Resource allocation with local QoS: Flexible loads in the power grid

Austin R. Coffman*,†, Matthew Hale*, and Prabir Barooah*

Abstract—Loads that can vary their power consumption without violating their Quality of service (QoS), that is flexible loads, are an invaluable resource for grid operators. Utilizing flexible loads as a resource requires the grid operator to incorporate them into a resource allocation problem. Since flexible loads are often consumers, for concerns of privacy it is desirable for this problem to have a distributed implementation. Technically, this distributed implementation manifests itself as a time varying convex optimization problem constrained by the QoS of each load. In the literature, a time invariant form of this problem without all of the necessary QoS metrics for the flexible loads is often considered. Moving to a more realistic setup introduces additional technical challenges, due to the problems' time-varying nature. In this work, we develop an algorithm to account for the challenges introduced when considering a time varying setup with appropriate QoS metrics.

I. INTRODUCTION

Relying more and more on renewable generation is the envisioned future for the power grid. However, this goal is not without its challenges; renewable sources, such as solar and wind, are highly volatile. Moreover, supply and demand of power must always be in equilibrium, and when renewable generation cannot ensure this, controllable generation sources must ramp to ensure equilibrium. Economically, for a Balancing Authority (BA) (the institution responsible for ensuring supply and demand are balanced in a given geographical area), ramping generators or utilizing batteries for this is not feasible. This has motivated the recent investigation of a new resource to help where conventional generators and batteries fall short: flexible loads.

Flexible loads can deviate from a baseline level of consumption without violating the Quality of Service (QoS) of the load. From the perspective of the BA, flexible loads deviating from baseline are identical to a battery discharging and charging. Due to this, flexible loads are often said to provide "Virtual Energy Storage" (VES) [1]. More importantly, grid support from flexible loads is more cost effective than batteries [2]. Some examples of flexible loads include residential air conditioners [3], water heaters [4], refrigerators [5], commercial HVAC systems [6], and pumps for irrigation [7] and pool cleaning [8].

To utilize flexible loads, the BA in some way must incorporate them into a resource allocation problem. In a centralized framework, the resource allocation problem involves a central authority accounting for all of its resources and their constraints, and then allocating its needs to each resource based on the constraints. The problem is typically solved for a specific future duration. For instance, the BA allocates its resources for the next day [9].

In contrast to the BA solving a centralized resource allocation problem, it is possible to decentralize and have each flexible load solve a portion of the centralized problem. Furthermore, this distributed algorithm can run in real time. The advantage of a distributed solution is that (i) privacy is protected, as each load only needs to know its own QoS and (ii) the solution is more robust to modeling error as no one entity is making decisions for the ensemble based on models of the ensemble; each member of the ensemble makes decisions for itself based on a combination of its local and global information.

Solving the resource allocation problem in a distributed fashion and real time falls under the framework of *time-varying optimization*. There are two main challenges in this framework: (C1) shifting to a real time solution is problematic for constraints with "memory", e.g. dynamic systems or rate constraints that require past state values to evaluate, and (C2) at each instant in time typically only one iteration of the optimization algorithm can be applied. While the effects of point (C2) are indirectly/directly analyzed in virtually all works on real time optimization, point (C1) is often not considered. That is, most works on time varying optimization focus only on static constraint maps [10] or unconstrained problems [11]. Unfortunately, the QoS of flexible loads is specified by constraints with memory.

In addition to the literature on *time-varying optimization*, there is a subfield of literature focused on the distributed resource allocation for flexible loads [12]–[19] in the smart grid. While there is a library of work [20]–[22] on how to model the QoS of flexible loads for the purpose of resource allocation, only a few works on distributed resource allocation take this into account [12], [16].

To summarize, much of the past work on time-varying optimization is focused on problems of different structure than the resource allocation problem for flexible loads. Thus the algorithms developed are not directly applicable. Whereas, many of the past works focused on distributed resource allocation for flexible loads do not account for the entirety of the loads' QoS.

