energies

Article

Generic Framework for the Optimal Implementation of Flexibility Mechanisms in Large-Scale Smart Grids

Alejandro J. del Real [1,*], Andrés Pastor [2] and Jaime Durán [2]

[1] Department of Systems and Automation, University of Seville, 41092 Seville, Spain
[2] IDENER, 41300 Seville, Spain; andres.pastor@idener.es (A.P.); jaime.duran@idener.es (J.D.)
* Correspondence: adelreal@us.es

Abstract: This paper aims to provide the smart grid research community with an open and accessible general mathematical framework to develop and implement optimal flexibility mechanisms in large-scale network applications. The motivation of this paper is twofold. On the one hand, flexibility mechanisms are currently a hot topic of research, which is aimed to mitigate variation and uncertainty of electricity demand and supply in decentralised grids with a high aggregated share of renewables. On the other hand, a large part of such related research is performed by heuristic methods, which are generally inefficient (such methods do not guarantee optimality) and difficult to extrapolate for different use cases. Alternatively, this paper presents an MPC-based (model predictive control) framework explicitly including a generic flexibility mechanism, which is easy to particularise to specific strategies such as demand response, flexible production and energy efficiency services. The proposed framework is benchmarked with other non-optimal control configurations to better show the advantages it provides. The work of this paper is completed by the implementation of a generic use case, which aims to further clarify the use of the framework and, thus, to ease its adoption by other researchers in their specific flexibility mechanism applications.

Keywords: large-scale systems; aggregated terms; flexibility mechanisms; model predictive control; centralised MPC; decentralised MPC

check for updates

Citation: del Real, A.J.; Pastor, A.; Durán, J. Generic Framework for the Optimal Implementation of Flexibility Mechanisms in Large-Scale Smart Grids. *Energies* **2021**, *14*, 8063. https://doi.org/10.3390/en14238063

Academic Editor: Ricardo J. Bessa

Received: 28 October 2021
Accepted: 30 November 2021
Published: 2 December 2021

1. Introduction and Literature Review

The implementation of flexibility mechanisms in large-scale power and energy systems [1] is a paradigm of great importance for today's industry, which has become more widespread as systems have grown [2]. Thanks to the development and integration of such automation and optimisation algorithms, the efficiency of the smart grids improves while saving costs and supporting the environment.

In particular, early attempts to solve the complex problem of scheduling in the frame of flexibility mechanisms were mostly based on heuristics, which is a methodology that does not guarantee optimality and is difficult to generalise [3]. More recently, the researchers have adopted approaches based on mathematical optimisation [4], which provides the means to achieve optimality and generalisation of the solutions.

Under such an approach, researchers usually make use of linear programming (LP) and mixed integer linear programming (MILP) that can be applied to optimise schedules based on different criteria under different operating conditions [5]. More sophisticated approaches than LP and MILP also includes hybrid mixed-integer quadratic optimisation [6], multi-objective mixed-binary linear programming [7], and other extensions of MILP [8].

However, despite the sophistication of such optimisation techniques and to the best of this authors' knowledge, most of them rely on ad hoc realisations that are particularly adapted to the specific configuration of the smart grid configuration and flexibility mechanisms under research. This fact hinders their direct extrapolation to other different configurations and usually forces the researchers to develop new formulations from scratch. As an alternative, this paper proposes a structured, general mathematical optimisation framework for the optimal and generic implementation of flexibility mechanisms that is based on model predictive control (MPC).

Historically, MPC has been one of the most applied control algorithms for large-scale systems within the industry [9–11]. The main reasons behind the success of such kind of controller are its well-known robustness and reliability [12,13], since MPCs are capable of reaching an optimal solution taking into account not only the current state of the system but also its future behaviour throughout the use of a model of the process [14]. Indeed, MPCs can find an adequate and optimal control behaviour even when the model is not perfectly accurate regarding the real-world process [15].

In addition, another key characteristic of MPCs is their flexibility when modelling large-scale systems. MPCs allow to mathematically subdivide, thanks to the use of state-space models (such as energy hub methods [16]), large systems into smaller subsections, usually called "nodes" [17].

Thanks to such advantages, the use of MPC for the implementation of flexibility mechanisms have the potential to provide the researchers with an efficient and easy-to-generalise framework for further applications. To illustrate the potential of MPC to provide an efficient and easy-to-generalise framework for the implementation of flexibility mechanisms, the rest of this paper is organised as follows. First, Section 2 briefly introduces the concept of model predictive control to provide some contextualisation. Section 3 describes, in detail, the mathematical framework proposed by the authors along with the explicit consideration of generic flexibility mechanisms. Section 4 shortly defines a benchmarking example based on heuristics. Later, Section 5 presents a generic use case for validating the authors' proposed approach against the benchmarking example. Section 6 discusses the application of the proposed framework to the implementation of smart grids' flexibility mechanisms applications. Finally, Section 7 presents the authors' conclusions about the results achieved and insights about the upcoming works on the topic.

2. Model Predictive Control

Model predictive control (MPC) is an advanced process control method that has been used in large-scale systems since the 1980s [18]. MPCs are widely used in the industry since they allow to consider both the current and also the future system status in order to satisfy the system constraints and control objectives [19]. This is achieved by minimising an objective function, which allows defining the cost associated with the use of certain variables throughout a prediction time horizon [20].

At each instant, the set of future control values is calculated by following the next criteria:

Minimise	An objective function (also called "cost function") in terms of actions over the prediction horizon.
Subject to	Dynamics of the system over the prediction horizon; constraints of the system; measurement of the initial state of the system at the beginning of the current control instant.

Therefore, MPC is based on an iterative optimisation, as shown in Figure 1. At instant k, the information about the current state of the system is received, and a control strategy that minimises the cost is calculated (using a mathematical solver) for a time range from the current instant (k) to a future instant ($k + H$). Specifically, a solution that emerges from the current state and guides the system through a time range $[k,\ k + H]$ is found (always complying with objective function cost minimisation and constraints). Then, only the next step (that corresponds to $k + 1$) of the control strategy is implemented. Then, when the next instant $k + 1$ arrives, the process starts again.

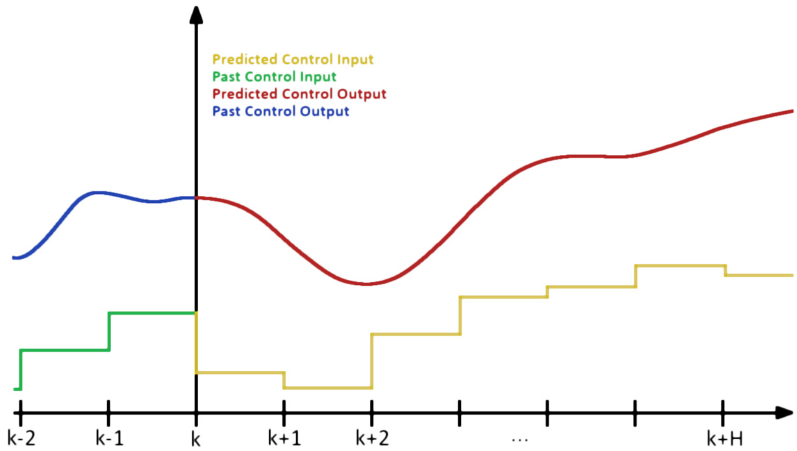

Figure 1. MPC scheme.

