

Robust Real-Time Control of Power Grids in the Presence of Communication Network Non-Idealities

Wajeb Saab*, Roman Rudnik*, Lorenzo Reyes-Chamorro[†], Jean-Yves Le Boudec* and Mario Paolone[†]

* Laboratory for Communications and Applications 2, {wajeb.saab, roman.rudnik, jean-yves.leboudec}@epfl.ch

[†] Distributed Electrical Systems Laboratory, {lorenzo.reyes, mario.paolone}@epfl.ch
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Abstract—Deploying a power grid controller in the field makes it susceptible to message losses caused by the inherent uncertainties and non-idealities of communication networks, especially when the control action is taken at a sub-second time-scale. We consider a centralized power grid controller that monitors and controls resources in real-time. The resources send *advertisements* that contain information about their state, and an estimation of their behavior in the time horizon when the control action is expected to be implemented. The controller uses this information to compute and issue setpoints that are thus only valid for this time horizon. An occasional loss of one or more advertisements might render the controller incapable of issuing valid setpoints. We introduce advertisements with a longer-term prediction interval, which are constantly sent along with the short-term ones, and can be used by the controller when it is missing information from some or all resources. We show the advantages of using such an approach on a controller that, by exploiting local resources flexibilities, performs frequency support on the CIGRÉ benchmark low-voltage microgrid.

Index Terms—Robust, Reliability, Uncertainty, Real-Time, Power Grids, Networks

I. INTRODUCTION

A. Background

The trend in power distribution-grid control is to aggregate in real-time the network flexibilities in order to achieve local and global objectives. Controllers that implement real-time (sub-second) functionalities, will occasionally experience communication network non-idealities during deployment, such as message losses and delays. This increases the uncertainty in the operation of the controller, and limits its ability to maintain a feasible control over the grid resources. Given the mission-critical nature of power grid control, and the possible consequences that might arise in case of failure [1], it is essential for power grid controllers to be robust in the presence of such uncertainties.

Several controller designs presented in literature assume an ideal communication network [2], [3], [4] since, as they are designed for real-time, they can quickly counteract an occasional missing package. However, despite advances in improving the resiliency and reliability of communication networks in power grids [5], [6], [7], non-idealities cannot be eliminated due to the stochastic nature of wide-spread

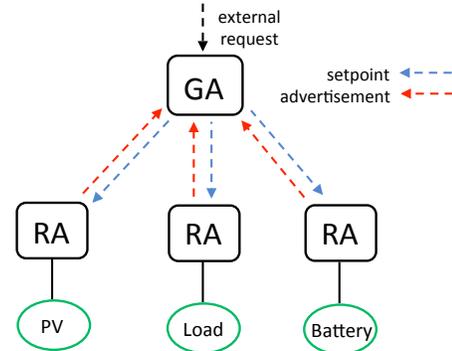


Fig. 1. COMMELEC Architecture

communication networks. This is especially true for real-time applications, in which low latencies are required.

As an example, in this paper we consider the COMMELEC framework [2], initially designed with the assumption of ideal communication, and we propose a method that makes it robust to message losses and delays. COMMELEC is a multi-agent framework for real-time control of distribution power grids using explicit power setpoints. However, the method presented in this paper is not limited to this framework and can be applied to any application with real-time decisions.

Such a method complements the traditional redundancy approach in achieving a higher degree of reliability [8], [9].

The COMMELEC architecture is shown in Figure 1. It consists of a main controller, henceforth the *Grid Agent* (GA), and several local controllers, each in charge of a specific resource, henceforth the *Resource Agents* (RAs). An RA captures the inner state of its resource and sends it as an *advertisement* to the GA. The GA uses the received advertisements to compute and issue explicit power *setpoints* that the RAs will instruct their resources to implement. The setpoints are computed such that their implementation: (1) maintains the grid in a feasible state (i.e. within voltage and current bounds), (2) tracks an upper-level request, such as dispatching a power profile or providing frequency support, and (3) tries to yield to the desired operation of the resources.