In this paper, we develop an algorithm for distributed resource allocation that allows loads to account for a wide variety of QoS metrics. In doing so, our algorithm incor-

porates principled techniques to overcome the challenges (C1) and (C2) listed above. To overcome (C1), we employ a *state augmentation* technique that augments past fictitious state values that act as surrogates for the previous states. To overcome (C2), we utilize predictions of the time varying quantities to facilitate a benefit similar to warm start technique in centralized optimization. With all features of the algorithm accounted for, we prove an Input to State stability (ISS) result for when the time varying aspect is arbitrary (but in some sense bounded). This stability result is guaranteed under gain conditions that are specified in terms of the readily available problem data.

In numerical experiments we validate our theoretical results and compare our proposed method to a past method in the literature. In the time-varying setting our proposed method is able to successfully have flexible loads solve the resource allocation problem in a distributed/hierarchical fashion. Additionally, it is shown that the past method, based on dual ascent, can lead to integrator windup in the same time-varying setting.

The paper proceeds as follows: in Section II the problem setup and requirements are described and in Section III optimization basics are introduced. In Section IV our proposed method is introduced and it is analyzed in Section V. We give numerical examples in Section VI and conclude in Section VII.

II. Needs of the Loads and the Power Grid

A. Notation

We let the index $i \in \{1, \ldots, N\}$ denote the ith load, where N is the total number of loads. The index $t \in \mathbb{N}$ is the discrete time index. We reserve lowercase letters for vectors/scalars and uppercase letters for matrices. The notation $x[j]$, when x is a vector, will refer to the jth element of the vector x.

The power consumed by load i at time t is denoted $d_{t|t}^i$. Furthermore, the quantity $d_{t+j|t}^i$ is the power consumption that at time t load i predicts it will consume at time $t + j$, where $j \leq N_p$ and N_p is the prediction horizon. For convenience, we also define $N_p^- := N_p - 1$. The required total power from all loads, i.e., the reference signal, at time t is denoted s_t.

We consider two "stacked" vectorized versions of the scalar quantities $d_{t+j|t}^i$. The first is the *load perspective stacking* where we stack the scalars $d_{t+j|t}^i$ into a vector and denote it as $x_t^i \triangleq [d_{t|t}^i, \ldots, d_{t+N_p^-|t}^i]^T$. The second is the *grid perspective stacking* where we stack over all loads, forming $x_{j|t} \triangleq [d_{t+j|t}^1, \ldots, d_{t+j|t}^N]^T$. In any case, for a fixed N_p we refer to to the following $x_t \triangleq [(x_t^1)^T, \ldots, (x_t^N)^T]^T$, which contains all the elements of x_t^i and $x_{j|t}$. The purpose for introducing both stacked forms is for ease of exposition.

B. BA's Needs: Reference tracking (global goal)

The BA employs support from flexible loads to help mitigate supply and demand mismatch. Using the previously
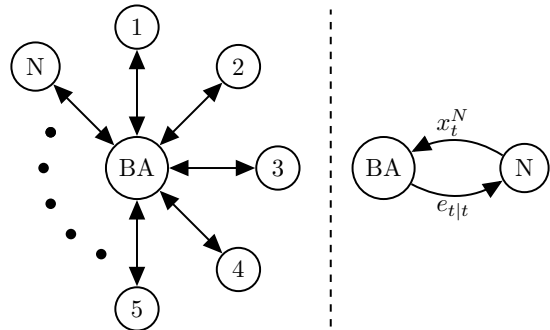

Fig. 1: The information structure considered, which is representative of the structure of a utility (BA) in the USA. The numbers represent the flexible load index.

defined variables, this goal is captured by requiring the following to be small:

$$e_{\tau|t} = \sum_{i=1}^{N} d_{\tau|t}^i - s_\tau, \quad J_G(x_{\tau|t}) = e_{\tau|t}^2, \ \tau \leq t + N_p^-. \quad (1)$$