3. Proposed Approach

In the first part of this section, the mathematical framework proposed by the authors is presented and described in depth. In addition, all the variables, constraints, matrixes, and equations that define the optimisation problem are enunciated. The second part of this section introduces the flexibility mechanism and details how it is integrated into the optimisation problem.

3.1. Centralised MPC

A Centralised MPC relies on a control Agent that manages the whole large-scale system by itself. This Agent considers the state variables, inputs, outputs, constants, constraints, disturbances and dynamics of all the elements of the network to satisfy a single global objective function for the entire system. Since this MPC knows every piece of information of every node of the grid, it ensures the convergence and the optimality (global optimum) of the solution [14].

However, some past literature [18] indicated that the disadvantages of a centralised MPC include the difficulty of modelling the entire system and the complication of isolating parts of the system in case one part fails.

Nevertheless, the authors have followed some methodologies that reduce the impact of the disadvantages. First, the system has been modelled as an aggrupation of different and independent state-space models. Accordingly, the system can be easily modelled and the nodes can be rapidly isolated if necessary.

Figure 2 shows a large-scale system composed of N parts, also called "nodes", where each node has a set of inputs $u_i(t)$, state variables $x_i(t)$ and outputs $y_i(t)$.

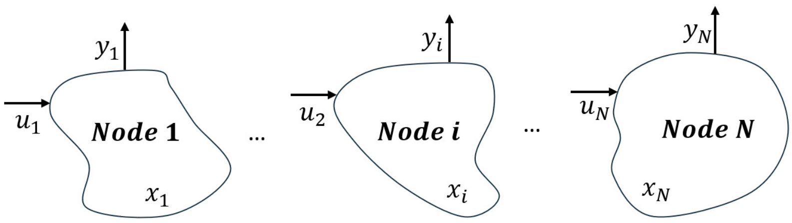

Figure 2. General network of N nodes.

According to the formulation:

$$x_i = \left[x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,\alpha}\right]^T \tag{1}$$

$$u_i = \left[u_{i,1}, \ldots, u_{i,j}, \ldots, u_{i,\beta}\right]^T \tag{2}$$

$$y_i = \left[y_{i,1}, \ldots, y_{i,j}, \ldots, y_{i,\gamma}\right]^T \tag{3}$$

with i being the number of nodes, α the number of states of the node, β the number of inputs of the node and γ the number of outputs of the node.

In general, the dynamics of each node are given by the continuous, non-linear and time-varying description:

$$\dot{x}_i\ (t) = f_i(x_i(t), u_i(t)) \tag{4}$$

$$y_i\ (t) = g_i(x_i(t), u_i(t)) \tag{5}$$

without loss of generality, each node is modelled by a discrete, linear, time-invariant description with a state-space model [21]:

$$x_i(k+1) = A_i \cdot x_i(k) + B_i \cdot u_i(k) \tag{6}$$

$$y_i\ (k) = C_i \cdot x_i(k) \tag{7}$$

where:

$$A_i = \begin{bmatrix} a^i_{1,1} & \cdots & a^i_{1,\alpha} \\ \vdots & \ddots & \vdots \\ a^i_{\alpha,1} & \cdots & a^i_{\alpha,\alpha} \end{bmatrix} \tag{8}$$

$$B_i = \begin{bmatrix} b^i_{1,1} & \cdots & b^i_{1,\beta} \\ \vdots & \ddots & \vdots \\ b^i_{\beta,1} & \cdots & b^i_{\beta,\beta} \end{bmatrix} \tag{9}$$

$$C_i = \begin{bmatrix} c^i_{1,1} & \cdots & c^i_{1,\gamma} \\ \vdots & \ddots & \vdots \\ c^i_{\gamma,1} & \cdots & c^i_{\gamma,\gamma} \end{bmatrix} \tag{10}$$

The local control problem for each node i is then written as:

$$\underset{\widetilde{x}_i(k+1), \widetilde{y}_i(k), \widetilde{u}_i(k)}{min} \phi_{local,i}(\widetilde{x}_i(k+1),\ \widetilde{u}_i(k),\ \widetilde{y}_i(k)) \tag{11}$$

$$\begin{array}{rl} subject\ to & x_i(k+1+l) = A_i x_i(k+l) + B_i u_i(k+l) \\ & y_i(k+l) = C_i x_i(k+l) \end{array} \tag{12}$$

where k is the execution or simulation instant, l is the prediction window instant and H is the control horizon:

$$\widetilde{x}_i(k+1) = \left[x_i^T(k), \ldots, x_i^T(k+H)\right]^T \tag{13}$$

$$\widetilde{u}_i(k) = \left[u_i^T(k), \ldots, u_i^T(k+H-1)\right]^T \tag{14}$$

$$y_i(k) = \left[y_i^T(k), \ldots, y_i^T(k+H-1)\right]^T \tag{15}$$

$$x_i(k+1+l), u_i(k+l), y_i(k+l)\ \in\ \Omega_{x_i}, \Omega_{u_i}, \Omega_{y_i} \tag{16}$$

$$l = [0, 1, \ldots, H-1] \tag{17}$$

$$k \in N \tag{18}$$

This mathematical approach is also supported and derived from the work of other authors [14,16,21,22]. Thanks to this mathematical description of a node, a centralised MPC can be efficiently defined for a large-scale system.

3.2. Introducing the Flexibility Mechanism into the Centralised MPC

As mentioned above, this paper intends to develop and validate an MPC controller capable of working with flexibility mechanisms. A "flexibility mechanism" is the need for several nodes in a large-scale smart grid to compensate or comply with constraints together [23,24]. This means that a set of variables from different nodes of a smart grid must participate together to satisfy a specific constraint.

In general, this constraint is exemplified by ensuring that the sum of the inputs of the various nodes in the system is not greater than a specific value. For the specific application to smart grids, such constraint would be used to ensure, for example, that the combined electricity consumption of several houses does not exceed a value.

Specifically, a centralised MPC applied to a large-scale system of N nodes with a flexibility mechanism can be represented as follows (Figure 3).