The advertisements sent by the RAs consist of, among other entries, the following two fields. (1) A *PQ profile* which represents the feasibility region of the resource, the region in the *PQ*-plane (for active and reactive power) in which the

resource can be instructed to operate. (2) A *Belief function* $\mathcal{B}(u)$, which captures the uncertainty of the resource. It maps every point $u = (P, Q)$ in the PQ profile to a set of points in the PQ -plane, which the resource might implement if instructed to implement u . The above fields are constructed by RAs such that they are only valid for a short-horizon λ , i.e. they need to be updated in the next setpoint implementation.

Under ideal communication network conditions (i.e. in the absence of messages losses and delays), the GA receives the latest advertisement from each RA, and performs the setpoint computation. For each resource, the GA determines a power setpoint from its PQ profile, such that the set of power setpoints for all resources results in a feasible grid state and maximizes the tracking of the requested external signal. This computation relies on the prediction in the advertisement, which is only valid for the horizon λ . An occasional lost or delayed advertisement from a resource might render the GA incapable of computing setpoints that satisfy the mentioned constraints. In principle, the GA waits until the advertisement is received, causing a loss of control for that period.

As an example, consider a grid-tied microgrid, as shown in Figure 1, that consists of a battery, a photovoltaic plant (PV), and a load. Let us suppose that the GA has the objective of providing primary frequency support to the main grid by controlling the battery power flow injection/absorption. It also needs to ensure that the bus-voltage and line-current magnitudes are within the safety limits, despite the stochastic profile of the PV injections and load consumption. A quick change in the frequency signal, PV production, or load consumption, coupled with a loss of advertisements, renders the GA unaware of the present and future state of the grid resources, and thus incapable of computing valid setpoints in the next cycle.

B. Contributions

In this paper, we propose a method that augments the RA advertisement, enabling the GA to become robust to communication network non-idealities. More precisely, we introduce two fields into the advertisements, constituting long-term information valid for a period greater than λ . We modify the GA in order to store the latest received advertisement, and make use of the long-term fields in the stored advertisement when the present advertisement is lost from some resource. The newly added fields are constructed in a way that ensures the safety of the grid, when they are used in a setpoint computation.

We validate our method and compare it to alternative methods, using a virtual commissioning tool that simulates the CIGRÉ benchmark low-voltage microgrid [10] consisting of a battery, a PV plant, and a load. This tool enables us to emulate non-ideal network conditions and study the behavior of the actual COMMELEC implementations under such conditions.

The structure of this paper is as follows. Section II focuses on the details of our proposed methodology, highlighting the properties of the newly added fields. In Section III, we describe the setup under study. Section IV provides the results of the

validation and comparative experiments. Finally, we conclude the paper and discuss future work in Section V.

II. METHOD

We augment the COMMELEC advertisements to include two new fields: (1) a long-term PQ profile (\mathcal{A}_l), and (2) a long-term Belief function (\mathcal{B}_l). The original PQ profile (\mathcal{A}) and Belief function (\mathcal{B}) are henceforth referred to as *short-term fields*. As mentioned earlier, the short-term fields estimate the behavior of the resource in the horizon that the control action is expected to be implemented, λ . This horizon should be short enough to allow the main controller to cope with the fastest dynamics in the system. On the other hand, long-term fields must be valid for a longer horizon Λ , taking into account all possible control actions and internal/external changes that might occur during this time. In practice, Λ should be chosen to ensure that a setpoint will be received during that horizon even in the presence of communication network non-idealities.

In general, the choice of a time horizon (λ or Λ) is a trade-off between several factors. A shorter horizon requires less time to compute, provides a more accurate prediction, and exports less uncertainty to the GA. However, as the horizon is shorter, the GA is not robust to losses. Therefore, sending both short-term and long-term fields allows us to take advantage of the accuracy of the short-term prediction, and the robustness of having a longer-term time horizon.¹

Note that since the PQ profile and the Belief function have a time horizon in which they are valid, they implicitly contain a time argument that represents their construction time, i.e. the time from which they are valid. This argument is made explicit in the rest of this section.

A. Properties of long-term fields of an advertisement

Formally, the following properties must hold for the long-term fields of an advertisement.

