frequency is relatively low compared to the sampling frequency of the control loop.

The reactive power control can be obtained modifying the v_g^{1h} waveform. The default is to maintain this waveform in phase with the voltage waveform, to avoid exchange of reactive power with the grid. The amplitude of this waveform is unitary.

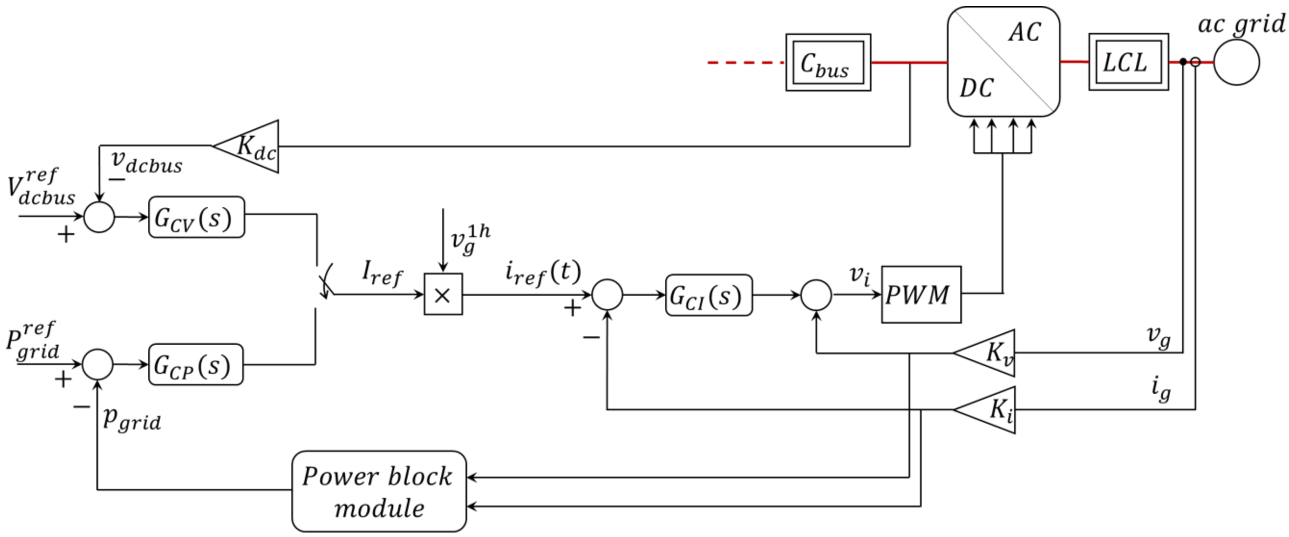


Fig. 3-14 Control loops for the grid-connected inverter

This simple control loop has been modified to allow harmonic compensation in [79]. A parallel repetitive control has been used to obtain a current with only a first order harmonic.

The GSSA model has been compared with the simulation results of a simulated grid-connected inverter. This inverter is simulated in Simulink using standard circuital components. The data of the model are in Table 3-2.

The simulation for both analytical and physical models is done according to the following points:

- The goal of the converter is to control the dc bus voltage;
- The dc bus voltage reference changes at t=3s from 400 to 420 V;
- The dc current load changes at t=5s from 0 to 1 A.

Table 3-2 Parameters of the simulated grid-connected inverter

Component	**Value**
Cbus	3e-3 F
L	3 mH
Rl	0.7 Ohm

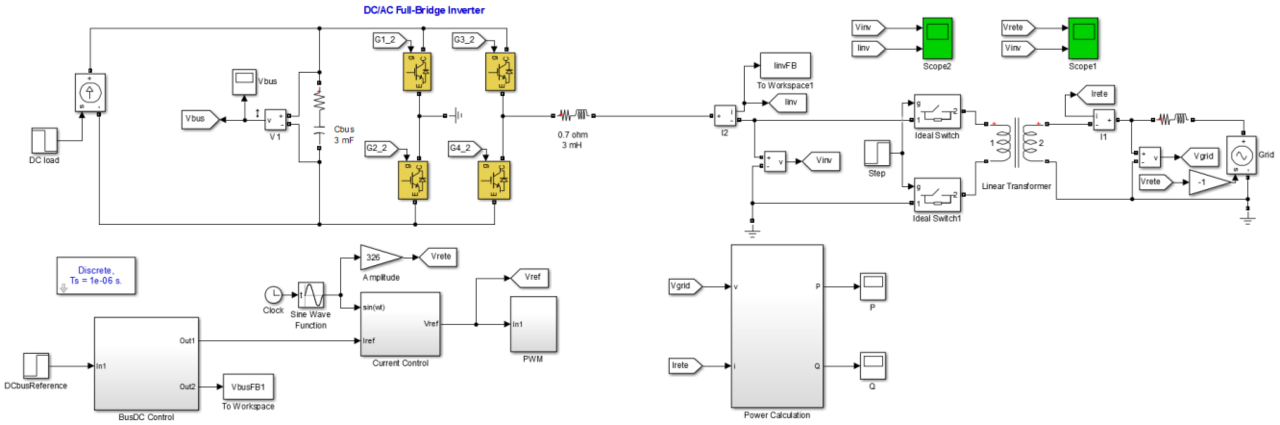


Fig. 3-15 Simulated grid-connected inverter with L-filter

The goal of the controller is to maintain the dc bus voltage to 400 V. The reference value steps from 400 to 420 V at t=3s. The controller tracks the reference value; thus, the new operating mode has a 420 V dc bus voltage. When t=5s the dc load increases and the controller absorbs power from the grid to restore a stationary operating mode. The resulted dc bus voltage for the GSSA model and for the simulated physical model are in Fig. 3-16.

The GSSA model considers the ac current between the inverter and the grid. The used model has only the first harmonic, to maintain the simple structure of the model. The current for both models is in Fig. 3-17. The current of the physical model differs from the analytical model, especially for low currents, because the low filter capability of the L-filter. However, the first harmonic of the analytical model, the only one considered in the model, closely approximates the current of the simulated inverter.

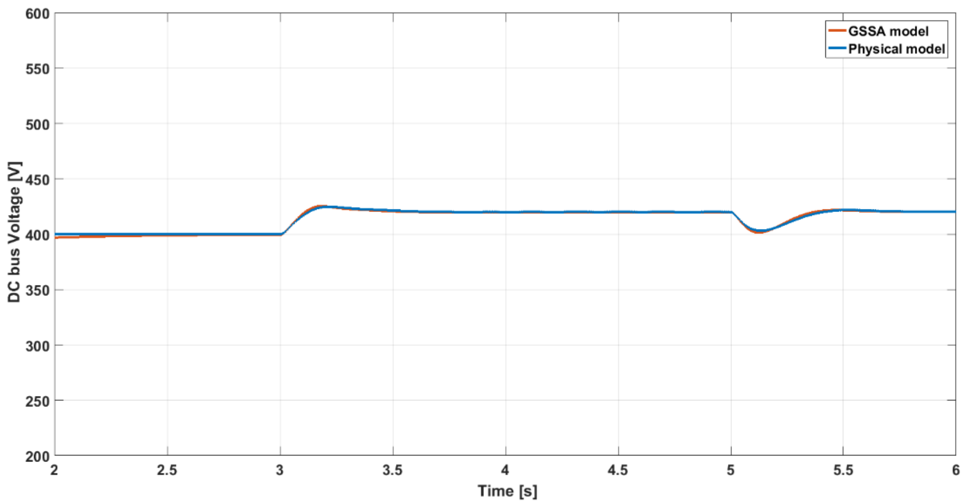


Fig. 3-16 DC bus voltage controlled by the inverter

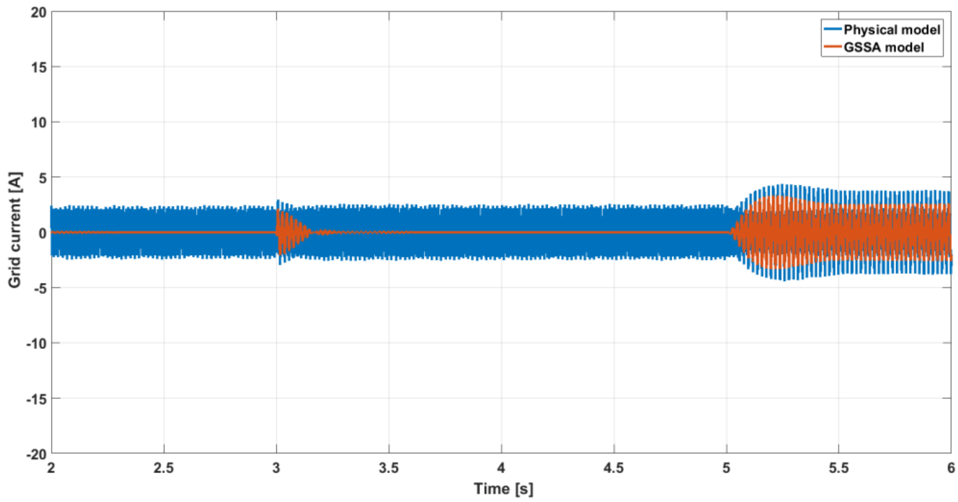


Fig. 3-17 Current between the inverter and the ac utility grid

The limit of this model is in the considered output filter. Indeed, the common practice is to use a LCL filter in grid-connected inverters instead of a simple L filter [80]. The LCL filter provides a better decoupling between the filter and grid impedance (it reduces the dependence of the filter on the grid parameters) and a lower current ripple. The drawback is that the resultant filter is a third-order filter and introduces a resonance frequency, which value depends on the filter parameters, according to:

$$fris = \frac{1}{2\pi}\sqrt{\frac{L_1 + L_2}{L_1 L_2 C}}$$

This frequency could degrade the performance of the converter and the response of its control loop, especially if a proportional-resonant controller is implemented. However, even if the model with a L inductor doesn't consider this resonance it could be used under certain circumstances. Indeed, the resonance frequency can be damped; the damping brings a similar frequency response for L and LCL filters.

The Kirchoff's laws for the filter in Fig. 3-18 are:

$$\begin{aligned} i_i - i_c - i_g &= 0 \\ v_i - v_c &= i_i(sL_1 + R_1) \\ v_c - v_g &= i_g(sL_2 + R_2) \\ v_c &= i_c\left(\frac{1}{sC} + R_1\right) \end{aligned}$$

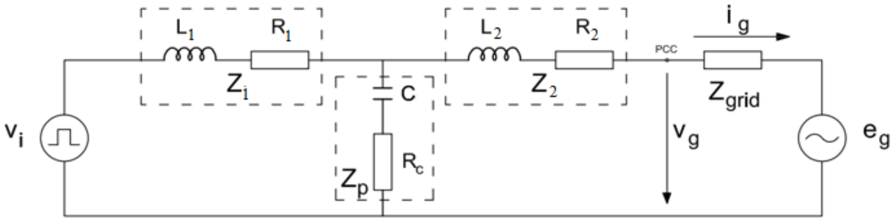


Fig. 3-18 Circuital representation of the LCL filter

Assuming an ideal grid voltage source, so that is a short circuit for harmonic frequencies and applying small signal perturbations, the transfer function $H_{LCL} = i_g/v_i$ is expressed by:

$$H_{LCL} = \frac{1 + sR_cC}{s^3L_1L_2C + s^2C\left(L_2(R_c + R_1) + L_1(R_c + R_2)\right) + s\left(L_2 + L_1 + C\,(R_cR_2 + R_cR_1 + R_2R_1)\right) + R_2 + R_1}$$

A Bode diagram of the H_{LCL} transfer function is in Fig. 3-19. The resonance can be clearly noted.

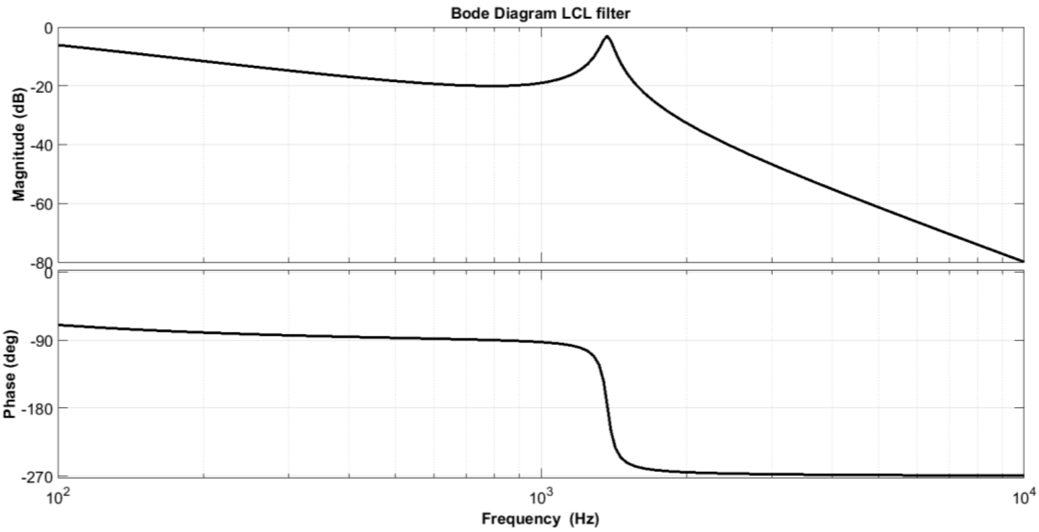


Fig. 3-19 Bode diagram of the filter transfer function: Ri=0.3 Ohm, Li=2.33e-3 H, Cf=25e-6F, Rc=0 Ohm, Rg=0.4 Ohm, Lg=0.7e-3 H.

To avoid the resonance peak a number of techniques can be adopted [81], [82]. These methods can be divided into two categories: active and passive methods. The passive methods add one or more passive components in the filter; this implies smaller efficiency. On the other hand, the active methods increase the complexity of the control structure and the number of measurement. Besides, most of the active methods make use of the filter value to shave the resonance peak. By inserting the filter values directly into control loops, inherent unstable elements of the system can be cancelled. Inherent poles in LCL transfer function are cancelled in the closed-loop transfer function. As a consequence, the resonance peak in LCL filter is effectively damped without affecting filter performance in high-frequency range; the effect is similar to the addition of a damping resistance in the LCL filter. However, with the aging of the components and with temperature changes, both L and C values will drift from the original values, causing a negative effect on the damping performance. This is a serious drawback because if the value is not exact the frequency response of the filter may get worse.

The most interesting method to damp the frequency resonance is to emulate a passive component, such as a resistor, in the filter stage. A passive resistor damps the resonance peak; if the value is big enough the transfer function can be similar to a transfer function of the first-order filter, simplifying the design of the controller. However, bigger is the value of the resistor higher are the power losses. If the effect of the resistor in the transfer function of the LCL filter can be emulated, the resonance can be avoided. Besides, introducing in the control loop a component that doesn't depend on the LCL physical values make the system more robust.

The similar frequency response of a LCL filter with physical resistor and of a LCL with an emulated resistor is now demonstrated. The block diagram of a simple LCL filter is in Fig. 3-20. There are two inputs: the inverter voltage and the grid voltage. The output is the current between the filter and the grid.

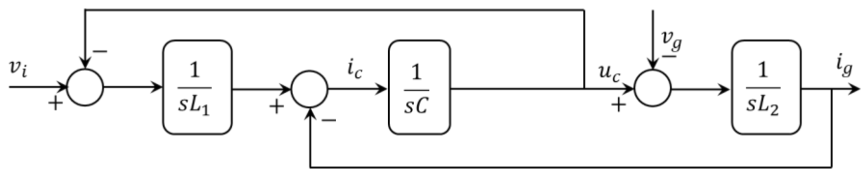


Fig. 3-20 Block diagram of a LCL filter without damping

If a physical resistor is inserted in series with the filter capacitor the block diagram is modified, as in Fig. 3-21. The current series resistor is the same current of the filter capacitor: the voltage on the resistor is added to the voltage capacitor.

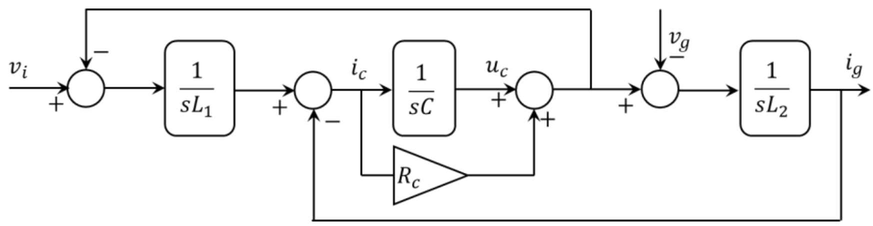


Fig. 3-21 Block diagram of a LCL filter with passive damping

If the Manson's rule is applied to the block diagram the transfer function is inferred. Two possible paths exist between v_i and i_g. The sum of the transfer functions along these paths are the numerator of the final transfer function.

$$F1 = \frac{1}{s\,L_1}\frac{1}{s\,C}\frac{1}{s\,L_2}$$

$$F2 = \frac{1}{s\,L_1}R_c\frac{1}{s\,L_2}$$

The denominator is 1-Δ, where Δ is the summation of the transfer function of all the loops present in the block diagram. In Fig. 3-21 there are four loops and each loop produces a term:

$$L1 = -\frac{1}{s\,L_1}\frac{1}{s\,C}$$

$$L2 = -\frac{1}{s\,L_1}R_c$$

$$L3 = -\frac{1}{s\,L_2}\frac{1}{s\,C}$$

$$L4 = -\frac{1}{s\,L_2}R_c$$

Therefore, the overall transfer function is:

$$G_{LCL_{pr}} = \frac{i_g}{v_i} = \frac{\frac{1}{s\,L_1}\frac{1}{s\,C}\frac{1}{s\,L_2} + \frac{1}{s\,L_1}R_c\frac{1}{s\,L_2}}{1 - \left(-\frac{1}{s\,L_1}\frac{1}{s\,C} - \frac{1}{s\,L_1}R_c - \frac{1}{s\,L_2}\frac{1}{s\,C} - \frac{1}{s\,L_2}R_c\right)} = \frac{\frac{1}{s^3C\,L_1L_2} + \frac{R_c}{s^2\,L_1L_2}}{1 + \left(\frac{L_1 + L_2 + sR_cCL_2 + sR_cL_1C}{s^2\,L_1L_2C}\right)}$$

$$= \frac{\frac{1 + sR_cC}{s^3C\,L_1L_2}}{\frac{s^2\,L_1L_2C + L_1 + L_2 + sR_cCL_2 + sR_cL_1C}{s^2\,L_1L_2C}}$$

$$= \frac{1 + sR_cC}{s^3\,L_1L_2C + sL_1 + sL_2 + s^2R_cCL_2 + s^2R_cL_1C}$$

$$= \frac{1 + sR_cC}{s^3\,L_1L_2C + s\,(L_1 + L_2) + s^2R_c(L_1 + L_2)}$$

The series resistor is both in numerator and denominator. In the denominator the R_c produces a second-order term. Thus, if in the control loop the current has a gain equals to R_c and the capacitor voltage is augmented by the voltage across the resistor R_c the system response will show a shaved resonance peak.

However, the current measurement of the capacitor current is not a trivial task, is not practical, not to mention the fact that a one more measurement is needed. Thus, let's try to produce a similar result with the voltage measurement instead of the current measurement. The method is also described in [81], but a formal description and a demonstration are now provided.

The block diagram of the previous Fig. 3-21 is modified with an additional loop, as in Fig. 3-22. The capacitor voltage is multiplied by sCK, where s is the Laplace variable, and the result is added to the output control signal, i.e. to the voltage inverter. To compare the result in the frequency domain the Manson's rule is applied again. All terms are now reported:

$$F1 = \frac{1}{s\,L_1}\frac{1}{s\,C}\frac{1}{s\,L_2}$$

$$L1 = -\frac{1}{s\,L_1}\frac{1}{s\,C}$$

$$L2 = -\frac{1}{s\,L_1}\frac{1}{s\,C}\;s\,C\,K$$

$$L3 = -\frac{1}{s\,L_2}\frac{1}{s\,C}$$

If $K = \frac{L_1+L_2}{L_2}\,R_c$, then:

$$G_{LCL_{vr}} = \frac{i_g}{v_i} = \frac{\frac{1}{s\,L_1}\frac{1}{s\,C}\frac{1}{s\,L_2}}{1 - \left(-\frac{1}{s\,L_1}\frac{1}{s\,C} - \frac{K}{s\,L_1} - \frac{1}{s\,L_2}\frac{1}{s\,C}\right)} = \frac{\frac{1}{s^3C\,L_1L_2}}{\frac{s^2\,L_1L_2C + L_1 + L_2 + sR_cC(L_2 + L_1)}{s^2\,L_1L_2C}}$$

$$= \frac{1}{s^3\,L_1L_2C + s(L_1 + L_2) + s^2\,R_cC(L_2 + L_1)}$$

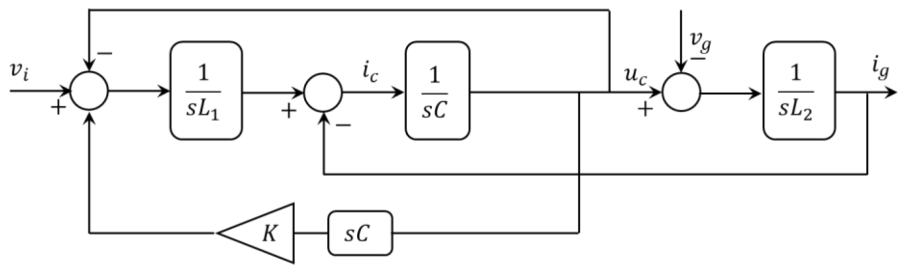


Fig. 3-22 Block diagram of a LCL filter with virtual damping

$G_{LCL_{vr}}$ has the same denominator of the transfer function with the physical resistor $G_{LCL_{pr}}$, i.e. the same poles. The result frequency response is not affected by the resonance peak and is similar, although not identical, to the transfer function with physical resistor. The difference is due to a zero in $G_{LCL_{pr}}$. As shown in Fig. 3-23 the behaviour differs only to high frequency after the knee of the magnitude plot.

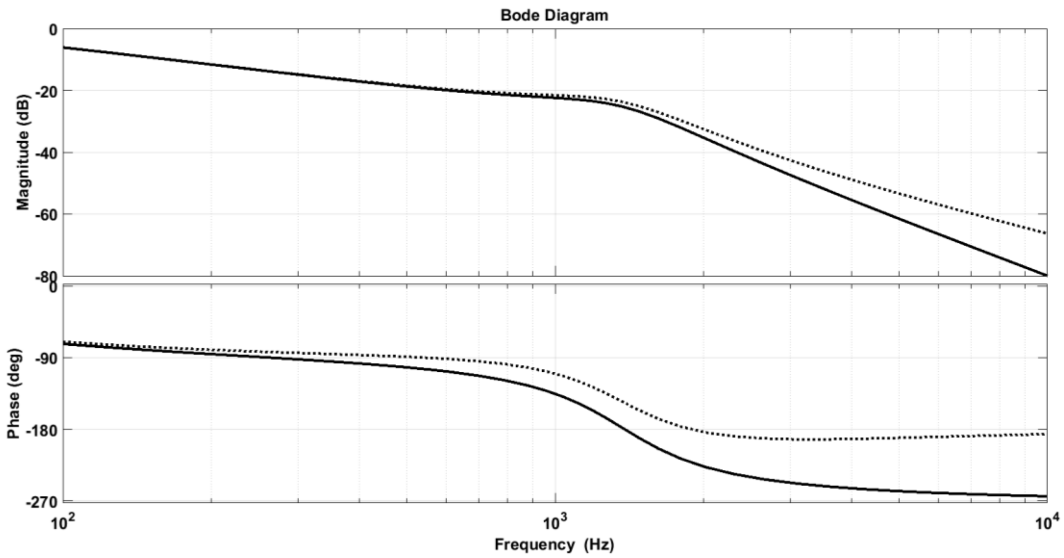


Fig. 3-23 Frequency response of the LCL filter with: passive damping (continuous line) and virtual damping (dotted line)

After that the resonance peak has been shaved, it is interesting to evaluate if the simple GSSA model is still valid, i.e. if it can be used also when the inverter has a LCL-filter. The previous simulation model has been changed, introducing the LCL filter (Fig. 3-24). The values of the new filter are in Table 3-3.

Table 3-3 Parameters of the inverter simulation with LCL filter

Component	Value
Cbus	3 mF
Li	2.3 mH
Rli	0.5 Ohm
Cf	25 uF
R_{Cf}	3 Ohm
Lg	0.7 mH
Rlg	0.2 Ohm

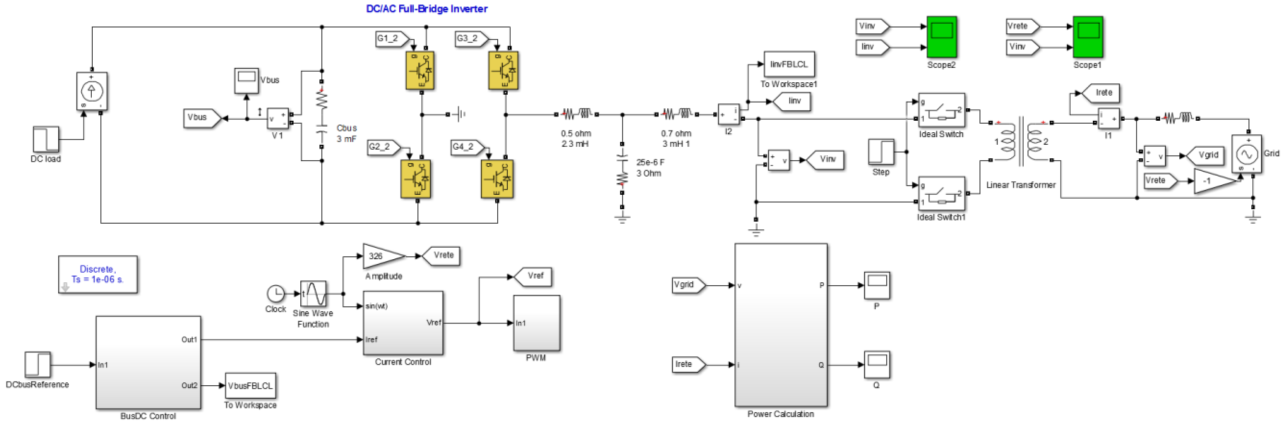


Fig. 3-24 Simulated physical inverter with LCL-filter and passive damping

The damped filter allows to consider the dynamical system like a first-order system. Fig. 3-25 and Fig. 3-26 show the results of the simulation. Now, instead of the L-filter for the simulated physical system is used a LCL filter with a physical resistor, which needs to damp the resonance. The invert dynamic behaviour is now similar to a first-order system.

The current of the analytical model in Fig. 3-26 foresees accurately the current of the simulated physical system. The current ripple is significantly reduced, thanks to the action of the LCL filter.

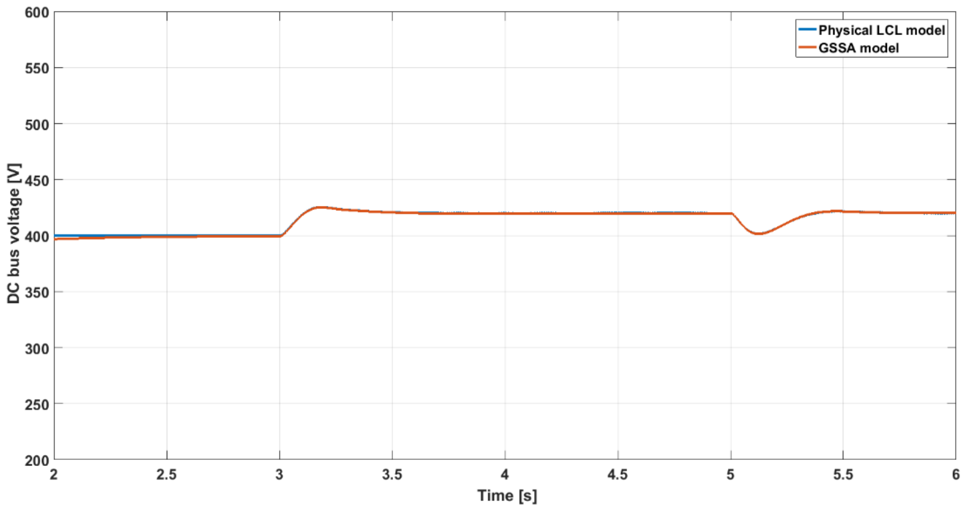


Fig. 3-25 Dc bus voltage: simulated model (blue) and GSSA model (orange)

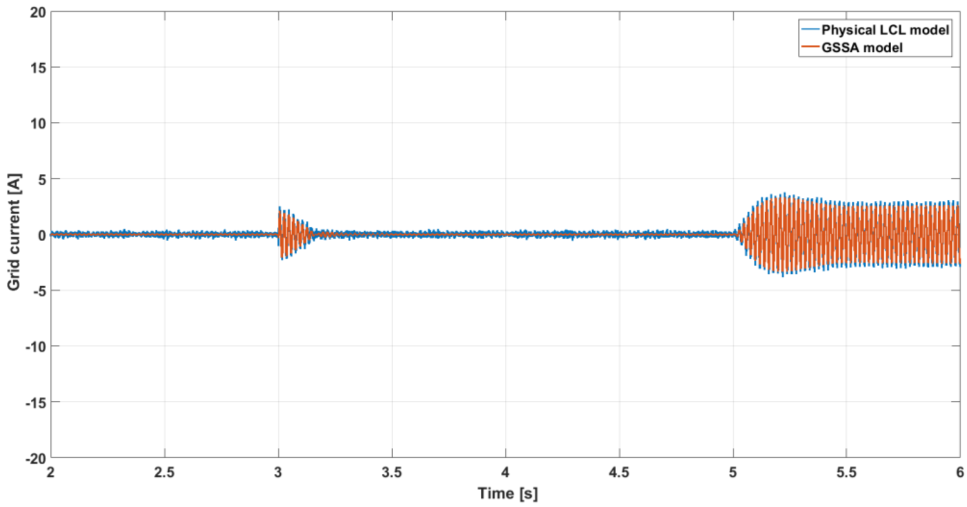


Fig. 3-26 Grid current: simulated model (blue) and GSSA model (orange)

3.3.4 Photovoltaic converter

The PV converter is a dcdc half-bridge converter, the same hardware structure used for the storage converter. However, the applied algorithm is different. The produced analytical model is still useful to describe the dynamic behaviour of the converter; in particular, it is useful that both input and output voltages appear explicitly in the model.

The PV converter has two operating modes (Fig. 3-27). The first is the MPPT mode: the algorithm changes the input voltage, i.e. the voltage on the plant, to extract the maximum power from the PV plant. The second mode is the master mode, in which the PV modulates the power from the PV plant, to supply the loads in the nanogrid during a critical condition.

The algorithm for the PV converter is selected by the discrete controller. The MPPT mode has a P&O MPPT algorithm to change the reference voltage on the voltage plant. A feedback controller changes the duty cycle of the converter to follow the reference voltage given by the MPPT algorithm.

In some critical situations, if there is a surplus in the NG the task of the discrete controller is to reduce the energy produced by the PV plant, so that the voltage control for the PV converter is selected. The controller $G_{CV}(s)$ is a controller acting directly on the duty-cycle of the half-bridge converter. It's important to highlight that this controller is critical to avoid a dangerous overvoltage in the NG, when a sudden event changes the NG configuration. For example, when the rated PV power is fed into the grid and the same grid goes down suddenly, then the voltage increases in a few ms over the maximum working voltage for the NG. At the best the nanogrid controller shuts down the NG to avoid damages. The shutdown is a highly undesirable event for the system. Thus, the $G_{CV}(s)$ controller has to be designed carefully according to the dc bus capacitor and the output filter stage of the converter.