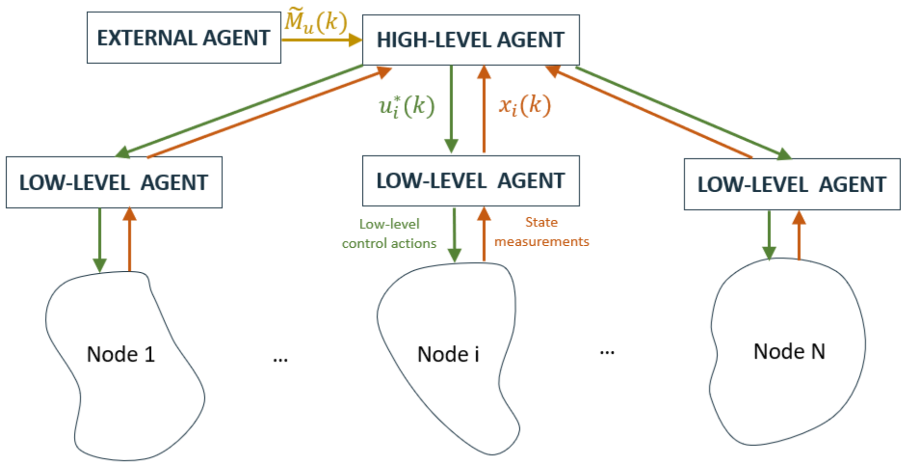

Figure 3. Generic centralised MPC diagram.

Under this approach, an External Agent defines the flexibility mechanism that all the nodes must jointly satisfy. A general flexibility mechanism is defined by an upper limit $\widetilde{M}_u(k)$, which is further explained later in this section.

The diagram includes a High-Level Agent, which implements the centralised MPC and performs the calculation of the optimal control actions to solve the optimisation problem. Once the optimal control actions are computed, the High-Level Agent sends the corresponding set-points to the Low-Level Agents, which are assumed to perform the low-level control actions (and thus, can be omitted in the high-level approach discussed herein).

This way, the control problem of a large-scale system of N interconnected nodes can be seen as an aggregation of the local control problems plus the constraints of the flexibility mechanisms. Its equations can be extrapolated from Equations (6)–(18) as follows:

$$\min_{\widetilde{x}_i(k+1),\widetilde{y}_i(k),\widetilde{u}_i(k)} \sum_{i=1}^{N} \phi_{local,i}(\widetilde{x}_i(k+1),\ \widetilde{u}_i(k),\ \widetilde{y}_i(k)) \tag{19}$$

$$\begin{aligned} subject\ to \quad \boldsymbol{X}(k+1+l) &= \boldsymbol{A}\boldsymbol{X}(k+l) + \boldsymbol{B}\boldsymbol{U}(k+l) \\ \boldsymbol{Y}(k+l) &= \boldsymbol{C}\boldsymbol{X}(k+l) \\ \boldsymbol{K}[\cdot]\cdot\boldsymbol{U}(k+l) &\leq \widetilde{M}_u(k+l) \end{aligned} \tag{20}$$

where:

$$\boldsymbol{A} = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_N \end{bmatrix} \tag{21}$$

$$\boldsymbol{B} = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_N \end{bmatrix} \tag{22}$$

$$\boldsymbol{C} = \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_N \end{bmatrix} \tag{23}$$

$$\boldsymbol{X}(k+1) = \begin{bmatrix} x_1^T(k+1) & \cdots & x_N^T(k+1) \end{bmatrix}^T \tag{24}$$

$$\boldsymbol{U}(k) = \begin{bmatrix} u_1^T(k) & \cdots & u_N^T(k) \end{bmatrix}^T \tag{25}$$

$$\boldsymbol{Y}(k) = \begin{bmatrix} y_1^T(k) & \cdots & y_N^T(k) \end{bmatrix}^T \tag{26}$$

$$\widetilde{M}_u(k+l) \in R^+ \tag{27}$$

3.3. Benchmark Example: Decentralised MPC

As highlighted in Section 1, one of the simplest and widely adopted methods to ensure multiple nodes in a network to comply with an overall constraint is to heuristically distribute this constraint between all the nodes of the grid. That way, each of the nodes must independently satisfy their respective assigned part of the overall constraint. In this paper, such a heuristic approach is formulated as a "Decentralised MPC" [25].

Under the decentralised MPC, the whole large-scale system is split into small subsystems. Each of these subsystems is totally independent and ignores (there is no communication between agents) the rest of the nodes. Each one has its own MPC agent, which is responsible for fulfilling the constraints, dynamics, and objective function of the model of its particular subsystem [26]. Since the nodes do not communicate with each other, the only way to fulfil the flexibility mechanisms is to distribute them heuristically among the agents. A direct way to distribute the limits imposed by the flexibility mechanism $\widetilde{M}_u(k)$ is its equal split among the N Agents, that is, $\widetilde{M}_u(k)/N$.

It is easy to infer that, under such an approach, a global optimum is not guaranteed. A decentralised MPC applied to a large-scale system of N nodes with a flexibility mechanism can be represented as follows (Figure 4).

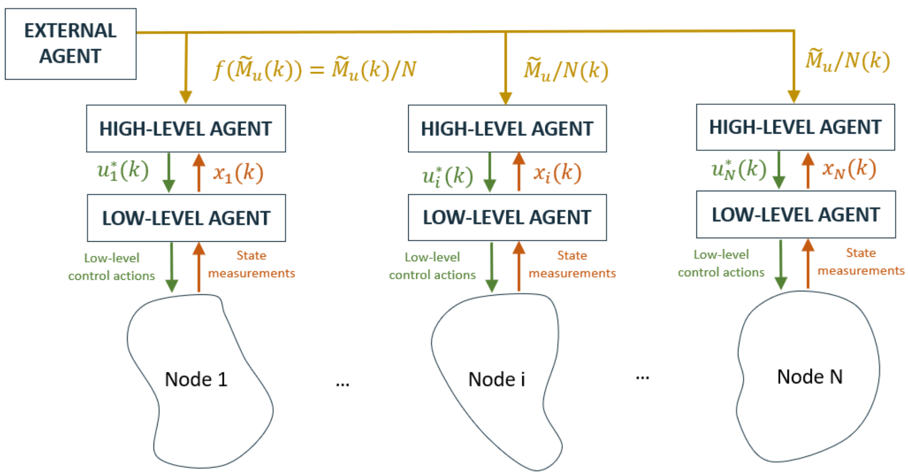

Figure 4. Generic decentralised MPC diagram.

The mathematical formulation of the benchmarking decentralised MPC is derived from previous Equations (6)–(18), and results as follows:

$$\min_{\widetilde{x}_i(k+1),\widetilde{y}_i(k),\widetilde{u}_i(k)} \sum_{i=1}^{N} \phi_{local,i}(\widetilde{x}_i(k+1),\ \widetilde{u}_i(k),\ \widetilde{y}_i(k)) \tag{28}$$

$$\begin{aligned} \text{subject to} \quad x_i(k+1+l) &= A_i x_i(k+l) + B_i u_i(k+l) \\ y_i(k+l) &= C_i x_i(k+l) \\ K_i[\cdot]\cdot u_i(k+l) &\le \frac{\widetilde{M}_u(k+l)}{N} \end{aligned} \tag{29}$$

4. Use Case

This section presents a use case to illustrate the implementation of the proposed formulation to a relatively simple network composed of four nodes, each of them with two inputs ($u_{i,1}$ and $u_{i,2}$), two states ($x_{i,1}$ and $x_{i,2}$) and one output ($y_{i,1}$), as depicted in Figure 5.