Property 1 (Long-term PQ profile):

$$\forall t' \in [t, t + \Lambda], \mathcal{A}_l(t) \subseteq \mathcal{A}(t')$$

In other words, the long-term PQ profile should be a subset of all short-term PQ profiles that lie within its horizon. This ensures that any setpoint in \mathcal{A}_l lies within the flexibility region of the resource for the entire long-term horizon. Therefore, if the GA chooses a setpoint from $\mathcal{A}_l(t)$, the implementation of this setpoint at the RA is guaranteed to be feasible if it is received at a time $t' \leq t + \Lambda$.

Property 2 (Long-term Belief function):

$$\forall t' \in [t, t + \Lambda], \forall u \in \mathcal{A}_l(t), \mathcal{B}(t')(u) \subseteq \mathcal{B}_l(t)(u)$$

As the belief function encapsulates the uncertainty of the resource when instructed to implement a setpoint $u = (P, Q)$, the long-term belief set of a setpoint should contain all the short-term belief sets of that setpoint in the long-term horizon. This ensures that any actual implementation lies within the

¹Note that, the proposed method is independent of the choice of λ and Λ .

long-term belief set of the issued setpoint. Therefore, if the GA computes a setpoint that is valid when considering the uncertainty advertised in $\mathcal{B}_l(t)$, then it is valid in the actual uncertainty $\mathcal{B}(t')$, for $t' \leq t + \Lambda$.

Note that the condition in Property 2 must hold for all u in $\mathcal{A}_l(t)$. Given that the domain of $\mathcal{B}_l(t)$ is $\mathcal{A}_l(t)$ and the domain of $\mathcal{B}(t')$ is $\mathcal{A}(t')$, then Property 1 guarantees that all the elements of $\mathcal{A}_l(t)$ are in the domain of $\mathcal{B}(t')$ as well.

B. Constructing long-term fields

Here, we define how the long-term fields can be constructed for the three types of resources of our case study, namely a battery, an uncontrollable PV, and an uncontrollable load. The method presented builds on the method defined in [11] for constructing the original COMMELEC advertisement (containing only the short-term fields).

1) *Batteries*: To compute the short-term fields, the battery agent makes use of the battery model proposed in [12]. Note that, assuming that the batteries are fully controllable, there is no uncertainty to deploy a setpoint. Thus, only the PQ profile needs to be continuously updated but not the Belief function. In particular, only the minimum and maximum active power, P_{min} and P_{max} (that depend on the state-of-charge of the battery, for details see [11]), need to be estimated. In general, the battery advertisement is defined by:

$$\begin{aligned} \mathcal{A} &= \{(P, Q) \in \mathbb{R}^2 \mid P_{min} \leq P \leq P_{max}, \sqrt{P^2 + Q^2} \leq S_r\}, \\ \mathcal{B}(P, Q) &= \{(P, Q)\}, \end{aligned} \quad (1)$$

where S_r represents the rated power of the battery converter.

Now, assuming that the state-of-charge (SoC) of the battery changes little in a λ -horizon, the battery agent is able to compute P_{min}^λ and P_{max}^λ , which represent the power limits that can be applied at horizon λ . Note that, this computation depends on the last implemented setpoint P . The pseudo-algorithm of this process is detailed in Algorithm 1.

Algorithm 1 Compute Power Bounds for the λ -horizon

Function: getShortTermBounds(P, λ, SoC)

- 1: Get the corresponding DC power p from the setpoint P implemented at t , using a converter model
 - 2: Use p and SoC to compute the SoC^λ : the state of the battery at $t + \lambda$
 - 3: Compute the DC short-term power bounds ([11]), $p_{min}^\lambda, p_{max}^\lambda$, that respect the DC voltage and current limits
 - 4: Use the converter model to get $P_{min}^\lambda, P_{max}^\lambda$
 - 5: **return** $P_{min}^\lambda, P_{max}^\lambda, \text{SoC}^\lambda$
-

In order to compute the PQ profile at the horizon Λ , we use Algorithm 1, $2(n - 1) + 1$ times, with:

$$n = \left\lceil \frac{\Lambda}{\lambda} \right\rceil \quad (2)$$

In other words, after computing the short-term power bounds, we use them as inputs for the computation of the bounds at the next λ -horizon (since they are valid). Note that, this is a

worst-case analysis since the actual implemented power will always respect the pre-computed bounds. We repeat the same until reaching the $n\lambda$ -horizon, using the $P_{min}^{i\lambda}$ as input for computing the $P_{min}^{(i+1)\lambda}$ bound (and similarly for P_{max}). In general, the computed power bounds are not monotonic with time, hence we take:

$$\begin{aligned} P_{min}^\Lambda &= \max_{i \in [1, n]} P_{min}^{i\lambda}, \\ P_{max}^\Lambda &= \min_{i \in [1, n]} P_{max}^{i\lambda} \end{aligned} \quad (3)$$

This will ensure that $\mathcal{A}_l \subseteq \mathcal{A}$, satisfying Property 1 from Section II-A. We summarise this process in Algorithm 2.