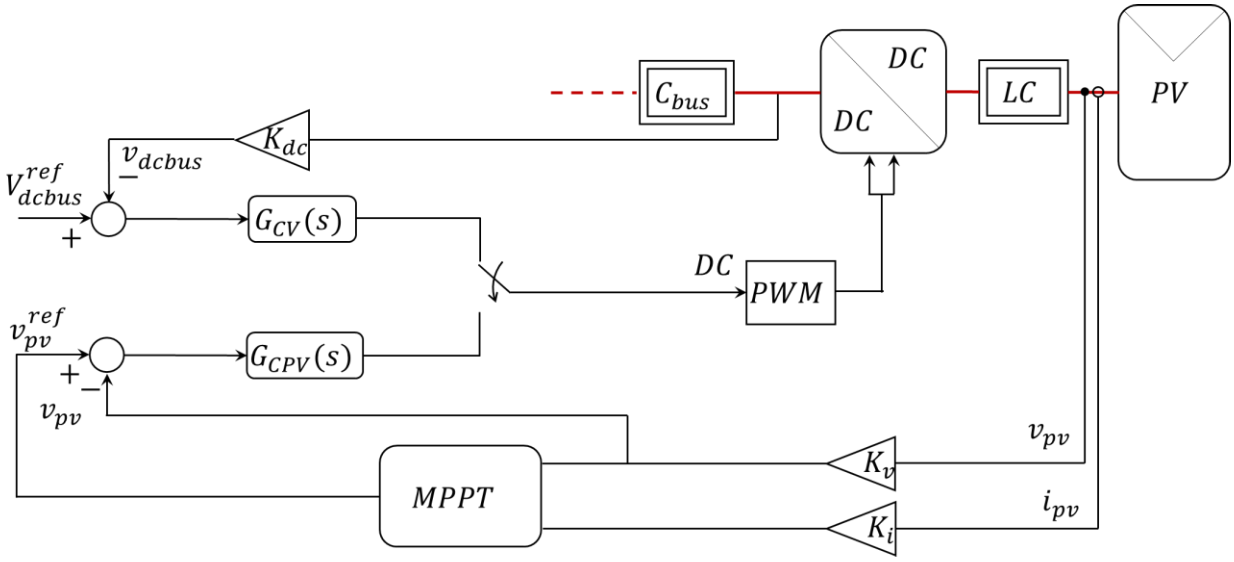


Fig. 3-27 Control loops for the PV converter

The action of the controller and the transition from the MPPT task to dc bus voltage control task is evaluated. The controller for the dcdc converter has been simulated in Simulink; the implemented model is in Fig. 3-28. The controller of Fig. 3-27 is implemented in the S-function block of the simulation model. This solution has been preferred because allows to use the same code, in the C programming language, for the simulation and for the processor of the converter in the real system.

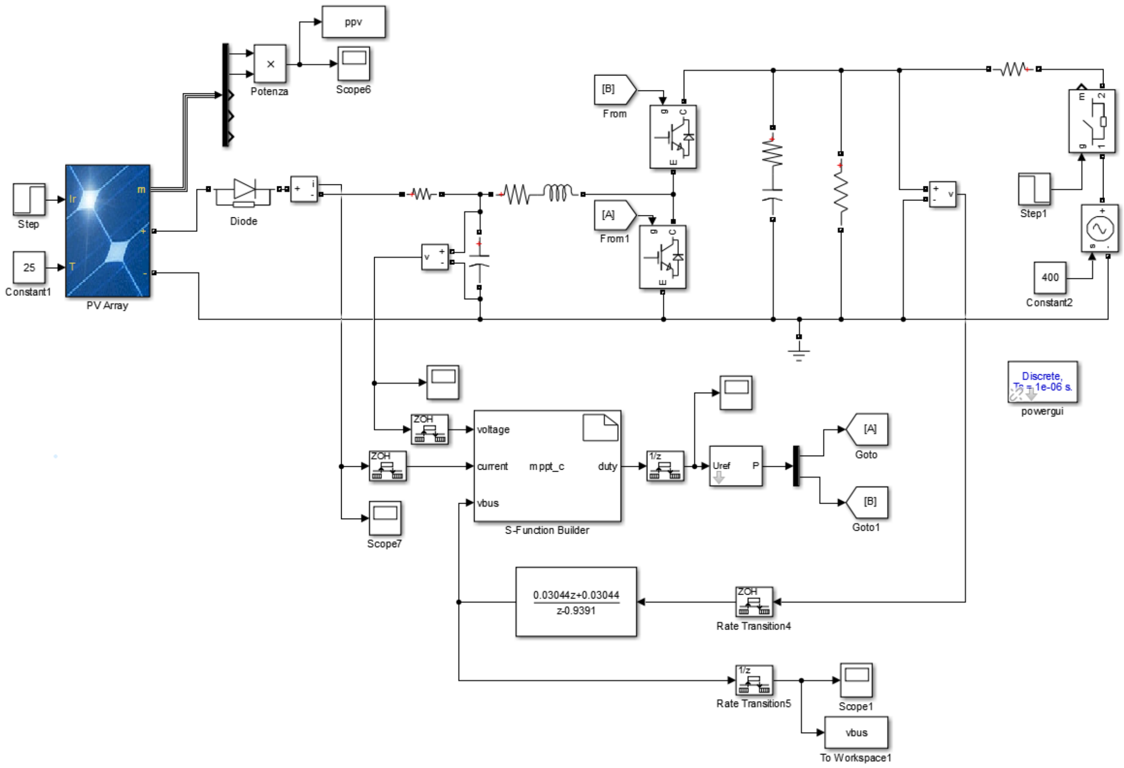


Fig. 3-28 Simulink model of the dcdc PV converter switching from MPPT to DC bus control

The PV plant specifications are plotted in Fig. 3-29; the PV characteristic is plotted for two values of irradiance: 800 and 1200 W/m^2.

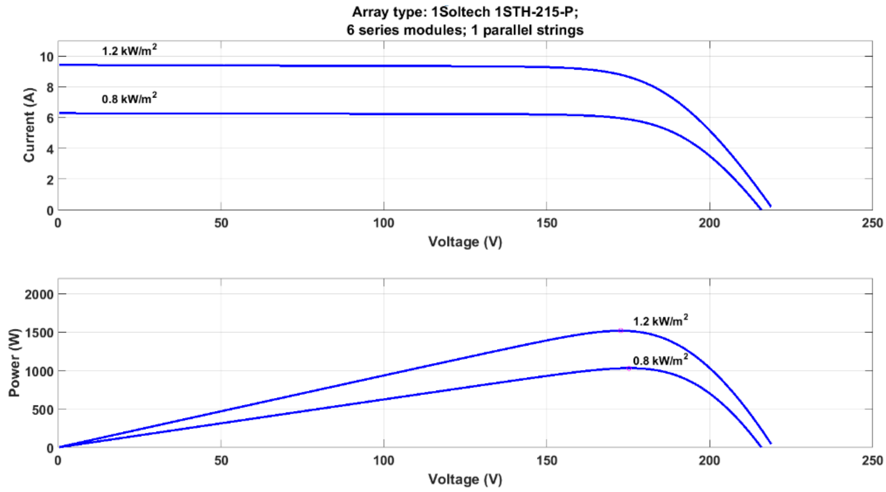


Fig. 3-29 PV plant characteristic used in the simulation model

The simulation starts with the dc bus voltage fixed with a constant voltage source to 400 V. The irradiance is 800 W/m^2. The PV converter extracts the maximum power from the PV plant, that is about 1000 W. When the irradiance jumps from 800 to 1200 W/m^2 at t=1 s the extracted power is 1500 W (Fig. 3-31).

Until the dc bus voltage is under 430 V the PV converter extracts the maximum power from the PV plant.

To evaluate the capability of the controller to fix the dc bus voltage in some critical situations the dc voltage source is disconnected at t=2s, as shown in Fig. 3-30. The master mode of the PV converter is selected only when a critical condition is evaluated, as when the voltage overcomes 430 V. When the logic detects the potential dangerous the MPPT algorithm is stopped and the dc bus controller is selected. This controller fixes the dc bus voltage to 430 V, reducing the generated power from the PV plant.

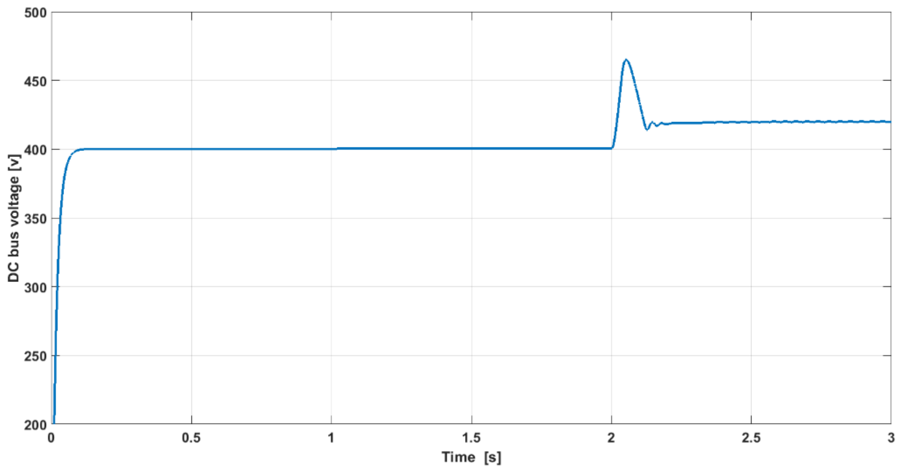


Fig. 3-30 DC bus voltage of the simulated PV system

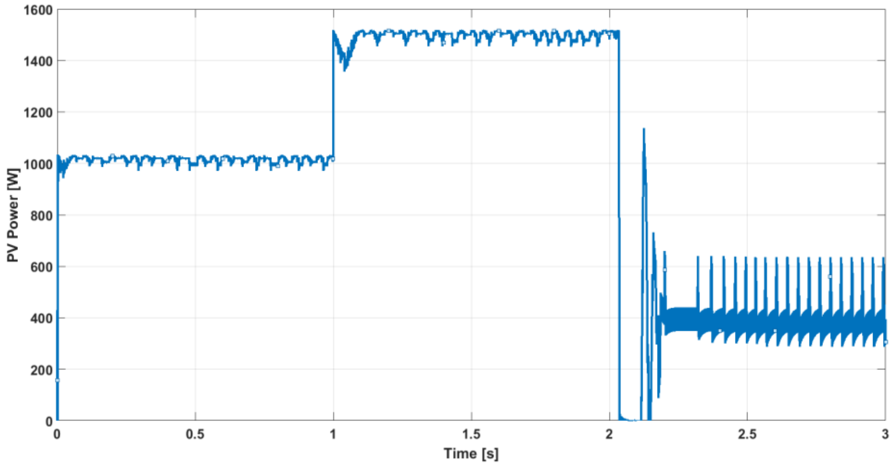


Fig. 3-31 Power extracted from the PV plant

3.3.5 Stirling converter

Among the sources considered in the nanogrid there is a free-piston stirling generator (Fig. 3-32). It is a cogeneration unit that produces hot-water and electrical energy from a hot source. Free-piston engines typically generate electric power with a linear alternator formed by the oscillatory motion of the power piston in a magnetic field. The displacer and power piston typically move as tuned spring-mass-damper systems in response to pressure differences. Free-piston machines are mechanically simple but dynamically and thermodynamically complex. For power generation applications, electricity is transferred from a linear alternator connected directly to the power piston; for cooling applications, the power piston is driven by an integral linear motor. This is done without dynamic seals that may leak or fail [83].

Fig. 3-32 The free-piston Stirling generator

The control of the stirling generator is a difficult task for three reasons. The first reason is due to the narrow constraints in terms of power quality needed by the stirling controller. The second reason is related to the grid forming capability needed by the stirling inverter. The last reason is related to the high inertia of the generator. Because of this inertia the source is practically uncontrollable. The word “uncontrollable” is used because the long-time constants of the stirling system are not compatible with the fast dynamic of the nanogrid, which make the stirling an uncontrollable source. When there is a positive unbalance, i.e. there is a generation surplus in the nanogrid, first the controllable sources are reduced; however, if this is not enough, the presence of the uncontrollable sources can bring to an overvoltage error. This error can produce the nanogrid shutdown or, in extreme cases, dangerous situations and irreversible damages. To avoid the overvoltage status, the power surplus must be dissipated.

For this reason, there is a breaking resistor connected to a dcdc converter in Fig. 3-33. The dcdc converter monitors the dc bus voltage in background and when the voltage overcomes the defined threshold it operates to restore the power balance in the nanogrid.

The most critical condition is when the stirling is the only source in a stand-alone configuration. Another critical condition is when the energy produced from the stirling is fed-into the utility and a sudden disconnection from the grid occurs. The dynamic response of the chopper and the action of the discrete controller guarantee the continuous operating of the nanogrid and of the generator, avoiding damages and sudden risky shutdown.

The braking resistor control is used in emergency condition. In emergency the power surplus is dissipated on the breaking resistor through the dcdc converter.

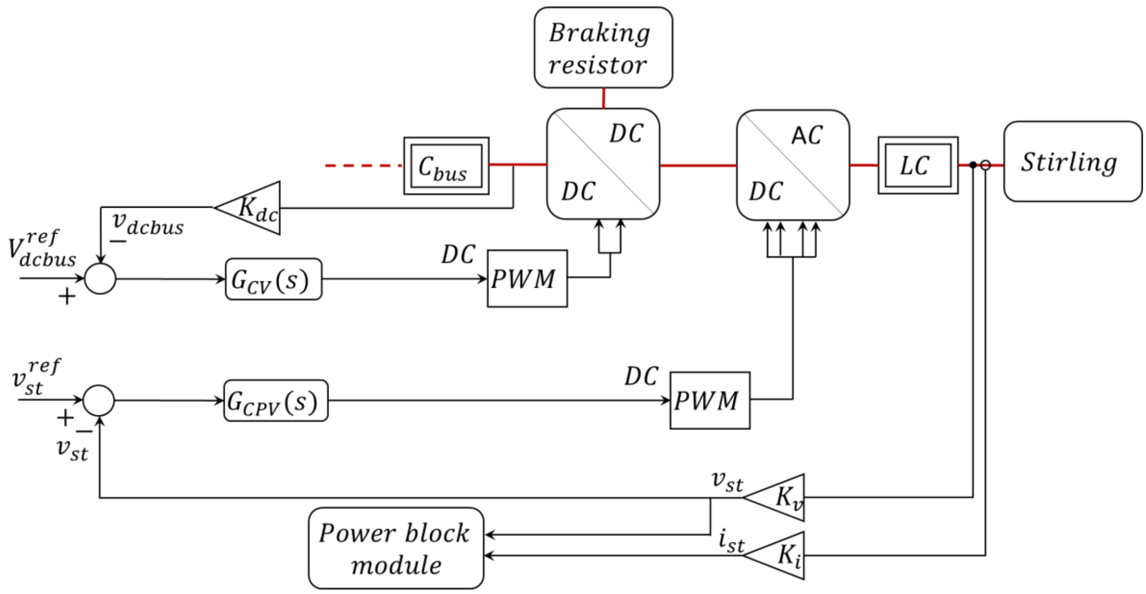


Fig. 3-33 Control loops for the stirling system

The controller for the braking resistor has been designed according to the internal model control, to achieve the performance requirements for the specific application. The requirements are narrow, mainly in the desired step-response, such as rise-time and overshoot. The analytical model and the physical system are simulated in Simulink (Fig. 3-34).

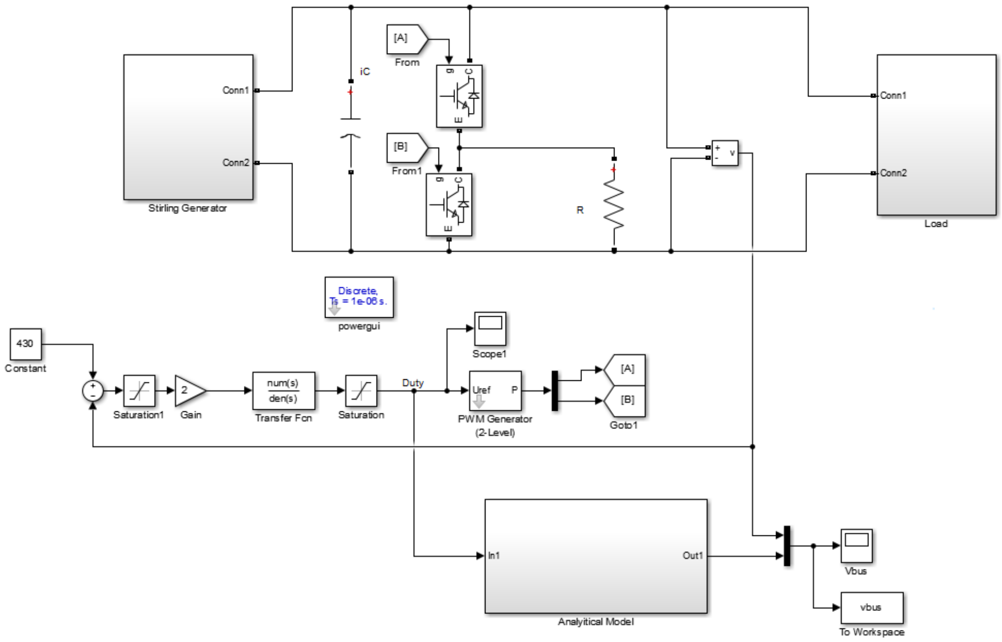


Fig. 3-34 Simulated Stirling system: physical system and analytical model

The stirling generator, acting as a constant current source, produces the power needed by the load on the dc bus. At t=1s the load is suddenly disconnected. The dc bus voltage starts to increase with a fast rate, until the critical voltage of 430 V is exceeded. At this point the dcdc converter starts to dissipate the power surplus to restore the balance in the nanogrid and to avoid the overvoltage error. The implemented controller has been designed with the main goal to reduce the rise time and to avoid overshoot. Indeed, in Fig. 3-35, although the huge unbalance and the intervention delay, the dc bus voltage exceeds the set point reference of 430 V only of a few volts.

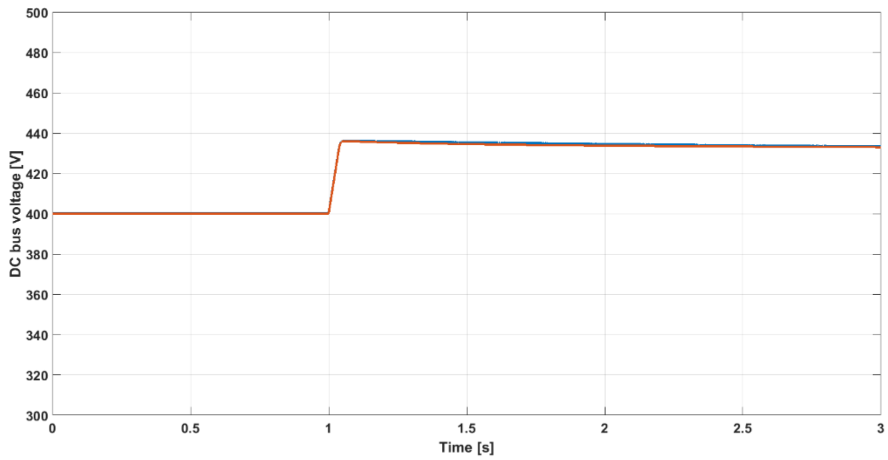


Fig. 3-35 Step response of the dcdc converter when a sudden unbalance causes a power surplus: physycal model (blue) and analytical model (orange)

3.4 Discrete Control Layer

The desired behaviour of the nGfHA is coded in the control architecture. The design and the implementation of the CA should not be a complex, time-consuming and error-prone activity. The control architecture should show modularity, flexibility, reliability, safety and robustness. Moreover, given the innovation and the research stage of the presented system, a lot of flexibility in the control architecture is required in order to make easier software corrections and changes during the experimental stages. The implemented code for the control architecture should show maintainability, modularity and reusability.

The control architecture is the algorithm used to control the discrete layer inside the hybrid control. This is the control layer where the decision-making algorithm leads to the ability of the controller to decide what to do.

An innovation introduced in this work is the application of the behaviour tree as discrete controller for the NG. This solution has been made possible to achieve all the requirements of the control problem. Previous attempts to build this smart/autonomous system with other control architectures were failed because the control became early too complex and hard to manage and implement. For example, though the FSM has an intuitive structure, the transitions number became soon unmanageable. In each state, the controller had to check for all possible events, reading all the n-inputs of the system. The design of a FSM is a trivial task for a system with a few inputs, but it becomes very hard to code in a more complex system given that in every state any possible condition may happen.

The NG has a clear predisposition for renewable energy sources, even if other sources can be included. Another critical point is the battery energy system management. Anyway, each source follows different rules and is managed accordingly. The requirements gathering define the desired behaviour of the system; they are given considering the following sources/loads: ac utility grid, renewable sources, not-dispatchable sources, storage systems, essential and critical loads.

The desired requirements for the control system are:

1. The NG starts the operation autonomously. The nanogrid controller selects the source to use for the starting procedure, then the goal is to reach a stationary operating condition and to start the other sources in the NG.
2. The exploitation of RESs should be the main goal of the control system. The renewable sources should work in maximum power mode for as long as possible. However, in particular conditions, the controller should try to modulate the power of these sources rather than shut down completely.
3. The storage energy management can follow different kinds of strategy, with different time horizons. In its basic operating mode, the storage should charge when local energy produced by RESs is available; the discharge should happen to avoid to absorb from the main grid or to smooth the sources and loads variability. In stand-alone mode the discharge should happen following two operating-mode, normal and emergency discharge; in emergency mode the operating mode should be communicated to the home automation system, to set a load-shedding technique and save energy.
4. The energy exchange with the grid should be done according to the following rules:
    a. If the produced energy is more than the local load, the surplus is feed-in to the grid, while if there is a deficit of the local production the energy is absorbed from the grid;
    b. The exchanged power with the grid is a value according to a received remote request. The request can be derived from the aggregator or the grid distributor operator.

   The operating mode priority must be selected.

5. Some cogeneration units work if supplied by the ac 50Hz grid; if a blackout occurs the generator is immediately stopped in emergency mode. This operation, which is potentially dangerous for some generators, entails the absence of domestic hot-water, other than the absence of electrical energy. In the NG, the cogeneration units should continue to produce electrical energy and hot-water, even if the utility grid is down. The difficult task is to maintain the power balancing inside the NG, especially in presence of a grid failure.
6. The control system should avoid power supply discontinuity of critical loads.

7. More NGs should be able to interconnect to build a microgrid. The interconnection allows to exchange energy between NGs and to share energy resources, with the final goal to maximize the financial benefits of the coalition and to increase power supply reliability.
8. A sudden event, such as a disconnection of one or more sources should not invalidate the operating of the NG, if there is energy enough to supply all the loads in the grid. Take, for example, a commercial pv inverter: even if there is the PV plant able to produce energy, the inverter can't supply power to local load if the utility grid is not available. Thus, all the energy production during the off-grid period is lost.
9. Each converter is autonomously controlled. This autonomy allows to prioritize the sources according to the status of each source and to take decisions about the source locally; the independence is a big advantage for the simultaneous control of different converters, however it could be a drawback if not accurately managed.
10. The control system should be able to manage different kinds of sources. Furthermore, the NG should be able to reconfigure the role of the converters, as necessary.
11. In case of a power deficit, a mechanism to notify the need to start a backup source, in an extreme event, should be provided.

There are several ways and control architectures useful to cover all the requirements, but the decision-making, in this work, is entrusted to a BT. The reasons are related to the inherent benefits of the BT over other control architectures, but also because its execution requires low computational power and it can be executed with a high frequency. The performance of a behaviour tree depends on the tasks within it. A tree that is O(1) in performance and memory will be O(n) in memory and O(log n) in speed, where n is the number of nodes in the tree [71]. Moreover, in this application of the BT, the algorithm is executed with hard real-time constraints, unlike other applications.

The designed BTs are now presented. Two revisions are discussed. The proposed controls are designed assuming that the resources available in the NG are: an ac utility grid, a PV plant, an energy storage system and a stirling generator. Moreover, uninterruptible ac loads are supplied from the dc bus, using an inverter.

The first revision, despite the good result provided, had a big drawback: the difficulty in managing interconnected NGs. The goal of the design is to have only one BT that runs on each NG and not different BTs for each NG. The nanogrid controller should be able to work autonomously, but also to work in cooperation with other NGs if are connected to the same dc bus.

Anyway, rather than work on this BT structure, a new BT structure with a new point of view has been developed in the second revision; the BT tries to model all the system, rather than focus on the single converter. However, some of the nodes developed for the first revision of the BT have been reused in the new design; the new BT, taking advantage from the developed experience and from the already implemented nodes, has been developed and implemented (control and software design plus code generation) in a few days. Then, more features have been added, thanks to the modularity of the tree, time after time.

3.4.1 Revision 1.0

This paragraph describes the first attempt in designing and implementing a discrete control inside the second layer of the NG, using the BT. Even if this specific design was abandoned, its description is useful to show some properties of the developed structure.

Each converter executes its local control; during a stationary condition the converters establish the power sharing among all units. To this end, a local power-sharing strategy, the master-slave strategy, has been implemented: among all converters, one is chosen to operate as master converter. The master converter controls the dc bus voltage, while the other converters control the supplied or absorbed power from the dc bus, following the orders received by the discrete control.

The Root node generates the *tick*, which spreads from the root to children node; *tick* is nothing but a formal abstraction used in the BT description to say that a node is called and, thus, executed. The function that implements the root calls the "*blackboard*" function; this function acquires all the information from the processor inputs (like shut-down, faults, grid presence, ecc…), unless the controlled variables that are locally acquired by the control algorithm.

The tick spreads from root to leaves and from left to right. The BT, with a low granularity to improve the readability, is in Fig. 3-36. Two branches leave the root. Each branch corresponds to an operating mode: grid-connected and islanded. For each operating mode there is a master converter: the PEI for the grid-connected mode and the Power Electronic converter of the Battery (PEB) for the islanded mode. Thus, in this model of BT, there is a master converter for every main branch of the tree. To add another master converter is sufficient to add another branch in the BT; adding this new feature doesn't require changes to the previous work.

A. Grid Connected

The left branch controls the operation in grid-connected mode. The first sequence node calls successively, from left to right, the child nodes. The first child node to be invoked is *GridMasterMode*; this node is another sequence node, which contains a sub-BT, that provides all the actions necessary to start the NG from the utility grid and to set the PEI as master converter. The *CheckGrid* node, condition node, checks the presence of the utility grid and if the rms voltage and the frequency are in the operating range. After the start it executes an anti-islanding detection algorithm. If there aren't anomaly conditions, the node returns Success, otherwise Failed. The returned result is used to advance in the tree. If the result is Success its parent continues to call in sequence the child nodes; otherwise the grid can't be used and the BT switch to the islanded mode.

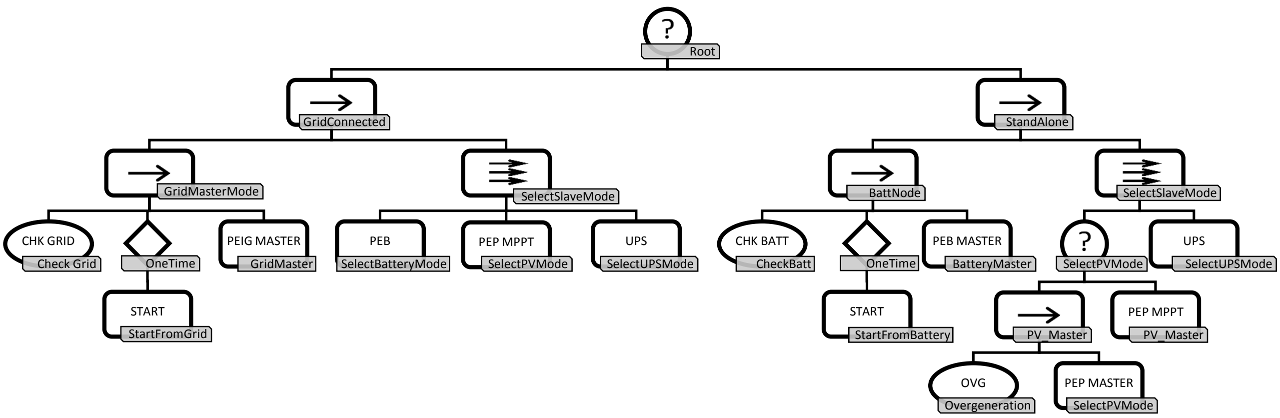


Fig. 3-36 Revision 1.0 of the BT implemented in the nGfHA

If *CheckGrid* is successful, the *GridMasterMode* calls the *StartFromGrid* node. This node activates all the procedures to start the NG through the utility grid. The *StartFromGrid* node starts a pre-charching procedure of the dc bus to avoid overcurrents in the grid: the precharging procedure closes the switch between the NG and the main grid and keeps the inverter disabled with the power switches open. Through free-circulation diodes the current starts to charge the capacitor on dc bus. To avoid overcurrents and rapid aging a series resistor is used to limit the current during this period. After 5 times the time constant of the circuit the precharging phase can be considered terminated. However, the voltage on the dc bus is not over sqrt(2) times the rms value, that is not enough to operate the PEI converter. Thus, to increase the voltage, the inverter is enabled and the dc bus voltage increased. Reached a stationary operating point the node *StartFromGrid* returns Success.

During all the precharging period the BT has not been stopped waiting for the *StartFromGrid* success; indeed, the BT is executed with a period equals to $1/f_{tick}$, that is smaller than the period to precharge the dc bus. Thus, each $1/f_{tick}$, the *StartFromGrid* node return Running, which indicates a transient operating condition.

The positive outcome is registered by the *OneTime* decorator node. This node, will return Success for all the other calls, without calling the *StartFromGrid* node anymore: indeed, the *StartFromGrid* procedure has to be executed only one time.

If the *StartFromGrid* node has not failures, the BT ticks the *GridMaster* node. This node sets the PEI as master converter and gives it the reference value for the dc bus voltage. All the other converters are still waiting in stand-by mode.

The operating mode of each converter is established under the parallel node. The parallel node allows to execute parallel tasks. This powerful feature allows to decouple the behaviour of each source/converter. The subBTs foresees all the possible sources that the device can manage; according to the availability of the specific implementation the sources may be independently selected. Moreover, the modularity of the BT is reflected on the NG: if a particular source is not foreseen in the initial project, it can be easily integrated adding another parallel subBT, without affecting other sources: obviously, its goal has to be coherent with other converters' goal. The execution order is not important for parallel nodes.

The parallel subBTs for storage, PV and loads are now discussed.

The storage subBT, represented by a node in Fig. 3-36, is exploded in Fig. 3-37. The subBT monitors the battery status, checks for anomalies, as short-circuit currents or dangerous voltages in the *CheckBatt* node; if one anomaly occurs the converter is disabled and the switch that connect the storage is opened for security reasons. If the storage doesn't present problems the *CheckBatt* node returns Success and the task for the storage converter has to be established. The task is selected through a selector node. When the grid is in master mode, the storage can choose to charge, discharge or stand by.

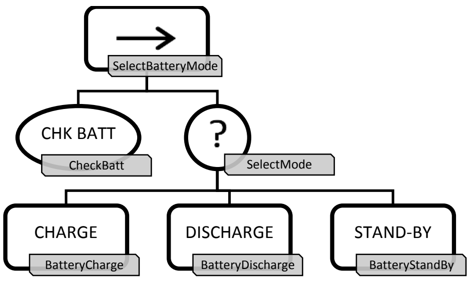


Fig. 3-37 sub-BT for the battery control

The charge operating mode is controlled by another subBT, shown in Fig. 3-38. The charge is started if some conditions are verified. The constraint is to avoid the use of energy from the utility grid to charge the storage, unless there is an explicit request from DSO or the aggregator.

