Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-6 (2022) 704–709

A distributed scenario-based stochastic MPC for fault-tolerant microgrid energy management

Vittorio Casagrande * **Francesca Boem** *

* *Dept. of Electronic and Electrical Engineering, University College London, UK. (e-mail:{vittorio.casagrande.19, f.boem} @ucl.ac.uk)).*

Abstract: This paper proposes a fault-tolerant energy management algorithm for microgrid systems composed of several agents. The method stems from the necessity to design an algorithm that takes explicitly into account the possibility of faults and their consequences to avoid solutions which are excessively conservative. A tree of possible fault scenarios is built in a completely distributed way by all the agents of the network; then the resulting optimization problem is solved through a distributed algorithm which not only does not require a high computational power for each agent, but keeps also private all local data and decision variables. The effectiveness of the proposed method is proved through simulation results.

Keywords: Energy management system, Fault-tolerant control, Microgrid, Stochastic MPC, Distributed optimization

1. INTRODUCTION

In this paper we propose a novel distributed and fault-tolerant method for energy management in microgrid systems. The Energy Management System (EMS), as a part of the microgrid tertiary control layer, is the controller that computes the power flows to provide a stable delivery of power to loads, while optimizing energy production and other operational goals (Yoldaş et al. (2017)). Multiple variants of optimization-based algorithms for energy management have been proposed, in particular, methods based on Model Predictive Control (MPC) are very successful thanks to their ability to efficiently compensate uncertainty and handle constraints. Although being easier to implement than decentralized algorithms, centralised MPC schemes may not be suitable for large-scale systems due to communication and computational constraints or in the case agents do not want to share their performance indices and decision variables with a central unit. Indeed, in recent years privacy issues have become increasingly important: for example, sharing the load demand profile can reveal details of a manufacturing process as illustrated in Mak et al. (2019). Instead, our method allows a distributed implementation of the MPC controller preserving privacy of load and generation profiles, local decision variables, objective functions and constraints. Distributed methods have been proposed not only to increase the privacy but also to distribute the computation among the agents of a network. Examples of distributed algorithms for EMS design can be found in Dehghani-Pilehvarani et al. (2019), Zheng et al. (2017). The main drawback of such methods is that either they are not fully distributed or they do not guarantee privacy. Instead, the method we propose does not require any coordination unit, nor the exchange of power profiles and sensitive information. Panteli and Mancarella (2015) clearly show the importance of a reliable electrical infrastructure. Control algorithms can increase the resilience of electrical systems to deal with the uncertainties that characterize microgrid operation, such as renewable power production, as explained in Hussain et al. (2019). A resilient controller not only has to predict possible faults and take their effects into account, but also has to ensure that the system is in appropriate safe conditions when the fault occurs to deal with its evolution. Although there is an extensive literature on energy management, a few papers consider the possible occurrence of faults. Jongerden et al. (2016), Ghasemieh et al. (2015), Haessig et al. (2019) and Prodan et al. (2015) propose to store a certain amount of backup energy in the storage system to sustain the system operation during fault events. The main drawback of such centralised methods is that a certain amount of stored energy is always committed to fault tolerance, hence the storage capacity cannot be used for economic goals or peak shaving/valley filling. On the other hand, in this paper we propose a method to deal with faults which is based on a distributed scenario-based stochastic MPC (SMPC) which allows to deal with uncommon events and uncertainty avoiding excessively conservative solutions. In particular, scenario-based SMPC, proposed by Calafiore and Campi (2006) and Bernardini and Bemporad (2009), has been used in Hans et al. (2015) to schedule the microgrid operation taking into account the stochastic nature of renewable generators. However, they do not consider the possible presence of faults and being the method centralized, it suffers from computational complexity issues. In this paper instead, by leveraging distributed optimization algorithms, it is possible to deal with the complexity which is typical of scenario optimiza-

⋆ *This work has been partially supported by the PHC Alliance Hubert Curien Programme (project code: 43234RH) and Engineering and Physical Sciences Research Council (grant reference: EP/R513143/1)