Algorithm 2 Compute Power Bounds for the Λ -horizon

Function: getLongTermBounds(P, λ, Λ)

- 1: Compute $n = \lceil \Lambda/\lambda \rceil$
 - 2: $[P_{min}^\lambda, P_{max}^\lambda, \text{SoC}^\lambda] = \text{getShortTermBounds}(P, \lambda, \text{SoC})$
 - 3: $\underline{\text{SoC}}^\lambda = \overline{\text{SoC}}^\lambda = \text{SoC}^\lambda$
 - 4: **for all** $i \in [1, n - 1]$ **do**
 - 5: $[P_{min}^{(i+1)\lambda}, -, \underline{\text{SoC}}^{(i+1)\lambda}] = \text{getShortTermBounds}(P_{min}^{i\lambda}, \lambda, \underline{\text{SoC}}^{i\lambda})$
 - 6: $[-, P_{max}^{(i+1)\lambda}, \overline{\text{SoC}}^{(i+1)\lambda}] = \text{getShortTermBounds}(P_{max}^{i\lambda}, \lambda, \overline{\text{SoC}}^{i\lambda})$
 - 7: **end for**
 - 8: $P_{min}^\Lambda = \max_{i \in [1, n]} P_{min}^{i\lambda}$
 - 9: $P_{max}^\Lambda = \min_{i \in [1, n]} P_{max}^{i\lambda}$
 - 10: **return** $P_{min}^\Lambda, P_{max}^\Lambda$
-

2) *Uncontrollable PV*: In the case of an uncontrollable resource, the GA does not count on the flexibility to request a setpoint different to what the resource is able to do. Thus, the short- and long-term PQ profiles will be same and equal to the forecasted power injection (P_f, Q_f) . In general, the uncontrollable-PV advertisement is defined by:

$$\begin{aligned} \mathcal{A} &= \{(P_f, Q_f)\}, \\ \mathcal{B}(P_f, Q_f) &= \{(P, Q) \in \mathbb{R}^2 \mid P_{min} \leq P \leq P_{max}, \\ & \quad Q_{min} \leq Q \leq Q_{max}\} \end{aligned} \quad (4)$$

For computing the long-term Belief function we use the same method as the one used for the short-term one, as proposed in [13]. This method predicts the interval where the injected power will lie in a given time-horizon. It consists of a training stage where it learns from past data, sampled at the desired horizon. Then, in an on-line stage, it uses the results of the training stage and the current measured value to estimate a prediction interval for the pre-defined horizon. As in practice longer term PV dynamics are larger than short-term ones, this method ensures that $\mathcal{B} \subseteq \mathcal{B}_l$ by the use of historical data in the training stage, satisfying Property 2 of Section II-A. Finally, we deploy two parallel predictions for both the short- and the long-term horizons.

3) *Uncontrollable Load*: As in the previous case, this agent only needs to update the Belief function. Likewise, the advertisement can be defined as in Eq. (4).

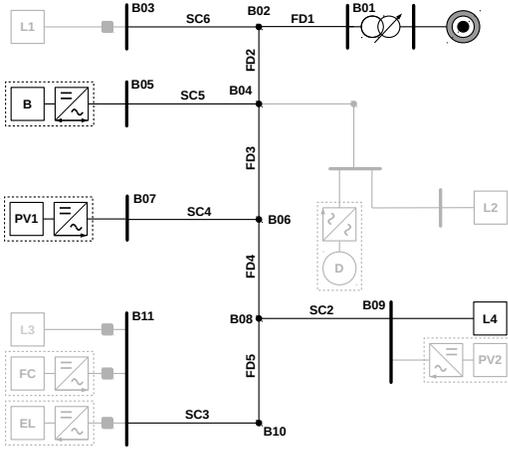


Fig. 2. CIGRÉ low-voltage benchmark microgrid. The resources not used for our experiments are greyed-out.