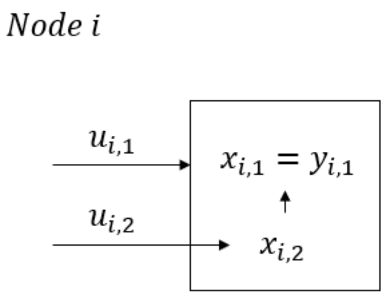

Figure 5. Diagram and variables of the nodes of the network defined in the use case.

4.1. Use Case for the Centralised MPC

The application of the centralised MPC to the specific use case is graphically synthetised in Figure 6.

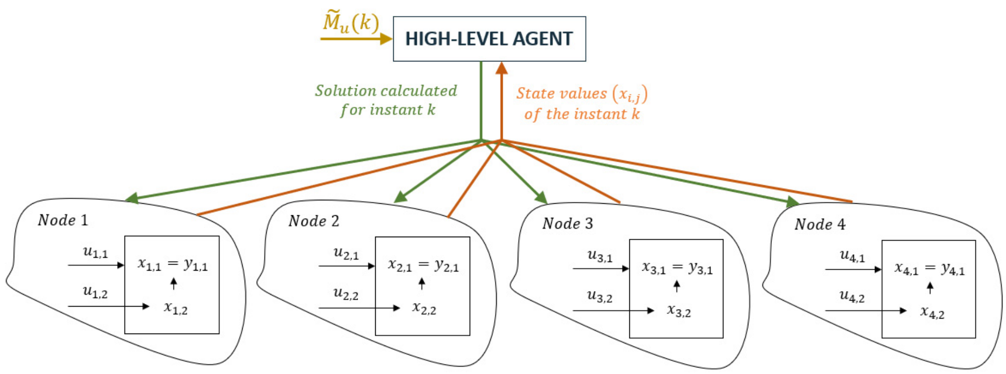

Figure 6. Centralised MPC applied to the use case.

The specific state-space models for each node are:

$$\text{Node 1} \begin{cases} \underbrace{\begin{bmatrix} x_{1,1}(k+1) \\ x_{1,2}(k+1) \end{bmatrix}}_{x_1(k+1)} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} x_{1,1}(k) \\ x_{1,2}(k) \end{bmatrix}}_{x_1(k)} + \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}}_{B_1} \underbrace{\begin{bmatrix} u_{1,1}(k) \\ u_{1,2}(k) \end{bmatrix}}_{u_1(k)} \\ \underbrace{[y_{1,1}(k)]}_{y_1} = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C_1} \underbrace{\begin{bmatrix} x_{1,1}(k) \\ x_{1,2}(k) \end{bmatrix}}_{x_1(k)} \end{cases} \tag{30}$$

$$\text{Node 2}\begin{cases}\underbrace{\begin{bmatrix}x_{2,1}(k+1)\\x_{2,2}(k+1)\end{bmatrix}}_{x_2(k+1)}=\underbrace{\begin{bmatrix}1&1\\0&0.7\end{bmatrix}}_{A_2}\underbrace{\begin{bmatrix}x_{2,1}(k)\\x_{2,2}(k)\end{bmatrix}}_{x_2(k)}+\underbrace{\begin{bmatrix}-0.2&0\\0&-0.2\end{bmatrix}}_{B_2}\underbrace{\begin{bmatrix}u_{2,1}(k)\\u_{2,2}(k)\end{bmatrix}}_{u_2(k)}\\ \underbrace{[y_{2,1}(k)]}_{y_2}=\underbrace{\begin{bmatrix}1&0\end{bmatrix}}_{C_2}\underbrace{\begin{bmatrix}x_{2,1}(k)\\x_{2,2}(k)\end{bmatrix}}_{x_2(k)}\end{cases}\tag{31}$$

$$\text{Node 3}\begin{cases}\underbrace{\begin{bmatrix}x_{3,1}(k+1)\\x_{3,2}(k+1)\end{bmatrix}}_{x_3(k+1)}=\underbrace{\begin{bmatrix}1&0.9\\0&0.8\end{bmatrix}}_{A_3}\underbrace{\begin{bmatrix}x_{3,1}(k)\\x_{3,2}(k)\end{bmatrix}}_{x_3(k)}+\underbrace{\begin{bmatrix}-0.8&0\\0&-0.8\end{bmatrix}}_{B_3}\underbrace{\begin{bmatrix}u_{3,1}(k)\\u_{3,2}(k)\end{bmatrix}}_{u_3(k)}\\ \underbrace{[y_{3,1}(k)]}_{y_3}=\underbrace{\begin{bmatrix}1&0\end{bmatrix}}_{C_3}\underbrace{\begin{bmatrix}x_{3,1}(k)\\x_{3,2}(k)\end{bmatrix}}_{x_3(k)}\end{cases}\tag{32}$$

$$\text{Node 4}\begin{cases}\underbrace{\begin{bmatrix}x_{4,1}(k+1)\\x_{4,2}(k+1)\end{bmatrix}}_{x_4(k+1)}=\underbrace{\begin{bmatrix}1&0.8\\0&0.7\end{bmatrix}}_{A_4}\underbrace{\begin{bmatrix}x_{4,1}(k)\\x_{4,2}(k)\end{bmatrix}}_{x_4(k)}+\underbrace{\begin{bmatrix}-1.7&0\\0&-1.7\end{bmatrix}}_{B_4}\underbrace{\begin{bmatrix}u_{4,1}(k)\\u_{4,2}(k)\end{bmatrix}}_{u_4(k)}\\ \underbrace{[y_{4,1}(k)]}_{y_4}=\underbrace{\begin{bmatrix}1&0\end{bmatrix}}_{C_4}\underbrace{\begin{bmatrix}x_{4,1}(k)\\x_{4,2}(k)\end{bmatrix}}_{x_4(k)}\end{cases}\tag{33}$$

The aggregation of the individual nodes results in the following formulation:

- Aggregated objective function:

$$\begin{aligned}&\min_{\tilde{x}_i(k+1),\,\tilde{y}_i(k),\,\tilde{u}_i(k)}\sum_{i=1}^{N}\phi_{local,i}(\tilde{x}_i(k+1),\,\tilde{u}_i(k),\,\tilde{y}_i(k))\\ =&\min_{\tilde{x}_i(k+1),\,\tilde{y}_i(k),\,\tilde{u}_i(k)}\sum_{i=1}^{N}\begin{bmatrix}\boldsymbol{X}(k+1+l)\\\boldsymbol{U}(k+l)\\\boldsymbol{Y}(k+l)\end{bmatrix}^T\cdot\boldsymbol{Q}\cdot\begin{bmatrix}\boldsymbol{X}(k+1+l)\\\boldsymbol{U}(k+l)\\\boldsymbol{Y}(k+l)\end{bmatrix}+\boldsymbol{f}^T\cdot\begin{bmatrix}\boldsymbol{X}(k+1+l)\\\boldsymbol{U}(k+l)\\\boldsymbol{Y}(k+l)\end{bmatrix}\end{aligned}\tag{34}$$