The energy used to charge the storage system is the energy produced locally by DERs, especially coming from renewable sources. The charging phase is started when there is an over-generation inside the NG; this condition is evaluated inside the *Overgeneration* node. After a positive outcome from the *Overgeneration* node another condition has to be fulfilled, i.e. the empty status of the storage. If the SOC is under a defined threshold, which depends of the particular technology of storage system, the Success state is returned and the charge can be started with the execution of the *BatteryCharge* node. The energy stored is equal to the available surplus if this value is lower than the maximum charging current established by the physical limit of the storage, it is the maximum charging current otherwise. The dynamic current limits are selected by the BT, while the low-level control saturates the current to the current limits of the power devices. The BT saturates dynamically the charging current to a value established by the BMS or by the same BT in relation to the actual SOC.

If there isn't an over-generation, but a remote request to absorb energy is received, then the *RemoteRequest* node draws up and returns Success, if the request is correct: the system try to charge the power given by the external command, if the *BatteryEmpty* node agrees. Worth noting that:

- the priority is given to the local production;
- the received command is related to the power exchanged with the grid, because it is the master converter.

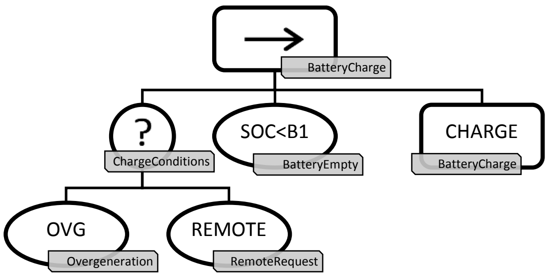


Fig. 3-38 Charge battery

A similar structure is adopted by the discharge subBT (Fig. 3-39). The conditions that can start the discharging operating mode are under-generation or remote request. If one of these nodes return Success, then *BatteryNotEmpty* verifies that the SOC level is greater than a defined threshold, that is the condition for which the storage has energy available. If also the *BatteryNotEmpty* returns Success, then the discharging process can start. Otherwise the subBT returns Failed and the storage can select another operating mode. The discharging power is the difference between the local production and local demand if *Undergeneration* has returned Success, is the value received otherwise.

The *BatteryCharge* node has priority over the *BatteryDischarge* node. For instance, if a discharging remote request is received and there is a local overgeneration, the structure of the tree chooses to recharge. However, if the storage doesn't need to charge, i.e. it is capable to give energy, the converter follows the remote request.

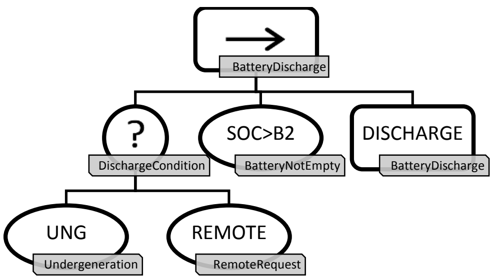


Fig. 3-39 Discharge battery

If both the *BatteryCharge* and *BatteryDischarge* node returns a negative outcome then the storage is kept in stand-by, with no energy exchange with the dc bus.

The subBT of the PV converter is simpler. If there is voltage at the terminals the BT commands the converter to execute the MPPT algorithm: this one is then executed in the lower layer by the low-level control. The voltage is constantly monitored in order to check the plant or the solar radiation presence. The BT disconnects the PV plant with the aim to avoid dangerous in case of high dc bus voltages. After a period, if an irreversible error is not detected, the BT tries to reconnect the PV and check if the anomaly is vanished.

The last converter is the ups inverter. The BT commands the dc ac converter to control the rms voltage and the frequency on the ac side: voltage and current are continually monitored. If an anomaly occurs the node returns Failed and the supply of the ac loads is stopped. However, the main task of this node is to keep supplied the loads as long as possible, because this converter supplies critical loads.

B. Stand-Alone

If *CheckGrid* in Fig. 3-36 returns Failed, then the islanded branch of the BT is executed. The battery converter becomes the master converter in the NG. The *checkBatt* node checks the storage capability to supply or absorb energy. If this node doesn't return Failed the storage can provide the black-start of the NG if the start has not already done. If the NG is already started can be checked in two ways: controlling the dc bus voltage or using the same decorator node of the first branch mode.

If both nodes return Success, then the storage converter is selected as master. The *BatteryMaster* also defines the current limits in both charging and discharging: the limits can be communicated by a BMS, if present, or can be evaluated by the same BT. This option can be selected in the initial configuration. Thus, if the storage is full charged, the charging current is set to zero to avoid an overcharging. In master mode the converter supply or absorb energy to control the dc bus voltage, according to the current limit of the storage system.

The parallel node defines the operating conditions for the other converters. The parallel node calls the PV and UPS subBTs. In this condition, the grid inverter is offline because the grid is down. However, it is easy to add a child node to the parallel node to set the inverter as voltage controlled inverter, to supply the essential loads: these loads, then, can be shaded if there is a deficit in the NG.

In islanded mode, in some operating conditions, unfortunately, the PV power has to be modulated to avoid the shutdown for an overvoltage error. The switching between the MPPT task to voltage control task is decided by the subBT in Fig. 3-40. The *Overgeneration* node checks if there is an overgeneration: if it returns Success then the PV converter starts to control the dc bus voltage modulating the produced power. The voltage reference value is higher than the reference value of the storage converter. Otherwise, if there is a balance in the grid and *Overgeneration* returns Failed the dcdc converter performs the classical MPPT task.

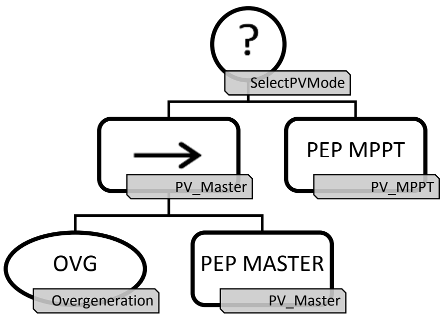


Fig. 3-40 PV power modulation

The last node is the UPS node. This one is the same node for the ups in the grid-connected branch. Thus, the switching between grid-connected to islanded happens with continuity for the dcac inverter, which the only task is to assure a high power quality to critical ac loads.

The extended BT of the revision 1.0 is in Fig. 3-41.

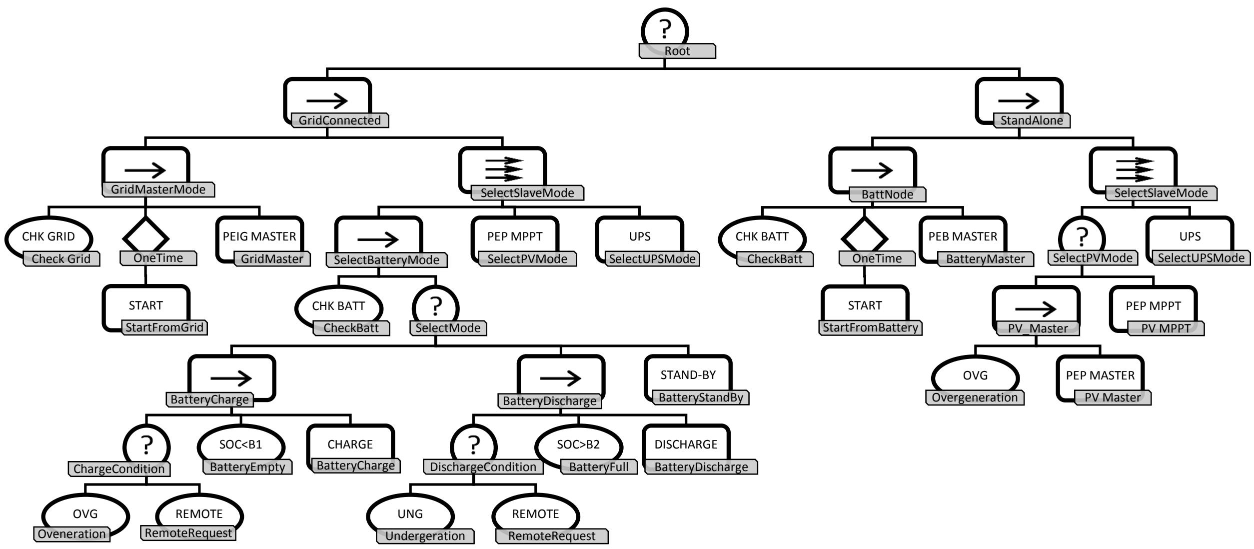


Fig. 3-41 Complete behaviour tree: revision 1.0

3.4.2 Revision 2.0

The first BT has highlighted the potentiality of the tool with its versatility, abstraction, modularity and it has proved the ease of translation the logic from natural to BT language. In the new revision, the NG is able to start and operate autonomously and cooperate with other NGs connected to the same dc bus.

Each NG has to be configured, selecting the sources connected to power terminals of the NG. The same BT is implemented in each NG: the difference is only in the configuration file.

Thus, the BT doesn't implement anymore a pure master-slave logic – almost static. Instead, the BT realizes a distributed control structure where each BT knows what to do using only the information transmitted with the voltage on the dc bus, implementing thus the DBS control. The BT from the dc bus voltage information takes decisions and chooses the operating point of each converter. In this logic at least a converter is voltage controlled. This architecture allows to each NG to execute the same BT and to more NGs to connect on the same dc bus.

The enclosed version of the implemented BT is in Fig. 3-42. The BT has two main branches: the left one manages the start of the NG, while the right one implements the DBS control.

The root node invokes the selector node. This node increases the attraction region towards the started status region of the NG, where the NG operates in a stationary condition. This node checks, in a priority order from left to right, if there is at least one source able to start the NG. As soon as one source gives a positive outcome the BT starts the starting procedure. The *oneTimeSuccess* node is transparent during the execution of the branch, until one of its child returns Success. This event is registered and the decorator doesn't execute the bottom nodes anymore, because the starting procedure has to be executed only one time.

The BT first checks if there is another NG ready for the interconnection already started on the same DC bus with the condition node *CheckSource*. If there is voltage on the bus, then the *StartSource* node closes the switch to physically connect the NG to the energized bus dc, starting the charging procedure. The NG is a load for the common bus dc in this situation. This starting mode allows to connect more NGs in parallel to build a wider grid, without the need of any kind of external communication.

If the first choice has a negative outcome, the BT checks for the grid presence. If the grid voltage is within the voltage and frequency expected range, then *CheckGrid* returns Success and the precharging procedure starts.

The failed outcome of *CheckGrid* entails the call to *CheckBatt*. If the storage is present and has the capability to charge the bus dc the node returns Success; the following invoked node takes care of the starting procedure, which first connect the source to the dc bus with the converter in a passive status and then enables the converter to step-up the voltage on the dc bus to a desired value.

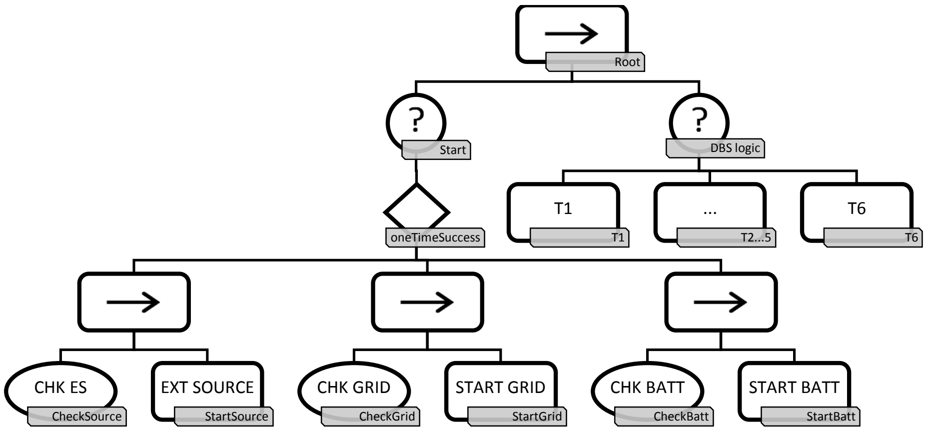


Fig. 3-42 BT structure of the second revision, implementing a DBS strategy

If a new source wants to be introduced with the ability to black-start the NG, like a pv plant, the new subBT can be easily connected in series with the other starting blocks without changes in the BT structure. The first condition node checks the source availability to charge the bus dc; then, with a positive outcome, the selector node ticks the *StartFromSourceX*, which provides to actuate all the actions for charging the dc bus; it closes the switch, returns Running during the precharging time, enables the converter to regulate the dc bus voltage to a desired value.

The multi-sources start is an important feature for a NG: this procedure allows to start more NGs connected on the same dc bus. But mainly the black-start allows to start the NG in remote areas or in emergency conditions after a natural disaster, where the utility grid can’t be used as power source. While the BT is waiting for the charging phase, it remains alive and ready to react if some conditions change. When the system is started, the decorator node registers the Success returned from one starting procedure. After that, the decorator does not allow anymore to execute its child nodes; the decorator returns directly Success, without invoking its child nodes.

The second branch of Fig. 3-42 implements the DBS control algorithm. The algorithm uses the dc bus voltage to select the operating mode: the implemented DBS has 6 thresholds, named from T1 to T6 (Fig. 3-43). The thresholds number is a design parameter. This numbers defines the number of the operating mode of the NGs. Each threshold corresponds to an operating point of the nanogrid, i.e. each threshold has its own state space in which the system can evolve. The BT is the discrete controller that drives the system from a state space to another.

The structure of the sub-BT used to define the behaviour in each threshold is in Fig. 3-43. To enable DBS logic, the first task of the controller is to acquire the information passed through the dc bus and associate the respective threshold. Thus, the controller assigns the operating mode and the reference values to each converter in the NG. The complexity of the behaviour of each converter is enclosed in their subBTs. Indeed, the subBt of each converter is ticked under a parallel node. The design, the updating and the corrections for a converter are made independently from the other converters. The presence, the absence, the addition or the elimination of a source and its converter doesn't correspond to a change in the BT structure; the structure is highly modular and flexible.

A decorator node, which is a parent of the parallel one, assures that a Failed doesn't spread in the BT: this choice allows to contain each failure in the subBT and to solve it locally. For example, a sudden grid disconnection is a failure for the grid-connected inverter, but can't be a failure for the other converters, that have to continue their operation to maintain the power balance in the grid. However, in the implemented BT, there are some dangerous situations in which a leaf node can return Error: this status indicates a dangerous situation, an irreversible error, and so the grid is driven to the shutdown. Basically, this decorator node masks the Failed temporary condition of the various converters.

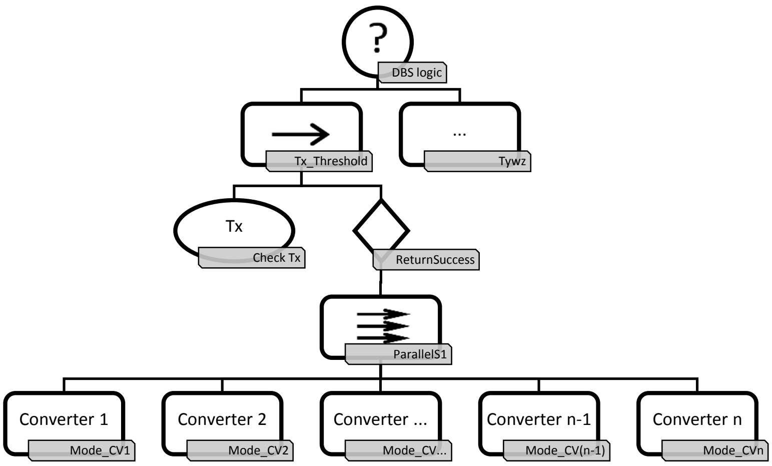


Fig. 3-43 Common structure of the subBT for each threshold

3.4.2.1 Threshold T1

T1 is the operating threshold that has the lowest voltage level. This threshold indicates a low production (or a high load) and the storage system with a low SOC, near to the minimum value acceptable. With insufficient power production, in this threshold, some non-critical loads can be disconnected. The storage converter operates as master converter, regulating the dc voltage to the T1 reference value, suppling the energy for the critical loads. If the storage reaches the lower limit of the SOC, then the system shuts down.

These rules are implemented in the BT in Fig. 3-44. The node *CheckT1* checks if the NG is in the T1 threshold: if it returns Success then the NG is working in T1, otherwise the threshold is another one. All the node threshold checkers have a similar structure for the corresponding subBT: if the node Tx returns Success then the NG is in the Tx threshold, while if it returns Failed then the NG is in a different threshold Ty.

The parallel node ticks five subBTs, which are executed in parallel. Each subBT controls the behaviour of one power converter. The implementation considers five different converters; therefore, five branches leave the parallel node. Of course, not all the converters must be present and each possible configuration between the five converters may be selected. The number of sources/converters is limited only by the I/O number of the processor and by its capability to run the low level controllers. If one source has not been considered, the only change in this BT is to add another branch under the parallel node, i.e. another subBT, able to control the activity of the new source.

The first subBT concerns the operation of the grid-connected inverter. The node *CheckGrid* returns Success if the parameters of the utility grid are inside the operating range, Failed otherwise.

With a positive outcome from *CheckGrid* the node *GridToT3* is ticked: the duty of this node is to take the dc bus control and to establish the T3 threshold. The activation of this node can occur in two situations:

- When the grid is restored after a blackout period and the storage is empty.
- When the grid-connected inverter supplies power from the storage to the utility grid; the remote request can discharge the storage, but when the storage is almost empty and the T1 threshold is activated, the power control of the grid inverter is stopped for a period, until a local source will start to produce or the storage will be recharged.

In both cases, the permanence in this threshold is temporary in presence of the utility grid.

With a negative outcome from *CheckGrid*, *Grid_Off* is activated. Indeed, if the utility is down the inverter is disabled and disconnected from the grid. In this threshold the supply of the essential loads is not provided because this threshold indicates a power deficit inside the NG.

The second subBT under the parallel node is the branch that controls the converter of the storage. The converter works in voltage control mode, therefore the storage supplies the power deficit of the NG. The first *CheckBattery* node gives the status of the storage: this determines the spread of the tick in the subBT. If *CheckBattery* returns Success, then the *SelectMode* node is invoked, otherwise the converter is turned off in *BatteryOff*.

BattStatus evaluates the energy capacity of the storage. If the SOC is over the B1 value then the voltage control mode for the T1 threshold is established in *BatteryMaster*: the discharging current is limited, according to the BMS information, to avoid damages to the storage system. However, if the SOC is under the B1 value the NG shuts down in *Shutdown* because no other source has the availability to supply the local loads. An alternative to this solution is the start of a backup generator when this situation occurs or when the T1 threshold is triggered.

The *PV_MPPT* node ensures that the PV converter extract the maximum possible power from the PV plant. Some basic checks on the plant are done inside this node.

StirlingControl is responsible for the operation of a stirling generator. The node activates the low-level control and enables the production of electrical energy. The generator is governed by heat sources, so it is characterized by high-inertia in modulating the produced power and it is controlled by the boiler.

The *UPS_Control* node commands the UPS inverter to generate a sinusoidal voltage with fixed amplitude and frequency and to supply the ac critical loads. According with the requirements of the loads, the voltage amplitude can be modulated or the supply could be stopped to avoid the shutdown due to low energy production, notified by the permanence in this threshold.

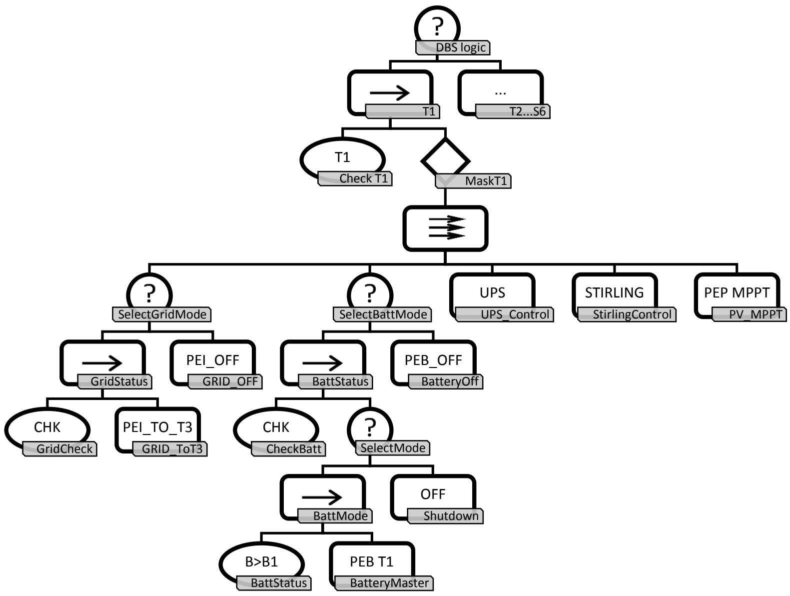


Fig. 3-44 subBT of the first threshold T1

Worth noting that the *UPS_Control* and other nodes encapsulate a more complex behaviour. In Fig. 3-44 only one node is shown to avoid additional complexities in the graphical representation. This feature allows to show the logic of a complex system with simple and clear nodes. Moreover, this is a big advantage for the programmer. The code that executes the BT could reuse a whole branch as a single node. The programmer can use a subBT without knowing implementation details, but using it to build new BTs. This ability to enclose allows to generate code in short time, to reduce coding errors and to produce code to implement sophisticated behaviours, starting from very simple nodes (or modules).

3.4.2.2 Threshold T2

In T2 the storage converter is the master converter in the NG. Unlike T1, the storage is not empty and has enough energy to operate as master converter for a reasonable period. The dc bus voltage reference for T2 is higher than the reference value for T1.

The utility grid could be present. If it is available, the grid converter operates in power control mode. If in T2 the reference power for the PEI inverter is zero, then the storage and the other local sources supply the local loads. When the reference power is a value $Pref_{grid}$, then the grid-connected inverter feeds in the utility grid a power equals to $Pref_{grid}$. Thus, in this condition the storage supplies a power equals to:

$$P_{storage} = P_{ref_{grid}} + P_{loads} - P_{pv} - P_{stirling}$$

If the remote request is not feasible, then *RemoteRequest* returns Failed and the grid inverter takes the control of the dc bus. This can cause some interactions between the two converters that are controlling the bus simultaneously. This conflict is avoided thanks to dynamic current saturation limits. For example, T2 is the threshold in which the storage discharges, thus the charging current is set to zero (unless 5 to 10 % of hysteresis). This trick allows to different converters to take the control of the dc bus and to cause a threshold change. The current limits are set dynamically by the BT.

The storage discharges in this threshold. The SOC value is higher than a B2 value: this condition is necessary to operate the converter in T2. If the SOC decreases under B2 the node *BatteryStandBy* is ticked. This node disables the storage converter, which stops to operate as master converter. If the storage converter releases the dc bus control in T2, then the dc voltage will decrease. The lower T1 threshold will be activated and the same converter will resume the dc bus control, but with some differences. Indeed, this choice is due to the need to communicate to all systems "listening" on the bus that the storage is almost discharged: as a consequence, some loads can be disconnected or a backup source can be activated to avoid the NG shutdown.

In T2 the renewable sources operate in maximum power, while backup sources are not involved in the production because there is enough energy to supply all the requests. This behaviour is coded in the subBT in Fig. 3-45.

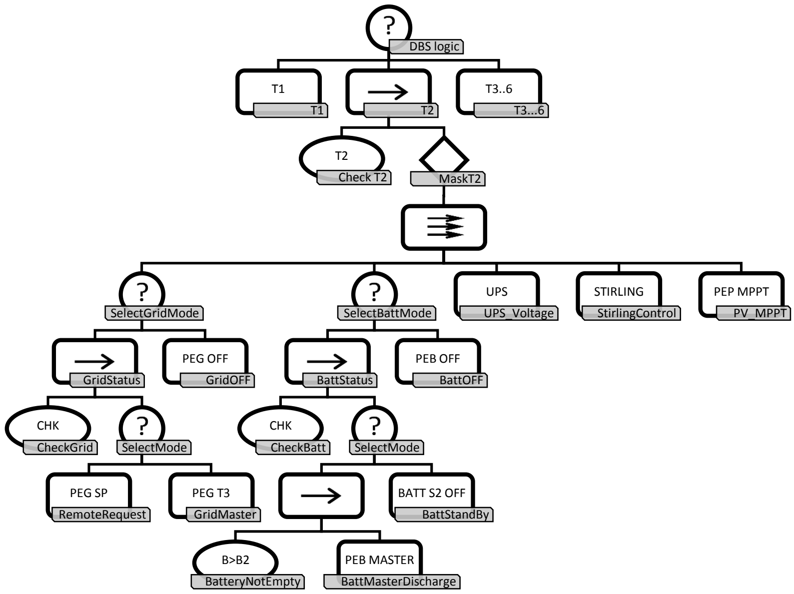


Fig. 3-45 subBT of the second threshold T2

3.4.2.3 Threshold T3

In the T3 threshold the power difference between the local production and the local demand is provided by the utility grid, while the storage is in stand-by. An external request can change the control mode of the inverter from voltage control to power control. If the request to the PEI inverter is activated, then the threshold may change: starting from a stationary operating point if the request is to feed-in energy in the grid, then the voltage will decrease and the lower threshold will be activated. Instead, if the request is to absorb energy from the grid the voltage will increase and the higher threshold will be activated.

The selector node *SelectMode* chooses the operating mode of the grid inverter (Fig. 3-46). The *RemoteRequest* node sets the power control mode for the inverter, after that the received value has been validated. If this mode is selected, the dc voltage is free to evolve, according to the NG condition. If there isn't a valid request the node *GridMaster* sets the grid inverter in voltage control mode: in T3 the inverter can only absorb from the grid, therefore the current limit is saturated to avoid that the energy is fed into the grid. If there is an overgeneration then the dc voltage will increase and other thresholds will be activated.

The storage converter stays in stand-by in this threshold. The other sources continue to operate as the previous thresholds.

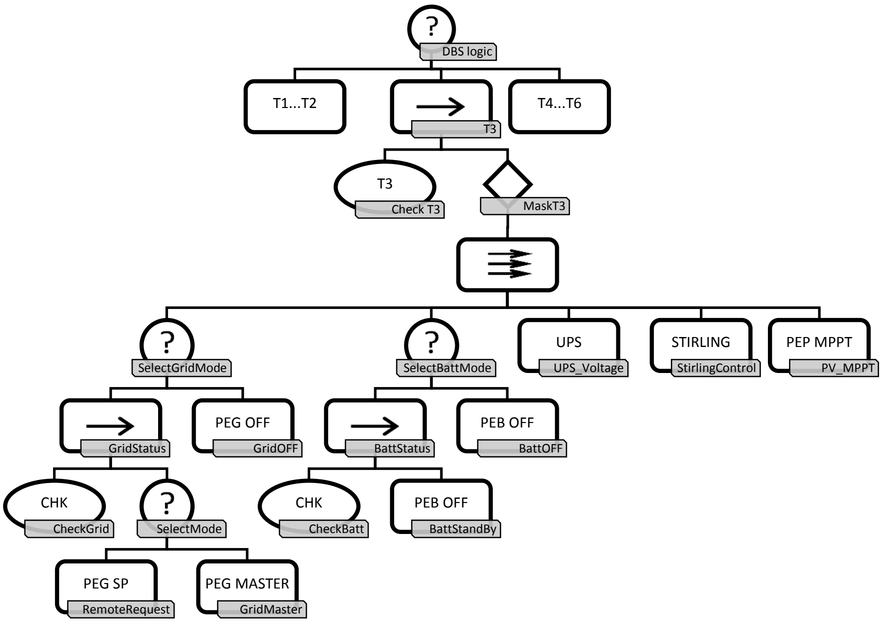


Fig. 3-46 subBT of the third threshold T3

3.4.2.4 Threshold T4

In the T4 threshold the storage stores energy to keep the dc bus balanced, since there is a surplus inside the NG. The activation of this threshold is related to the physical capability of the storage system to absorb energy (Fig. 3-47).

The subBT of the grid inverter is similar to subBT of T2. In presence of the utility grid the inverter is enabled only in power mode control; otherwise, if the *CheckGrid* node or the *RemoteRequest* returns Failed the inverter is disabled.

The storage can become master if the tick reaches *BatteryCharge*, that is possible only if the SOC is lower than the value B3 (*BatteryNotFull*). In this threshold, the absorbed energy is equal to the energy required to balance production and demand, i.e. the surplus on the dc bus. The upper limit in the absorbed power is limited only by the storage technology, the SOC value and the BMS information. If the SOC exceeds the value B4, that is the full charged condition, the converter is disabled and the energy flow is stopped by the *BatteryStandBy* node. If the two first conditions are not verified the converter stays in the last status modified: thus, if it starts the charge, then for all values between B3 and B4 the storage will continue to charge. If the SOC is already between B3 and B4, then the converter stays disabled. In these two conditions, the branch of the storage converter returns Running, to state the waiting status for a successful condition of the sub BT.

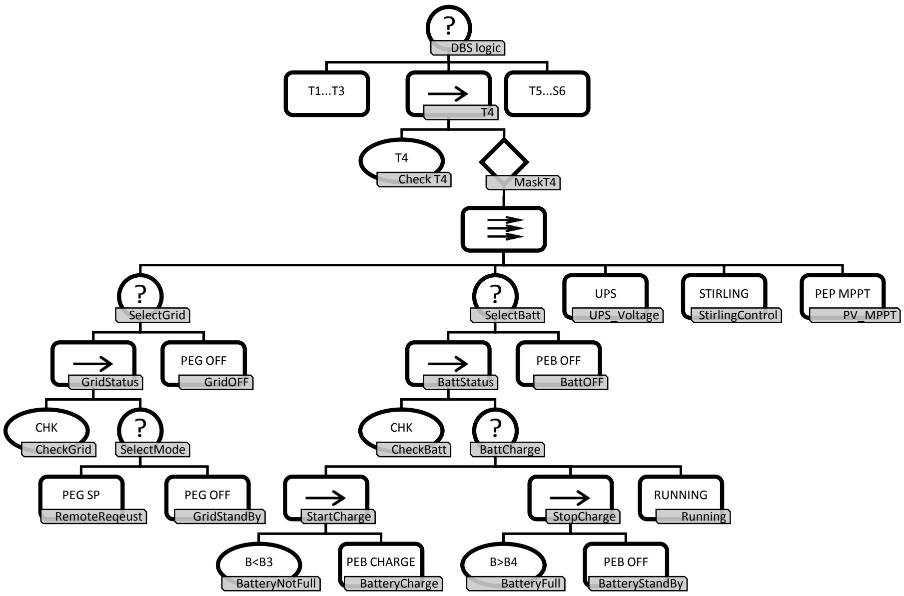


Fig. 3-47 subBT of the fourth threshold T4

3.4.2.5 Threshold T5

The T5 threshold indicates that the energy production is greater than the local demand. The power surplus is fed into the grid. The grid inverter is the voltage controlled converter.

The subBT of the grid inverter sets the master operation of the inverter, taking into account that the inverter in this threshold can only give energy. Thus, if a load connects to the NG while the inverter works in this threshold, to maintain the power balance, the energy for the new load has to be absorbed by the utility grid. But the inverter can’t absorb energy from the utility in this threshold. As a consequence, the dc bus voltage decreases and the T3 threshold will be triggered. This new threshold notifies to all the converters on the dc bus that there is low production from renewable sources, or anyway from local sources. For example, the battery will not charge in this new condition.

The storage converter in T5 can start the charge or it can continue to charge if the charging has already started. The behaviour is similar to the previous one. The difference is that in T5 the converter is current controlled. Thus, while in T4 the energy absorbed is obtained by the dc bus control, in T5 the energy is decided by the storage status, according to the information transmitted by the BMS. If the power required by the storage is greater than the available surplus the threshold T4 will be activated, because the energy from the utility grid is not used to charge the electrical storage.