C. Individual Needs: The QoS set (local constraints)

We describe the requirements of the loads through a QoS set. These constraints are taken from the vast literature on "capacity characterization" of flexible loads [20]–[22]. The constraints on the power for the ith load, x_t^i, are:

$$\mathcal{D}^i(d_{t-1|t-1}^i) \triangleq \quad (2)$$
$$\left\{ x_t^i : \forall \ j \in \{t, \ldots, t + N_p^-\}, \right.$$
$$\textbf{Power}: \quad d_L^i \leq d_{j|t}^i \leq d_H^i, \quad (3)$$
$$\textbf{Rate}: \quad r_L^i \leq d_{j|t}^i - d_{j-1|t}^i \leq r_H^i, \ j > t \quad (4)$$
$$\textbf{Rate-IC}: \quad r_L^i \leq d_{t|t}^i - d_{t-1|t-1}^i \leq r_H^i, \quad (5)$$
$$\left. \textbf{Energy}: \quad e_L^i \leq \sum_{j=t}^{t+N_p^-} d_{j|t}^i \leq e_H^i \right\}. \quad (6)$$

Each constraint (3)-(6) has a specific meaning as illustrated by the labels given. An additional Rate-IC constraint is included to emphasize that previous data is required to evaluate this constraint. Furthermore, it is necessary to define the QoS set (2) over a time horizon, otherwise enforcing the constraint (6) would not be possible. The constraints (3)-(6) model various classes of flexible loads, e.g. batteries, HVAC systems in commercial buildings, thermostatically controlled loads (TCLs) [21], and pool pumps [9].

However, while the QoS set specifies maximum limits it does not mean that it is desirable to operate at these limits. Thus, the loads are also interested in making the following quantity small,

$$J_L(x_{\tau|t}) = \sum_{i=1}^{N} (d_{\tau|t}^i)^2 \zeta^i, \quad \tau \leq t + N_p^-, \quad (7)$$

where $\zeta^i > 0$ for all $i \in \{1, \ldots, N\}$. The quantity (7) can be thought of as a regularization term.

Proposition 1. *For each i the set $\mathcal{D}^i(d_{t-1|t-1}^i)$ is compact, convex, and non empty.*

There are two important points about the set $\mathcal{D}^i(d_{t-1|t-1}^i)$: (i) the constraints (4)-(6) require more than one instant of time to appropriately evaluate, and (ii) the constraint (5) has memory, at time t the set $\mathcal{D}^i(d_{t-1|t-1}^i)$ is a function of $d_{t-1|t-1}^i$.

Comment 1. *The constraint set (2) in its abstract form captures the heterogeneity of the load. For example, load $i = 10$ could be a Walmart and load $i = 5$ could be a classroom on a university campus, both shifting their load to help the grid. Put explicitly, our proposed optimization problem and solution method tolerate arbitrary high degrees of heterogeneity.*

D. Information structure

The information structure considered is depicted in Figure 1, which is a hierarchical communication structure with distributed computation. For each time t, the loads are allowed to communicate exactly *once* to the BA in order to receive global information, the signal $e_{t|t}$ (1). The loads can then use this global information to apply *one* iteration of an optimization algorithm to achieve the global goal, tracking the reference s_t. However, at the next time, $t+1$, the reference s_t will change and hence the optimization problem the loads are attempting to solve is operating in "real time."

III. OPTIMIZATION BASICS

To better understand our contribution we review how to solve a constrained optimization problem, of special structure, in a distributed fashion using projected gradient descent (PGD). Consider the following optimization problem,

$$\min_{z \in \mathcal{Z}} f(z; t), \quad z \in \mathbb{R}^q, \quad \mathcal{Z} = \mathcal{Z}^1 \times \cdots \times \mathcal{Z}^q, \quad (8)$$

with $z^i \in \mathcal{Z}^i$ only. The PGD method to solve (8) is,

$$z_{t+1}^i = \Pi_{\mathcal{Z}^i}\left(z_t^i - \alpha \nabla f^i(z_t)\right), \quad \forall\, i \in \{1, \ldots, q\}, \quad (9)$$