10.1016/j.ifacol.2022.07.210

tion. A distributed scenario-based approach is proposed in Long et al. (2014), for house temperature control, and in Velarde et al. (2019), for water resources management. In particular, in the last paper a scenario tree is built in a centralized way and then each local controller considers its relevant scenario only. Unfortunately such approach cannot be implemented in microgrids since in these systems each scenario affects the operation of all agents. Moreover, the aforementioned distributed algorithms do not consider privacy related issues, fundamental for microgrids. Scenario-based MPC has already been exploited by Bø and Johansen (2014) to design fault tolerant controllers in centralized frameworks. This paper builds on recent preliminary work (Casagrande et al. (2021)). Differently from existing scenario-based SMPC approaches, in this paper we propose a new method for each agent to compute a scenario tree in a distributed way and we leverage the distributed algorithm proposed in Falsone et al. (2017) to solve the resulting optimization problem. There are several advantages in the proposed method. Firstly, a potentially large-scale optimization problem is solved in parallel by a number of processors, requiring a small computational power. Secondly, it allows to keep private local information (as opposed to Long et al. (2014) and Velarde et al. (2019)). Thirdly, stochastic methods allow to obtain less conservative solutions of the control problem than the aforementioned robust methods (e.g. Prodan et al. (2015)), allowing to store an amount of energy to sustain the microgrid during faults based on the fault probability of occurrence. The controller is reconfigured at each time step to take into account the effects of the fault after its occurrence. The remainder of the paper is organized as follows. In Section 2 the microgrid model, the communication network and the possible faults are described. In Section 3 the proposed energy management algorithm is presented. Finally, in Sections 4 and 5 simulation results are given and conclusions are drawn.

2. SYSTEM MODEL

We assume that the microgrid system is composed of four types of agents: loads, renewable generators, storage systems and connections to the main grid. Each agent is assumed to be equipped with a local controller with computation and communication capabilities.

Microgrid Each load agent is characterized by a power demand at each time step t which is composed of a critical and a non-critical part:

$$d_i^m(t) \le P_{l,i}(t) \le d_i^M(t) \tag{1}$$

where $P_{l,i}(t)$ is the power drawn by load i at time t, $d_i^m(t)$ is the minimum (or critical) load demand and $d_i^M(t)$ is the target power demand. Each load agent will try to draw the target power demand depending on the power availability in the microgrid, hence the finite horizon objective function of each load is set to:

$$J_{l,i}(t) = \sum_{k=t}^{t+T-1} w_{l,i} \left[P_{l,i}(k) - d_i^M(k)\right]^2 \tag{2}$$

where T is the time horizon and $w_{l,i}$ is a weight term used to balance all the objectives.

Renewable generator agents produce the power $P_{r,i}$ that is injected in the microgrid. The maximum amount of power that can be injected is bounded by the maximum produced power $P_{r,i}^M$:

$$0 \le P_{r,i}(t) \le P_{r,i}^M(t). \tag{3}$$

The goal of each renewable generator agent is to maximise the power sold to the other agents, hence its objective function can be written as:

$$J_{r,i}(t) = \sum_{k=t}^{t+T-1} w_{r,i}\gamma^k \left[P_{r,i}(k) - P_{r,i}^M(k)\right]^2, \tag{4}$$

where $w_{r,i}$ is a weight term used to balance the objective functions, T is the time horizon and $\gamma \in [0;1]$ is a weight used to reduce progressively the importance of time steps that are more uncertain.

Storage systems are used along with renewable generators to compensate their intermittence and to prevent load fluctuations. For energy management purposes the specific employed storage technology is not significant, hence we model the storage system as in Prodan et al. (2015) and Parisio et al. (2014) as a first order linear system:

$$s_i(t+1) = s_i(t) + \mu_{i,c/d} T_s P_{s,i}(t), \tag{5}$$

where $s_i(t)$ is the state of charge of storage i, $\mu_{i,c/d}$ is the energy conversion efficiency for charging and discharging ($\mu_{i,c} < 0$ and $\mu_{i,d} > 0$), T_s is the sample time of the controller and $P_{s_i}(t)$ is the power exchanged with the microgrid. The amount of energy which can be stored in a storage system is limited by a maximum and a minimum value:

$$s_i^m \le s_i(t) \le s_i^M. \tag{6}$$

In order to increase the battery lifetime and avoid a quick degradation of its performance the lower limit of charge is higher than zero (the usable energy is typically between 80% and 95% of the total energy). The maximum value of power that can be exchanged with the grid is denoted by $P_{s_i}^M$:

$$-P_{s,i}^M \le P_{s,i}(t) \le P_{s,i}^M. \tag{7}$$

The objective of the storage system agent is expressed in terms of power and state of charge:

$$J_{s,i}(t) = \sum_{k=t}^{t+T-1} w_{P_{s,i}} \left[P_{s,i}(k) - \bar{P}_{s,i}(k)\right]^2 + w_{s,i}\left[s_i(k) - \bar{s}_i(k)\right]^2, \tag{8}$$

where $w_{P_{s,i}}$ and $w_{s,i}$ are weighting parameters and $\bar{P}_{s,i}(k)$ and $\bar{s}_i(k)$ are the target values. Such objective function allows to penalise the battery setting $\bar{P}_{s,i}(k)$ to zero.

Typically, a microgrid is connected to the main distribution grid in one or more points so that power can be bought or sold. The amount of power which flows through the utility grid connection is limited by a maximum and a minimum value:

$$P_{g,i}^m(t) \le P_{g,i}(t) \le P_{g,i}^M(t). \tag{9}$$

The power is assumed to be negative when the microgrid absorbs it from the utility grid. The goal of the external tie agent is to maximise the profit due to energy trading while meeting the requirement of the microgrid:

$$J_{g,i}(t) = -\sum_{k=t}^{t+T-1} \lambda_g(k) P_{g,i}(k), \tag{10}$$

where $\lambda_g(t)$ is the electricity price at time t and T is the time horizon. The minus sign is introduced since the power sign is positive when energy is sold to the utility grid, hence we penalize buying.

All the components of the microgrid are coupled by the power balance constraint, that is, the sum of powers exchanged with the microgrid has to be equal to zero. Such constraint is expressed as follows:

$$\sum_{i=1}^{N_l} P_{l,i}(t) + \sum_{i=1}^{N_s} P_{s,i}(t) + \sum_{i=1}^{N_g} P_{g,i}(t) = \sum_{i=1}^{N_r} P_{r,i}(t), \quad (11)$$

where N_r is the number of renewable generators, N_l is the number of loads, N_s is the number of storage systems and N_g is the number of connections to the utility grid.

Communication network Agents communicate with each other through the communication network which is modelled as a undirected graph $G(\mathcal{V}, \mathcal{E})$ in which $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node represents an agent of the network, hence the total number of agents of the network denoted by $N = |\mathcal{V}| = N_r + N_l + N_s + N_g$.

Faults We consider distribution grid faults and renewable generator faults. In the first case, the maximum power that flows through the microgrid connection to the utility grid is reduced. This can happen for example, in the case of a damage on the transformer that interconnects the microgrid with the utility grid or, in the case of a power outage, thus making the microgrid operate in island mode. This can be mathematically modelled by introducing two factors $\alpha_i, \beta_i \in [0; 1]$ in the constraint (9):

$$\alpha_i P_{g,i}^m(t) \le P_{g,i}(t) \le \beta_i P_{g,i}^M(t), \;\; \forall \;\; t \in \left[\tau_{g,i}^i, \tau_{g,i}^f\right], \quad (12)$$

where $\tau_{r,i}^i$ and $\tau_{r,i}^f$ are the time steps in which the utility grid fault starts and ends respectively.

In the second case, the maximum amount of power that can be supplied by the renewable generator decreases. This can be modelled by introducing a factor $\epsilon_i \in [0; 1]$ in constraint (3):

$$0 \le P_{r,i}(t) \le \epsilon_i P_{r,i}^M(t) \;\; \forall \;\; t \in \left[\tau_{r,i}^i, \tau_{r,i}^f\right], \quad (13)$$

where $\tau_{r,i}^i$ and $\tau_{r,i}^f$ are the time steps in which the renewable generator fault starts and ends, respectively.

3. ENERGY MANAGEMENT SYSTEM

In this section the novel SMPC-based algorithm employed to solve the energy management problem is described.