This is coded in the BT shown in Fig. 3-48.

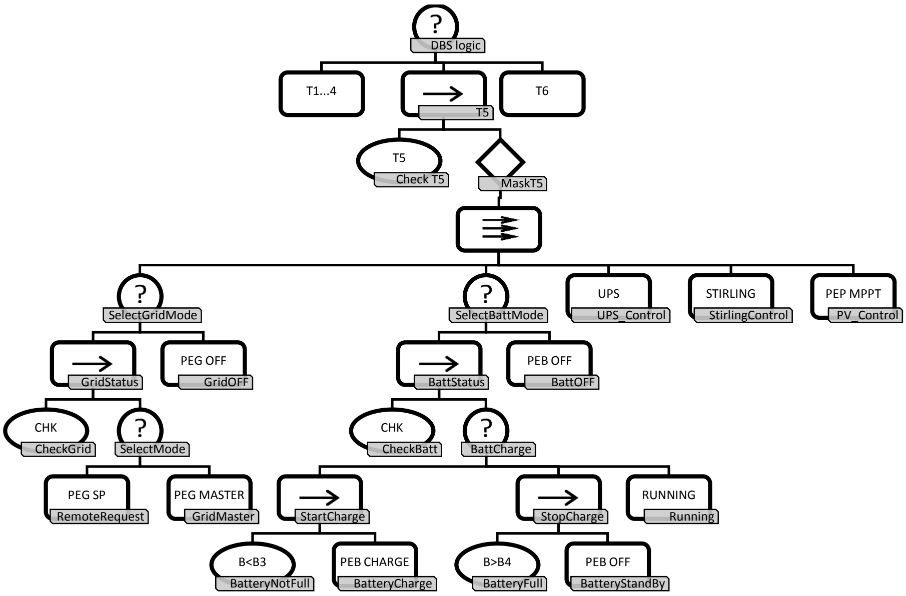


Fig. 3-48 sub BT of the fifth threshold T5

3.4.2.6 Threshold T6

The last implemented threshold is T6. This threshold is the threshold with the highest voltage, and it is activated when the surplus can't be supplied to the grid or can't be used to charge the storage system, which corresponds to an overgeneration. The surplus increases the dc voltage, until the T6 threshold is triggered.

If the *CheckGrid* is restored after a blackout, the node returns Success; the positive outcome activates the *GridToT5* node, which commands to the grid connected inverter to activate the voltage controlled mode and to restore the T5 threshold.

The storage converter works as in the threshold T5.

In this threshold *PV_Master* is ticked, therefore the power converter of the PV controls the dc bus voltage. The PV converter doesn't work in MPPT anymore, but it modulates the power to maintain the power balance in the grid and to stop the overgeneration.

The stirling node may command to the stirling generator to shut down. However, the time required to stop the production from the stirling generator is not compatible with the fast dynamic on the dc bus; however, the system has to manage this power surplus. If the voltage will increase over the T6 voltage reference value, that corresponds to a situation in which the PV production is zero, a braking resistor will start to dissipate the power production of the stirling generator. A dcdc converter controls the voltage to a voltage higher than the T6 voltage reference value, acting as master converter. This allows to the stirling generator to produce continuously electrical energy, but above all will continue to produce heat water. The continuous production of the heat water, in some critical conditions as a long blackout in a cold winter, is fundamental in some islanded applications.

This behaviour is implemented by the subBT in Fig. 3-49.

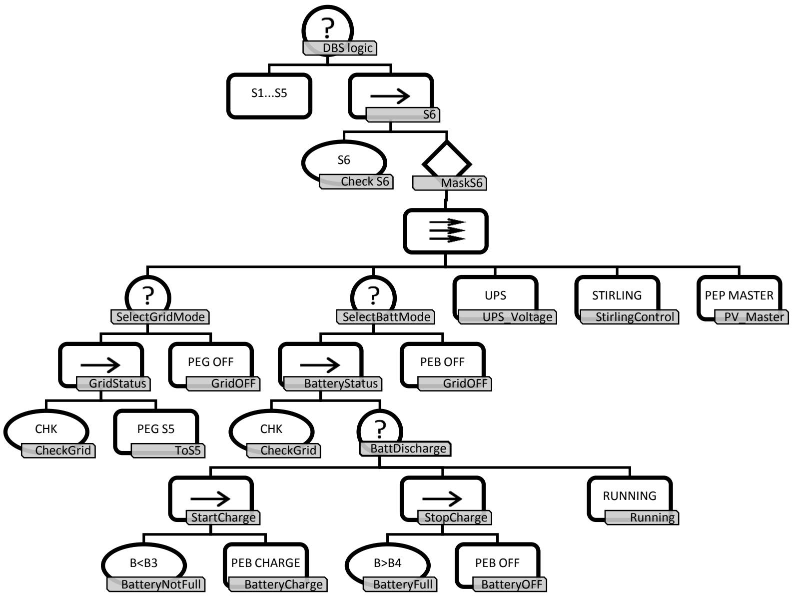


Fig. 3-49 subBT of the sixth threshold T6

3.4.2.7 Final Implementation

All the subBTs together build the overall BT, that is in Fig. 3-50. Some details are omitted in favour of maintaining a clear and readable graphical representation. Theoretically, the BT can be extended until a single node performs a very simple action, as the change of the state of an output digital pin. What it is important to understand is that from the very simple actions a really sophisticated algorithm can be produced thanks to this tool.

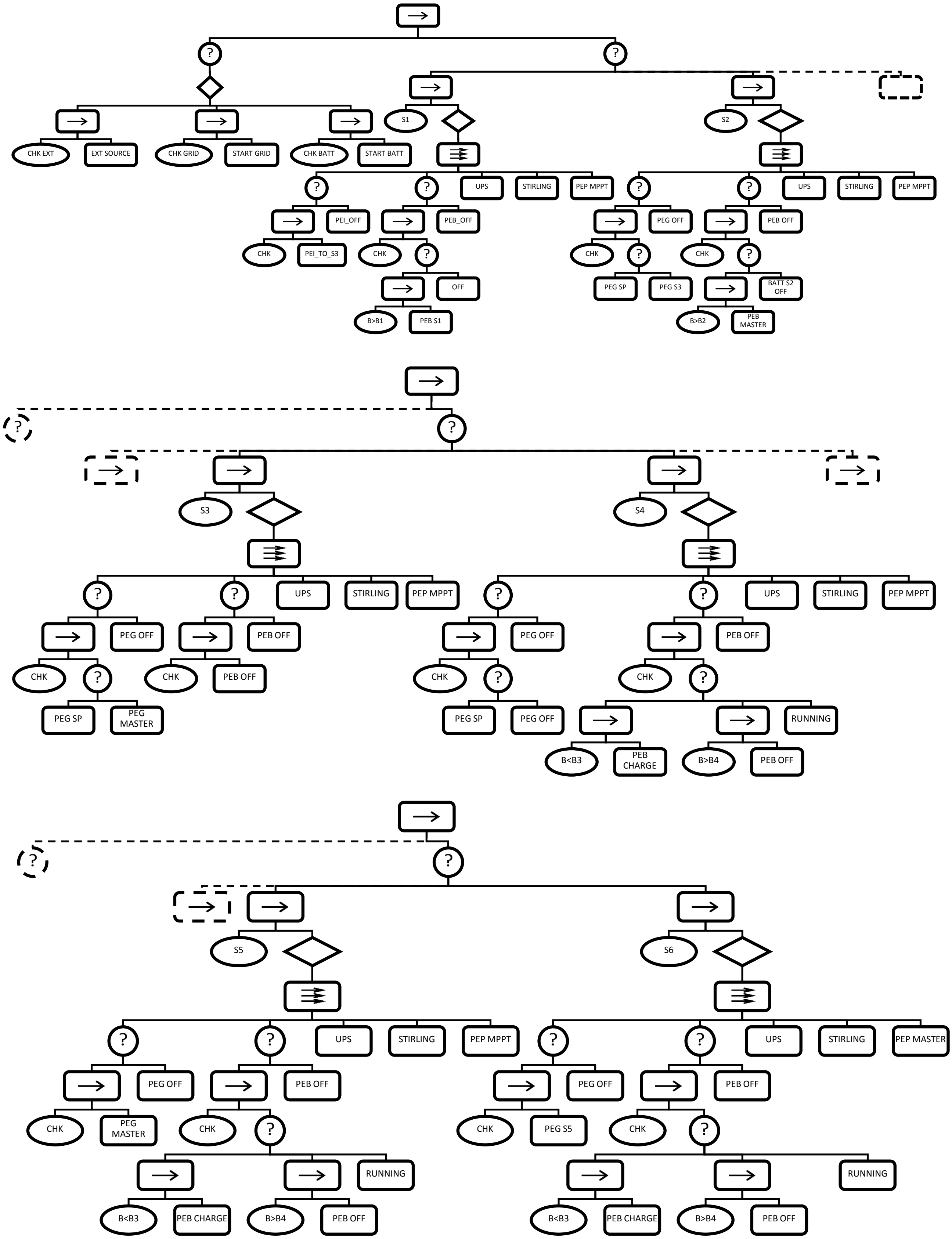


Fig. 3-50 Revision 2.0 of the BT used to control the nGfHA.

4 Implementation of the proposed control structure

The control algorithm is executed inside a processor, which is responsible for the management of all the converters in the system. The processor runs the discrete control and the converters continuous controllers. A detailed analysis of the dynamic behaviour of the converters and their interactions has been carried out. The used analytical models and the dynamic constraints are illustrated; they represent a great value of the research because are the result of many hours of analysis, simulations and laboratory tests. The design of feedback controllers uses common techniques for SISO systems and the specific controller is related to the specific implementation of the converter.

A detailed description about the implemented code in the NG controller is provided. The software design has been focused on developing a modular, easily changeable and reconfigurable software in a bare-metal environment. The software has been used for a number of applications with different resources and configurations, to demonstrate the reached flexibility.

Also the supervisor software is presented. The specific implementation used to report the result of the research is illustrated, but the idea is to generalize the structure to embrace other devices. The used IDE allows access to powerful libraries, which have been used to develop a versatile environment, where different kind of objects can interact to pursue a common goal, i.e. the home energy management.

4.1 Software architecture of the nanogrid controller

The NG controller has been implemented on the EVK1100 development board (Fig. 4-1). The EVK1100 is a fully functional embedded computer system [84]. The EVK1100 embeds an Atmel Dataflash (8MBytes) and a SDRAM (32MBytes). The on-board memories can be expanded with a MMC or SD card through the SD/MMC slot. Several communication interfaces are available on the EVK1100: RS232, USB and ETHERNET. The JTAG connector is used to program the on-chip flash and to debug. The expansion connectors provide developers access to unused interfaces from the AT32UC3A0512 microcontroller.

Fig. 4-1 The evk1100 evaluation board

The AVR32UC3A is a complete System-On-Chip microcontroller based on the AVR32 UC RISC processor running at frequencies up to 66 MHz. The AVR32UC is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high performance. The processor implements a Memory Protection Unit (MPU) and a fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems. Higher computation capabilities are achievable using a rich set of DSP instructions. The AVR32UC3A incorporates on-chip Flash and SRAM memories for secure and fast access. The Peripheral Direct Memory Access controller (PDCA) enables data transfers between peripherals and memories without processor involvement. PDCA drastically reduces processing overhead when transferring continuous and large data streams between modules within the MCU. The Timer/Counter includes three identical 16-bit timer/counter channels. Each channel can be independently programmed to perform frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation. The PWM modules provides seven independent channels with many configuration options including polarity, edge alignment and waveform non overlap control. One PWM channel can trigger ADC conversions for more accurate closed loop control implementations. The AT32UC3A also features many communication interfaces for communication intensive applications. In addition to standard serial interfaces like UART, SPI or TWI, other interfaces like flexible Synchronous Serial Controller, USB and Ethernet MAC are available [85].

The IDE used to program and debug the implemented code is Atmel Studio, which is an IDE derived from the well-known Visual Studio. The programming language is C. The C code can take advantage of the Atmel framework, known as Atmel Studio Framework (ASF).

The bare-metal approach doesn't allow a practical method to manage different tasks in the same controller, prioritize and execute them in the right order. To this end, a use of a ROS would allow a better method in managing different tasks inside the same controller; however, the ROS background activity require additional power processing.

The software architecture design has been focused on developing a modular, easily changeable and reconfigurable software in a bare-metal environment. The limited power processing and the resources required by the control algorithm aren't compatible with the adoption of a ROS to configure the activity execution.

The code flow is realized by three function: the main function that is the entry point for the software activity and the two interrupt functions. The interrupts are triggered by hardware events, to realize a real-time system.

Inside the main function the peripheral configurations and variables initializations are provided. After the setup, the main function enters in an infinite while loop where performs basic non-RT activities, such as LCD updates.

The first interrupt, generated by the timer counter (TC), executes the continuous control, i.e. the closed-loop control algorithms for the converters. The interrupt is generated from one of the three internal timer, with a sampling period compatible with the dynamic of the control loops. The best option is to generate this interrupt with the shortest period possible, which is the PWM period. In this interrupt the first layer control module of the presented architecture is executed.

The second interrupt is triggered by the real-time clock (RTC). The priority is lower than the first interrupt priority, because this function doesn't perform closed-loop activities, such as for example current control. However, the activity has a control and supervision role and can be executed at lower rate, even if the execution certainty must be guaranteed. In this interrupt the second layer module is executed.

4.1.1 Nanogrid controller code flow

The code flow normally follows the instructions in the main function. This execution is interrupted periodically by two timers. Each timer triggers an interrupt. The generated interrupt request is taken over by the INTC module. In the interrupt handling of the TC the first layer control is executed, while in the interrupt handling of the RTC, which has a lower frequency, the second layer is executed.

The main function has two sections. The first section provides to configure all the peripherals involved in the project and to initialize variables and data structures. The second section is an infinite loop, where the periodic tasks are executed. In this section the LCD available on the evk1100 is updated and the communication with external entities is provided. Periodically, the transfer is enabled and the buffer from the memory is transferred to the serial port. The buffer is then sent to the supervisor, which elaborates the request and generates the reply. The reply received from the supervisor is stored directly in the memory, with no processor intervention, by the PDCA controller. The processor in the main function provides to check if the reply has been sent back. The communication protocol is a custom serial protocol used to exchange different kind of information. This will be discussed in the following.

4.1.1.1 Main function

The flowchart of the main function is in Fig. 4-2. The first task is the peripherals setup. The configuration is platform-dependent, so the setup is specific for the specific processor. For the AVR32UC3A the following setups are provided:

- The clock source is the PLL0, fed by the oscillator OSC0, running at 66 MHz;
- TC is setup in Waveform Mode and used to generate the first periodic interrupt;
- RTC is used to generate the second periodic interrupt at lower frequency;
- ADC has a 10-bit resolution and a Throughput Rate of 294kSPS with all channel enabled;
- PWM has enabled 6 channel, left-aligned, with a frequency of 15 kHz;
- PDCA enables the USART TX and RX between internal memory and the USART module with two interrupt enabled.

Then, the interrupt handling functions are registered, so when the interrupt is generated the CPU is addressed to the handler and the interrupt can be processed. After that, the variables are initialized and the configuration to handle the hardware board is applied.

After the first section, the infinite loop starts and the periodic tasks are executed. In this section, mainly the communication and not-critical tasks are executed. Every second the LCD is update with new values.

Every 5 seconds the data transfer through the PDCA module is enabled from the memory to the USART TX. The processor is not interested by this transfer thanks to the PDCA, so it is free to do other (critical) work. The data transferred is formulated as a request for the supervisor. This request can be a reading or writing request. The requests and the replies to them are formulated according to a custom developed serial protocol. The protocol foresees 5 kinds of possible requests:

- VSS: the processor sends information to the supervisor, as the internal state of the NG, the measured values (bus dc voltage, currents and voltages of the converters present) and calculated variables (active and reactive power, rms values, ecc…);
- SP: the controller requires the optional set-points for the converters with the enabled/disabled commands;
- BMS: the battery data can be obtained directly by a third-party BMS. The required values are: charging current, discharging current, SOC, temperature and cluster voltage;
- ERR: a dangerous condition or an error status are notified with an error message;
- BT: all the states of each node of the BT are sent. This information is useful to monitor the BT state in real-time. This information is useful for debugging purpose.

The kind of requests to the supervisor may be independently selected. The correct and autonomous operation of the NG is independent from this correct communication (if the BMS is absent or if it is directly connected to nanogrid controller).

After that the request has been sent, the controller waits for the reply. The wait for the response is not a blocking-activity: the data transfer happens without CPU intervention from the serial peripheral USART RX to the internal memory. The CPU is only informed that new data are arrived by an interrupt generated by the PDCA. When new data are arrived and the CPU is inside the main function, the formal validity of the answer is verified.

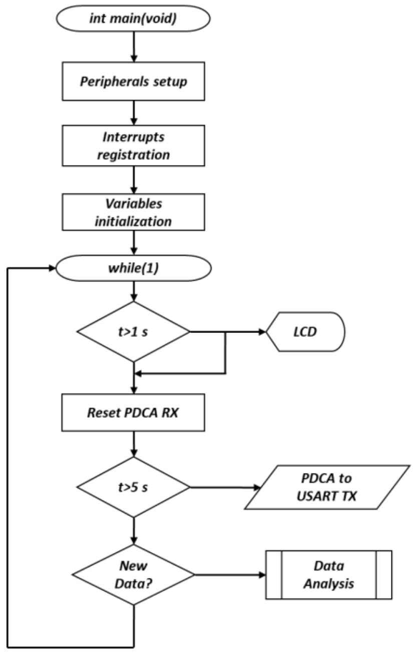


Fig. 4-2 Flowchart of the main() function

4.1.1.2 *First interrupt handler*

The main code flow is interrupted when one of the two interrupt is triggered. The interrupt of the timer counter, intTC, has higher priority than the interrupt generated by the timer counter, intRTC. The TC triggers the interrupt with a frequency of 5kHz.

The flowchart of the intTC interrupt handler is in Fig. 4-3. The interrupt source is cleared, to avoid that the INTC doesn't move on because the trigger stays enabled. Then, the values measured are read from the ADC registers. These values are the input for the feedback controllers, i.e. the first layer control. Each converter has a feedback controller with a goal; if a converter is not available in the NG the discrete control doesn't command its feedback controller.

Worth noting, the modularity of the BT allows to add an increasing number of converters, as shown in the previous chapter; however, when implementing the controllers for each converter is important to ensure that the computational power of the CPU is not saturated; this is one of the most critical aspect in the proposed implementation.

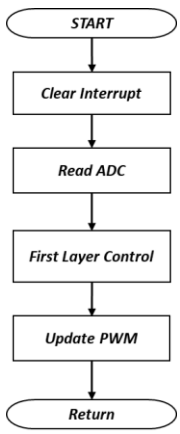


Fig. 4-3 Flowchart oh the first interrupt

4.1.1.3 *Second interrupt handler*

The second interrupt intRTC has a frequency of 1 kHz. This interrupt handler executes the BT used to control the operation of the NG and to decide the operating point, i.e. the state of the system. The BT is called in the interrupt handler through the Root node. The end of the execution of the BT coincides with a returned status from the Root node.

The flowchart of the second interrupt handling is in Fig. 4-4. The main task in this function is the execution of the second layer control. After the BT returns from the execution, the controller verifies the presence of some critical errors. If one or more errors are detected the controller starts the shutdown procedure. Every 100 ms the rms values, the active and reactive power are updated and stored in memory.

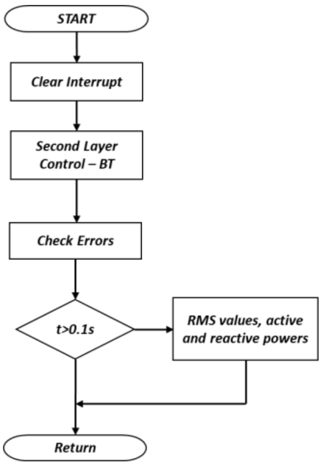


Fig. 4-4 Flowchart of the second interrupt

4.1.2 *Platform-dependent code and peripherals setup*

In this section the main peripherals used in the project are illustrated. A brief description is done for all of them. Then, the configurations adopted inside the project are illustrated. However, since this is platform-dependent the discussion will not go deep into the implementation details. Indeed, the main goal of the discussion is to highlight the stratification and modularity of the presented structure and how it is possible to implement it with a simple microcontroller.

Clock

The first configuration is the clock for the peripherals. The clocks of the AVR32 can be derived from several sources: an internal high precision 32 kHz; an RC oscillator (low-precision) at approximately 115 kHz; an off-chip crystal oscillator (OSC0 e OSC1). The Power Manager (PM) controls the oscillators and PLLs, and generates the clocks and resets in the device. The PM controls two PLLs; the PLLs can provide higher frequencies multiplying the clock from either oscillator. The PLL output is divided by a multiplication factor, and the PLL compares the resulting clock to the reference clock. The PLL will adjust its output frequency until the two compared clocks are equal, thus locking the output frequency to a multiple of the reference clock frequency. The PM masks the oscillator outputs during the start-up time, to ensure that no unstable clocks propagate to the digital logic.

The source used to feed the PLL0 is the OSC0. The OSC0 frequency of the oscillator mounted on the evk1100 is 12MHz. Then, the PLL0 has been configured as clock source for the CPU. A multiplier value of 10 and a divider equals to 1 have been chosen, so the frequency of the clock is $f_{vco} = 12 * (10 + 1)/1 = 132\ MHz$.

The final frequency fpll is divided by 2 in the option, so to achieve a frequency of 66MHz, which is the maximum admissible value for the CPU. Then the clock for buses are selected as follows: 16.5MHz for PBA and 66MHz for PBB and HSB.

ADC

The ADC is based on a Successive Approximation Register (SAR) 10-bit Analog-to-Digital Converter (ADC). A software or a hardware trigger can start a conversion. The software trigger is provided by writing the Control Register (CR) with the bit START at 1. Only one start command is necessary to initiate a conversion sequence on all the channels. The ADC hardware logic automatically performs the conversions on the enabled/active channels, then waits for a new request.

Basically, an ADC requires some time for startup, sampling, and holding and for conversion.

The startup time is the time required to guarantee the best converted value after the ADC has been enabled either for the first time or after a wake up from specific sleep modes. The Sample and Hold time is the time to charge the internal capacitor to a stable value in order to get accurate conversion results. Switching between multiple channels requires a specific amount of time, before beginning the sample and hold phase, in order to have accurate results. This is the settling time.

The combination of the sampling time and the holding time, is called conversion time. This is usually represented in number of clock cycles (Table 4-1). The conversion time is the primary parameter in deciding the speed of the ADC.

Table 4-1 ADC parameters

ADC clock frequency	4,125	MHz
Resolution	10	bit
Conversion Time	10	ADC clock cycle
Capacitor charging time $\boldsymbol{T_s}$	8,85	ns
$\boldsymbol{S\&H\ time}$	4	ADC clock cycles>Ts
Throughput Rate	294	kSPS

When an analog-to-digital conversion begins, one channel select switch is closed (MUX), allowing the sample and hold capacitor to charge. The charge applied to the sample and hold capacitor is then converted into a digital representation by the successive approximation register. When a source is connected to the ADC input, the conversion process has to be started after that the input capacitance has reached a stationary value. Let's consider this value be Vin ± (Vin/Resolution). The worst case appears when C_{in} is completely discharged and V_{in} is maximum ADC input voltage [86]. The capacitor charging time is related to the RC equivalent circuit.

$$T_s > (R_s + R_{in})C_{in} \ln(2^{10})$$

The S&H time has to be greater than the T_s. This time is related to the SHTIM bit in the MR register. From the datasheet the minimum value is 0,6us, so 4 cycles are required to complete the Track and Hold Acquisition Time. The final throughput rate with these settings is 294kSPS.

When a conversion is completed, the resulting 10-bit digital value is stored in the Channel Data Register (CDR) of the current channel. The channel EOC bit in the Status Register (SR) is set. Reading one of the CDR registers clears the corresponding EOC bit.

Interrupt Controller (INTC)

The INTC collects interrupt requests from the peripherals, prioritizes them, and delivers an interrupt request and an autovector to the CPU. The INTLEVEL field is associated to each group to give a priority level from INT0 to INT3, so 4 priority levels for regular, maskable interrupts are available (Fig. 4-5).

The request masking hardware maps each of the GrpReq lines to a priority level from INT0 to INT3 by associating each group with the INTLEVEL field in the corresponding IPR register. The GrpReq inputs are then masked by the CPU status register. Masking of the interrupt requests is done based on five interrupt mask bits of the CPU status register, namely interrupt level 3 mask (I3M) to interrupt level 0 mask (I0M), and Global interrupt mask (GM). Any interrupt group that has a pending interrupt of a priority level that is not masked by the CPU status register, gets its corresponding ValReq line asserted.

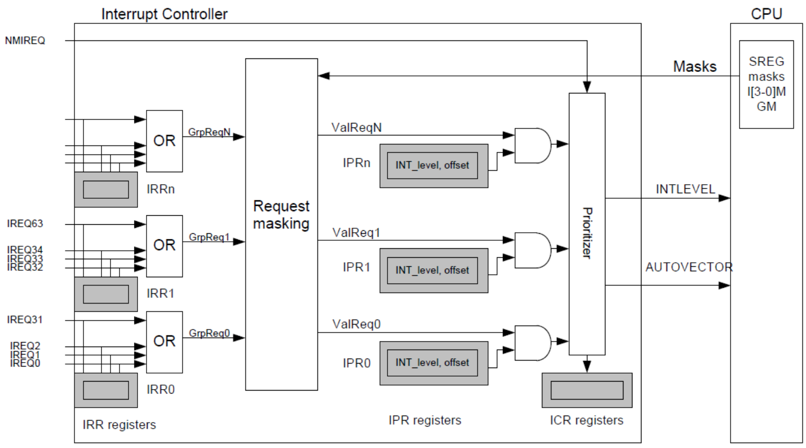


Fig. 4-5 Interrupt Controller: block diagrams

All interrupt signals in the same group share the same autovector address and priority level. If several interrupt lines are active in the group, an interrupt line must be selected. The highest number line is selected. This is to be coherent with the prioritization of interrupt groups performed by the hardware interrupt controller.

After a group is selected the handler of the active line has to be invoked. The function *_get_interrupt_handler* in intc.c, through the Interrupt Cause Register (ICR), identify the group that has a pending interrupt of the corresponding priority level.

```
__int_handler_get_interrupt_handler(uint32_t int_level)
```

The return is an interrupt handler address stored in a structure, when the handler has been registered. The correct handler is returned looking for the active line and its correspondent handler function. If more lines are active the first invoked handler is the one associated with the line with the highest number inside the same group.

To register the function, the following function from the ASF library can be used:

```
void INTC_register_interrupt(__int_handler handler, uint32_t irq, uint32_t int_level)
```

This function stores the handler address in the *structure_int_line_handler_table_x* the pointer to the interrupt handler, where all the register handlers are divided for group and, inside the same group, for lines. so that *_get_interrupt_handler* can retrieve it when the interrupt is vectored.

```
_int_line_handler_table[irq % AVR32_INTC_MAX_NUM_IRQS_PER_GRP]=handler;
```

Timer Counter (TC)

The Timer Counter (TC) includes three identical 16-bit Timer Counter channels. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts. A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

The following triggers are common to both modes:

- Software Trigger;
- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger.
- Compare RC Trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value.

Each channel can independently operate in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

One timer is used in Waveform mode and a periodic interrupt is generated when the timer reaches the RC compare register. The interrupt triggered allows the closed-loop control of the first layer software. The timer of the clock is derived by the PBA clock, and the RC register is TIMER_CLOCK/SamplingFrequency.

Real-Time Counter (RTC)

The Real Time Counter (RTC) enables periodic interrupts at long intervals, or accurate measurement of real-time sequences. The RTC is fed from a 16-bit prescaler, which is clocked from the 32 KHz oscillator.

The RTC can generate an interrupt when the counter wraps around the value stored in the top register, producing accurate periodic interrupts. The interrupt generated is used to allow the periodic execution of the second-layer discrete control.

Pulse Width Modulation (PWM)

The PWM peripheral is primarily composed of a clock generator module and 7 channels; each channel is clocked by one of the outputs of the clock generator and each channel can independently choose one of the thirteen clocks. Each channel generates an output waveform with attributes that can be defined independently for each channel through the user interface registers.

The different properties of output waveforms are:

- internal clock selection;
- waveform period;
- waveform duty cycle;
- waveform polarity;
- waveform alignment.

An example of a configuration channel is shown. In the example, the period of the generated PWM is: (2*550)/16.5MHz=66.667us, that is a 15 kHz frequency.

```
pwm_channel_0.cdty = 225;          /* Channel duty cycle, should be < CPRD. */
pwm_channel_0.cprd = 550;  /* Channel period. */
pwm_channel_0.cupd = 0;            /* Channel update is not used here. */
pwm_channel_0.CMR.calg = PWM_MODE_LEFT_ALIGNED; /* Channel mode. */
pwm_channel_0.CMR.cpol = PWM_POLARITY_LOW;      /* Channel polarity. */
pwm_channel_0.CMR.cpd = PWM_UPDATE_DUTY;        /* Not used the first time. */
pwm_channel_0.CMR.cpre = AVR32_PWM_CPRE_MCK_DIV_2; /* Channel prescaler. */
```

Peripheral DMA controller (PDCA)

The Peripheral DMA controller (PDCA) transfers data between on-chip peripheral modules such as USART, SPI, SSC and on- and off-chip memories. Using the PDCA avoids CPU intervention for data transfers, improving the performance of the microcontroller. The PDCA can transfer data from memory to a peripheral or from a peripheral to memory. This peripheral has been used to transfer data between the memory and the USART, in both directions.

When a transmit buffer is empty or a receive buffer is full, an interrupt request can be signalled. The PDCA has three interrupt sources:

- Reload Counter Zero: the Transfer Counter Reload Register is zero;
- Transfer Finished: both the Transfer Counter Register and Transfer Counter Reload Register are zero;
- Transfer Error: an error has occurred in accessing memory.

The two programmed channels use two interrupt to transmit and receive buffer from and to the USART module.

4.1.3 Nanogrid controller – Source code

In this section the source code is presented. The directory structure is in Fig. 4-6. The code for the evk1100 is inside the src directory. Inside the *src* folder there are four folders. The nanogrid folder contains the code used to implement the control of the system; indeed, the BT folder that implements the discrete control and the feedback controllers of the converters are inside this folder. For each converter there is a folder with all the available low-level controllers selectable by the BT.

The *ASF* folder includes the drivers provided by the Atmel Studio Framework used to configure and use the peripherals involved in the project.

The *config* folder includes some useful definition in some include files used inside the project and provided by the same ASF to keep track of some configuration constants.

The *Peripherals* folder includes all the files used to initialize all the used module inside the program and to interface the software with the external environment through the enabled peripherals. Besides, this side of the code aims to reach an abstraction between the code and the used processor, to implement the same code in another processor with a different architecture only modifying the code in this folder.