$$\nabla f^i(z_t) \triangleq \left.\frac{\partial f(z; t)}{\partial z^i}\right|_{z=z_t}, \quad \Pi_{\mathcal{X}}(x) \triangleq \arg\min_{y \in \mathcal{X}} \|y - x\|,$$

with $\alpha > 0$ a step size. The projected gradient method applied to time invariant problems has its origins in [23]. For an introduction to time varying convex optimization the paper [24] is a good reference. As we will see, the resource allocation problem naturally has a similar structure to (8).

The formulation of a resource allocation problem with all the appropriate QoS constraints in the QoS set (2), and an algorithm for its solution are the focus of the rest of the paper.

IV. PROPOSED METHOD

Largely, the limitation of the past works on resource allocation problem formulations/algorithms is that they do not consider appropriate load QoS metrics. Consequently, our proposed method is centered around including the approrpriate QoS constraints. When doing this, a technical trouble arises that we solve with a *state augmentation* technique, which we describe next.

A. Predictive Resource Allocation with memory

We define the *memory* objective at time $t \in \mathbb{N}$ as follows: $J_{\mathrm{m}}(x_{t-1|t}) \triangleq$

$$\sum_{i=1}^{N}(d_{t-1|t}^i - d_{t-1|t-1}^i)^2 \bar{\zeta}^i + \left(\sum_{i=1}^{N} d_{t-1|t}^i - s_{t-1}\right)^2. \quad (10)$$

We have introduced the variable $d_{t-1|t}^i$, which is a fictitious variable at time t that we desire to be close to $d_{t-1|t-1}^i$ (treated as a constant at time t). The augmented decision variable, z_t is then:

$$z_t^i \triangleq [d_{t-1|t}^i, (x_t^i)^T]^T, \text{ and } z_t \triangleq [(z_t^1)^T, \ldots, (z_t^N)^T]^T, \quad (11)$$

where, by construction, z_t contains all the elements in $x_{j|t}$, so where convenient we refer to $x_{j|t}$. However, within the scope of an optimization problem, the decision variable is z_t. With z_t^i it is now possible to redefine the QoS set (2) as independent of the previous state value. We denote this new set as:

$$\mathcal{D}^i \triangleq \left\{z_t^i : \text{ s.t. } (3), (4), \text{ and } (6)\right\}. \quad (12)$$

Comment 2. *In (12) the constraint (5) is evaluated with the decision variable $d_{t-1|t}^i$ and not an externally specified variable/parameter. The same methodology could be extended to handle any convex constraint with memory. Thus, the expanded state (11) has mitigated the challenge (C1) described in the introduction: the constraint set is neither time varying nor state dependent.*

With this, the predictive resource allocation problem with memory is the following:

$$\min_{z_t} \eta(z_t) = \frac{1}{2}\left(\sum_{\tau=t}^{t+N_p^-} J_L(x_{\tau|t}) + J_G(x_{\tau|t}) + J_{\mathrm{m}}(x_{t-1|t})\right)$$

$$\text{s.t.} \quad z_t^i \in \mathcal{D}^i, \quad \forall\, i \in \{1, \ldots, N\}. \quad (13)$$

We see that (13) is in a form applicable to the example algorithm (9). The solution to (13) is denoted z_t^* with optimal value $\eta_t^* = \eta(z_t^*)$.