In order to have control on the validity of the advertisements, we use in this case a simple predictor for both short- and long-term fields. This is, we use the persistence method as a point predictor for both horizons and we compute the power bounds (in this case for both active and reactive power) using:

$$\begin{aligned}
 P_{min}(t + \lambda) &= (1 - \alpha_P^\lambda) \hat{P}(t), \\
 P_{max}(t + \lambda) &= (1 + \alpha_P^\lambda) \hat{P}(t), \\
 Q_{min}(t + \lambda) &= (1 - \alpha_Q^\lambda) \hat{Q}(t), \\
 Q_{max}(t + \lambda) &= (1 + \alpha_Q^\lambda) \hat{Q}(t),
 \end{aligned} \tag{5}$$

where $(\hat{P}(t), \hat{Q}(t))$ is the measured power at time t , and the parameters $\alpha_P^\lambda, \alpha_Q^\lambda \in (0, 1]$. The same definition can be used for Λ , and in order to guarantee Property 2 of Section II-A, $\alpha^\Lambda \geq \alpha^\lambda$ must hold for both P and Q .

III. EXPERIMENTAL SETUP

We test our method on the CIGRÉ benchmark low-voltage microgrid [10], shown in Figure 2. The microgrid is connected to the main grid, and consists of a 25 kW uncontrollable PV, a 30 kW / 90 kWh battery, and a 5 kW uncontrollable load. The GA is instructed to provide frequency support to the main grid, in addition to tracking a pre-determined power profile.

In our setup, we compare four different implementations of the COMMELEC GA. (1) The *Normal* GA, which is the original implementation that keeps requesting advertisements until it receives them from all RAs. (2) The *Robust* GA, which is our proposed methodology with short- and long-term fields in the advertisement. This GA replaces any missing short-term advertisement with valid long-term fields from that resource, if available. (3) The *Only-long* GA, which is a variation of the Robust GA, in that it only uses long-term fields throughout its operation (i.e. just by replacing λ by Λ in the *Normal* GA). This decreases the size of an advertisement and simplifies the design of the GA. (4) The *Previous-short* GA, which replaces any missing advertisement with the latest previously received advertisement from that RA. This eliminates the need

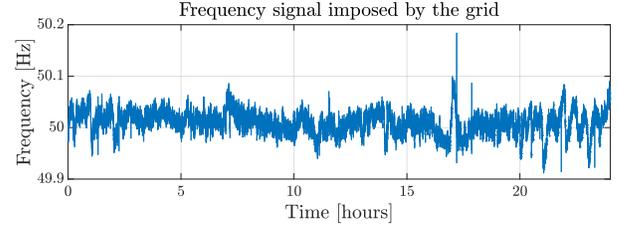


Fig. 3. Frequency signal imposed by the main grid used to provide frequency support.

Method / Loss rate	0%	5%	10%	15%	20%
Normal	121.65	198.95	317.48	541.52	1442.98
Robust	120.44	129.03	147.25	150.55	188.12
Only-long	121.65	130.36	153.84	154.99	178.85
Previous-short	121.65	122.18	139.40	154.77	451.17

TABLE I
ROOT MEAN SQUARE ERROR (IN WATTS) BETWEEN THE REAL POWER AT THE SLACK BUS AND THE REQUESTED TRACKING SIGNAL, FOR A 10-MINUTE INTERVAL

to construct, send, or handle long-term fields. In this paper, we consider $\lambda = 100$ ms and $\Lambda = 1$ s.

We use T-RECS [14], a virtual commissioning tool, to perform our tests on a simulated version of the grid. T-RECS enables us to use the actual GA and RA code, a simulated version of the resources, and the messages are exchanged over an emulated communication network. The topology consists of one router, with each software agent (GA and RAs) on a different subnet. Resources are on the same host machine as the RAs. With T-RECS, we are able to vary the link loss rate, and we analyze different values between 0% and 20%.