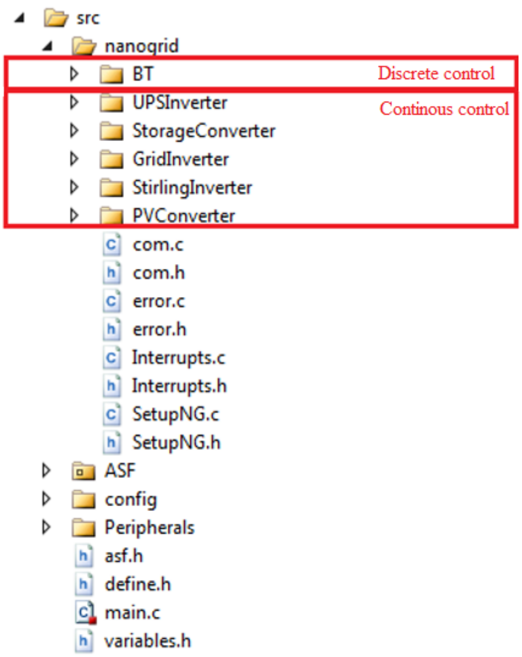


Fig. 4-6 Folder structure of the source code

A brief description of some files used in the project are in Table 4-2.

Table 4-2 Files description

Filename	Description
define.h	In define.h there are the configurations parameter to adapt the firmware to the specific implementation. This allows to use the same software for different hardware revisions. Besides, different sources can be selected to give flexibility among the sources available. The maximum voltages and currents for each source are defined. The presence of a third-party BMS is selected. The threshold values for the DBS are defined in this file. Three parameters have to be selected: Vref_min_DC_bus_signal is the value of the first threshold, DVTx is the width of the threshold and then DVth is a hysteresis value to allow the threshold transition. This allows to adapt the voltage level to different physical constraints among the possible applications that the NG can deal with.
variables.h	The file includes the global variables used inside the project. All available information is stored in a random access array: each position of the array has an assigned variable in which the value is stored. The status of the NG is stored in the VectorStatusSystem (VSS) array. The node states of the BT are stored in VSBT array. This is not mandatory, but is useful for debugging purpose. For each converter there is a structure that collect all the information of the converter. Two auxiliary enumerated types are used: err_conv and status_conv, where conv is the name of the converter. The err_ conv is used to store all possible error code, as overcurrent, overvoltage and so on. The status_conv stores all the planned operating mode of each converter, which are converter specific.
com.c	The file defines the functions needed to manage the serial communication with the external supervisor. The information to write and to read are stored in a data structure, where the kind of request and the values are temporary stored. Besides, functions to build the frame to write over serial and functions to read the frame from serial are implemented. The read data structure is temporary stored to check the integrity of the data received.
error.c	This file includes functions to handle errors and to build structures to alert that an error has occurred. The errors can be sent over the serial communication other than display them on the LCD.
interrupts.c	This file includes the two interrupt handlers used in the project.
setupNG.c	This file is used to reconfigure the NG input and output, following the declaration made in define.h. In this file is selected the kind of sources connected to the NG.

Each converter in the C code has a correspondent data structure. The converter in its structure has a possible errors list enumerated in *err_conv*:

```
typedef enum
{
        CONV_INPUT_OVERVOLTAGE,
        CONV_INPUT_UNDERVOLTAGE,
        CONV_INPUT_OVERCURRENT,
        CONV_OUTPUT_OVERVOLTAGE,
        CONV_OUTPUT_UNDERVOLTAGE,
        CONV_OUTPUT_OVERCURRENT,
} err_conv;
```

Specific errors for a converter can be united with the *err_conv* list.

Another enumerated structure is *status_conv* that enumerated the possible control modes of the converter:

```
typedef enum
{
        CONV_ABSENT,
        CONV_DISABLED,
        CONV_CURRENT_CONTROL,
        CONV_VOLTAGE_CONTROL,
        MASTER_MODE
} stato_conv;
```

All the information related to the converter and the variables used for the control of the specific converter are stored in the *cv_conv* structure:

```
//cv=converter
typedef struct cv_conv
{
       err_conv error;
       stato_conv status;
       double temperature;
       int input_voltage;
       int output_voltage;
       float setPointBusDC;
       float Imax;
       float Imin;
       double power;
       int channelPWM_A;
       int enabledPin;
}cv_conv;

cv_conv conv;
```

4.1.4 Implementation of the discrete control layer

Behaviour trees are made up of independent tasks, i.e. independent nodes, each with its own algorithm and implementation. All of them conform to a basic interface which allows them to call one another without knowing how they are implemented. The BT implementation is simple and efficient in object-oriented programming languages. However, most microprocessors are programmed in C, that is not an object-oriented language. So, how is it possible to "emulate" an object in C language to easily implement BTs? The adopted solution is to use pointers function. A control flow node calls its child through the pointer to the function that implements the child node. If the child node changes, the only change to do is to assign a different pointer. All child nodes are implemented in an array of pointers: each element of the array is the address of the function that implements the child nodes.

The declaration of the array is provided. The return value is an integer, that is the outcome of the execution of the nodes. The functions don't take any arguments. This is true for all nodes to keep inherent compatibility among nodes.

```
int(*ArrayPuntatori[n])()={PointerFunction_1, PointerFunction_2,…, PointerFunction_n};
```

For debug purpose, the outcome of each array is stored in another array, in which the element [0] corresponds to the outcome of the root node.

```
VSBT[N]=(*PointerArray[n])();
```

The control flow nodes call the child nodes; the spread of the tick is related to the outcomes of the leaf nodes. The code for selector, sequence and parallel node is now provided.

A. Selector node code

The selector node is a function in the C code. The function hasn't any arguments and returns an integer value. This value is the result of the execution of the node. In the example, this node has a number of child nodes equals to size. These child nodes are called inside a for-loop. The selector node calls in sequence its child nodes, i.e. the pointer to the function in the array, until one of its nodes returns a positive outcome. If all nodes return a negative outcome, then also the selector node returns the same outcome.

```
int (*pointerFunction[size])(void)={function1,function2,…,functionSize};
```

```
int selectorNodeNx(void)
{
	for (int w=0;w<size;w++)
	{
		VSBT[Nx]=(*pointerFunction[w])();

		if (VSBT[Nx]!=Failed)
		{
			return VSBT[Nx];
		}
	}
	return Failed;
}
```

B. Sequence node code

The same structure of the selector node is used for the sequence node. However, this time the for-loop is interrupted as soon as the outcome of a child node is not Succeed. The sequence node returns Success only when all its children return a positive outcome.

```
int (*pointerFunction[size])(void)={function1,function2,…,functionSize};


int sequenceNodeNx(void)
{
	for (int w=0;w<size;w++)
	{
		VSBT[Nx]=(*pointerFunction[w])();

		if (VSBT[Nx]!=Succeed)
		{
			return VSBT[Nx];
		}
	}
	return Succeed;
}
```

C. Parallel node code

In the parallel node there is not a direct correspondence between the outcome of one child and the status of the parallel node. Instead, the parallel node keeps track of all the outcomes of its children and then compare the number of successful and failure nodes with values defined by the designer. For example, if the successful nodes are greater than a value *X*, then the outcome of the parallel node is Success.

```
int (*pointerFunction[size])(void)={function1,function2,…,functionSize};

int parallelNx(void)
{
	static int stateParallelT1[size];
	int contSucc;
	int contFail;

	for (int i=0;i<size;i++)
	{
		stateParallelNx[i]=(*pointerFunction[i])();

		if (stateParallelNx[i]==Succeed)
		{
			contSucc+=1;
		}
```

```
                else if(stateParallelNx[i]==Failed)
                {
                        contFail+=1;
                }
        }

        if (contSucc>=size-y)
        {
                VSBT[N_parallelT1]=Succeed;
                return Succeed;
        }
        else if (contFail>=size-z)
        {
                VSBT[N_parallelT1]=Failed;
                return Failed;
        }
        else
        {
                VSBT[N_parallelT1]=Running;
                return Running;
        }

}
```

The last kind of used node is the decorator node. This node has not a defined implementation, because its behaviour is specific and defined for the specific use.

The possible outcomes returned are in Table 4-3. The outcomes have the same meaning of the classical theory of the BT illustrated in the previous chapter. However, compared to the classical implementation two additional possible outcomes are used: Start and Error. The nodes associated with the "Start" value are not-ticked node. The nodes that return "Error" are affected by a not-reversible issue. The difference between Error and Failed is that the "Failed" status indicates a reversible or temporary status, while Error is a dangerous not-reversible status.

Table 4-3 Node returned states

Outcome	Value	Description
Start	0	Not-ticked node
Succeed	1	Action performed correctly/Condition true
Running	2	Action that requires some time before returning an outcome
Failed	3	Action not correctly executed/ Condition False
Error	4	Not-reversible condition – shutdown

The BT execution starts calling the first node of the BT, the root node:

```
VSBT[ROOT]=(*Root[n])();
```

The Root node, is a sequence node cyclic executed with a frequency f_{tick} equals to1 kHz. The root node, calls the *"blackboard"* function, in which global analogic and digital input are acquired. The local analogic variables, for control purpose, are acquired by the control algorithm, to respect the sampling frequency constraints, according to the frequency response of the converter.

```
int (*NanoGridManagement[2])(void)={startNG, DBS};


int Root(void)
{
        Blackboard();

        for (int n=0;n<2;n++)
```

```
        {
                VSBT[ROOT]=(*NanoGridManagement [n])();

                if (VSBT[ROOT]!=Succeed)
                {
                        return VSBT[ROOT];
                }
        }
        return Succeed;
}
```

The VSBT[ROOT] stores the outcome of the execution of the BT. The VSBT is an array and it is used to store all the outcomes of the nodes in the BT. This is a really useful way to track the status of the BT, but mainly to debug the control software during the testing stages. Indeed, if the BT structure has 1 to 1 correspondence in the code, it can become very useful for real-time debugging. Besides, if an error is revealed during the testing stage the designer can easily go up to the error position in the code which implements the BT only by looking the graphical representation of the tree. The correction can be made without affecting other parts of the code.

The graphical representation is also a valid tool to confront stakeholders involved in the design and the development stage. The BT allows to reduce the complex original problem in a number of smaller problems: the overall behaviour is given by all the small actions, decisions and actuations.

4.2 Software architecture of the nanogrid supervisor

The supervisor of the NG runs on a single-board computer. This board gathers all the information of the entities involved in the energy management of the building, such as the home automation system, and the entities that interact directly with the NG, such as a third party BMS.

The board runs a Linux-based OS. The application is created in the Qt IDE. Qt is a cross-platform application framework that is used for developing application software that can be run on various software and hardware platforms with little or no change in the underlying codebase, while still being a native application with native capabilities and speed. Qt is used for developing multi-platform applications and graphical user interfaces (GUIs); however, programs without a GUI can be developed, such as command-line tools and consoles for servers.

The developed application uses a declarative scripting language called QML, included in the same Qt. The logic inside the QML code is provided by the possibility to use JavaScript code. The classes in the Qt QML module enable QML objects to be loaded and manipulated from C++, and the nature of QML engine's integration with Qt's meta object system enables C++ functionality to be invoked directly from QML. This allows the development of hybrid applications which are implemented with a mixture of QML, JavaScript and C++ code. Integrating QML and C++ provides a variety of opportunities, including the ability to:

- Separate the user interface code from the application logic code, by implementing the former with QML and JavaScript within QML documents, and the latter with C++
- Use and invoke some C++ functionality from QML
- Access functionality in the Qt QML or Qt Quick C++ API
- Implement your own QML object types from C++ — whether for use within your own specific application, or for distribution to others.

QObject-derived C++ class can be registered as the definition of a QML object type. Once a class is registered with the QML type system, the class can be declared and instantiated like any other object type from QML code. Once created, a class instance can be manipulated from QML; the properties, methods and signals of any QObject-derived class are accessible from QML code.

To register a QObject-derived class as an instantiable QML object type, call qmlRegisterType() to register the class as QML type into a particular type namespace. The namespace has to be imported in order to use the type:

qmlRegisterType<ObjectQML>("namespace", 1, 0, " ObjectQML ");

4.2.1 Code structure and interaction diagram

The supervisor coordinates the activities between the involved entities inside the NG. The code structure allows to add new devices, which can be easily integrated with the existing objects. The supervisor gathers all the information from the entities involved in the energy management of the building, interacting between them to reach a common goal.

The supervisor manages the interactions between the objects in the “supervisor” class. The interaction happens mostly using the signal/slot mechanism provides by Qt.

The supervisor provides a direct interface with the NG controller. Inside the C++ code, the object NG ensures the interface between the physical hardware of the NG controller, represented by a serial optocoupled port, and the Qt application. The implemented custom serial protocol is optimized to allow a high throughput, while the integrity of the frame transmitted is verified. Five different messages can arrive through the serial communication. For each message a proper action is provided and an appropriate reply is generated. The information transmitted from the NG for other entities are sent to the Qt application, which provide to direct the messages to recipients. Each kind of message is summarized in Table 4-4.

Table 4-4 Possible requests sent by the nanogrid controller to the supervisor through the custom serial protocol.

Request	***Description***
VSS	The values arrived (NG state, voltages, currents, …) with a VSS message are sent to the GUI, a remote database, a local server
SP	The answer to a SP message is the set-point for the NG converters. The value of the set-point can be taken from a remote server, from the local database where the result of the EMS activity is stored. Other sources can be integrated. The source of the set-point is masked to NG by the supervisor class.
BMS	The BMS message asks a reply with the BMS information. The information are taken from the BMS and given to the NG class by the supervisor class.
ERR	If errors are detected, then the error code(s) is shown on the GUI and logged locally.
BT	This option is introduced to display graphically the BT logic for debugging purpose.

Not all the entities in Fig. 4-7 are present in all the applications and other entities can be easily integrated following the developed architecture. Fig. 4-7 refers to the specific implementation of the experimental setup, that will be presented in the last chapter. In this setup, the foreseen entities are enumerated in an enumerated structure in the source code, that is:

```
typedef enum object {REMOTE_DB, SEAG, ECU_STIRLING, BMS, DASHBOARD} object;
```

The presence of each entity is declared in the *config.txt* file, inside the *config* folder. This file is read when the supervisor class is instantiated; this class provides to create the objects declared in the *config.txt* file. The presence of each entity is stored in a QList<object>. This class is registered to be used in the QML like QML object and it is the interface between the GUI and the application code, that runs underneath the hood. The GUI is update each time the signal *updateGUI()* is emitted, that gather by the QML engine allows to update the values displayed on the GUI (Fig. 4-8). Moreover, the user interaction is allowed thanks to an optional touchscreen mounted on the single-board: each time the GUI emits a signal the C++ code calls the correspondent slot. The GUI can be seen as another entity interacting with the supervisor class.

The optional entities involved in the specific presented implementation are in the following briefly described. Each entity is an object inside the code, which interact with the supervisor class through the signal/slot mechanism.

Battery Management System

This class implements a ModBus/TCP client. The client periodically queries the ModBus server that runs inside the third-party BMS of the storage system, adopted in the presented project. Because a lot of data are provided by the BMS, in this class, the most important data are selected and aggregated. These data are sent to the NG controller.

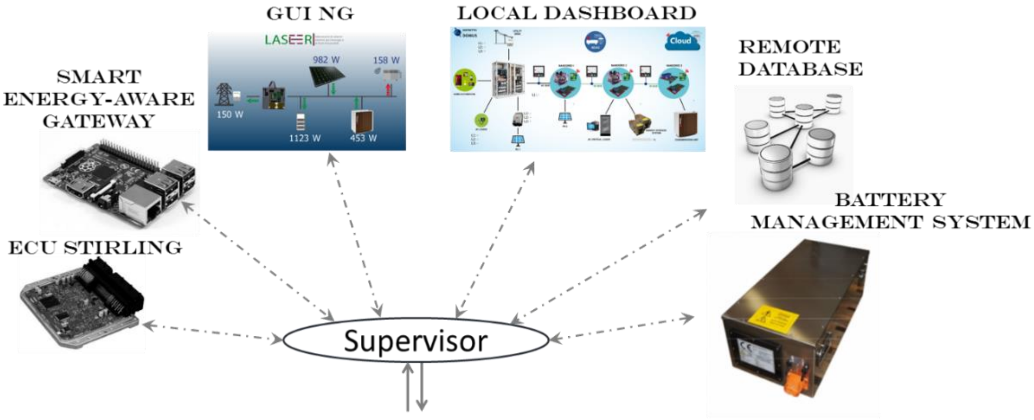


Fig. 4-7 Interaction of the supervisor with external objects to support the nanogrid controller in the decision-making process

Smart Energy-Aware Gateway

This class manages the communication between the supervisor and the Smart Energy-Aware Gateway (SEAG). The SEAG coordinates the objects of the physical layer with the cloud. The cloud is used to store aggregated data and to obtain useful information to establish the optimal behaviour of the physical system, through the invocation of services and complex applications. The optimal behaviour is obtained solving a local optimization problem. The physical system is represented by three nanogrids, the stirling generator and a home automation system.

The result of the optimization problem is stored in a local database; the supervisor queries the database through a TCP connection. The nanogrid supervisor, once a day, queries directly the DB to obtain the set-point for the following 24 hours. These 24 values are stored as a 24 elements array and sent to the NG through the NG class, each time the NG controller asks for the SP message.

The data from the NGs are sent to the SEAG through a TCP socket. The SEAG stores the information on its local database and then, use the stored information to solve the optimization problem.

ECU Stirling

This is the interface between the serial port of the ECU that controls the stirling generator and the supervisor. This object reads the data from the ECU through a serial port. The data are used to visualize various variables, such as temperatures, flow rate, electrical power. The supervisor acts as a gateway, sending the collected data to SEAG, which uses this information to solve an offline optimization problem. Each time new data arrive a signal is emitted.

Remote Database

Each time arrive useful data from one of the entities connected to the gateway controller, these are sent to a remote server for data-logging purpose. The class implements a HTTP client. The class has the task to elaborate and manipulate data before send them over internet.

Local Dashboard

All the information of the system is gathered by a local server. The information is used to update a Graphical User Interface, where there are visualized the power flows inside the system. This GUI will be illustrated in the next chapter.

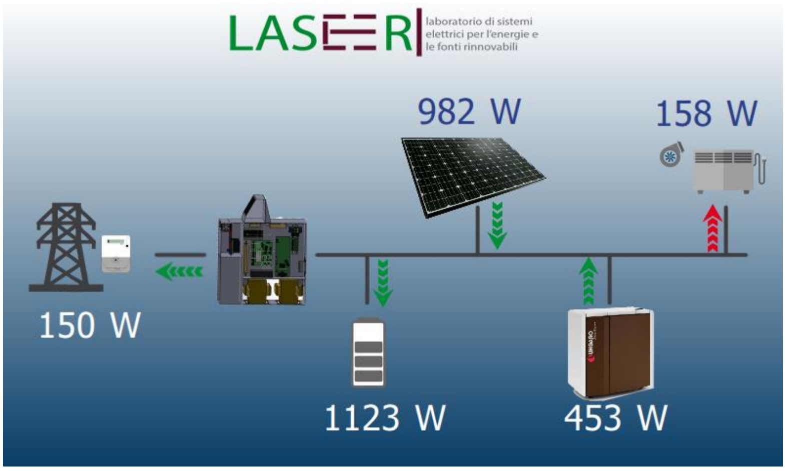


Fig. 4-8 Supervisor GUI launched on a 10 inches touchscreen

To represent the control flow between the various objects inside the system, in Fig. 4-9 is shown the correspondent Sequence Diagram. This diagram is one of the diagrams used inside the Unified Modeling Language (UML). The Sequence diagram, which belongs to the Interaction Diagrams, has the goal to visualize the interactive behaviour of the system. The diagram shows the relations among the objects and how they cooperate and interact between them to realize the behaviour of a use case. They help to describe, in a dynamical manner, what happens inside the system, when an action is performed; they emphasize the time sequence of the messages exchanges between the objects.

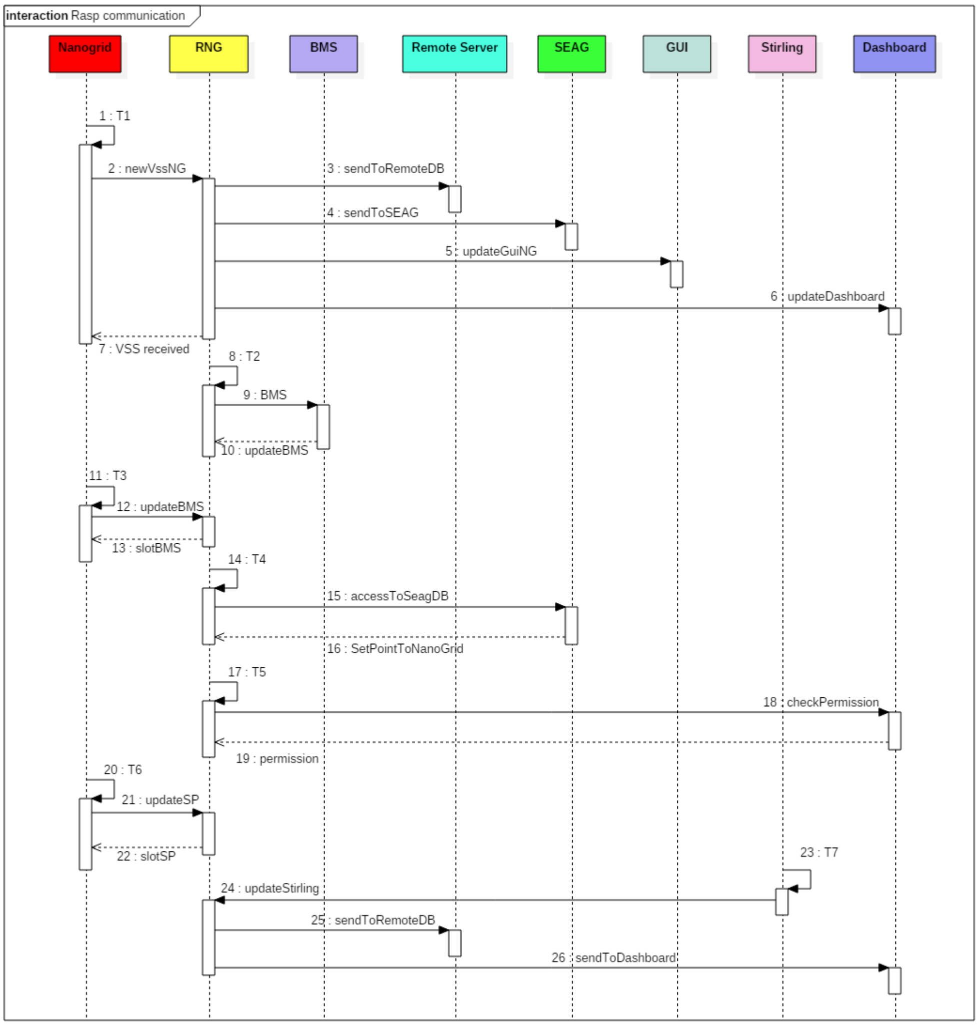


Fig. 4-9 Interaction diagram of the supervisor software

5 Experimental results

The effectiveness of the proposed control structure is demonstrated in a number of prototypes developed for different kind of applications. In this work, a real application of the nanogrid control system is illustrated. The application shows the flexibility and modularity of the control structure. The installed system is in the campus of the University of Calabria and it is used to give electrical energy in some spaces inside a building of the campus. The building is a residential building for the students.

The system is realized connecting three nanogrids to a common dc-bus. The first nanogrid is connected to the utility grid and to a 4.5 kW PV plant. The second nanogrid has a 232V/22 kWh storage system and an inverter to supply ac critical loads. The last nanogrid is used to interface a stirling generator. The rated power of the first two NGs is 7 kW, while the rated power of the last NG is 1 kW.

Each NG can be autonomously started or they can be started together. The NGs interact communicating through the dc bus power connection, realizing a dc-bus signal control. Each NG has a supervisor. The supervisor is configured according to the nanogrid specifications.

All the supervisors communicate with a local dashboard and with a Smart-Energy-Aware-Gateway (SEAG). The SEAG gathers data from the NGs and has access to cloud services as load and production forecasting. All this information allows to solve a local optimization problem. The result of the optimization is a 24h ahead scheduling for the entities involved and for the nanogrids. Each supervisor has access to the results of the optimization and send this information to the nanogrid controller; then, the nanogrid controller decides to actuate the result of the optimization.

5.1 Evolution of the control board and control structure

In this section, the evolution of the nGfHA project is briefly summarized.

5.1.1 Revision 0

The nGfHA project has started three years ago. The very first fully operational prototype is in Fig. 5-1. The prototype had four different controllers to manage the grid connection, the stirling generator, a lead acid storage. Each controller had its own task. But they had to share some information, as the presence of the utility grid to work correctly through a communication link.

The experience developed showed the bad design of this first prototype, but it was useful to improve the overall design. The strong interest in developing an industrial product pushed to the reduction of costs, i.e. the number of used components.

An interesting aspect in this prototype was the use of a dual active bridge converter (DAB), with the structure presented in [87]. The HF transformer guaranteed an electrical insulation between the storage and the rest of the system. Besides, this converter gave the possibility to use a low-voltage storage system.

5.1.2 Revision 1

To reduce the cost production and the number of components the DAB converter was abandoned. The control number was reduced from four to one. The device had only one board, where control signals and power coexist (Fig. 5-2).

The storage converter is a unidirectional buck during the charging period. The discharge happened trough a diode connected directly to the dc bus. Thus, the storage intervenes every time the dc bus voltage is lower than the dc storage voltage. This solution assures reliability (the storage is ready to sustain the dc bus), but gives low flexibility in managing the storage system.

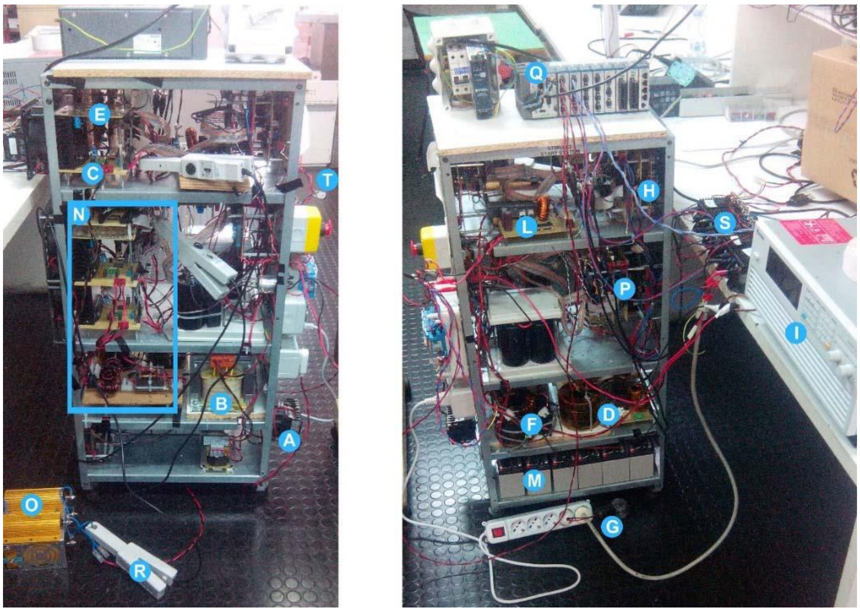


Fig. 5-1 Revision 0 of the nGfHA prototype

Using only one controller ensures simplicity and responsibility in the same time. All the information is available inside the same controller, so there is no need to interact with an external communication infrastructure. However, given all the information available the controller has the task to coordinate the converters. The high number of information inside the same controller make the task difficult. For this version a FSM was designed. However, this control architecture resulted in a difficulty in the design due to the high number of if-then-else nested structure, which makes the implementations of this structure for this device a time-consuming and an error-prone activity.

This experience leads to focus on different control architectures. And from now on the BT starts to become the preferred solution inside the project.

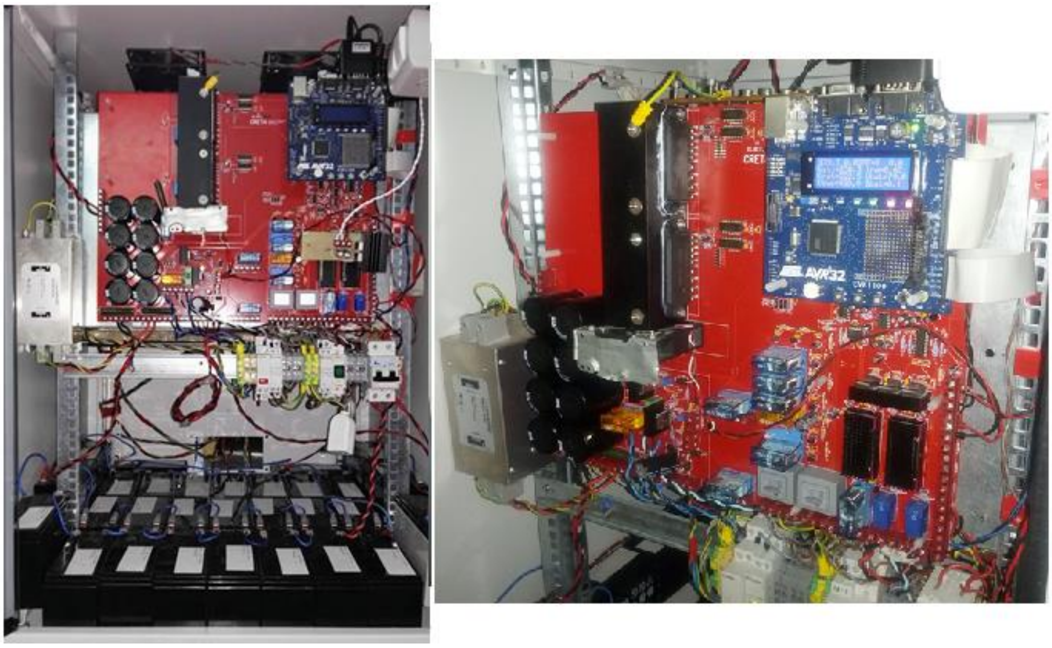


Fig. 5-2 Revision 1 of the nGfHA prototype

5.1.3 Revision 2

The second revision has two power control board (PCBs), one for the control and the other for the power. The device has a control board that allows to adapt measurements and signals to different powers. This structure has allowed the implementation of prototypes, for different applications, with rated power between 1 to 7 kW. The power limit is not introduced by the control board, but by the installed power switches, which are independent from the control board [75].

The storage converter is a bidirectional buck/boost converter. The nanogrid controller is governed by the first BT revision.

However, the interconnection of more NGs with this solution is not trivial. Thus, the next solution implements a DBS strategy to share power and to coordinate more NGs; this is possible thanks to the implementation of the second revision of the BT. Worth noting that this second BT has retro compatibility, so it can be implemented in some of the previous revisions.

Two implementations of this revision are shown in Fig. 5-3.