3.1 Scenario-based SMPC

In the SMPC framework the objective is to obtain less conservative results with respect to robust MPC algorithms when dealing with uncertainties, by taking advantage of known probability distribution of the disturbances. In particular, at each time step a stochastic optimal control problem is solved in which the goal is to minimize the expected performance over the prediction horizon T:

$$\mathbb{E}\left[\sum_{k=t}^{t+T-1} J(k)\right]. \quad (14)$$

The scenario-based SMPC framework requires first to build a scenario tree which is composed of several nodes from the root (corresponding to the current time step) to the leaves where each edge corresponds to a possible scenario. Before delving into the algorithm some useful definitions and notation regarding scenario trees are presented. The tree in Figure 1 is used as an example. The index k denotes the time step in the prediction horizon $k \in [t, t+1, \dots T+t-1]$. The set of all nodes of the tree

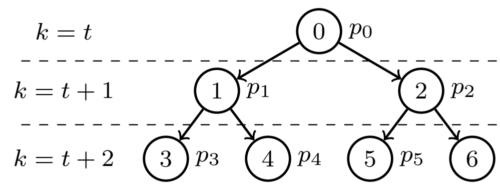

Fig. 1. Example of a scenario tree. The node ID index j is written inside each node, whereas probabilities p_j are the probabilities to reach node j.

is denoted by S, whereas the set of nodes related to time step k is denoted as S_k. The nodes are numbered from the root node (node 0 for $k = t$) to the leaves nodes (at $k = t + T - 1$), hence the set $S_0 = \{0\}$ in the example in Figure 1. The set C_m denotes the set of children of node m, for example, referring to Figure 1, $C_1 = \{3, 4\}$. A probability p_j is associated to each node j of the tree. In this work we model the transition probability to pass from one node to the next as a discrete time Markov process. Hence, the expected performance (14) is expressed as a sum over all the possible nodes of the scenario tree weighted by the probability to reach the node:

$$\sum_{k=t}^{t+T-1} \sum_{j \in S_k} p_j J_j(k), \quad (15)$$

where J_j is the cost associated to node j.

3.2 Scenario tree generation

In this paragraph, the novel algorithm employed by the agents of the network to compute the scenarios is outlined. Differently to Bø and Johansen (2014) where a centralized architecture is considered, the scenario tree is built by a network of interconnected agents based on the local fault states and transition probabilities.

Assumption 1. Each agent i of the microgrid is provided with a unique identification number and its functionality is characterized by a current fault state $f_i(t) \in F_i$. The set F_i is the set of the n_i possible fault states associated to agent i:

$$F_i = \{\varphi_i^1, \varphi_i^2, \dots \varphi_i^{n_i}\} \quad (16)$$

where $j \in [1, \dots n_i]$ is the fault state index and φ_i^j may represent the healthy state, different faults or increasingly serious levels of faults. Moreover, we assume that each local fault state model is known by agent i, i.e. variables α_i, β_i (for each external tie agent) and ϵ_i (for each renewable generator agent) corresponding to each fault state φ_i^j.

Assumption 2. The transition probability between fault states can be modelled as a discrete time Markov process. Furthermore, we assume that the probability matrix of agent i, whose entries are the time varying transition probabilities between the current and the future fault states, is known to agent i and it is denoted by $P_i(t) \in \mathbb{R}^{n_i \times n_i}$.

The algorithm that is used by each agent i to locally build the scenario tree needed for the proposed distributed SMPC is now explained, considering the perspective of agent i at time step t.

(1) Agent i shares with the network: (i) its identification number i; (ii) its current transition probability matrix $P_i(t)$; (iii) the index $j \in [1, \dots n_i]$ corresponding to its current fault state $f_i(t)$ of set F_i. Such information can be shared, for example, running the flooding algorithm described in Tanenbaum (2006).

(2) Once all the previous information is received from all the agents of the network, agent i computes:
- the combinations of failure states labels $\bar{F}$, as the Cartesian product of the set of possible failure states labels $\bar{F} = \bar{F}_1 \times \bar{F}_2 \times \cdots \times \bar{F}_N$. The sets $\bar{F}_n$, $n = 1 \dots N$, only contain the indexes or labels of the fault states and are identified by a bar because the real sets F_i are not shared. However the cardinality of set F_i can be inferred by the number of rows (or columns) of the matrix P_i (which is shared).
- the global transition probability matrix $P(t)$, as the Kronecker product of the transition probabilities:

$$P(t) = P_1(t) \otimes P_2(t) \otimes \cdots \otimes P_N(t) \qquad (17)$$

Remark 1. The constraints corresponding to the fault state are implemented locally by each agent via equations (12) and (13) hence keeping information about the current fault state private.