We use the root mean square error (RMSE) as a metric to measure the performance of the different GA implementations. The RMSE is calculated between the measured power at the slack and the frequency support signal (or dispatch plan signal). This shows how well each implementation can track the signal, and how robust each is to message losses.

IV. RESULTS

In this section, we illustrate the performance of the proposed method of handling communication network non-idealities under different conditions, and compare the results for the four different GAs described in Section III. Several scenarios are considered in order to highlight the conditions under which each method performs well.

A. Frequency support with non-binding grid constraints

We first study the performance of the methods when the grid state is far from the operational limits in terms of bus voltages and line currents. The GA is instructed to provide frequency support to the main grid, based on the frequency signal of Figure 3, which represents quick dynamics. As we are interested in studying the effects of the losses in the network, we vary the link loss rate between the GA and the RAs in the range [0%, 20%].

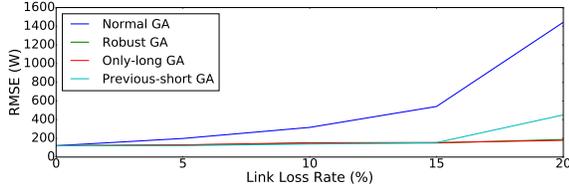


Fig. 4. Root mean square error (in Watts) between the real power at the slack bus and the requested tracking signal, for a 10-minute interval

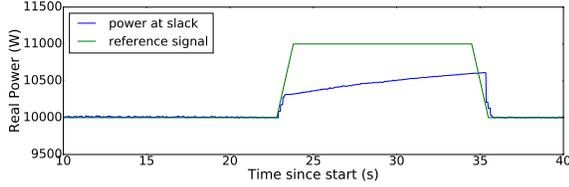


Fig. 5. Tracking experiment of *Only-long* GA with binding grid conditions and a 2% loss rate

Table I and Figure 4 show the resulting RMSE for the different methods across the different link loss rates, for an interval of 10 minutes. The RMSE is calculated between the actual power at the slack, and the result of $S = -\sigma(f - f_0)$, where S is the expected power at the PCC when providing frequency support, computed by multiplying the droop parameter σ with the divergence of the grid frequency f from the reference frequency $f_0 = 50$ Hz.²

We observe that the performance of the *Normal* GA rapidly deteriorates as the link loss rate increases. This follows directly from the fact that it is extremely sensitive to the amount of available information, and fails to follow the request in our quick dynamic scenario. The *Previous-short* GA maintains a good level of tracking until the loss rate is too high. This is expected as the information it uses in case of a loss (the previous advertisements) is invalid, and as the loss rate increases, tracking the quick frequency changes becomes increasingly unlikely.

The *Robust* and *Only-long* GA manage to provide frequency support even under 20% link loss rate, although the *Only-long* GA obtains worse performance throughout, especially for lower loss rates. This stems from the fact that its computations are always conservative, as they all use advertisements with a long-term horizon Λ . The effects of this are not drastic in such a scenario, but will appear when the grid conditions are binding, as presented in the next section.

B. Tracking a power profile with binding grid constraints

In order to study the behavior of the *Robust* and *Only-long* GA under binding grid conditions, we consider a scenario in which the GA is instructed to follow a pre-computed dispatch plan. The slower dynamics in this experiment allow us to better visualize the tracking performance. Moreover, we artificially limit the ampacity of the line connecting the microgrid to the main grid (FD1) to 16 A, i.e. a power limit of c.a. 11 kVA.

²We take $\sigma = 100$ kW/Hz in our experiments.

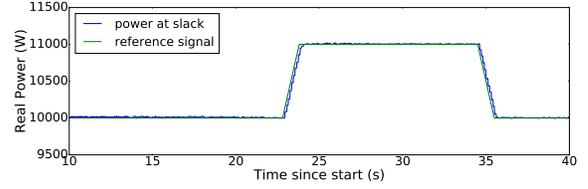


Fig. 6. Tracking experiment of *Robust* GA with binding grid conditions and a 2% loss rate

Figures 5 and 6 show the tracking results of *Only-long* GA and *Robust* GA, respectively, when the link loss rate is 2%. We observe that, although both manage to track the 10 kW request fully (as it results in a current far away from the ampacity limit of line FD1), only the *Robust* GA manages to track the 11 kW signal. The *Only-long* GA uses advertisements with larger uncertainty, and is thus conservative in order to avoid current violations. The *Robust* GA maintains tracking as it can safely do so without risking violation, due to the accuracy of the short-term advertisements it uses.