B. Proposed algorithm

To solve the problem (13), we propose the following algorithm. The ith load updates its state with:

$$z_{t+1}^i = \Pi_{\mathcal{D}^i}\left(\hat{P}\left(z_t^i - \alpha \nabla \eta^i(z_t)\right)\right) = \Pi_{\mathcal{D}^i}\left(\hat{P}\psi_t^i\right), \quad (14)$$

$$\psi_t^i \triangleq z_t^i - \alpha \nabla \eta^i(z_t),$$

where $\alpha > 0$ is a step size common to all loads, and $\hat{P}$ is the following circulant matrix,

$$\hat{P} = \begin{bmatrix} \mathbf{0}_{N_p \times 1} & I_{N_p} \\ 1 & \mathbf{0}_{1 \times N_p} \end{bmatrix}. \tag{15}$$

The primary purpose of the matrix $\hat{P}$ is to help mitigate challenge (C2) described in the introduction. It does this by "shifting" the data to facilitate a benefit similar to warm start techniques in optimization. In fact, a flavor of this idea was included in [25], among others, to speed up the solution time for real time Model Predictive Control.

Recall, for each load i, the quantity z_t^i is a vector in $\mathbb{R}^{N_p+1}$ where N_p is the prediction horizon. The algorithm (14) is an update rule for the entire vector z_t^i, the value that the load i *actually* consumes at time t is then $z_t^i[2] = d_{t|t}^i$. The vectorized form of the algorithm (14) is: z_{t+1}

$$= \left[\Pi_{\mathcal{D}^1}(\hat{P}\psi_t^1), \dots, \Pi_{\mathcal{D}^N}(\hat{P}\psi_t^N) \right]^T = \Pi_{\mathcal{D}}\left(P\psi_t \right), \tag{16}$$

$$\mathcal{D} = \mathcal{D}^1 \times \cdots \times \mathcal{D}^N, \quad \psi_t = [(\psi_t^1)^T, \dots, (\psi_t^N)^T]^T, \tag{17}$$

$$P = I_N \otimes \hat{P}, \tag{18}$$

where $\times$ denotes Cartesian product, $\otimes$ denotes matrix Kronecker product [26], and z_t is a vector in $\mathbb{R}^{(N_p+1)N}$. Since the Cartesian product operation preserves convexity and by Proposition 1 each $\mathcal{D}^i$ is convex, the set $\mathcal{D}$ is also convex. The vectorized form (16) is useful for analysis, however during implementation each load has the ability to update its own local variable z_t^i by solely using (14).

Proposition 2. *Let z_t^* be the optimal solution to problem* (13) *and $\psi_t^* = z_t^* - \alpha\nabla\eta(z_t^*)$, both at time $t \in \mathbb{N}$, then*

$$z_t^* = \Pi_{\mathcal{D}}\left(z_t^* - \alpha\nabla\eta(z_t^*) \right) = \Pi_{\mathcal{D}}(\psi_t^*).$$

V. STABILITY

A. Preliminaries

We list a few results that will be useful for the analysis of the proposed algorithm (16).

Proposition 3. *The Hessian $\nabla^2\eta$ and gradient $\nabla\eta(z_t)$ can be expressed in the following form, letting $H^i \triangleq diag([\bar{\zeta}^i, \zeta^i, \dots, \zeta^i]) \in \mathbb{R}^{N_p+1}$, for all $z_t \in \mathbb{R}^{(N_p+1)N}$*

(i) $\nabla^2\eta = \mathbf{1}_N \otimes \left(\mathbf{1}_N^T \otimes I_{N_p+1} \right) + \bigoplus\limits_{i=1}^{N} H^i$,

(ii) $\nabla\eta(z_t) = (\nabla^2\eta)z_t - u_t$,

where $\bigoplus$ denotes the Kronecker sum of matrices [26], $diag(a)$ denotes the diagonal matrix of the vector a, $\mathbf{1}_N \in \mathbb{R}^N$ is the column vector of all ones, and the vector $u_t \in \mathbb{R}^{(N_p+1)N}$ is,

$$u_t = [(u_t^1)^T, \dots, (u_t^N)^T]^T \text{ with,} \tag{19}$$

$$u_t^i = [d_{t-1|t-1}^i + s_{t-1}, s_t, \dots, s_{t+N_p^-}]^T. \tag{20}$$

We have dropped the dependence of z_t on the Hessian, as the Hessian is a constant matrix, where additionally, based

on the form given in Proposition 3, it is symmetric, i.e., $\nabla^2\eta = (\nabla^2\eta)^T$ and positive definite.