Remark 2. The identification number transmitted at step 1 ensures that all agents compute the Kronecker product (which is not commutative) of (17) in the correct order.

(3) Agent i builds the scenario tree, given the MPC prediction horizon length T, the set of combinations of possible failure states labels $\bar{F}$ and the transition probability matrix of the current time step $P(t)$. In particular, the probability to reach node $l \in C_m$ from node m can be computed applying the recursive rule $p_l = p_m P_{[c;d]}$ where $P_{[c;d]}$ is the entry of matrix P that represents $P_{[c;d]} = \mathbf{Pr}\left[f(k+1) = \bar{\varphi}^c | f(k) = \bar{\varphi}^d\right]$ in which $\bar{\varphi}^j$, $j = 1, \dots, |\bar{F}|$, are the elements of $\bar{F}$. For the root node we set (probability to reach the current node): $p_0 = 1$. Clearly such scenario tree has two properties: (i) the sum of the probabilities to reach any of the nodes at the optimisation step k is one: $\sum_{j \in S_k} p_j = 1$; (ii) the probability to reach any children node of m from node m is equal to the probability to reach node m itself: $\sum_{j \in C_m} p_j = p_m$.

Remark 3. The fault state index transmitted at step 1 is required to use the correct line of the matrix P to compute the transition probabilities from the current fault state.

Once the scenario tree is built, each agent uses the probabilities to reach each node as weights in their local objective function.

3.3 Distributed optimization-based MPC

The energy management problem is formulated as an SMPC problem to be solved using a distributed optimization algorithm. At each time step the following constrained-coupled optimization problem (Notarstefano et al. (2019)) is solved:

$$\min_{\{\mathbf{x}_1, \dots \mathbf{x}_N\}} \quad \sum_{i=1}^{N} f_i(\mathbf{x}_i) \qquad (18a)$$

$$s.t. \quad \mathbf{x}_i \in X_i, \qquad (18b)$$

$$\sum_{i=1}^{N} \mathbf{g}_i(\mathbf{x}_i) \leq 0. \qquad (18c)$$

The local decision variable $\mathbf{x}_i$ is the future power profile of agent i over the prediction horizon for all the nodes $j \in S$ of the scenario tree (realizations of the faults):

$$\mathbf{x}_i = \{P_i^0(t), \dots P_i^j(t+k), \dots P_i^{n_{dv}}(t+T-1)\} \qquad (19)$$

The global objective function to minimize is the expected value of the sum of local objective functions of all agents:

$$\sum_{i=1}^{N} f_i(\mathbf{x}_i) = \mathbb{E}\left[\sum_{i=1}^{N_r} J_{r,i} + \sum_{i=1}^{N_l} J_{l,i} + \sum_{i=1}^{N_s} J_{s,i} + \sum_{i=1}^{N_g} J_{g,i}\right]. \qquad (20)$$

The coupling constraint (18c) is used to model the microgrid interconnection (which is equation (11)) and it is expressed through the function $\mathbf{g}_i$ that has to be specified for each agent. Clearly this constraint has to be enforced for all the nodes of the scenario tree, hence the number coupling constraints is equal to the number of decision variables n_{dv} of each agent. The optimization problem is solved using the distributed algorithm proposed in Falsone et al. (2017). The details of the algorithm are not given due to space constraints, however it consists of three steps from the perspective of agent i: (1) to gather the dual variables of the optimization problem from the neighbours; (2) to compute the local primal variable x_i by minimizing the i-th part of the Lagrangian; (3) to update the local estimate of the dual variable.