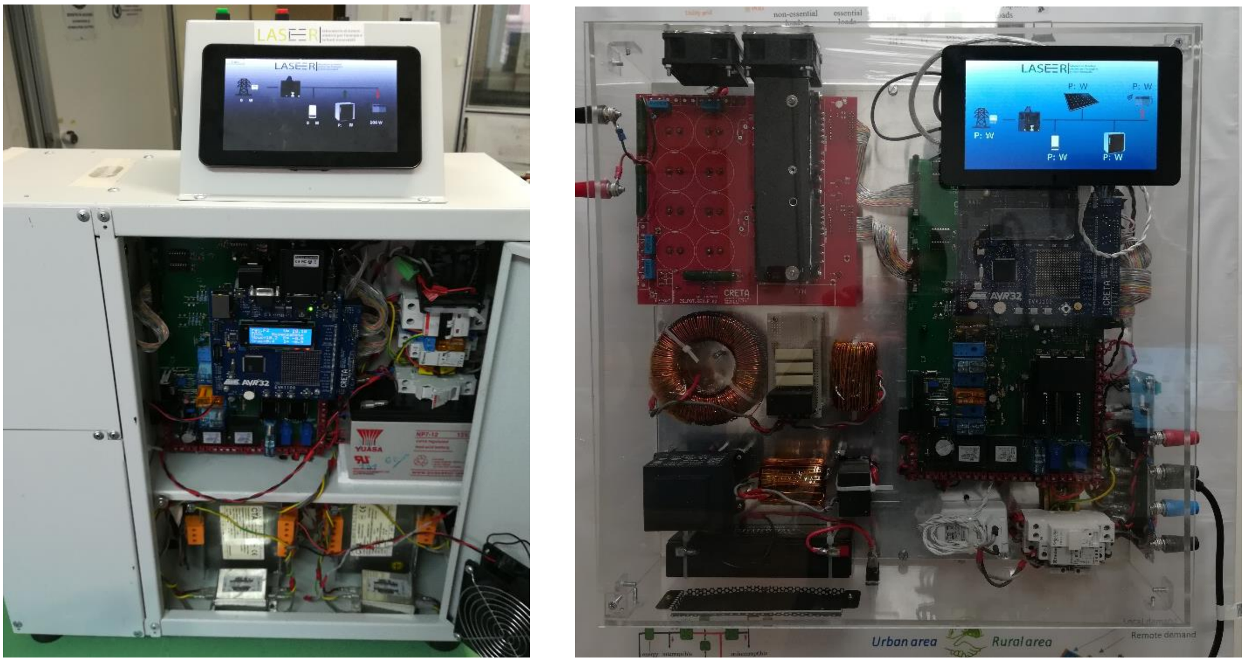


Fig. 5-3 Two implementations of the nanogrid revision 2. The right is a demo built to present the work in seminars and conferences

5.2 Setup description

The installation is realized in some spaces of a building allocated for student’s residences inside the campus of University of Calabria (Fig. 5-4).

The building has two PV plants of 4.5 kW each. Inside the building, one room is dedicated to the installation of the electrical system of these spaces, as shown in Fig. 5-5. From right to left, the setup is provided with:

- A supervisory control and data acquisition system;
- A 22 kWh lithium-based storage system;
- A conventional three-phase PV inverter connected to the first PV plant and directly to the utility grid;
- An electrical protection panel with motorized switches and two energy meters;
- Three nanogrids connected to the same dc bus;
- A cogeneration unit, realized with a stirling generator inside a pellet boiler.

Fig. 5-4 View of the building where the system is installed

The critical loads are not in this room, but are 30 m far from the electrical panel. The acquisition system is realized in LabView using a compactRio for electrical variables. The supervisory and data acquisition system has been realized in Qt, running on a local PC, for the information gathered from the nanogrid supervisors and the other objects of the installation.

Fig. 5-5 The setup inside the building of the University of Calabria

The three nanogrids are installed in the electric panel, as shown in Fig. 5-6. The first nanogrid NG_1, positioned in the upper shelf, is the only one nanogrid grid-connected; the inverter has a rated power of 7 kW. The NG_1 has also a PV plant of 4.5 kW.

The second nanogrid NG_2, positioned in the middle shelf (not highlighted), is used to interface a storage system and the inverter to supply ac critical loads. The storage system is a 232V/22kWh lithium battery (Fig. 5-7). Two clusters are parallel connected. The battery storage has an internal BMS and a log unit to collect data. The BMS and the NG_2 exchange information through the nanogrid supervisor, using a ModBus TCP connection. The ac critical loads are server and communication devices, which are fundamental to provide connectivity and services inside the building. These loads are in another room, as shown in Fig. 5-7 .

The last nanogrid is used to control the Stirling generator inside the boiler. The rated power is 1 kW. At the present, this is the only task for the NG_3.

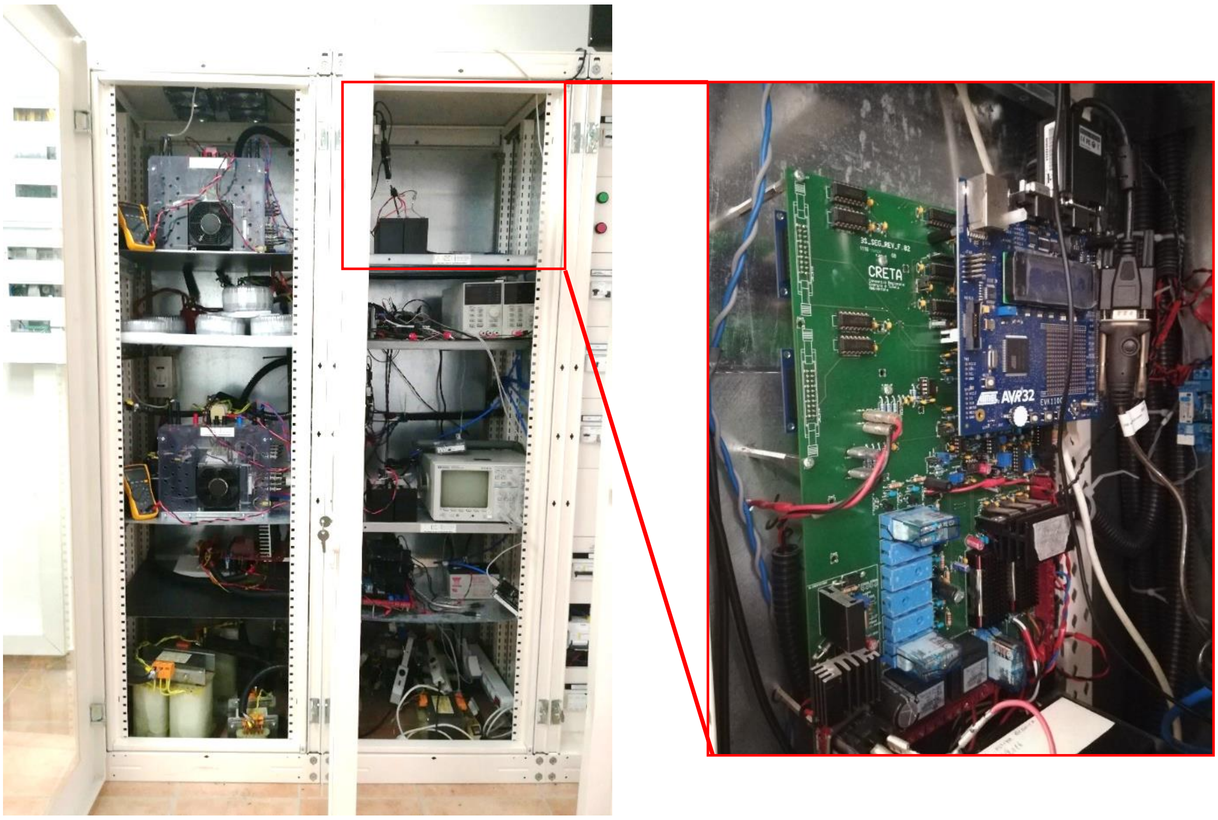


Fig. 5-6 Electrical panel with the installation of the three nanogrids. Only the first nanogrid control board is highlighted (right).

Fig. 5-7 Battery storage system (left) and critical loads (right)

The wiring diagram of the setup is shown in Fig. 5-9.

The classical PV system is directly connected to the grid. Also the not-essential loads are connected directly to the utility grid; in case of a grid failure the non-essential loads are not supplied.

The essential loads are supplied from a second ac-bus. When the system works in islanded the essential loads can be supplied by the grid-inverter of NG_1, which changes from current to voltage control. The other two NGs don't have the connection to the utility grid.

5.3 Experimental results

In this section, the functioning of the system is described from the black-start. Each NG runs the software presented in the previous chapter. The second revision of the BT is the algorithm implemented in the NG controller. The DC-bus signal is adopted to exchange information between NGs. The voltage on DC bus is between 340 (T1) and 415 V (T6). The voltage difference between each threshold is 15 V. However, to jump from a threshold to the other a hysteresis of 5 V has to be crossed.

The power flows inside the system are summarized on a local dashboard, specifically developed. The dashboard collects information from the NG supervisors, from the traditional PV inverter and from an energy meter installed on the electrical panel. This application runs on a local server. The dashboard is shown in Fig. 5-8. The electrical system of the building is three-phase, such as the traditional PV inverter. The AC loads are divided between the three phases. Instead, the grid inverter of the NG1 is connected to the L1 phase. The "AC loads" in Fig. 5-8 are the non-essential loads in Fig. 5-9.

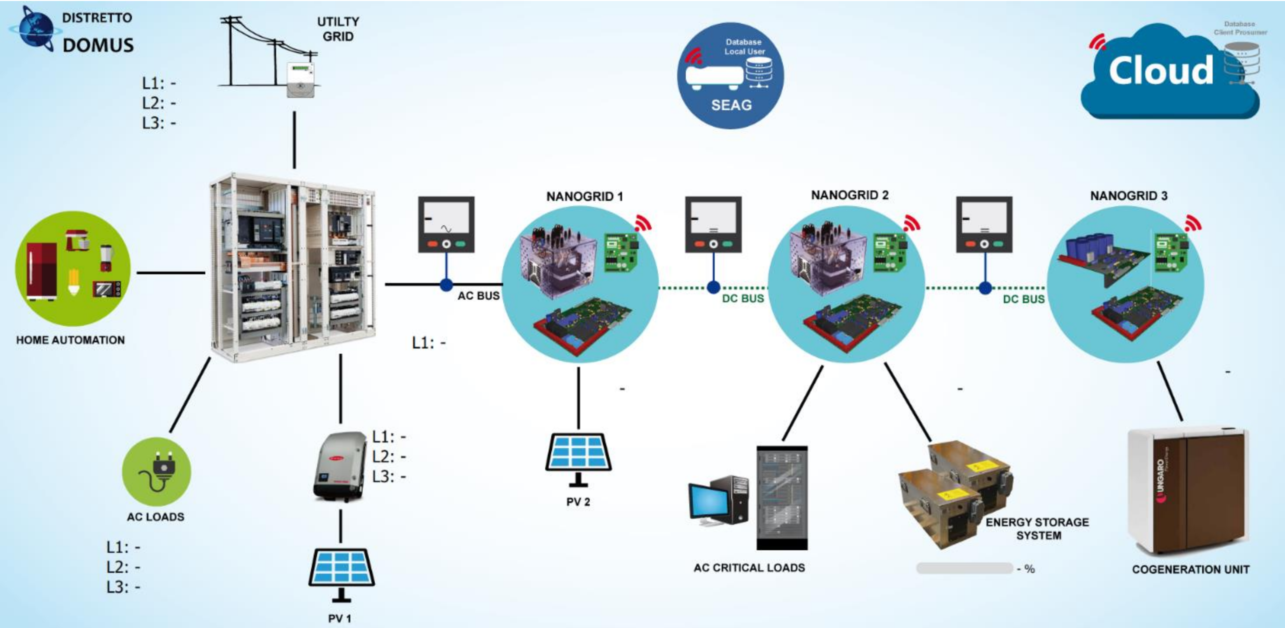


Fig. 5-8 Graphical User Interface of the local dashboard developed to describe the power flows inside the system

A. STEP 1

All the NGs logic is turned on. Each NG waits for the connection of a source that allows the dc-bus charging. To show the stand-alone and the black-start capability, the storage system is firstly connected. The storage system has its own BMS that interact with the NG controller thanks to the nanogrid supervisor. The storage presence starts the charge of the dc bus in two steps: firstly, the circuit breaker is closed and the converter is disabled. After this transient, the converter is enabled and the voltage is regulated to the reference value of the second threshold (T2), that is 355V (Fig. 5-11). The utility grid is disconnected. The connection between the utility grid and the NG_1 is manually governed.

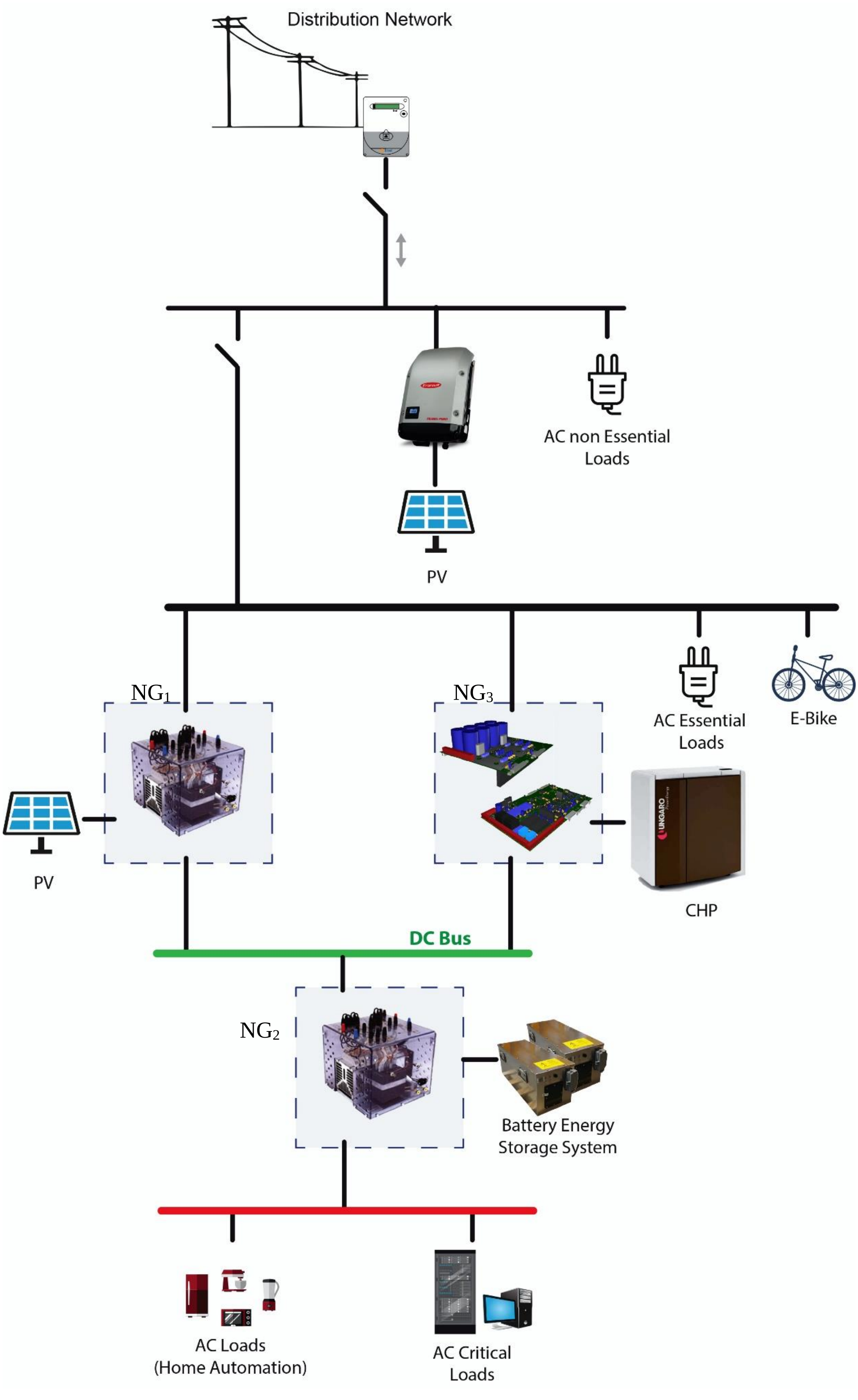


Fig. 5-9 Wiring diagram of the electrical setup: in the presented results the home automation loads of NG_2 and the connection to the grid of NG_3 are not considered, but are shown for further information

When the other two NGs detect the dc bus voltage, they start up too. The storage provides the power for all the systems. The control logic of the stirling ECU and the hydraulic circuit of the cogeneration unit are the only loads for the system, as shown in Fig. 5-10. The traditional PV inverter feeds into the utility grid 2850W. The “ac loads” are the loads of the buildings as lights, office equipment, and other generic load not controlled by the system. The loads controlled by the automation system are temporally turned off, so these loads don’t require power.

The net electrical power exchanged with the utility grid is given by a power meter inside the electrical panel. In Fig. 5-10 the power of the traditional PV inverter is fed into the utility grid. The net power to the grid is the produced power minus the power used by the ac loads.

When NG_2 ends the start-up procedure, the converter used to supply the critical loads starts its operation, as shown in Fig. 5-11. However, the critical loads are not connected yet.

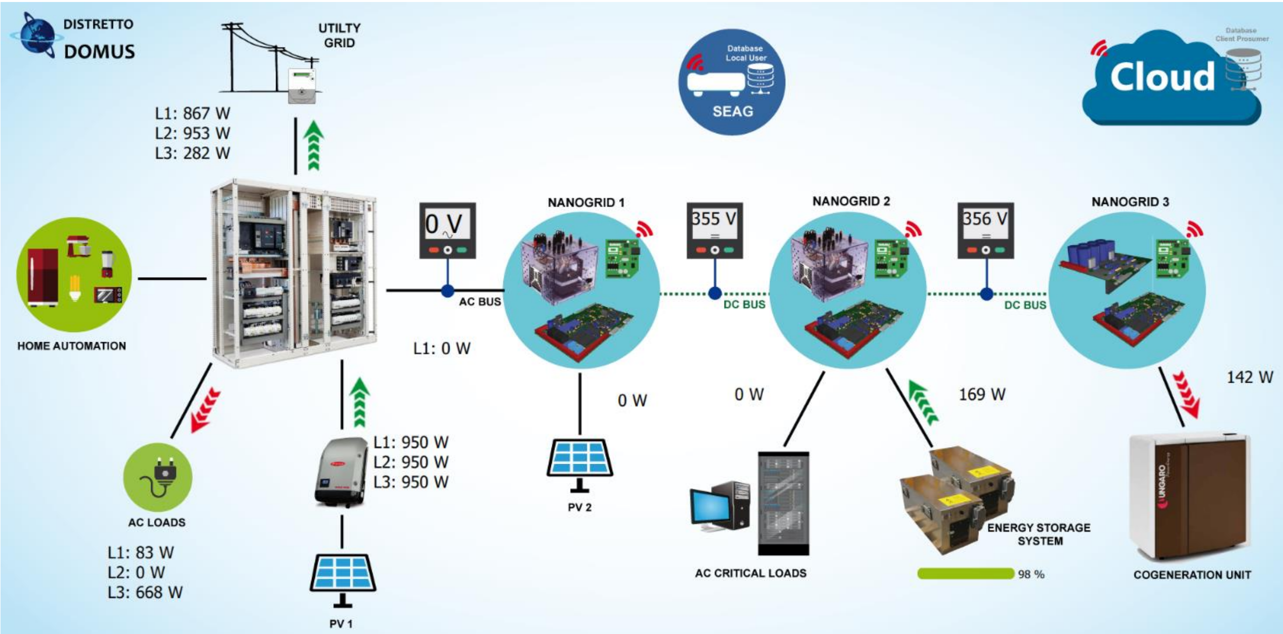


Fig. 5-10 The system is started as a stand-alone system: the storage is the only source to energize the dc bus (T2).

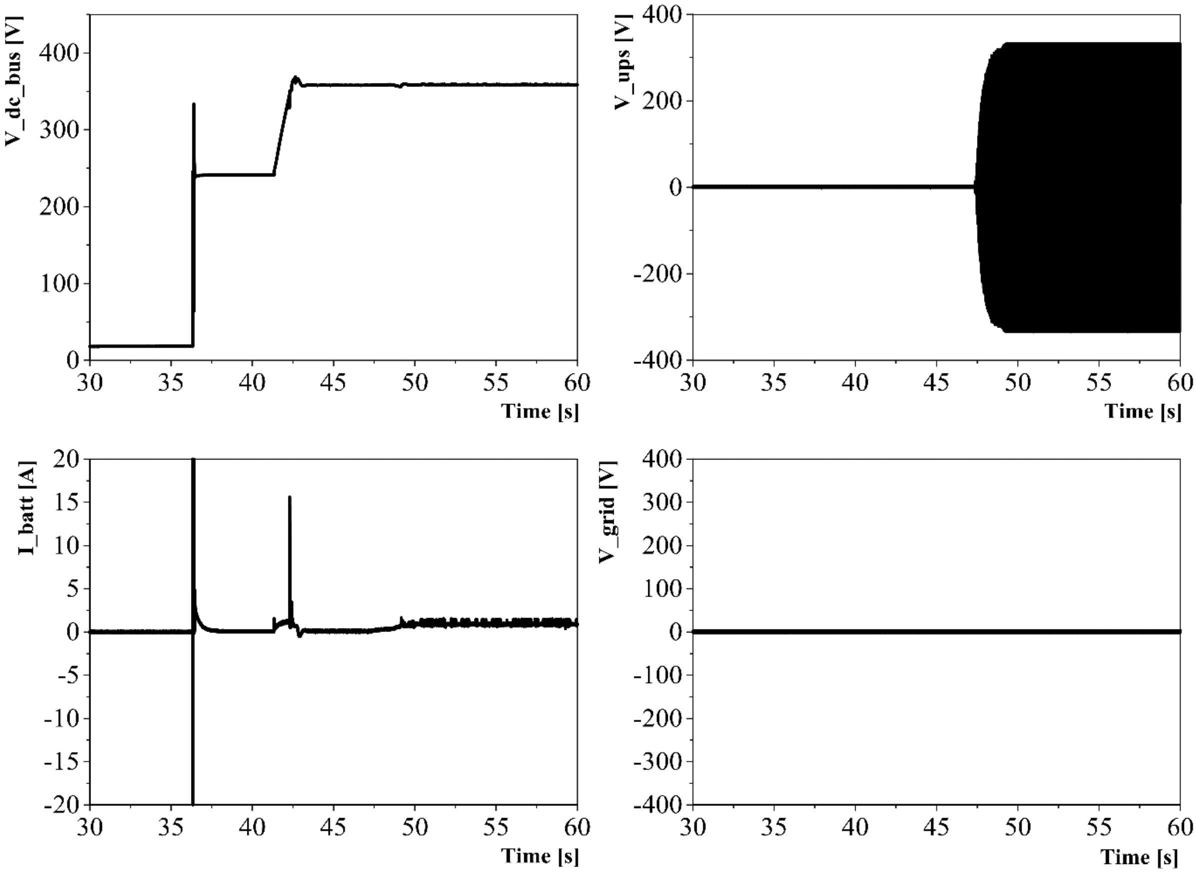


Fig. 5-11 Starting procedure from the local storage: dc bus voltage (top left),ups voltage (top right), grid voltage (bottom right), battery current(bottom left)

B. STEP 2

The NG_1 is manually connected to the utility grid. After about 5s the grid inverter takes control over the dc bus and the threshold T3 is established to the reference value of 370V. From the T2 threshold the behaviour tree commands the grid inverter to start the voltage control. The jump of the grid controller state from a stand-by to voltage control cause an overshoot of 5V. When the jump is detected the converter of the storage system switches from voltage control to stand by, so the current goes to zero. The reference voltage for T3 is 370 V. The batteries are now in standby (Fig. 5-13).

In this condition, the power for the three NGs is absorbed from the utility grid. The power is absorbed from the L1 phase. The traditional PV inverter feeds in 948 W for each phase (Fig. 5-12).

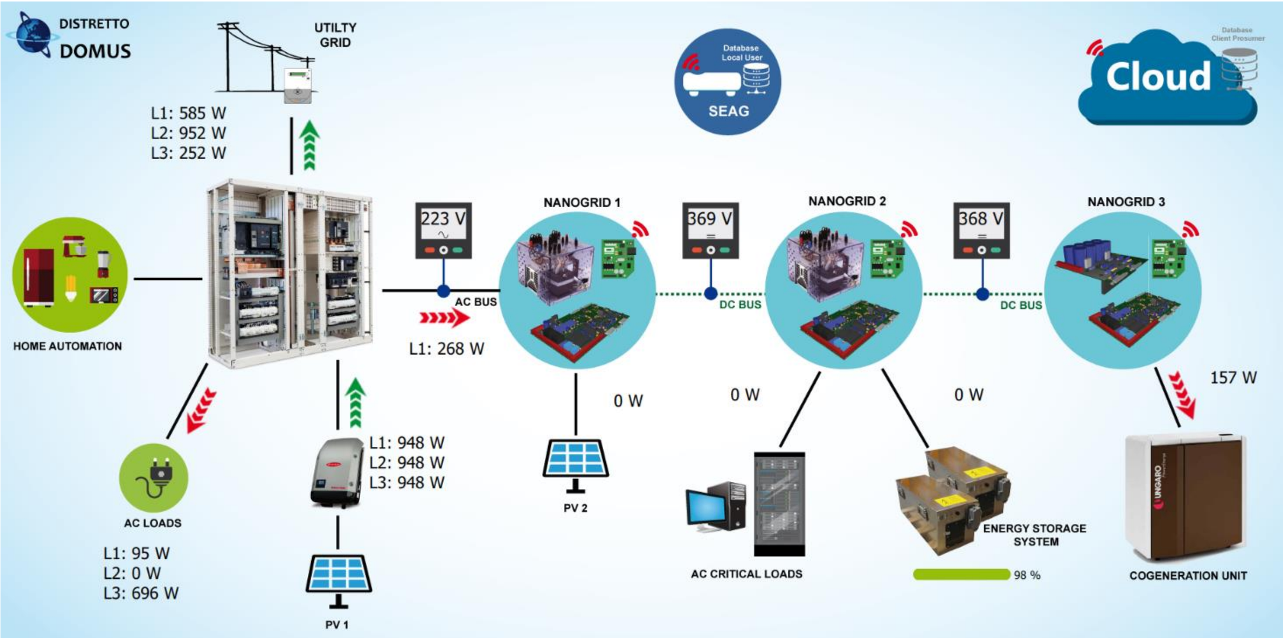


Fig. 5-12 The power from the three NGs is absorbed from the ac utility grid (T3).

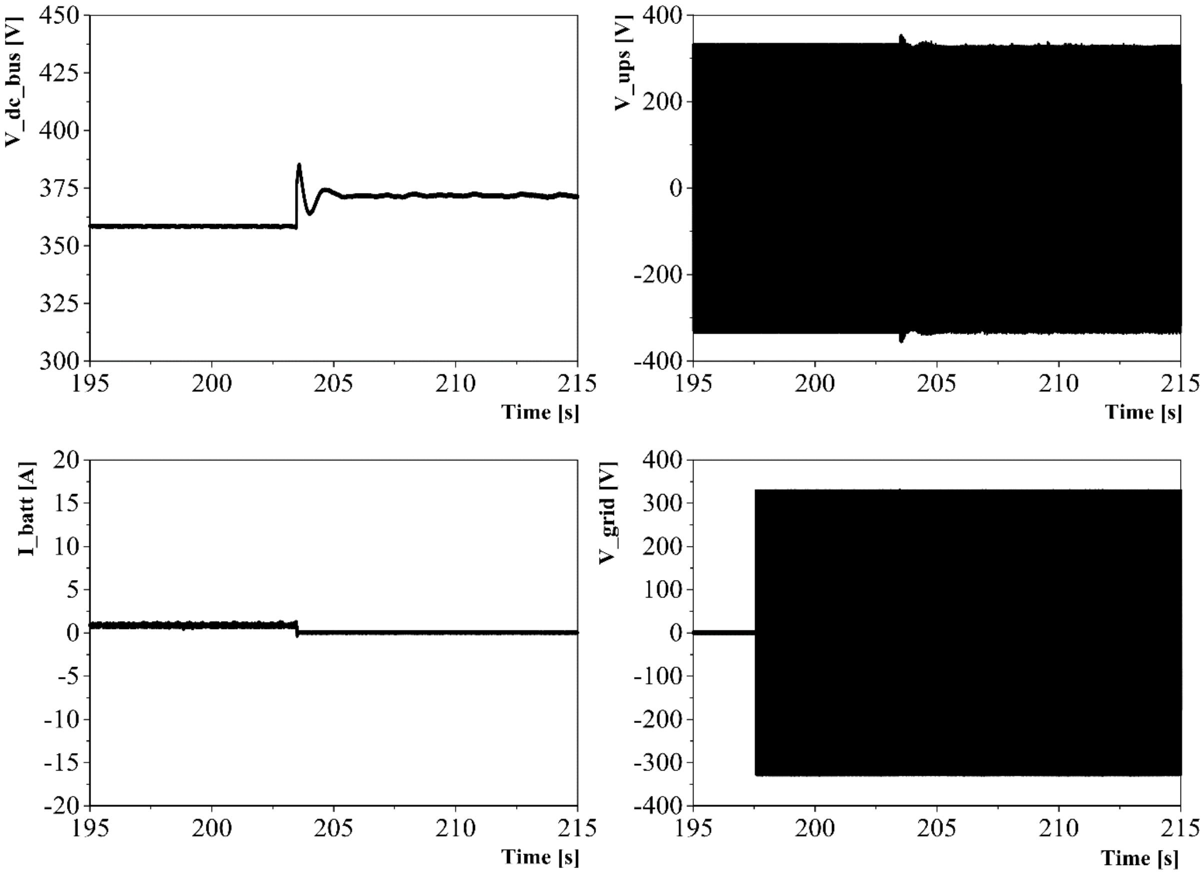


Fig. 5-13 Manual connection of the grid: the dc bus voltage goes from T2(355 V) to T3(370 V)

C. STEP 3

The system is working in T3. The PV plant is now connected. The PV plant has three strings connected in parallel; all of them are connected. The NG_1 controller, when detects the PV connection, closes the breaker switch and start the MPPT algorithm. The permanence in this threshold is now related to the PV production. If the PV power is greater than the load inside the three NGs, then the dc voltage will start to increase, until another threshold is triggered. In this test, the next triggered threshold is T5 (Fig. 5-15). After the new threshold is triggered the PV production increases, as can be seen from Fig. 5-15; to note the increasing of the dc bus oscillations due to the single-phase configuration of the inverter.

The PV production is 2766 W, while the grid inverter of NG_1 feeds into the utility grid 2306 W (Fig. 5-14).

The threshold T4 has not been triggered when the operating state transitions from T3 to T5. The T4 is the threshold in which the storage converter absorbs power from the dc bus as master converter; however, the trigger of this threshold is related to the SOC of the storage system. The SOC is equal to 98%; according to the initial setup, the BT decides that is unnecessary to charge. Obviously, the minimum SOC value to start the charge can be configured according to the user needs.