Proposition 4. *Let $\bar{\zeta}^i = \zeta^i$ for all $i \in \{1, \dots, N\}$, then*

$$\|P\nabla^2\eta - \nabla^2\eta P\| = 0.$$

Lemma 1 (Theorem 2.1, [27])**.** *For any $s, \tau \in \mathbb{N}$, the following bound holds,*

$$\frac{1}{N}\|z_s^* - z_\tau^*\| \le \frac{\bar{u}_{s,\tau}^*}{\lambda_{min}(\nabla^2\eta)}, \tag{21}$$

where $\bar{u}_{s,\tau}^* = \|u_s^* - u_\tau^*\|$.

Lemma 2. *For all $t \in \mathbb{N}$ the following holds,*

$$\frac{1}{N}\|Pz_{t-1}^* - z_t^*\| \le \frac{\bar{g}_t^*}{\lambda_{min}(\nabla^2\eta)},$$

where $\bar{g}_t^* = \bar{u}_{t,t-1}^* + 2\tilde{u}_t^*$, $\tilde{u}_t^* = \|u_{t-1}^* - u_t^{*,0}\|$ and $u_t^{*,0}$ is the value that produces an optimal solution of all zeros.

Proof. See appendix. $\qquad\square$

This result will render itself useful for the stability analysis. Also necessary in our stability results is the class of $\mathcal{K}$ and $\mathcal{KL}$ functions, that hold their usual definitions as seen, e.g. in [28].

B. Stability: Main result

Our main theoretical results for our proposed algorithm (14) is summarized in Theorem 1. If we treat the value $\|z_t - z_t^*\|$ as the "state" and an upper bound on the time varying aspects to the optimization problem as the "input", then Theorem 1 is a global input to state stability (ISS) result.

Practically, we want the magnitude $\|z_t - z_t^*\|$ to be small, as the optimal solution z_t^* represents the value that optimally satisfies all of the specified criteria. The theorem below requires the following boundedness assumptions:

A1: for all $t \in \mathbb{N}$, $\bar{g}_t^* < \bar{g} < \infty$,
A2: for all $t \in \mathbb{N}$, $\ell < t$, $\|Pu_{t-\ell} - u_t^*\| < \Delta < \infty$.
Then we denote $\bar{u} \triangleq \left(\frac{N\bar{g}}{\alpha\lambda_{min}(\nabla^2\eta)} + \Delta \right)$.

Theorem 1 (Global-ISS)**.** *If assumptions **A1** and **A2** are satisfied, the step size α satisfies,*

$$\alpha \in \left(0, \frac{1}{\zeta^{max} + N} \right), \quad \text{where} \quad \zeta^{max} = \max_{1 \le i \le N} \zeta^i,$$

and $\bar{\zeta}^i = \zeta^i$ for all $i \in \{1, \dots, N\}$, then for all $z_0 \in \mathbb{R}^{(N_p+1)N}$ there exists a $\Gamma \in \mathcal{K}$ and an $\Omega \in \mathcal{KL}$ such that

$$\|z_t - z_t^*\| \le \Omega(\|z_0 - z_0^*\|, t) + \Gamma(\bar{u})$$

where z_0 is the initial iterate of (16).

Proof. See appendix. $\qquad\square$

In Theorem 1 we have developed conditions on the step-size in terms of the readily available problem data that will give a stability result for time varying reference signals.