3.4 On the computational complexity

The number of nodes of the scenario tree grows with the number of fault states combinations $\bar{\varphi}^j \in \bar{F}$ and with the MPC prediction horizon length. Hence, in order to set the MPC prediction horizon and the number of fault states one has to consider the computational power available to each agent. The number of nodes of the scenario tree, which corresponds to the number of optimization variables that each agent has to compute, can be calculated as $n_{dv} = \sum_{i=0}^{T} |\bar{F}|^i$, in which $|\bar{F}|$ denotes the cardinality of set $\bar{F}$. An alternative approach to reduce the complexity of the optimization problem is to adopt a threshold so that, if the probability of a failure state is lower than the threshold, then such faulty state is not considered (for example, if the transition between faulty state "a" and "b" occurs with a probability 0.05 and the threshold is set to 0.1).

4. SIMULATION RESULTS

In this section the results obtained by applying the proposed stochastic EMS algorithm is applied to the microgrid of Zafeiratou (2020) which is composed of four

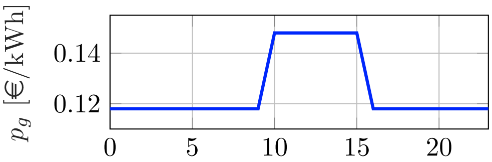

Fig. 2. Electricity price profile

components: (i) agent 1 is the external tie; (ii) agent 2 is the renewable generator; (iii) agent 3 is the load; (iv) agent 4 is the storage system. The proposed stochastic MPC controller considers in its scenario tree that fault might occur in the future on the external tie or the renewable generator agent. The sets F_i and the matrices P_i are set as: $F_1 = F_2 = \{n, d\}$, $F_3 = F_4 = \{n\}$, $P_1 = P_2 = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix}$ and $P_3 = P_4 = \begin{bmatrix} 1 \end{bmatrix}$. Intuitively, setting P_1 and P_2 as above means that, if the system is "n" (normal functioning state), there is a 50% probability that it will be damaged (i.e. pass to state "d") at the next time step and, if the system is faulty, it will remain faulty in the optimization horizon. This controller, denoted as C1, will be compared with other two controllers. Controller 2 (denoted as C2), which is the controller presented in Casagrande et al. (2021), is a MPC controller based on distributed optimization which does not consider the possible occurrence of a fault. Controller 3 (denoted as C3) is a robust MPC controller designed to maintain a minimum charge in the storage systems at each time step sufficient to supply the load demand for a certain number of future time steps. In particular, we define $\tilde{r}_\nu(t) = \sum_{\tau=t}^{t+\nu-1} T_s d^m(\tau)$ as the critical energy demand of the loads for the next ν time steps. The following constraint is then added to the local optimization problem of the storage agent:

$$s(t+k) \geq \tilde{r}_\nu(t+k) \quad \forall \quad k \in [0, T-1] \tag{21}$$

As soon as a fault is detected, such constraint is deactivated in order to allow energy to flow from the storage to the load. Controller C3 is robust since it maintains the minimum charge level $\tilde{r}_\nu(t)$ regardless the fault occurrence probability and it assumes that the storage will be the only component that can support the load operation during faults. Using C2 and C3 each agent has to compute one optimization variable for each step of the prediction horizon. The main simulation parameters and the electricity price profile are given in Table 1 and Figure 2.

Param.	Value	Param.	Value	Param.	Value
T	4	w_s	0	ν	4
T_s	1h	w_{P_s}	0.1	$\mu_{i,c/d}$	0.98/1.02
γ	0.9	w_r	1	w_l	1
α_i	0	β_i	0	ϵ_i	0

Table 1. Simulation parameters.

The following performance indices are defined to compare the aforementioned controllers:

- energy delivered to loads during fault events with respect to the maximum load demand: $E_l^{\%} = \frac{E_l}{E_l^M} \times 100$, where E_l is the energy supplied to loads during faults (i.e. in the time interval $t \in \left[\tau_{g,i}^i, \tau_{g,i}^f\right]$) and E_l^M is the maximum demand of loads during faults;
- energy drawn from the renewable generator with respect to the total available energy: $E_r^{\%} = \frac{E_r}{E_r^M} \times 100$, where E_r is the energy drawn by renewable generators and E_l^M is the maximum energy produced by the renewable generators over the whole simulation period;
- battery utilization over the simulation period: $|P_s| = \sum_{i=1}^{N_s} \sum_{k=0}^{23} |P_{s,i}(k)|$. Since repeated charges and discharges of the battery decrease its lifetime, a controller with low battery utilization is to be preferred;
- total energy cost: $\mathrm{C} = \sum_{i=1}^{N_g} \sum_{k=0}^{23} \lambda_g(k) P_{g,i}(k)$,.