The aggregation of the four nodes results in the following set of overall equations:

- Aggregated model:

$$\begin{gathered}\boldsymbol{A}=\begin{bmatrix}A_1&&&\\&A_2&&\\&&A_3&\\&&&A_4\end{bmatrix}\ \boldsymbol{B}=\begin{bmatrix}B_1&&&\\&B_2&&\\&&B_3&\\&&&B_4\end{bmatrix}\ \boldsymbol{C}=\begin{bmatrix}C_1&&&\\&C_2&&\\&&C_3&\\&&&C_4\end{bmatrix}\\ \boldsymbol{X}=\begin{bmatrix}x_{11}&x_{12}&x_{21}&x_{22}&x_{31}&x_{32}&x_{41}&x_{42}\end{bmatrix}^T\\ \boldsymbol{U}=\begin{bmatrix}u_{11}&u_{12}&u_{21}&u_{22}&u_{31}&u_{32}&u_{41}&u_{42}\end{bmatrix}^T\\ \boldsymbol{Y}=\begin{bmatrix}y_{11}&y_{21}&y_{31}&y_{41}\end{bmatrix}^T\end{gathered}\tag{35}$$

- Flexibility mechanism:

$$\begin{gathered}\widetilde{M}_u(k+l)=4\,,\ \forall k,\ \forall l=[0,1,\ldots,H-1]\\ \boldsymbol{K}[\cdot]=\boldsymbol{1}_{1x8}\end{gathered}\tag{36}$$

- Initial condition ($k=0$):

$$\boldsymbol{X}(0)=\begin{bmatrix}8&7&6&5&4&3&4&5\end{bmatrix}^T\tag{37}$$

- Space of variables:

$$u_{ij}(k+l),\ x_{ij}(k+l+1),\ y_{ij}(k+l)\geq 0,\ 0,\ 0\tag{38}$$

- Prediction horizon:

$$H = 5 \tag{39}$$

4.2. Decentralised MPC Applied to the Use Case

The application of the decentralised MPC to the specific use case is graphically synthetised in Figure 7.

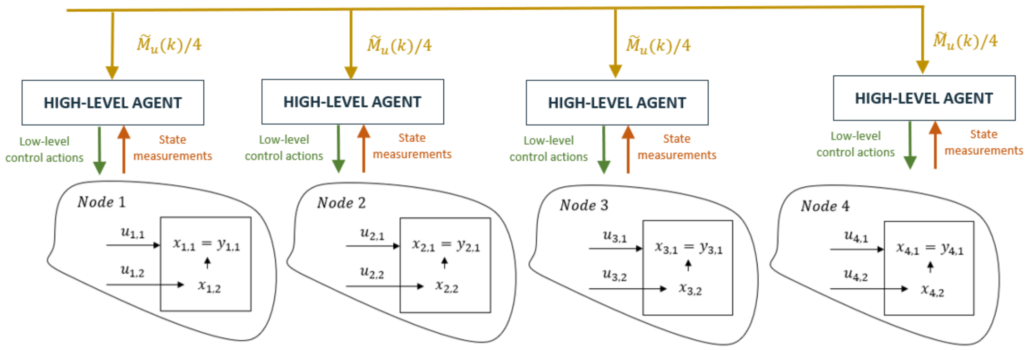

Figure 7. Decentralised MPC applied to the use case.

Under this approach, there is no aggregation and each local MPC implements the local problem defined as follows:

- Objective function:

$$\begin{aligned} &\min_{\widetilde{x}_i(k+1),\, \widetilde{y}_i(k),\, \widetilde{u}_i(k)} \phi_{local,i}(\widetilde{x}_i(k+1),\, \widetilde{u}_i(k),\, \widetilde{y}_i(k)) \\ &= \min_{\widetilde{x}_i(k+1), \widetilde{y}_i(k), \widetilde{u}_i(k)} \begin{bmatrix} x_i(k+1+l) \\ u_i(k+l) \\ y_i(k+l) \end{bmatrix}^T \cdot Q_i \cdot \begin{bmatrix} x_i(k+1+l) \\ u_i(k+l) \\ y_i(k+l) \end{bmatrix} + f_i^T \cdot \begin{bmatrix} x_i(k+1+l) \\ u_i(k+l) \\ y_i(k+l) \end{bmatrix} \end{aligned} \tag{40}$$

where Q_i is the matrix that contains the quadratic cost of each variable of the node, and f_i is the matrix that contains the linear cost of each variable of the node.

The use case data (Equations (30)–(33)) can be implemented in the Equations (28), (29) and (40) in such a way that:

- Model:

$$\begin{aligned} &Node\ 1 \rightarrow \{\ A_1 \quad B_1 \quad C_1\ \},\ x_1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix},\ u_1 = \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix},\ y_1 = [y_{11}] \\ &Node\ 2 \rightarrow \{\ A_2 \quad B_2 \quad C_2\ \},\ x_2 = \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix},\ u_2 = \begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix},\ y_2 = [y_{21}] \\ &Node\ 3 \rightarrow \{\ A_3 \quad B_3 \quad C_3\ \},\ x_3 = \begin{bmatrix} x_{31} \\ x_{32} \end{bmatrix},\ u_3 = \begin{bmatrix} u_{31} \\ u_{32} \end{bmatrix},\ y_3 = [y_{31}] \\ &Node\ 4 \rightarrow \{\ A_4 \quad B_4 \quad C_4\ \},\ x_4 = \begin{bmatrix} x_{41} \\ x_{42} \end{bmatrix},\ u_4 = \begin{bmatrix} u_{41} \\ u_{42} \end{bmatrix},\ y_4 = [y_{41}] \end{aligned} \tag{41}$$

- Flexibility mechanism:

$$\begin{aligned} \widetilde{M}_u(k+l) = 4\ ,\ \forall k,\ \forall l = [0, 1, \ldots, H-1] \\ K_i[\cdot] = [\ 1 \quad 1\ ] \end{aligned} \tag{42}$$

- Initial conditions ($k = 0$):

$$\begin{aligned} x_1(0) &= \begin{bmatrix} 8 & 7 \end{bmatrix}^T \\ x_2(0) &= \begin{bmatrix} 6 & 5 \end{bmatrix}^T \\ x_3(0) &= \begin{bmatrix} 4 & 3 \end{bmatrix}^T \\ x_4(0) &= \begin{bmatrix} 4 & 5 \end{bmatrix}^T \end{aligned} \tag{43}$$

- Space of variables:

$$u_{ij}(k+l),\ x_{ij}(k+l+1),\ y_{ij}(k+l) \geq 0,\ 0,\ 0 \tag{44}$$

- Prediction horizon:

$$H = 5 \tag{45}$$

With this, the decentralised MPC control problem is particularised for the use case.