TABLE I: Simulation Parameters

Par.	Unit	value		Par.	Unit	value
N	hundred	1		α	N/A	$\frac{0.99}{\zeta^{\max}+N}$
$\zeta^{\min}, \zeta^{\max}$	N/A	0.1, 4		$e_L^{\min}, e_H^{\max}$	kWh	0, 4
$d_L^{\min}, d_H^{\max}$	kW	0, 10		$r_L^{\min}, r_H^{\max}$	kW	-0.50, 0.50

VI. Numerical Examples

Here we offer numerical examples to validate the result from Theorem 1. This involves simulating the algorithm (14) on various types of data. We provide two scenarios for this: Scenario 1 (S1) where only the loads' power constraint (3) is considered and the loads are asked to track a step reference that is not feasible to track (the maximum value of the reference is larger than the sum of all the upper bounds in (3)) and Scenario 2 (S2) our proposed method tracking Bonneville Power Administrations (BPA) balancing reserves deployed (BRD) signal to illustrate the effectiveness of our algorithm tracking a time varying signal. Additionally, in (S1) we compare our method to a method that uses dual ascent from [29].

In both scenarios: (i) each load is given a set of parameter values obtained by a linear spacing between the maximum and minimum values found (along with the other relevant simulation parameters) in Table I and (ii) the sampling time is $T_s = 5$ minutes.

A. Scenario 1: Integrator Windup of dual ascent

Here we illustrate the "integrator windup" behavior of dual ascent based algorithms [29]:

$$\lambda_t = \lambda_{t-1} + \gamma e_{t-1|t-1}, \text{ and } d_{t|t}^i = \Pi_{[d_L^i, d_H^i]}\left(\frac{\lambda_t}{\zeta^i}\right), \quad (22)$$

when feasibility is lost. The result of this is shown in Figure 2. When the resource allocation problem is not feasible, the update for λ_t in (22) will continue to integrate non-zero area. It then takes dual ascent time to reach zero steady state error once feasibility is regained. It is worth noting that the two regions of integrated area in Figure 2 are equivalent.

For comparison we also utilize our proposed algorithm with solely the magnitude constraints (3) and $N_p = 0$, which does not suffer from integrator windup.
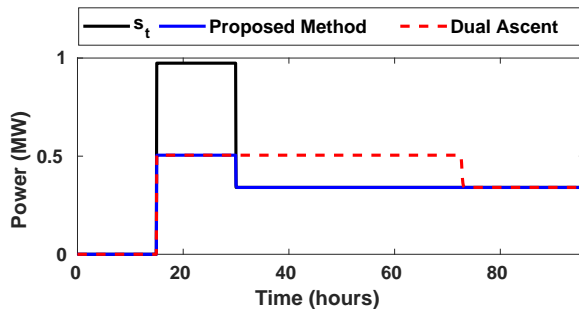

Fig. 2: Integrator Windup of dual ascent with step response reference.

B. Scenario 2: Tracking BPA's BRD

With our proposed method, we track a time-varying reference with a prediction horizon of $N_p = 5$; see Figure 3. Since the data obtained from BPA is on the order of GW, we scale the reference down to satisfy the magnitude constraint. However, this is *not* required for the success of the algorithm, only to aid in exposition of the results.

The 1-norm tracking error of the signal in Figure 3 is 16.3%, and can be attributed to 2 factors: (i) the reference is only guaranteed to satisfy the magnitude constraint (3) so it may not be feasible for the other constraints and (ii) the algorithm only guarantees ISS and not asymptotic tracking. However, from experience we believe (i) to be the contributing factor. Other numerical experiments conducted suggest that it is possible to make the error quite small by increasing N_p if the constraints are all feasible.
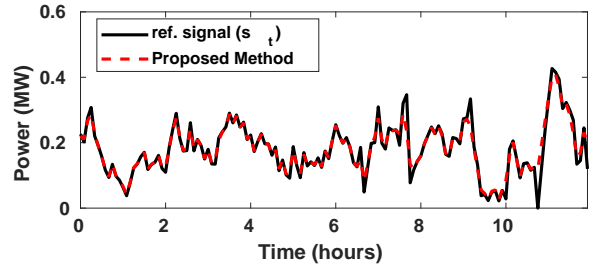

Fig. 3: Tracking the time varying reference with the proposed method.