The conservative nature of the *Only-long* GA is highlighted in such binding grid constraints. Similar results are observed for frequency support experiments.

The *Previous-short* GA is not conservative, and thus maintains tracking (under low loss rates) even in binding grid conditions. However, as it uses invalid information, it might cause voltage and/or current violations.

C. Validation

Finally, we validate the *Robust* method via a 24-hour frequency support experiment with a 2% link loss rate. The 24-hour simulation, with the profiles of the PV and the load taken from an actual experimental run, enables us to see the performance under different and realistic grid conditions. In this particular case, the battery power is used as the slack variable compensate for the PV power-variations and adapting to the frequency signal to provide frequency support. The initial state of charge of the battery (20%) is pre-defined by the forecasted PV and load powers the day before.

Figure 7 shows the results of the tracking, in addition to the power at the buses of the battery, the PV, and the load. Although the comparison between the reference signal and the measured power is not visible in the graph, the computed RMSE is 145.67 W for the entire day. We also measure the RMSE over a rolling window of 20 minutes, and the resulting average and the maximum RMSE are 142.40 W and 263.53 W, respectively. This shows the robustness of our method throughout the daily cycle. Furthermore, the state-of-charge of the battery during the experiment is shown in Figure 8, showing the capabilities of the battery to provide such an ancillary service to the main grid.

V. CONCLUSION

We present a method for real-time power grid control that is robust in non-ideal network conditions. We take an example of a state-of-the-art framework for power grid control, and

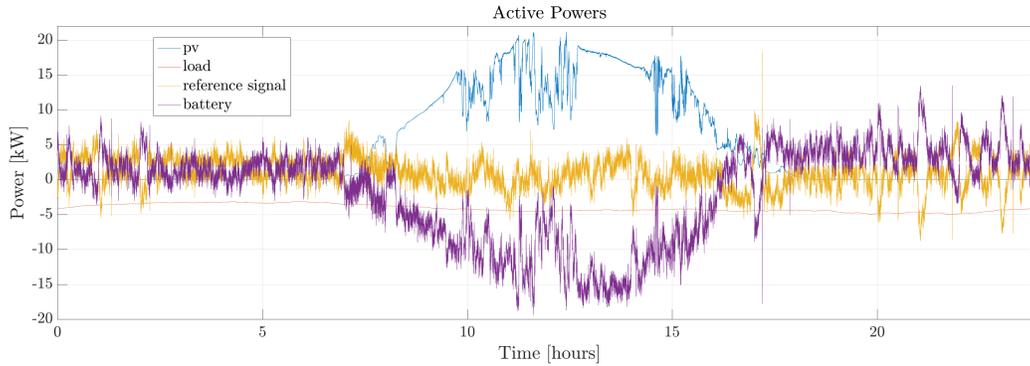


Fig. 7. 24-hour frequency support *Robust* experiment with a 2% link loss rate. The power at the slack is not visible since it is below the reference (tracking).

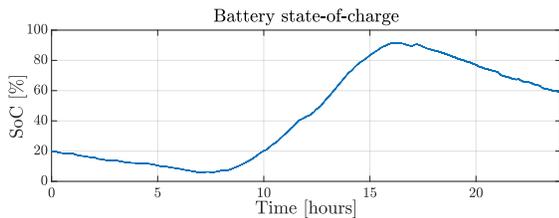


Fig. 8. Battery state-of-charge (SoC) during the 24-hour experiment

augment it such that it maintains its performance in spite of communication failures that lead to losses or delays of messages from the resources to the controller. This is done by having the resources send information about their state that is valid for a longer time horizon, in addition to the short-term information originally sent. The long-term information can be used to compute valid setpoints in cases when the short-term messages are lost due to a non-ideal network.

We show that our method guarantees grid safety, by construction. We also show a validation of our method over a 24-hour period, and provide a comparative analysis with alternative methods. We observe that the *Robust* GA is able to track the requested frequency signal better than alternative methods, under both binding and non-binding grid conditions.