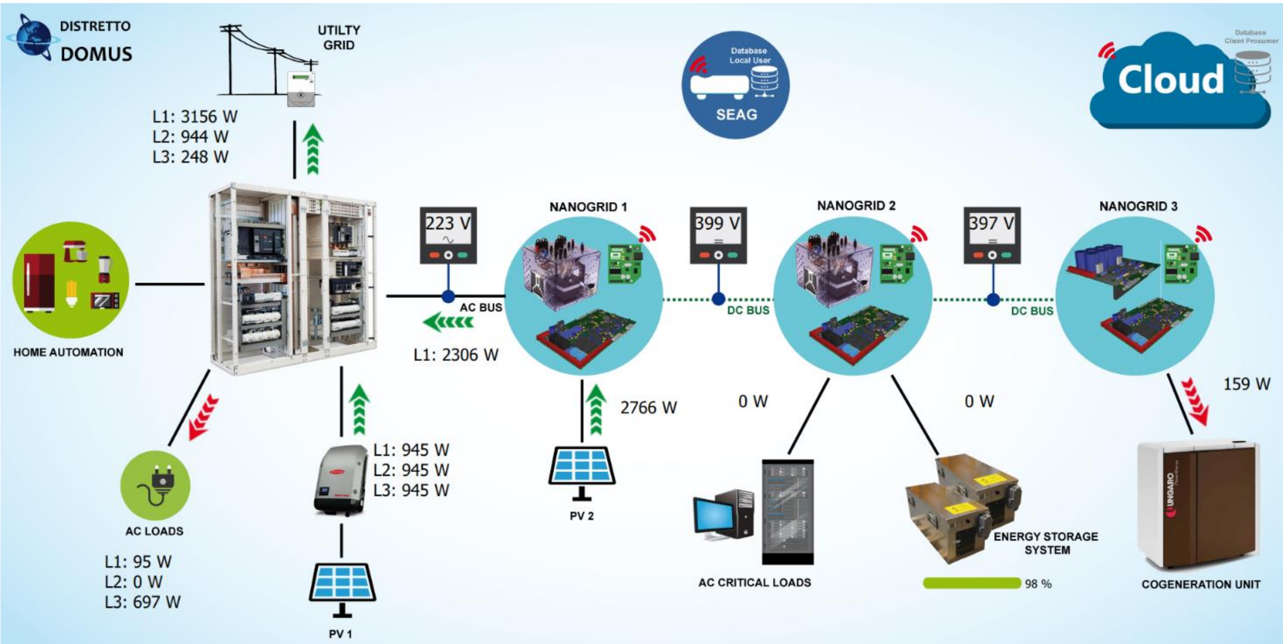


Fig. 5-14 The NG1 feeds into the grid, on the L1 phase, 2306 W (T5)

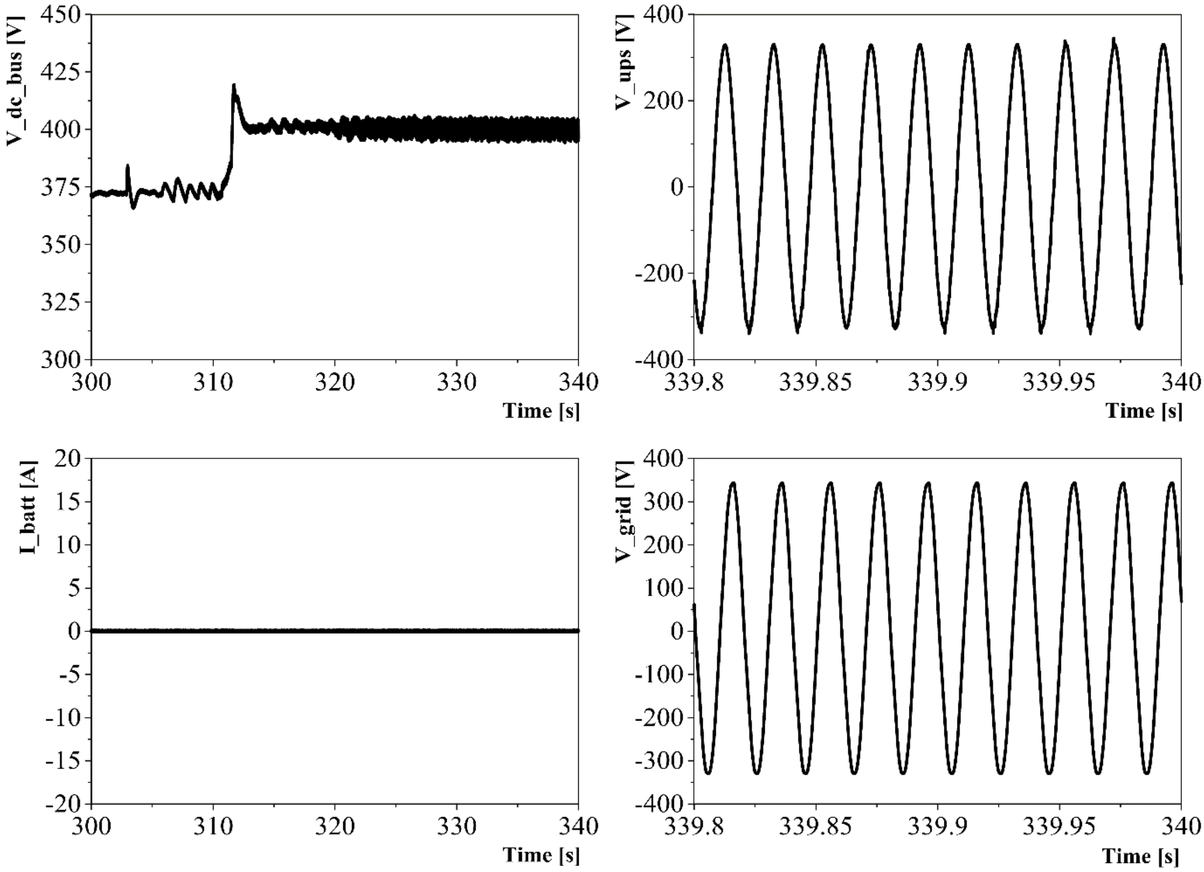


Fig. 5-15 Manual connection of three strings with an overall rated power of 4,5 kW. This event changes the threshold from T3 (370 V) to T5 (400 V)

D. STEP 4

Now, let's suppose there is a sudden reduction in the PV production. This situation is simulated manually opening two of the three strings connected in parallel. As a consequence, the production decreases from about 2.8 kW to 0.9 kW. Obviously, also the power given to the utility grid is reduced of about one third.

Now, suppose the DSO, or the other entity enabled to command remotely the system, ask to NG_1 to exchange 1000 W with the main grid. However, the PV production is lower than 1000 W and there are local loads and losses that need to be taken in account. The NG1 is unaware of the other sources in the system, as the storage system. However, it tries to feed 1000 W into the grid, if the dc bus voltage drops under the lowest threshold, then it stops to follow this command and will try after 5 minutes.

The BT changes the controller of the grid inverter from voltage control to power control, as a consequence the voltage drops. After a short transient, influenced by the manual operation of disconnecting the strings, the system starts to work in T2, where the storage is the master converter. The storage, with SOC equals to 98%, can give power to the system. Thus, the PV works in MPPT, while the battery converter gives the difference between the PV production and the power fed into the grid plus the local loads and the losses (Fig. 5-16).

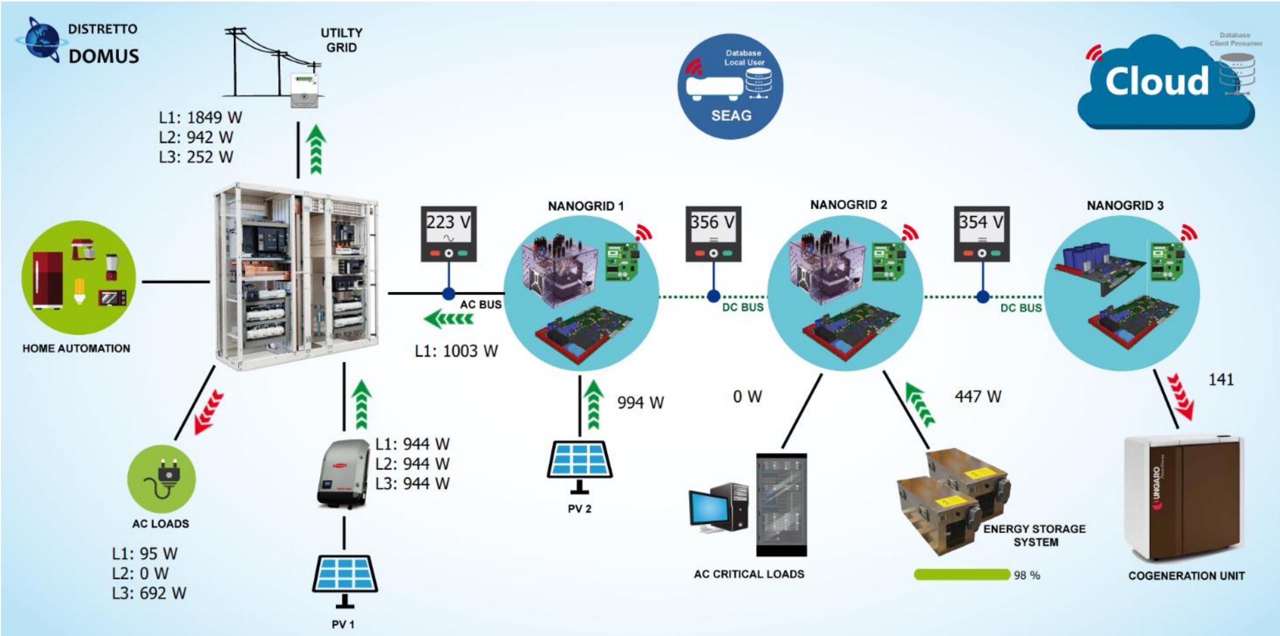


Fig. 5-16 NG1 receives the power set-point to feed into the grid 1000 W. Since the PV production is not enough the difference is given by the storage system

To show the capability of the system to follow remote commands the set-point is changed. The set-point is manually changed; this can be done in different ways thanks to the versatility of the nanogrid supervisor. For example, it can read the set-point on a remote server [88]. The sequence of the command given is: 1000 W, 2000 W, 500 W and 100 W (Fig. 5-17 a,b,c,d). From Fig. 5-17 it is possible to note that to a command change there is a correspondent change in the battery current; this is due to the master converter role of the storage. Worth noting that the grid inverter doesn't know any information about the storage, however it can take advantage of its presence, thanks to the distributed control. The last command is to reduce power from 500 to 100 W. This produces a state transition. Indeed, the PV production alone is higher than the local request and the storage doesn't need to be recharged. The implemented behaviour is to stop the power control of the inverter, giving priority to the extraction of the maximum power from the PV. This behaviour can be easily changed, according to the specific user needs. In this implementation, maximize the PV production has higher priority than following the remote command. This choice produces the jump from T2 to T5.

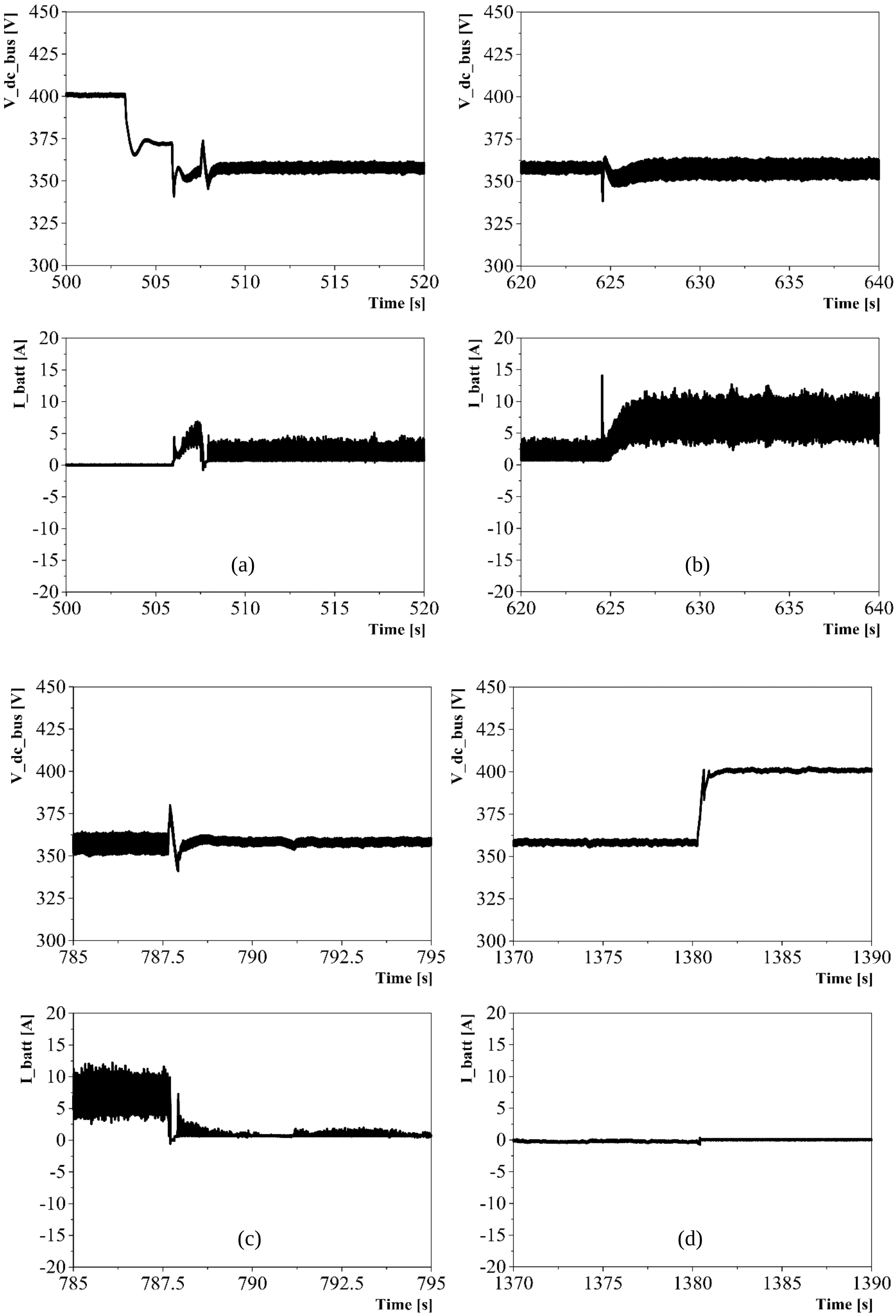


Fig. 5-17 Manual disconnection of two strings and remote request of 1000 W (a); 2000 W (b); 500 W (c); 100 W (d).

E. STEP 5

In this step, from the last condition, the critical loads are connected and the boiler with the stirling generator is activated. The critical loads are about 600 W, although this value is not constant. The cogeneration unit inside the boiler needs a relatively long time to start to produce electrical and thermal energy. Indeed, the electrical power generated from the stirling is not practically controllable, but it will be determined by the boiler. The boiler has a variable load during the start-up, with a mean power of about 400 W and power spikes over 1000 W.

To prove the resilience of the system against a sudden event, the grid is manually disconnected. Because the power produced is greater than the local demand the disconnection from the utility grid causes the dc voltage increment and the threshold T6 is activated (Fig. 5-19). In this threshold the PV production is modulated. Obviously, the classical PV inverter is now off and the PV production is terminated, while the NGs can still take advantages from the solar irradiation. This ensures a stand-alone capability.

Now, let's try to disconnect also the PV plant. The boiler and the critical loads are still supplied (Fig. 5-19). The voltage decreases until the threshold T2 is activated and the storage takes the responsibility to control the dc bus and give the needed power to supply all the loads. This condition is maintained until the SOC of the storage system is higher than 20%. The storage is the only source in the system and gives the power needed for critical loads and boiler (about 1.1 kW).

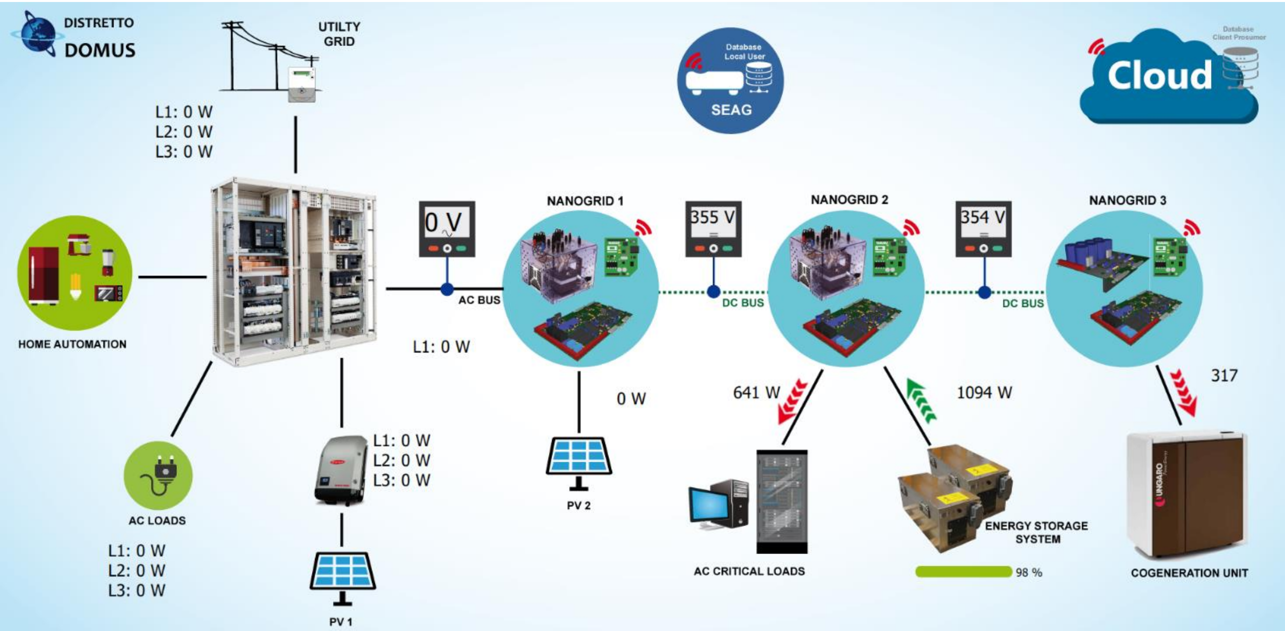


Fig. 5-18 After the manual disconnection of the utility grid and the PV plant the storage supplies the critical loads and the cogeneration unit

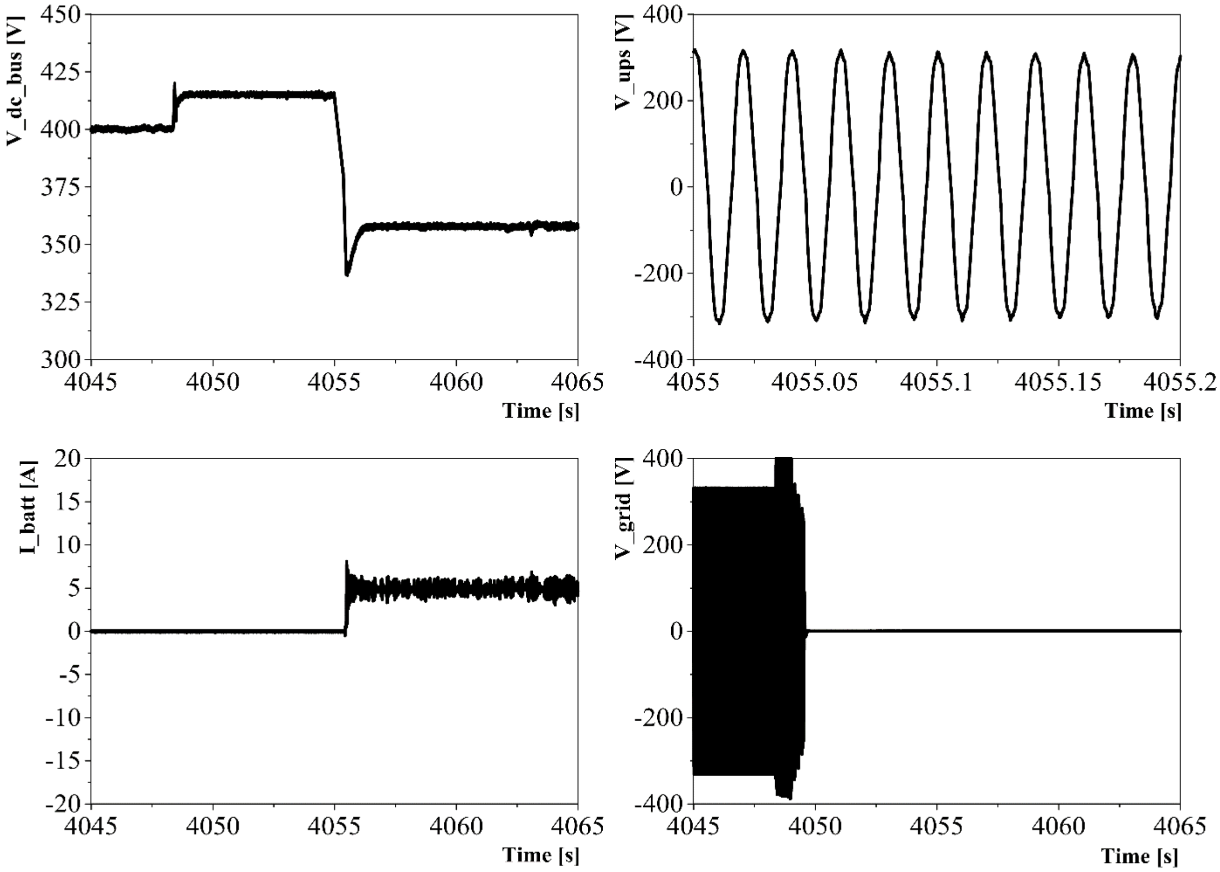


Fig. 5-19 From the threshold T5 after the main grid disconnection the T6 threshold is established; when also the PV plant is disconnected the threshold changes to T2

F. STEP 6

The charge of the battery can start only if the SOC is lower than 90%. To see how the storage behaves when it is not empty, the grid is restored to discharge the storage. The remote request to feed into the grid 3100 W is provided, to accelerate the discharge. For a period, the PV connection is not restored; this is done to use only the power from the storage system. Also the starting of the stirling generator is delayed, to allow the fast discharge of the storage system.

When the SOC is 87% the PV connection is also restored. Now, the power fed into the grid is the sum of the PV production and the storage system (Fig. 5-20).

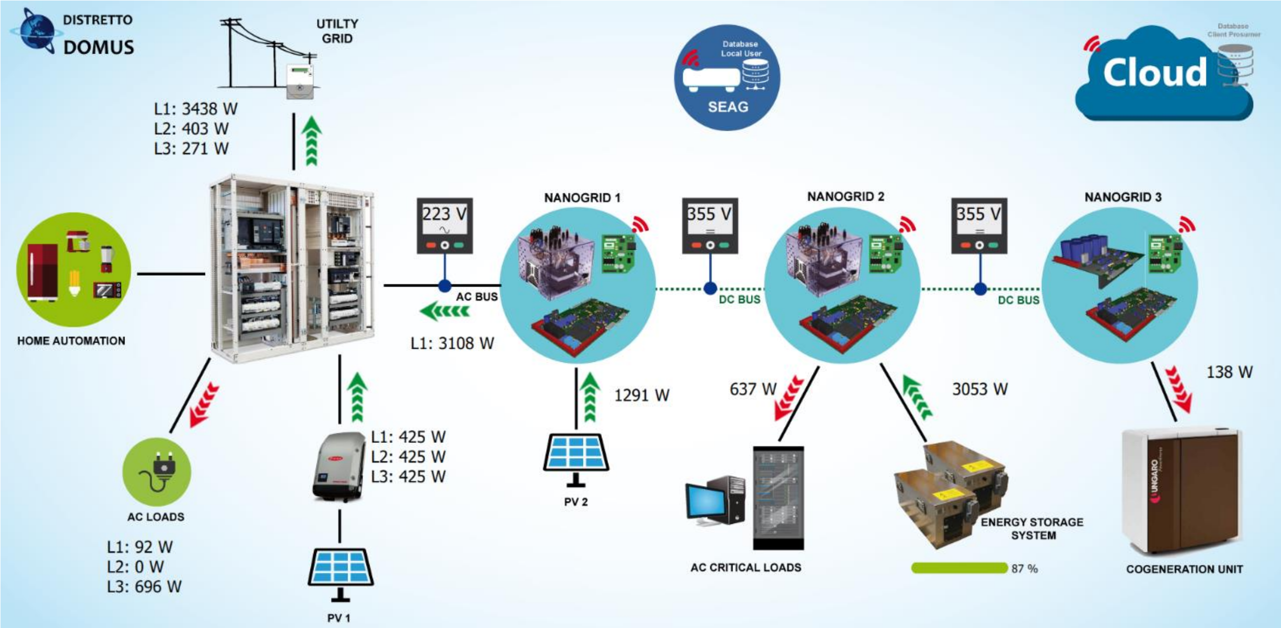


Fig. 5-20 The storage system is discharged giving the command to NG1 to supply 3100 W into the grid

G. STEP 7

The remote command is reduced from 3100W to 0W. Because the PV production is higher than the request the dc voltage increases and T4 is triggered (Fig. 5-22). Now, that the SOC is under 90%, the power surplus is used to charge the storage system. The power with the grid is controlled to zero. There are two loads: the critical loads and the boiler that is warming up. The sum of the power required is 686 W, while the PV is producing 1397 W. The difference is stored in the batteries, less losses (Fig. 5-21).

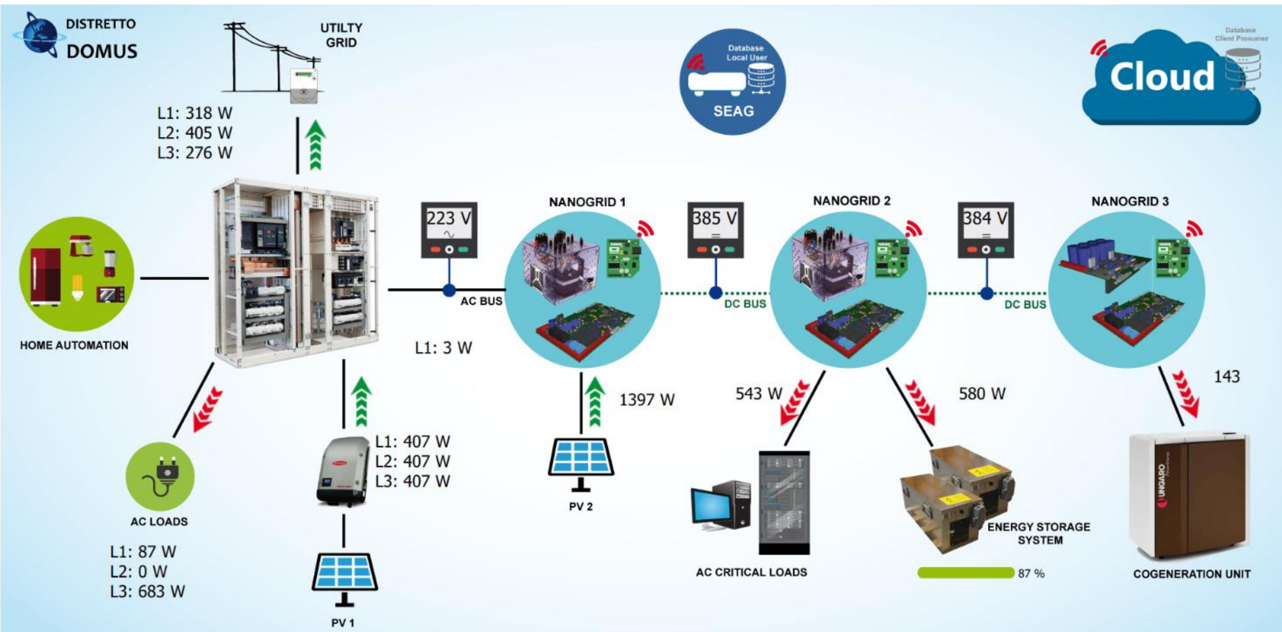


Fig. 5-21 The remote request is set to 0 W. The surplus of the PV production is stored in the storage.

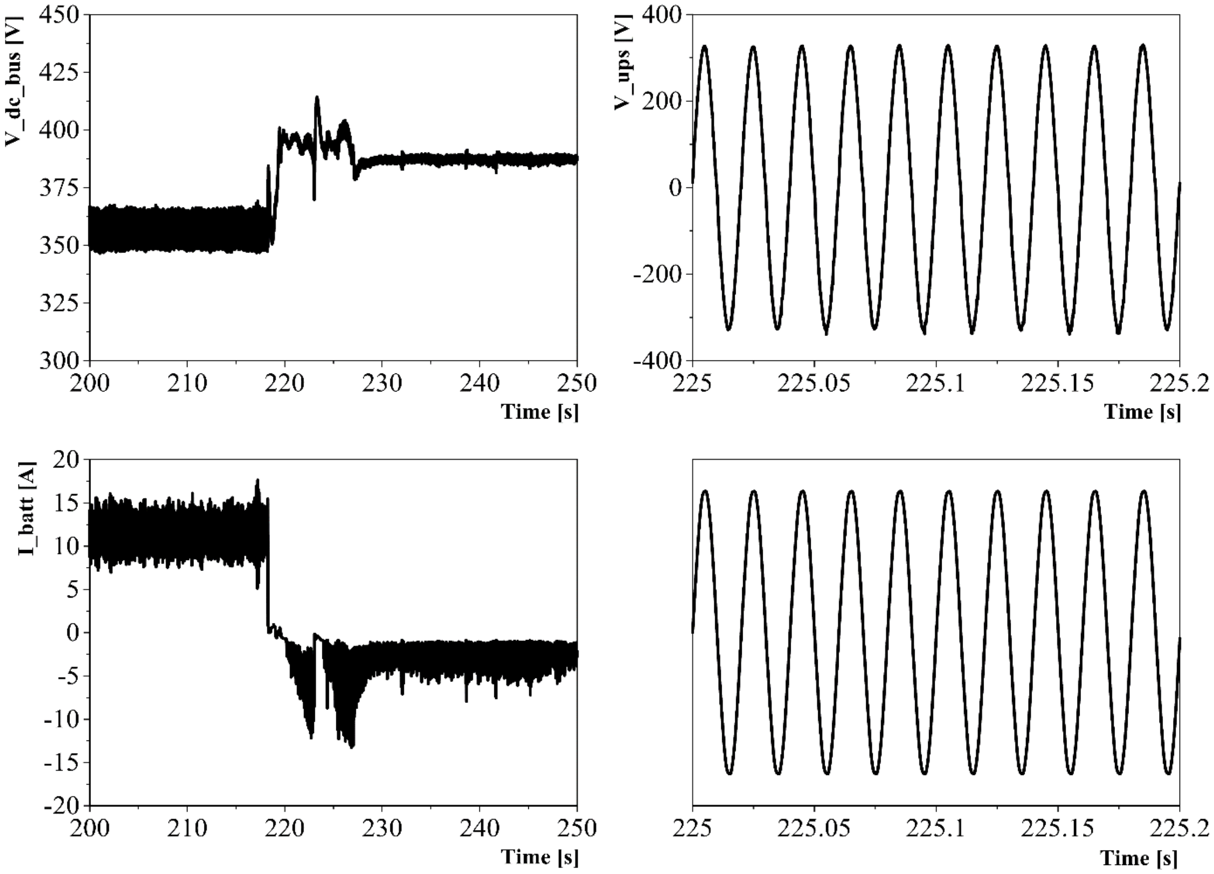


Fig. 5-22 When the request is changed from 3100 to 0 W, because the PV is producing and the SOC is under 90%, the threshold T4 (385 V) is triggered.

H. STEP 8

Until now the power from NG has been fed in the grid, following a remote request. Obviously, the grid inverter is bidirectional, so it can be used to absorb power from the grid. Thus, a new request, sent to the nanogrid controller, is to absorb from the ac utility grid 2000 W. From the previous step the power stored in the storage increases from about 580 W to 2204 W, as shown in Fig. 5-23. The current of the batteries increases consequently (Fig. 5-24).