5. Analysis of Results

Figures 8–10 show the results obtained from running the centralised MPC approach under the four-node use case. Specifically, Figure 8 shows the solution for the u-variables (inputs of the nodes), while Figure 9 shows the results for the y-variables (outputs of the nodes) and Figure 10 shows the evolution of the objective function cost throughout the simulation.

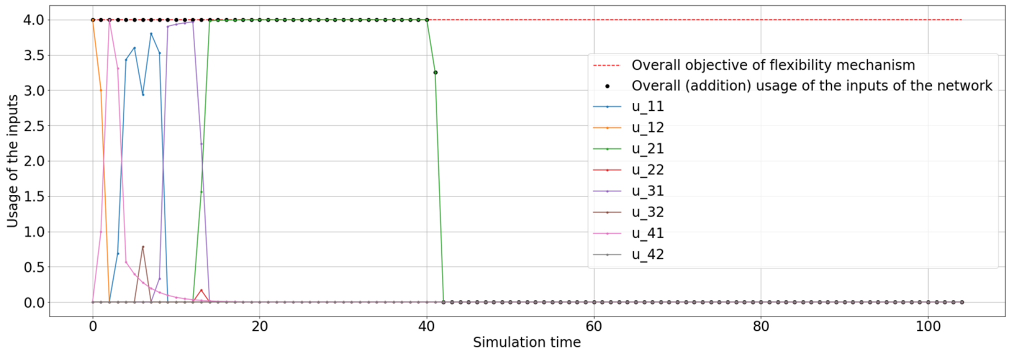

Figure 8. Solution for u-variables of the centralised MPC.

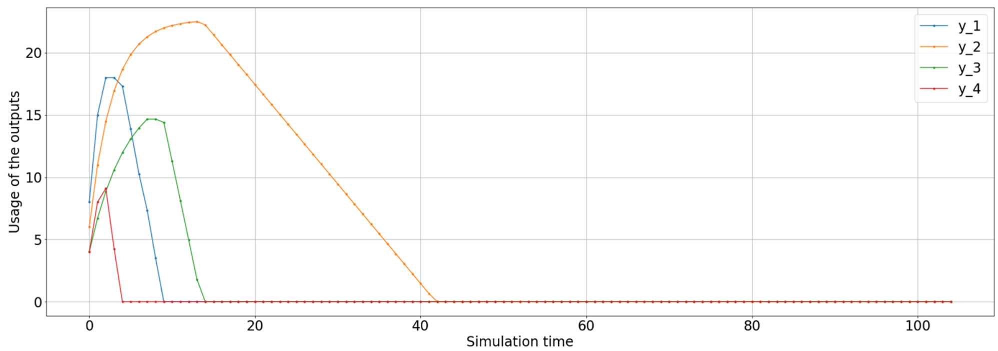

Figure 9. Solution for y-variables of the centralised MPC.

Under this optimal solution, provided by the proposed centralised MPC, the evaluation of the objective cost function is expected to decrease as quickly as possible (concretely, Figure 10 shows that the cost reaches 0 at simulation step 42). While doing so, Figure 8 shows that the constraint related to the flexibility mechanism is always fulfilled (concretely, the sum of the inputs is always below the maximum value 4). These trade-off results in the optimal sequential evolution of the outputs shown in Figure 9.

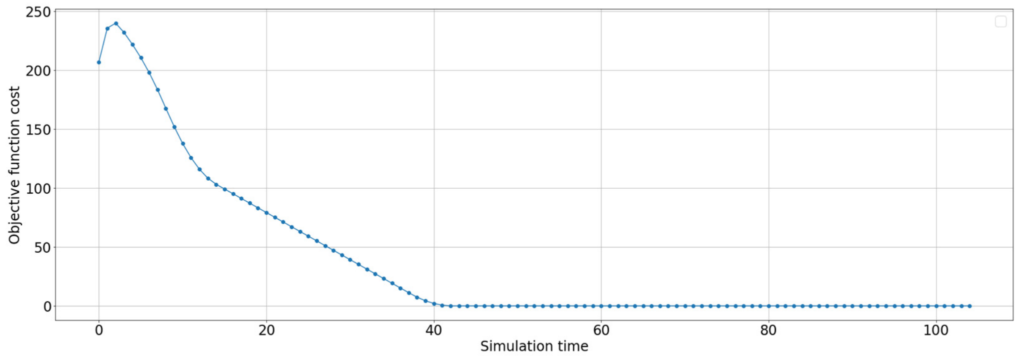

Figure 10. Objective function cost of the centralised MPC.

For benchmarking such an optimal solution, Figures 11–13 summarise the result of the non-optimal decentralised MPC. Specifically, Figure 11 shows the solution for the u-variables (inputs of the nodes), Figure 12 shows the results for the y-variables (outputs of the nodes), and Figure 13 shows the evolution of the sum of the local objective function costs under the decentralised MPC simulation in comparison with the aggregated objective function cost under the centralised MPC.

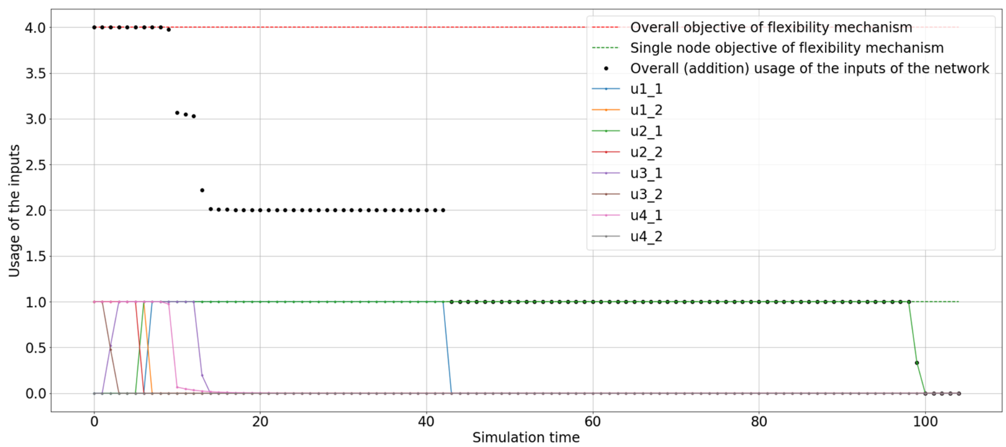

Figure 11. Solution for u-variables of the decentralised MPC.