For this simulation we enabled 4 out of 6 communication links. Simulations were implemented using DISROPT Python package (Farina et al. (2019)) and the number of iterations of the distributed optimization algorithm has been set to 1500. A distribution grid fault is simulated between time steps $t = 18$ and $t = 21$, the fault interval is highlighted in gray in Figure 3. At $t = 18$, when the fault occurs, controller C1 has stored a greater amount of energy than C2 and C3. When the fault occurs power cannot be exchanged with the utility grid anymore (top left plot). Since the renewable power production is close to zero, the load can draw energy only from the storage system. All the controllers manage to supply the critical power demand to the load, however since the C1 and C2 stored a greater amount of energy would be able to supply the load for longer time. Since constraint (21) is deactivated during faults, the storage energy drops below the lower limit $\tilde{r}_\nu(t)$ (bottom plot of Figure 3) for C3. Table 2 highlights the main differences of the three controllers. The amount of energy supplied to the load during the fault event is similar for C2 and C3 and it is lower than the one supplied by C1. Energy provided by the renewable generator is similar for the three controllers. The main difference is clearly in the battery utilization and in the total energy price. Although C2 allows to achieve the results with a lower energy price, the battery utilization using C1 is significantly lower. Moreover, the amount of stored energy at the end of the fault is higher using C1 than other controllers, hence allowing it to continue to provide energy in case of persistent faults. It is clear that all the controllers offer advantages and disadvantages which have to be taken into consideration when designing the EMS. While the C2 offers economical benefit in the short-term in the considered scenario, since the total energy cost is lower, in the long-term C1 may be beneficial since it allows a lower battery usage and thus an increased expected battery lifetime. For the sake of clarity the matrices P_i are assumed to be constant for C1 over the simulation, however such variable may change at each time step depending on the situation of each agent. Hence, when a fault is improbable, the behaviour of C1 would be similar to the C2, whereas, when the fault probability is high, it behaves accordingly combining the advantages of both the controllers.

EMS	$E_l^{\%}$ [%]	$E_r^{\%}$ [%]	$\lvert P_s \rvert$ [W]	C [€]
C1	30.6	97.8	2937	2.20
C2	25.8	95.6	3949	1.96
C3	26.6	93.3	3648	2.14

Table 2. Numerical results.

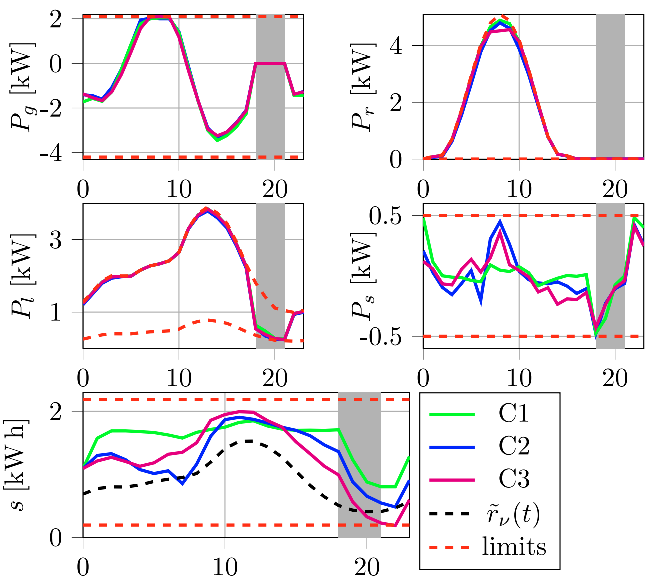

Fig. 3. Power profiles of scenario 2. The fault interval is highlighted in gray.

5. CONCLUSION

In this paper a distributed fault-tolerant algorithm for microgrid energy management has been presented. Such algorithm is based on scenario-based SMPC and it computes the future control actions based on the current probability of fault occurrence. The algorithm is fully distributed hence allowing to deal with possibly large-scale problems with a lot of involved agents while maintaining private local information. As a future work, we will investigate methods to reduce the number of nodes of the scenario tree to allow the increase of the prediction horizon and the number of faulty states without incurring in computational issues.