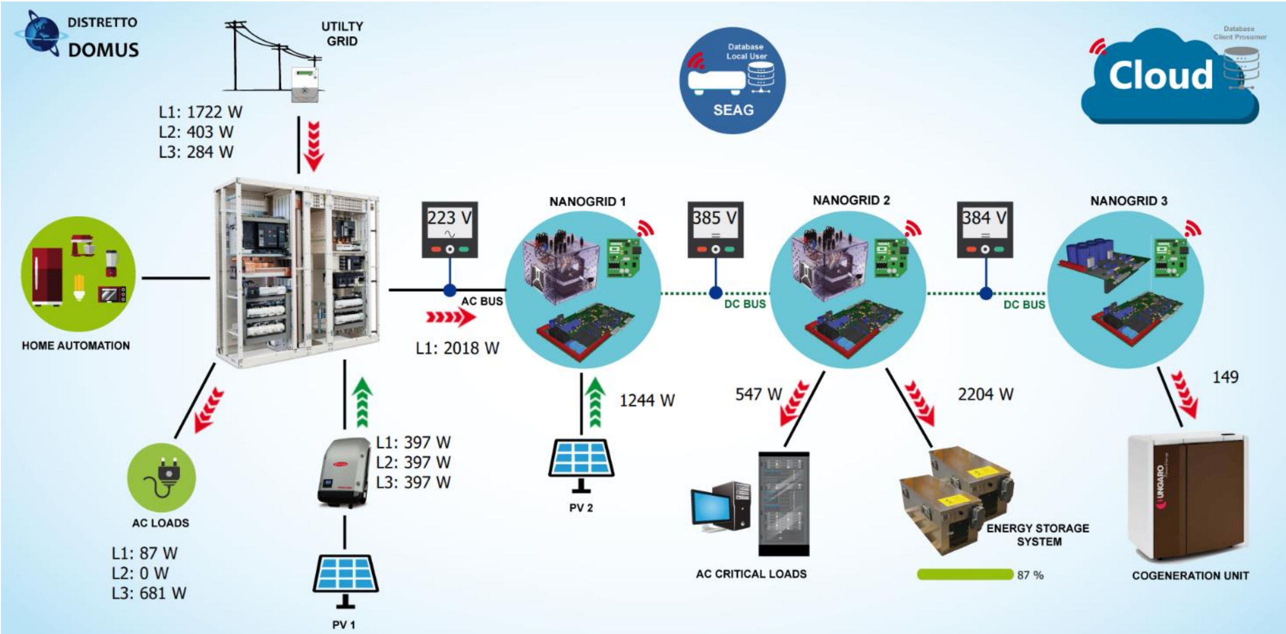


Fig. 5-23 The grid inverter follows the command to absorb 2000 W from the utility grid

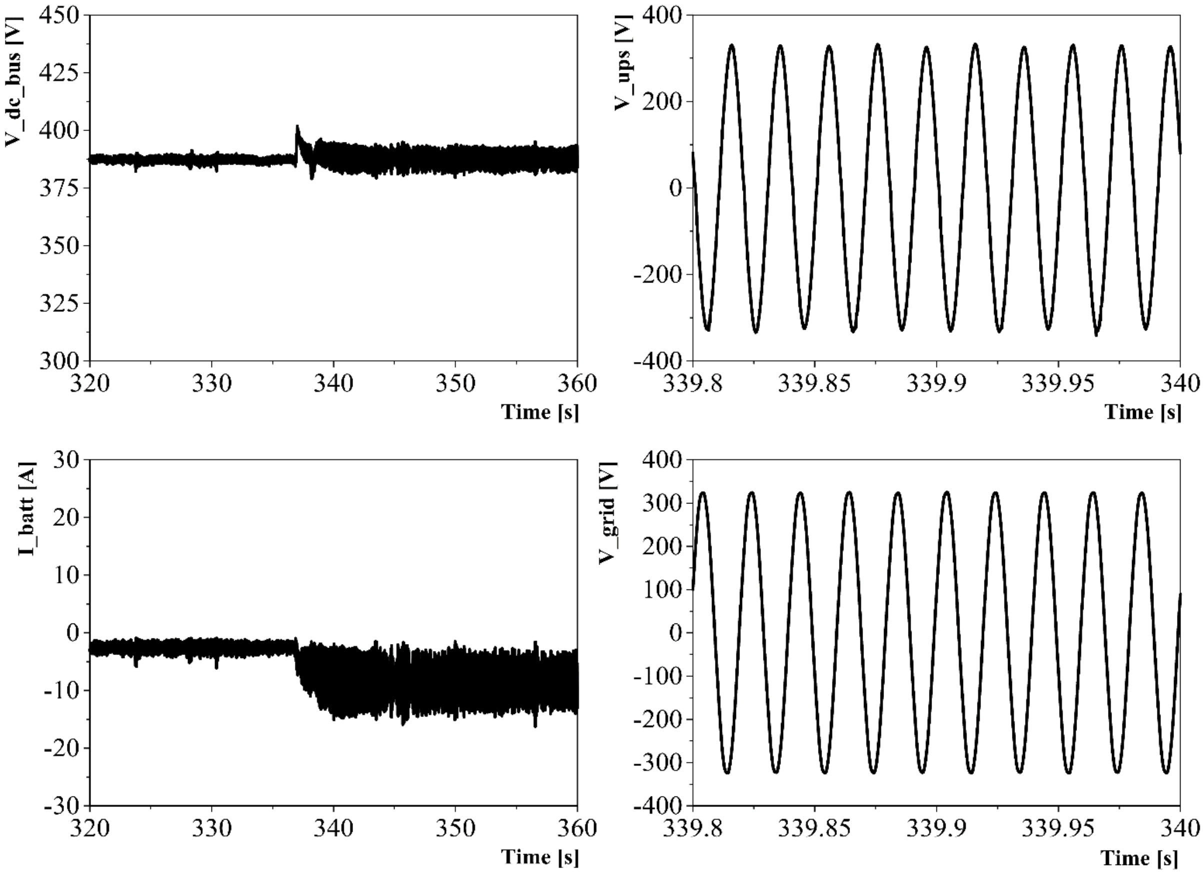


Fig. 5-24 Following the request to absorb 2000W the battery current increases. The threshold doesn't change.

I. STEP 9

In the last step the same situation is preserved. Meanwhile the condition changes because the stirling starts to generate electrical power. This power increases gradually, with a maximum generated power of 1000W. This power is also limited by the temperature of the hydraulic circuit. The generated power is used to increase the power stored in the batteries (Fig. 5-25).

The power stored can increase until the maximum admissible current for the storage, determined by the BMS, is reached. If the current needed to control the dc bus at the reference voltage is greater than the maximum current the converter saturates the current. As a consequence, the dc bus voltage will increase until when a higher threshold is activated.

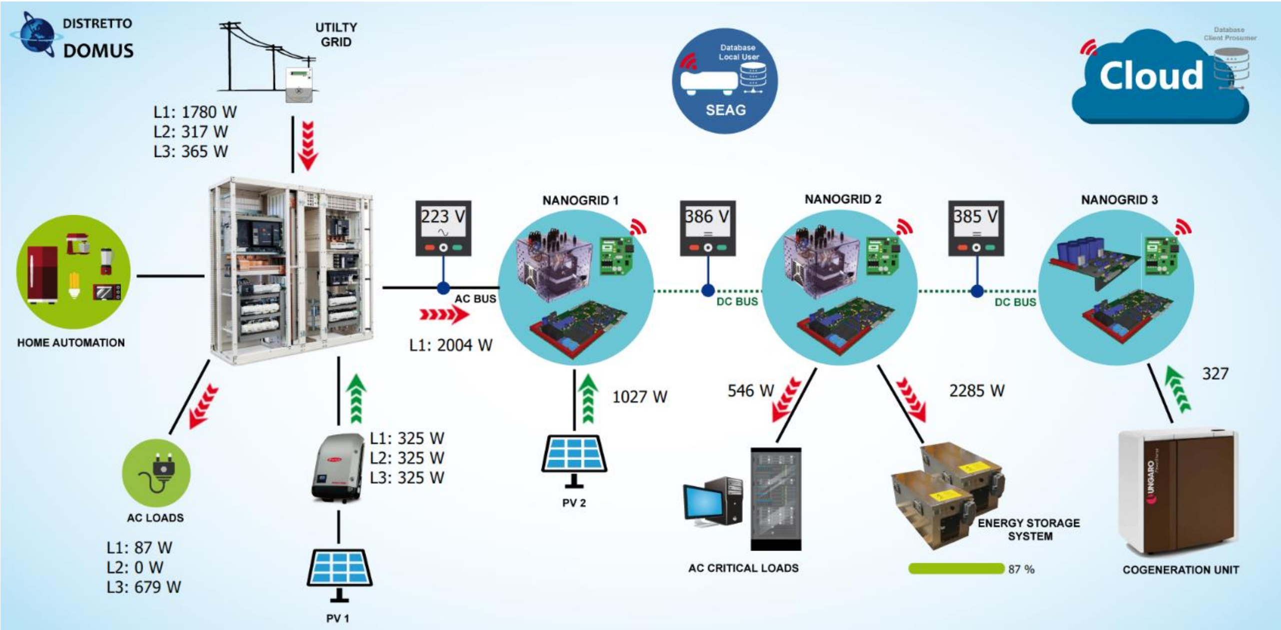


Fig. 5-25 The stirling generator starts and contributes to the local production

6 *Conclusions*

A nanogrid is a power distribution system for a single house or small building. In order to enable zero net-energy consumption and optimal power management for future homes or buildings, the dc nanogrid can integrate multiple renewable energy sources. However, the integration of all sources and loads in a simple, reliable and smart way is still challenging.

In this thesis a control system for a nanogrid has been developed and implemented in a real-life application. The activity aspires to boost the technology in both academic and industrial sectors, in order to spread the use of the nanogrids in the domestic environments. This work wants to be the proof-of-concept needed to attract the attention of industry, which is important to start the development of this new technology.

The developed nanogrid control system realizes a nanogrid:

- Flexible: the nanogrid optimizes the available sources, while manages different configurations and combinations;
- Modular: the nanogrid can easily adopt new sources and new behaviours;
- Reliable: the nanogrid ensures the safe operation in all possible conditions;
- Robust: the nanogrid is able to avoid failure regions of the state space, i.e. avoid dangerous and critical situations;
- Autonomous: the algorithm knows what to do in case of a sudden event without incurring in random, unnecessary and dangerous actions, while optimizes the available sources;
- Easily installable: the nanogrid is easily installable, thus cheap and simple to setup, also in an already existing power system. This feature is important because users are resistant to changes and reluctant to high investments.

The nanogrid system is an autonomous entity able to follow some predefined behaviours. The behaviour is the set of actions that the nanogrid adopts/implements/applies to optimise the power production, to exploit renewable energy sources, to supply continuously critical loads and to react to any sudden exogenous event that perturbs the physical system.

The sequence of the actions to apply are decided during the decision-making process. The actions are applied to power converters, the nanogrid building blocks. The power converters interface power sources and loads to the common dc bus of the nanogrid. The nanogrid controller controls the nanogrid behaviour according to the specified goal, selecting the operating mode for each converter. The high number of converters and the combinations of all possible operating modes of each converter ensure to put in place advanced control strategies, but at the same time complicate the control algorithm.

The developed nanogrid control system realizes the future home power system. The effectiveness and the validity of the developed control system for the nanogrid derives from two points:

- The theoretical background, the formal analysis of the system and the explicit design of the nanogrid decision-making process;
- The experimental results and the implemented prototypes prove the validity of the proposed control system.

In almost all the engineering activities some simplifications, under some hypotheses, are accepted in order to model a process. The simplifications eliminate some information, but ensure a simpler model. This simple model is used to analyse the system and to design a controller.

Most of the time, when designing a controller, the discrete side of the hybrid system is disregarded and only the continuous side is considered. This simplification is acceptable if the system control requirements are simple or if the practical implementation in a real physical system is not faced. Indeed, architectural design principles are often disregarded and when the implementation stage arrives, a negative choice of the architecture can significantly impede the system development.

The nanogrid system has been modelled like a hybrid dynamical system. The feedback controllers evolve in the continuous state space; the decision-making process evolves in the discrete state space. The careful design of the discrete control allows to the nanogrid controller to become an intelligent entity. This work shows the importance to formally face a complex control problem, designing a hybrid control, i.e. focusing on the discrete and continuous control designing stages.

A significant innovation introduced is the use of a Behaviour Tree (BT) to control the transitions from a state to another in the discrete system. The BT has been chosen because allows to develop intelligent agents, i.e. entities capable to react inside an unstructured environment, much better than other control architectures. The BT, in a more formal way, allows to design and implement the discrete control of a nanogrid. The BT defines the transactions between different kinds of activity, that is the switching between different controllers of the continuous low-level layer.

The nature of BTs is to code modular and reactive behaviours. Modularity is the capacity to divide and combine single behaviours. Reactivity is the ability to react to exogenous events. The behaviour is implemented in the control algorithm, which follows the simple structure of the tree. The whole behaviour is made by a sequence of sub-behaviours; the design of a sub-behaviour is independent from the overall behaviour. The sub-BTs can be designed recursively, adding more details (more smartness) time after time. Thanks to the BT the behaviour complexity is given by the composition of a high number of simple algorithms and not by the complexity of the algorithm. The interaction between the flow operators allows to provide the desired behaviour, through the relationships that are achieved between the state space of each node.

Avoiding a complex control algorithm to implement the nanogrid desired features is an important goal of the design. Indeed, high levels of complexity in software increases the chance of introducing defects when making changes. In more extreme cases, it can make modifying the software virtually impossible. If the system is too complex, then changes are hard to do and time-consuming. The difficulty in modifying the behaviour, that is the algorithm and the associated software, can produce obsolete products before that the product has been completely developed. The BT allows to implement a control logic as sophisticated as desired, avoiding an unmanageable complex algorithm and without increasing the complexity of the implementation. Other control architectures have been explored, but they showed critical limitations in the first place.

The BT has proved to be a powerful technique. It is efficient, clear and practice. The translation from requirements in the common language to the syntax rules of BT is almost immediate. The behaviour is described in the structure tree, which has a clear graphical representation. The tree structure improves the performance of algorithms. In this application the BT is executed under hard real-time constraints, with a relatively high frequency, by a microcontroller with modest performance.

Other than a theoretical description, the control problem is faced from a practical implementation. The thesis allows to move from a simulation environment to a laboratory setup toward a mature industrial product. Indeed, all the concerns not encountered during a prototype development have been faced and resolved thanks to the proposed control structure. The thesis wants to drive the reader from the requirements gathering to the final deployment. It is strongly pursued the purpose to present the work from a practical implementation side; indeed, the control algorithms and the software are illustrated in detail.

The features of the proposed control system are tested and proved in the presented testbed. The testbed is installed in a building in the campus of the University of Calabria. The building is a student residence of the university. The setup has three different NGs, connected to the same dc bus. The first nanogrid is connected to the utility grid and to a 4.5 kW PV plant. The second nanogrid is connected to a 232V/22 kWh storage system and supplies critical ac loads. The last nanogrid is used to interface the stirling generator.

In the testbed, among the various solution prospected, the attention is focused on a particular distributed strategy, known as dc-bus signal (DBS). The DBS is characterized by a reliable structure, because the communication infrastructure is the same power connection between the sources. This control structure is one of the most promising solution for NG applications, because ensures high reliability, inherent simplicity and low costs. The DBS logic is coded by the BT inside each nanogrid.

In conclusion, the theoretical and practical approach to the control problem have allowed to develop a system near to the commercial deployment. The nanogrid becomes an actuator and a reactive entity with distributed intelligence inside the smart grid, which gives the opportunity to enable advanced energy management strategies in the local system and advanced business models inside the energy community.

References

[1] P. Dondi, D. Bayoumi, C. Haederli, D. Julian, and M. Suter, "Network integration of distributed power generation," in *Journal of Power Sources*, 2002, vol. 106, no. 1–2, pp. 1–9.

[2] J. A. P. Lopes, N. Hatziargyriou, J. Mutale, P. Djapic, and N. Jenkins, "Integrating distributed generation into electric power systems: A review of drivers, challenges and opportunities," *Electr. Power Syst. Res.*, vol. 77, no. 9, pp. 1189–1203, Jul. 2007.

[3] ETP SmartGrids, *European technology platform smart grids: vision and strategy for Europe's electricity networks of the future*, vol. 19, no. 3. 2006.

[4] R. H. Lasseter, "MicroGrids," in *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309)*, vol. 1, pp. 305–308.

[5] B. Lasseter, "Microgrids [distributed power generation]," in *2001 IEEE Power Engineering Society Winter Meeting, PES 2001 - Conference Proceedings*, 2001, vol. 1, pp. 146–149.

[6] D. Burmester, R. Rayudu, W. Seah, and D. Akinyele, "A review of nanogrid topologies and technologies," *Renewable and Sustainable Energy Reviews*, vol. 67. Pergamon, pp. 760–775, 01-Jan-2017.

[7] R. Jayabalan, B. Fahimi, A. Koenig, and S. Pekarek, "Applications of power electronics-based systems in vehicular technology: state-of-the-art and future trends," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, pp. 1887–1894.

[8] "The Expanding Business of Nanogrids." [Online]. Available: https://www.navigantresearch.com/webinar/the-expanding-business-of-nanogrids.

[9] "Nanogrids and Microgrids are Keys to Decentralizing the Energy Infrastructure - Energy Manager Today." [Online]. Available: https://www.energymanagertoday.com/nanogrids-and-microgrids-are-keys-to-decentralizing-the-energy-infrastructure-0126982/. [Accessed: 29-Oct-2017].

[10] B. Automation, "Nanogrids : The Ultimate Solution for Creating Energy-Aware Buildings ?," pp. 8–11, 2016.

[11] M. Lawrence, "Executive Summary: Nanogrids," 2014.

[12] "Navigant Research Shares Its Nanogrid Solutions Definition." [Online]. Available: https://www.energymanagertoday.com/whats-a-nanogrid-099702/. [Accessed: 29-Oct-2017].

[13] R. Hebner, "Nanogrids, Microgrids, and Big Data: The Future of the Power Grid," 2017. [Online]. Available: https://spectrum.ieee.org/energy/renewables/nanogrids-microgrids-and-big-data-the-future-of-the-power-grid. [Accessed: 29-Oct-2017].

[14] D. Menniti *et al.*, "Nanogrids for Home Application in a Power Cloud framework," in *2016 AEIT International Annual Conference (AEIT)*, 2016, pp. 1–6.

[15] G. Brusco, A. Burgio, D. Menniti, A. Pinnarelli, and N. Sorrentino, "Energy management system for an energy district with demand response availability," *IEEE Trans. Smart Grid*, vol. 5, no. 5, pp. 2385–2393, Sep. 2014.

[16] Navigant Inc., "The energy cloud: Emerging opportunities on the decentralized grid," 2016.

[17] B. Nordman, "Nanogrids Evolving our electricity systems from the bottom up."

[18] B. P. Koirala, E. Koliou, J. Friege, R. A. Hakvoort, and P. M. Herder, "Energetic communities for community energy: A review of key issues and trends shaping integrated community energy systems," *Renewable and Sustainable Energy Reviews*, vol. 56. Pergamon, pp. 722–744, 01-Apr-2016.

[19] "Energy Management - Microgrid and Nanogrid Energy Management System - NEC-Labs." [Online]. Available: http://www.nec-labs.com/research/microgrid-and-nanogrid-energy-management-

system/90. [Accessed: 25-Oct-2017].

[20] Virginia Tech, "Nanogrids for Sustainable Buildings," 2012. [Online]. Available: https://ece.vt.edu/news/article/nanogrids_for_sustainable_buildings. [Accessed: 25-Oct-2017].

[21] B. Nordman, K. Christensen, and A. Meier, "Think Globally, Distribute Power Locally: The Promise of Nanogrids," *Computer (Long. Beach. Calif).*, vol. 45, no. 9, pp. 89–91, Sep. 2012.

[22] J. Schonberger, S. D. Round, and R. Duke, "Decentralised Source Scheduling in a Model Nanogrid Using DC Bus Signalling," *Aust. J. Electr. Electron. Eng.*, vol. Vol. 2, No. 3, pp. 183–190, 2005.

[23] J. Schonbergerschonberger, R. Duke, and S. D. Round, "DC-Bus Signaling: A Distributed Control Strategy for a Hybrid Renewable Nanogrid," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1453–1460, Oct. 2006.

[24] J. Bryan, R. Duke, and S. Round, "Decentralized generator scheduling in a nanogrid using DC bus signaling," in *IEEE Power Engineering Society General Meeting, 2004.*, vol. 2, pp. 977–982.

[25] D. Qu, M. Wang, Z. Sun, and G. Chen, "An improved DC-Bus signaling control method in a distributed nanogrid interfacing modular converters," in *2015 IEEE 11th International Conference on Power Electronics and Drive Systems*, 2015, pp. 214–218.

[26] S. I. Ganesan, D. Pattabiraman, R. K. Govindarajan, M. Rajan, and C. Nagamani, "Control Scheme for a Bidirectional Converter in a Self-Sustaining Low-Voltage DC Nanogrid," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6317–6326, Oct. 2015.

[27] A. Pinnarelli *et al.*, "A power management and control strategy with grid-ancillary services for a microgrid based on DC Bus," *Int. Rev. Electr. Eng.*, vol. 9, no. 4, p. 792, Aug. 2014.

[28] J. Schönberger, S. Round, and R. Duke, "Autonomous load shedding in a nanogrid using DC bus signalling," in *IECON Proceedings (Industrial Electronics Conference)*, 2006, pp. 5155–5160.

[29] D. Burmester, R. Rayudu, and W. K. G. Seah, "Distributed generation nanogrid load control system," in *2015 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2015, pp. 1–5.

[30] D. Menniti, N. Sorrentino, A. Pinnarelli, M. Motta, G. Barone, and G. Brusco, "A Laboratory-Scale Prototype of a Smart User Network with DBS control," *Int. Rev. Electr. Eng.*

[31] G. Barone *et al.*, "A real-life application case of a Smart User Network," in *IEEE International Conference on Clean Electrical Power (ICCEP)*, 2017.

[32] L. Zhang, T. Wu, Y. Xing, K. Sun, and J. M. Gurrero, "Power control of DC microgrid using DC bus signaling," in *2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2011, pp. 1926–1932.

[33] Xiaofeng Sun, Zhizhen Lian, Baocheng Wang, and Xin Li, "A Hybrid renewable DC microgrid voltage control," in *2009 IEEE 6th International Power Electronics and Motion Control Conference*, 2009, pp. 725–729.

[34] S. Sheng, P. Li, C. T. Tsu, and B. Lehman, "Optimal power flow management in a photovoltaic nanogrid with batteries," in *2015 IEEE Energy Conversion Congress and Exposition, ECCE 2015*, 2015, pp. 4222–4228.

[35] F. Shahnia, A. Ghosh, R. P. S. Chandrasena, and S. Rajakaruna, "Dynamic operation and control of a hybrid nanogrid system for future community houses," *IET Gener. Transm. Distrib.*, vol. 9, no. 11, pp. 1168–1178, Aug. 2015.

[36] D. Dong, F. Luo, X. Zhang, D. Boroyevich, and P. Mattavelli, "Grid-Interface Bidirectional Converter for Residential DC Distribution Systems—Part 2: AC and DC Interface Design With Passive Components Minimization," *IEEE Trans. Power Electron.*, vol. 28, no. 4, pp. 1667–1679, Apr. 2013.

[37] Dong, I. Cvetkovic, D. Boroyevich, W. Zhang, R. Wang, and P. Mattavelli, "Grid-interface

bidirectional converter for residential DC distribution systems - Part one: High-density two-stage topology," *IEEE Trans. Power Electron.*, vol. 28, no. 4, pp. 1655–1666, Apr. 2013.

[38] I. Cvetkovic *et al.*, "A testbed for experimental validation of a low-voltage DC nanogrid for buildings," in *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*, 2012, p. LS7c.5-1-LS7c.5-8.

[39] W. Zhang, F. C. Lee, and P.-Y. Huang, "Energy management system control and experiment for future home," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2014, pp. 3317–3324.

[40] H. Wu, S. C. Wong, C. K. Tse, and Q. Chen, "Analysis, control and design of a long-lifetime AC-DC bus converter within a nanogrid," in *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, 2015, vol. 2015–May, no. May, pp. 1855–1860.

[41] R. Adda, S. Member, O. Ray, and S. K. Mishra, "Synchronous-Reference-Frame-Based Control of Switched Boost Inverter for Standalone DC Nanogrid Applications," *IEEE Trans. Power Electron.*, vol. 28, no. 3, pp. 1219–1233, Mar. 2013.

[42] B. Nordman, "Local Grid Definitions: A white paper, developed by the Smart Grid Interoperability Panel, Home Building and Industrial Working Group," 2016.

[43] S. K. Ghai, Z. Charbiwala, S. Mylavarapu, D. P. Seetharamakrishnan, and R. Kunnath, "DC Picogrids: A Case for Local Energy Storage for Uninterrupted Power to DC Appliances," in *Proceedings of the the fourth international conference on Future energy systems - e-Energy '13*, 2013.

[44] A. Sannino, G. Postiglione, and M. H. J. Bollen, "Feasibility of a DC network for commercial facilities," *IEEE Trans. Ind. Appl.*, vol. 39, no. 5, pp. 1499–1507, Sep. 2003.

[45] P. Cairoli, I. Kondratiev, and R. A. Dougal, "Controlled power sequencing for fault protection in DC nanogrids," in *2011 International Conference on Clean Electrical Power (ICCEP)*, 2011, pp. 730–737.

[46] D. Dong *et al.*, "Passive filter topology study of single-phase ac-dc converters for DC nanogrid applications," in *2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2011, pp. 287–294.

[47] A. Goikoetxea, J. M. Canales, R. Sanchez, and P. Zumeta, "DC versus AC in residential buildings: Efficiency comparison," in *Eurocon 2013*, 2013, pp. 1–5.

[48] A. Werth, N. Kitamura, and K. Tanaka, "Conceptual Study for Open Energy Systems: Distributed Energy Network Using Interconnected DC Nanogrids," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1621–1630, Jul. 2015.

[49] B. Nordman and K. Christensen, "DC Local Power Distribution with microgrids and nanogrids," in *2015 IEEE 1st International Conference on Direct Current Microgrids, ICDCM 2015*, 2015, pp. 199–204.

[50] A. S. Morais and L. A. C. Lopes, "Interlink Converters in DC nanogrids and its effect in power sharing using distributed control," in *2016 IEEE 7th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, 2016, pp. 1–7.

[51] D. E. Olivares *et al.*, "Trends in Microgrid Control," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1905–1919, Jul. 2014.

[52] C. N. Papadimitriou, E. I. Zountouridou, and N. D. Hatziargyriou, "Review of hierarchical control in DC microgrids," *Electric Power Systems Research*, vol. 122. Elsevier, pp. 159–167, 01-May-2015.

[53] W. Su and J. Wang, "Energy Management Systems in Microgrid Operations," *Electr. J.*, vol. 25, no. 8, pp. 45–60, Oct. 2012.

[54] D. E. Olivares, C. A. Canizares, and M. Kazerani, "A centralized optimal energy management system for microgrids," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–6.

[55] H. Kanchev, D. Lu, F. Colas, V. Lazarov, and B. Francois, "Energy Management and Operational

Planning of a Microgrid With a PV-Based Active Generator for Smart Grid Applications," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4583–4592, Oct. 2011.

[56] J. K. Schönberger, "Distributed Control of a Nanogrid Using DC Bus Signalling," 2005.

[57] W. W. Weaver, R. D. Robinett, G. G. Parker, and D. G. Wilson, "Distributed control and energy storage requirements of networked Dc microgrids," *Control Eng. Pract.*, vol. 44, pp. 10–19, Nov. 2015.

[58] M. Colledanchise, "Behavior Trees in Robotics," *Dr. Thesis*, 2017.

[59] "Gamasutra - GDC 2005 Proceeding: Handling Complexity." [Online]. Available: https://www.gamasutra.com/view/feature/130663/gdc_2005_proceeding_handling_.php. [Accessed: 29-Oct-2017].

[60] M. S. Branicky, "Introduction to Hybrid Systems," in *Handbook of Networked and Embedded Control Systems*, Boston, MA: Birkhäuser Boston, 2005, pp. 91–116.

[61] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer, 2008.

[62] D. Isla, "Halo 3-building a better battle," *Game Dev. Conf.*, 2008.

[63] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ogren, "Towards a unified behavior trees framework for robot control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5420–5427.

[64] P. Ogren, "Increasing Modularity of UAV Control Systems using Computer Game Behavior Trees," in *AIAA Guidance, Navigation, and Control Conference*, 2012.

[65] Alex J. Champandard, "10 Reasons the Age of Finite State Machines is Over | AiGameDev.com," *http://aigamedev.com*. [Online]. Available: http://aigamedev.com/open/article/fsm-age-is-over/. [Accessed: 25-Oct-2017].

[66] R. De pontes pereira, "An Introduction To Behavior Trees," vol. 2014, p. 1/3, 2017.

[67] R. De pontes pereira, "An Introduction To Behavior Trees," vol. 2014, p. 1/3, 2017.

[68] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ogren, "Towards a unified behavior trees framework for robot control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5420–5427.

[69] M. Colledanchise, R. Parasuraman, and P. Ögren, "Learning of Behavior Trees for Autonomous Agents," 2015.

[70] "Understanding Behavior Trees | AiGameDev.com," 2007. [Online]. Available: http://aigamedev.com/open/article/bt-overview/. [Accessed: 29-Oct-2017].

[71] I. Millington and J. Funge, "ARTIFICIAL INTELLIGENCE FOR GAMES Second Edition," pp. 3–22, 1968.

[72] A. J. Champandard and P. Dunstan, "The Behavior Tree Starter Kit," in *in Game AI Pro., ed. S.Rabin. Boca Raton, FL, USA*, 2014, pp. 73–91.

[73] M. Colledanchise and P. Ogren, "How Behavior Trees Modularize Hybrid Control Systems and Generalize Sequential Behavior Compositions, the Subsumption Architecture, and Decision Trees," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 372–389, Apr. 2017.

[74] M. Colledanchise and P. Ogren, "How Behavior Trees modularize robustness and safety in hybrid systems," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1482–1488.

[75] G. Barone, A. Burgio, D. Menniti, M. Motta, A. Pinnarelli, and N. Sorrentino, "A printed circuit board suitable for controlling a 22.8kVA IGBT three-phase inverter for DC Nanogrids in a laboratory setup," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE*

Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2017, pp. 1–6.

[76] S. Bacha, I. Munteanu, and A. I. Bratcu, *Power Electronic Converters Modeling and Control*. 2014.

[77] R. W. (Robert W. Erickson and D. Maksimović, *Fundamentals of power electronics*. Kluwer Academic, 2001.

[78] M. Bhardwaj, "Modeling Bi-Directional Buck/Boost Converter for Digital Control Using C2000 Microcontrollers."

[79] G. Brusco, A. Burgio, D. Menniti, M. Motta, A. Pinnarelli, and N. Sorrentino, "A repetitive-based control for a single-phase shunt active power filter for harmonic voltage compensation," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, 2016, pp. 1–6.

[80] M. Prodanovic and T. C. Green, "Control and filter design of three-phase inverters for high power quality grid connection," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 373–380, Jan. 2003.

[81] Chi Zhang, T. Dragicevic, J. C. Vasquez, and J. M. Guerrero, "Resonance damping techniques for grid-connected voltage source converters with LCL filters — A review," in *2014 IEEE International Energy Conference (ENERGYCON)*, 2014, pp. 169–176.

[82] Chenlei Bao, Xinbo Ruan, Xuehua Wang, Weiwei Li, Donghua Pan, and Kailei Weng, "Step-by-Step Controller Design for LCL-Type Grid-Connected Inverter with Capacitor–Current-Feedback Active-Damping," *IEEE Trans. Power Electron.*, vol. 29, no. 3, pp. 1239–1253, Mar. 2014.

[83] D. Thimsen, "Stirling Engine Assessment," 2002.

[84] "Peripherals on board of EVK1100." [Online]. Available: http://www.atmel.com/webdoc/evk1100/evk1100.Introduction.html. [Accessed: 29-Oct-2017].

[85] "32-Bit Atmel AVR Microcontroller - AT32UC3A0512."

[86] "AVR32138: How to optimize the ADC usage on AT32UC3A0/1, AT32UC3A3 and AT32UC3B0/1 series."

[87] A. Burgio, D. Menniti, M. Motta, A. Pinnarelli, N. Sorrentino, and P. Vizza, "A two-input dual active bridge converter for a smart user network using integrated power modules," in *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, 2016, vol. 2016–Janua, pp. 1–6.

[88] G. Barone, A. Burgio, D. Menniti, M. Motta, A. Pinnarelli, and N. Sorrentino, "Remote control of nanogrids: A cost-effective solution in a laboratory setup," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, 2017, pp. 1–6.