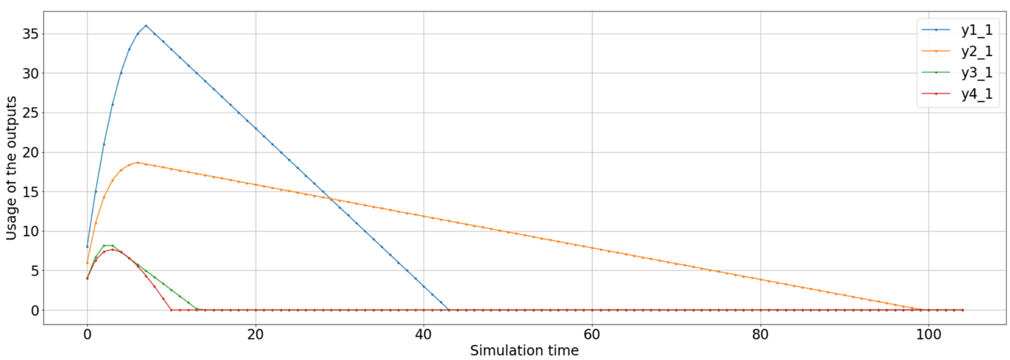

Figure 12. Solution for y-variables of the decentralised MPC.

As expected, the evolution of the sum of the local objective function costs under the non-optimal decentralised MPC is consistently higher than the objective function cost under the optimal centralised MPC (as shown in Figure 14). Specifically, the area under the blue line in Figure 14 (centralised MPC cost) is 3925.4, while the area under the red line in the same figure (decentralised MPC cost) is 9205.6.

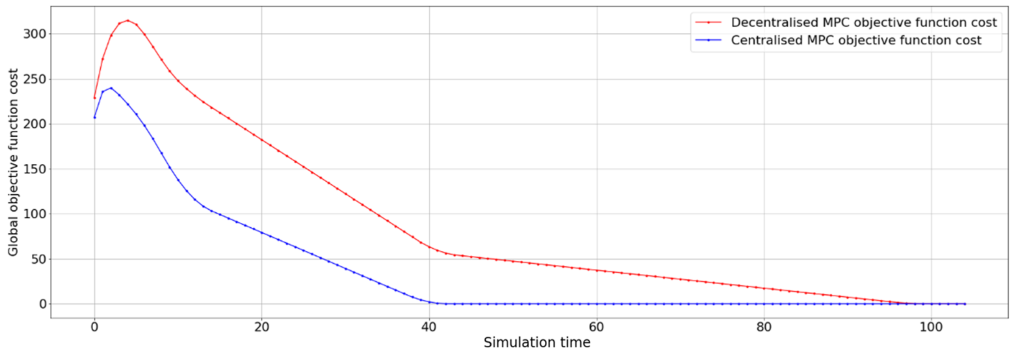

Figure 13. Sum of local objective function costs of the decentralised MPC and objective function cost of the centralised MPC.

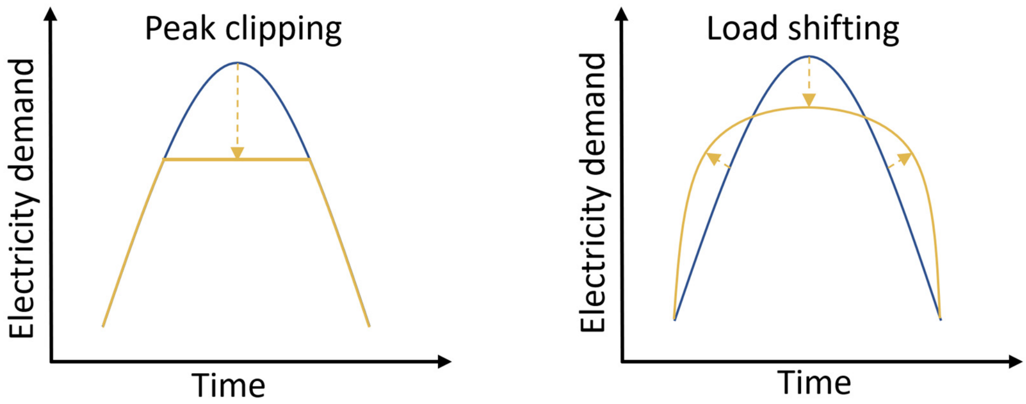

Figure 14. Examples of demand response services.

The higher costs associated with the non-optimal approach are caused by the infra-utilisation of the inputs by the decentralised MPC, as shown in Figure 11, where the sum of the inputs is far from the limit allowed by the flexibility mechanism. This comparison clearly shows the advantage of the proposed centralised MPC over heuristically-based approaches and confirms that the automatic flexibility mechanism designed by the authors is an optimal and efficient method to define joint constraints between several nodes in a large-scale system.

To conclude this section, the following Table 1 summarises all the characteristics of the algorithms that have been analysed in this document.

Table 1. Comparison between centralised MPC and decentralised MPC.

	Centralised MPC	Decentralised MPC
Convergence guarantee	Yes	Yes
Global optimality guarantee	Yes	No
Computational requirements	Medium (*)	Low
Execution time of one instant	Medium (*)	Low
Response time (number of execution instants to reach the optimum)	Low	High (**)
Fulfilment of flexibility targets	High	Medium

(*) It can be reduced with modern technologies. (**) Only guaranteed if the global optimum can be reached or if it can be reached in an amount of time considered reasonable.

6. Extrapolation of Use Case to Energy Flexibility Mechanisms

Thus far in this document, the proposed mathematical formulation has been described generally, so other researchers can easily extrapolate the formulation and conclusions achieved in this work to any kind of system. Nevertheless, the specific application of the proposed framework to energy flexibility mechanisms is straightforward.

On the one hand, the authors recommend adopting the so-called energy hub technique [16] for constructing the models of the smart grids' nodes. On the other hand, the flexibility mechanism can be directly included in the formulation, as presented in Section 3.2.

Specifically, when controlling such kind of large electrical network, it is a common practice to activate a demand response or flexibility mechanism services to ensure the overall consumption is in line with the needs of the grid. For instance, in the event of a consumption peak, the energy supplier asks different clients (nodes) of the network to temporality reduce their consumption to reduce the stress over the network. This is usually called "peak clipping". Another example can be the "load shifting", which consists of incentivising the displacement of the consumption to certain hours with lower demand favouring the equilibrium of the network, so the energy extracted from renewable sources is maximised while the one extracted from no renewable production (gas, carbon, oil, etc.) is not used.

7. Conclusions and Future Work

This research offers a general and effective method for automating the distribution of aggregate constraints between nodes of a large-scale system. Although the use case presented has been maintained as simple to allow for a straightforward analysis, the proposed method can be extrapolated to other more complex large-scale systems without losing effectiveness.

It was demonstrated that the designed centralised MPC achieves the optimal solution and produces more efficient results than other methods used in the control of large-scale systems with aggregate constraints between nodes. Effectiveness has been demonstrated through the benchmarking with a non-optimal decentralised MPC, which prevents communication between nodes and thus forces subdivisions of the flexibility mechanisms.

The flexibility mechanisms are optimally distributed across the instants of the prediction window in the MPC proposed by the authors (Figure 8). The performance achieved in this research presents promising results for industry sectors seeking to implement controllers in large-scale systems. One of the main characteristics of the proposed method, compared to traditional methods, is that it uses the advantages of today's technology without losing the benefits of node modelling. In this way, it is possible to reach the desired results and distribute the constraints efficiently, without complex modelling or mathematics, achieving the most optimal solutions.