For scenarios in which the dynamics are not as quick, such as power profile tracking, and under non-binding grid conditions, we find that the *Only-long* GA alternative is comparable with the proposed *Robust* method, but still performs worse. Our proposed methodology comes at no additional expense, and provides significant advantages in deployment conditions.

Here, we address losses of advertisements, and our proposed methodology focuses on dealing with communication failures that affect these messages, without explicitly dealing with the dual issue of lost setpoints. However, our experimental setup was designed to test for random communication failures, with the RAs maintaining the previous setpoint until a new one is issued. This problem will be studied further in future research.

REFERENCES

[1] G. Andersson, P. Donalek, R. Farmer, N. Hatzigiorgiou *et al.*, “Causes of the 2003 Major Grid Blackouts in North America and Europe, and

Recommended Means to Improve System Dynamic Performance,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1922–1928, Nov 2005.

[2] A. Bernstein, L. Reyes-Chamorro, J.-Y. Le Boudec, and M. Paolone, “A Composable Method for Real-Time Control of Active Distribution Networks with Explicit Power Setpoints. Part I: Framework,” *Electric Power Systems Research*, vol. 125, pp. 254–264, 2015.

[3] A. Lucas and S. Chondrogiannis, “Smart Grid Energy Storage Controller for Frequency Regulation and Peak Shaving, Using a Vanadium Redox Flow Battery,” *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 26–36, 2016.

[4] S. J. Crocker and J. L. Mathieu, “Adaptive State Estimation and Control of Thermostatic Loads for Real-Time Energy Balancing,” in *American Control Conference (ACC)*, 2016. IEEE, 2016, pp. 3557–3563.

[5] M. Popovic, M. Mohiuddin, D.-C. Tomozei, and J.-Y. Le Boudec, “iPRP – The Parallel Redundancy Protocol for IP Networks: Protocol Design and Operation,” *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1842–1854, 2016.

[6] E. Piatkowska, L. P. Bayarri, L. A. Garcia, K. Mavrogenou, K. Tsatsakis, M. Sanduleac, and P. Smith, “Enabling Novel Smart Grid Energy Services with the Nobel Grid Architecture,” in *2017 IEEE Manchester PowerTech*, June 2017, pp. 1–6.

[7] Y. Wu, J. Wei, and B. M. Hodge, “A Distributed Middleware Architecture for Attack-Resilient Communications in Smart Grids,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[8] M. Mohiuddin, W. Saab, S. Bliudze, and J.-Y. Le Boudec, “Axo: Masking Delay Faults in Real-Time Control Systems,” in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 2016, pp. 4933–4940.

[9] W. Saab, M. Mohiuddin, S. Bliudze, and J.-Y. Le Boudec, “Quarts: Quick Agreement for Real-Time Control Systems,” in *22nd IEEE International Conference on Emerging Technologies And Factory Automation*. IEEE, 2017.

[10] Taskforce C6.04.02, “Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources,” CIGRÉ, Tech. Rep., 2010.

[11] L. Reyes-Chamorro, A. Bernstein, J.-Y. Le Boudec, and M. Paolone, “A Composable Method for Real-Time Control of Active Distribution Networks with Explicit Power Setpoints. Part II: Implementation and Validation,” *Electric Power Systems Research*, vol. 125, pp. 265–280, 2015.

[12] M. Bahramipanah, D. Torregrossa, R. Cherkaoui, and M. Paolone, “Enhanced Equivalent Electrical Circuit Model of Lithium-Based Batteries Accounting for Charge Redistribution, State-of-Health, and Temperature Effects,” *IEEE Trans. Transport. Electric*, vol. 3, no. 3, pp. 589–599, Sept 2017.

[13] E. Scolari, F. Sossan, and M. Paolone, “Irradiance Prediction Intervals for PV Stochastic Generation in Microgrid Applications,” *Solar Energy*, vol. 139, no. Supplement C, pp. 116 – 129, 2016.

[14] J. Achara, M. Mohiuddin, W. Saab, R. Rudnik, and J.-Y. Le Boudec, “T-RECS: A Software Testbed for Multi-Agent Real-Time Control of Electric Grids,” in *22nd IEEE International Conference on Emerging Technologies And Factory Automation*. IEEE, 2017.