REFERENCES

Bernardini, D. and Bemporad, A. (2009). Scenario-based model predictive control of stochastic constrained linear systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 6333–6338. IEEE.

Bø, T.I. and Johansen, T.A. (2014). Dynamic safety constraints by scenario based economic model predictive control. *IFAC Proceedings Volumes*, 47(3), 9412–9418.

Calafiore, G.C. and Campi, M.C. (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51, 742–753. doi: 10.1109/TAC.2006.875041.

Casagrande, V., Prodan, I., Spurgeon, S.K., and Boem, F. (2021). A robust distributed-optimization-based mpc for microgrid energy management. In *2021 European Control Conference (ECC)*, 2549–2554. IEEE.

Dehghani-Pilehvarani, A., Markou, A., Ferrarini, L., et al. (2019). A hierarchical distributed predictive control approach for microgrids energy management. *Sustainable Cities and Society*, 48, 101536.

Falsone, A., Margellos, K., Garatti, S., and Prandini, M. (2017). Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica*, 84, 149–158.

Farina, F., Camisa, A., Testa, A., Notarnicola, I., and Notarstefano, G. (2019). Disropt: a python framework for distributed optimization. *arXiv preprint arXiv:1911.02410*.

Ghasemieh, H., Haverkort, B.R., Jongerden, M.R., and Remke, A. (2015). Energy resilience modelling for smart houses. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 275–286. IEEE.

Haessig, P., Bourdais, R., Guéguen, H., et al. (2019). Resilience in energy management system: A study case. *IFAC-PapersOnLine*, 52(4), 395–400.

Hans, C.A., Sopasakis, P., Bemporad, A., Raisch, J., and Reincke-Collon, C. (2015). Scenario-based model predictive operation control of islanded microgrids. In *2015 54th IEEE conference on decision and control (CDC)*, 3272–3277. IEEE.

Hussain, A., Bui, V.H., and Kim, H.M. (2019). Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience. *Applied energy*, 240, 56–72.

Jongerden, M.R., Hüls, J., Remke, A., and Haverkort, B.R. (2016). Does your domestic photovoltaic energy system survive grid outages? *Energies*, 9(9), 736.

Long, Y., Liu, S., Xie, L., and Johansson, K.H. (2014). A scenario-based distributed stochastic mpc for building temperature regulation. In *2014 IEEE international conference on automation science and engineering (CASE)*, 1091–1096. IEEE.

Mak, T.W., Fioretto, F., Shi, L., and Van Hentenryck, P. (2019). Privacy-preserving power system obfuscation: A bilevel optimization approach. *IEEE Transactions on Power Systems*, 35(2), 1627–1637.

Notarstefano, G., Notarnicola, I., and Camisa, A. (2019). Distributed optimization for smart cyber-physical networks. *arXiv preprint arXiv:1906.10760*.

Panteli, M. and Mancarella, P. (2015). Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies. *Electric Power Systems Research*, 127, 259–270.

Parisio, A., Rikos, E., and Glielmo, L. (2014). A model predictive control approach to microgrid operation optimization. *IEEE Transactions on Control Systems Technology*, 22(5), 1813–1827.

Prodan, I., Zio, E., and Stoican, F. (2015). Fault tolerant predictive control design for reliable microgrid energy management under uncertainties. *Energy*, 91, 20–34.

Tanenbaum, A.S. (2006). *Computer networks*. Prentice-Hall of India Private Limited.

Velarde, P., Tian, X., Sadowska, A.D., and Maestre, J.M. (2019). Scenario-based hierarchical and distributed mpc for water resources management with dynamical uncertainty. *Water Resources Management*, 33, 677–696. doi:10.1007/s11269-018-2130-2.

Yoldaş, Y., Önen, A., Muyeen, S., Vasilakos, A.V., and Alan, İ. (2017). Enhancing smart grid with microgrids: Challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 72, 205–214.

Zafeiratou, I. (2020). Hierarchical control of a meshed dc microgrid through constrained. *doctoral dissertation*.

Zheng, Y., Li, S., and Tan, R. (2017). Distributed model predictive control for on-connected microgrid power management. *IEEE Transactions on Control Systems Technology*, 26(3), 1028–1039.