The formulation and use case presented in this paper has been designed as generically as possible so that other researchers can use the proposed control algorithm in any large-scale smart grid system. Furthermore, the extrapolation of the proposed framework has been discussed in Section 6 to ease the extrapolation to a specific case. Regarding future work, the authors of this paper will extend the capabilities of the control solution, specifying their use for large-scale energy systems to other use cases.

Author Contributions: Conceptualisation, A.J.d.R.; software, J.D.; validation, A.J.d.R. and J.D.; writing—original draft preparation, A.P.; writing—review and editing, A.P., A.J.d.R. and J.D.; supervision, A.J.d.R. and J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data has been obtained from external sources.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fadaeenejad, M.; Saberian, A.M.; Fadaee, M.; Radzi, M.A.M.; Hizam, H.; AbKadir, M.Z.A. The present and future of smart power grid in developing countries. *Renew. Sustain. Energy Rev.* **2014**, *29*, 828–834. [CrossRef]
2. Kaldellis, J.K.; Zafirakis, D. Optimum energy storage techniques for the improvement of renewable energy sources-based electricity generation economic efficiency. *Energy* **2007**, *32*, 2295–2305. [CrossRef]
3. Shakouri, H.; Kazemi, A. Multi-objective cost-load optimization for demand side management of a residential area in smart grids. *Sustain. Cities Soc.* **2017**, *31*, 171–180. [CrossRef]
4. Shaban, A.; Maher, H.; Elbayoumi, M.; Abdelhady, S. A cuckoo load scheduling optimization approach for smart energy management. *Energy Rep.* **2021**, *7*, 4705–4721. [CrossRef]
5. Kaczmarczyk, V. Optimal Scheduling of Domestic Appliances via MILP. *Energ.—Open Access Energy Res. Eng. Policy J.* **2015**, *8*, 217–232. [CrossRef]
6. Chai, B.; Yang, Z.; Gao, K.; Zhao, T. Iterative Learning for Optimal Residential Load Scheduling in Smart Grid. *Ad Hoc Netw.* **2016**, *41*, 99–111. [CrossRef]
7. Foroozandeh, Z.; Ramos, S.; Soares, J.; Vale, Z. Energy Management in Smart Building by a Multi-Objective Optimization Model and Pascoletti-Serafini Scalarization Approach. *Processes* **2021**, *9*, 257. [CrossRef]
8. Melhem, F.Y.; Grunder, O.; Hammoudan, Z.; Moubayed, N. Optimal residential load scheduling model in smart grid environment. In Proceedings of the 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Milan, Italy, 6–9 June 2017; pp. 1–6. [CrossRef]
9. Arumugasamy, S.K.; Ahmad, Z. Model Predictive Control (MPC) and Its Current Issues in Chemical Engineering. *Chem. Eng. Commun.* **2012**, *199*, 472–511. [CrossRef]
10. Brooks, K.; Westcott, M.; Bauer, M. A Combined MPC for Milling and Flotation—A Simulation Study. *IFAC-PapersOnLine* **2019**, *52*, 24–29. [CrossRef]
11. Garcia-Torres, F.; Báez-Gonzalez, P.; Tobajas, J.; Vazquez, F.; Nieto, E. Cooperative Optimization of Networked Microgrids for Supporting Grid Flexibility Services Using Model Predictive Control. *IEEE Trans. Smart Grid* **2021**, *12*, 1893–1903. [CrossRef]
12. Nevistić, V.; Morari, M. Robustness of MPC-Based Schemes for Constrained Control of Nonlinear Systems. *IFAC Proc. Vol.* **1996**, *29*, 5823–5828. [CrossRef]
13. Allan, D.A.; Bates, C.N.; Risbeck, M.J.; Rawlings, J. B. On the inherent robustness of optimal and suboptimal nonlinear MPC. *Syst. Control Lett.* **2017**, *106*, 68–78. [CrossRef]
14. Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer: London, UK, 2007; ISBN 978-0-85729-398-5.
15. Rodrigues, M.; Odloak, D. MPC for Stable Linear Systems with Model Uncertainty. *Automatica* **2003**, *39*, 569–583. [CrossRef]
16. Geidl, M.; Koeppel, G.; Favre-Perrod, P.; Klockl, B.; Andersson, G.; Frohlich, K. Energy hubs for the future. *IEEE Power Energy Mag.* **2007**, *5*, 24–30. [CrossRef]
17. Jamshidi, M. *Controls, Large-Scale Systems. Encyclopedia of Physical Science and Technology*, 3rd ed.; Academic Press: Cambridge, MA, USA, 2003; pp. 675–686. [CrossRef]
18. Del Real, A. An integrated framework for distributed model predictive control of large scale networks. Applications to power networks. Doctoral Dissertation, University of Seville, Seville, Spain, 2011.
19. Du, Y.; Wu, J.; Li, S.; Long, C.; Paschalidis, I.C. Distributed MPC for Coordinated Energy Efficiency Utilization in Microgrid Systems. *IEEE Trans. Smart Grid* **2019**, *10*, 1781–1790. [CrossRef]
20. Ananduta, W.; Maestre, J.M.; Ocampo-Martinez, C.; Ishii, H. Resilient distributed model predictive control for energy management of interconnected microgrids. *Optim. Control. Appl. Methods* **2020**, *41*, 146–169. [CrossRef]
21. Del Real, A.J.; Arce, A.; Bordons, C. An Integrated Framework for Distributed Model Predictive Control of Large-Scale Power Networks. *IEEE Trans. Ind. Inform.* **2014**, *10*, 197–209. [CrossRef]
22. Valderrama Bento da Silva, P.H.; Camponogara, E.; Seman, L.O.; Villarrubia González, G.; Reis Quietinho Leithardt, V. Decompositions for MPC of Linear Dynamic Systems with Activation Constraints. *Energies* **2020**, *13*, 5744. [CrossRef]
23. Trodden, P.; Richards, A. Cooperative distributed MPC of linear systems with coupled constraints. *Automatica* **2013**, *49*, 479–487. [CrossRef]
24. Morosan, P.-D.; Bourdais, R.; Dumur, D.; Buisson, J. Distributed MPC for Multi-zone Temperature Regulation with Coupled Constraints. *IFAC Proc. Vol.* **2011**, *44*, 1552–1557. [CrossRef]
25. Gommers, S.; Lazar, M. Smart decentralized MPC for temperature control in multi–zone buildings. In Proceedings of the 29th Mediterranean Conference on Control and Automation (MED), Puglia, Italy, 22–25 June 2021; pp. 415–420. [CrossRef]
26. El Houda Hammami, D.; Maraoui, S.; Bouzrara, K. Implementation of MPC control for quadruple-tank system using LabView: A decentralized approach. In Proceedings of the 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 24–26 March 2019; pp. 41–46. [CrossRef]