VII. Conclusion

We propose a real time optimization algorithm with distributed computation and hierarchical communication structure for the resource allocation of flexible loads in the smart grid. Our algorithm has two key innovations: (i) the utilization of predictions and (ii) a state augmentation technique to handle dynamic constraints.

Future work includes: (i) analyzing further the effects of the state augmentation technique, similar to the penalty method technique applied in [16] and (ii) the development of asymptotic results for constrained time varying optimization.

References

[1] P. Barooah, *Smart Grid Control: An Overview and Research Opportunities*. Springer Verlag, 2019, ch. Virtual energy storage from flexible loads: distributed control with QoS constraints, pp. 99–115.

[2] N. J. Cammardella, R. W. Moye, Y. Chen, and S. P. Meyn, "An energy storage cost comparison: Li-ion batteries vs Distributed load control," in *2018 Clemson University Power Systems Conference (PSC)*, Sep. 2018, pp. 1–6.

[3] A. Coffman, A. Bušić, and P. Barooah, "Virtual energy storage from TCLs using QoS preserving local randomized control," in *5th ACM International Conference on Systems for Built Environments (BuildSys)*, November 2018, p. 10.

[4] M. Liu, S. Peeters, D. S. Callaway, and B. J. Claessens, "Trajectory tracking with an aggregation of domestic hot water heaters: Combining model-based and model-free control in a commercial deployment," *IEEE Transactions on Smart Grid*, 2019.

[5] J. Mathias, R. Kaddah, A. Buic, and S. Meyn, "Smart fridge/dumb grid? demand dispatch for the power grid of 2020," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 2498–2507.

[6] H. Hao, A. Kowli, Y. Lin, P. Barooah, and S. Meyn, "Ancillary service for the grid via control of commercial building HVAC systems," in *American Control Conference*, June 2013, pp. 467–472.

[7] A. Aghajanzadeh and P. Therkelsen, "Agricultural demand response for decarbonizing the electricity grid," *Journal of Cleaner Production*, vol. 220, pp. 827 – 835, 2019.

[8] Y. Chen, M. U. Hashmi, J. Mathias, A. Bušić, and S. Meyn, "Distributed control design for balancing the grid using flexible loads," in *IMA Volume on the Control of Energy Markets and Grids*, 2017, pp. 1–26.

[9] N. Cammardella, J. Mathias, M. Kiener, A. Bušić, and S. Meyn, "Balancing California's grid without batteries," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 7314–7321.

[10] C. Sun, M. Ye, and G. Hu, "Distributed time-varying quadratic optimization for multiple agents under undirected graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3687–3694, July 2017.

[11] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, 2016.

[12] J. Brooks and P. Barooah, "Coordination of loads for ancillary services with Fourier domain consumer QoS constraints," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6148–6155, 2019.

[13] ——, "Consumer-aware load control to provide contingency reserves using frequency measurements and inter-load communication," in *American Control Conference*, July 2016, pp. 5008 – 5013.

[14] G. Hug, S. Kar, and C. Wu, "Consensus + innovations approach for distributed multiagent coordination in a microgrid," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1893–1903, July 2015.

[15] C. Zhao, U. Topcu, N. Li, and S. Low, "Design and stability of load-side primary frequency control in power systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1177–1189, May 2014.

[16] A. Cherukuri and J. Cortés, "Distributed coordination of DERs with storage for dynamic economic dispatch," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 835–842, March 2018.

[17] L. Bai, C. Sun, Z. Feng, and G. Hu, "Distributed continuous-time resource allocation with time-varying resources under quadratic cost functions," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 823–828.

[18] C. Zhao, U. Topcu, and S. H. Low, "Optimal load control via frequency measurement and neighborhood area communication," *Power Systems, IEEE Transactions on*, pp. 3576–3587, 2013.

[19] T. T. Doan and C. L. Beck, "Distributed lagrangian methods for network resource allocation," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 650–655.

[20] A. Coffman, N. Cammardella, P. Barooah, and S. Meyn, "Aggregate capacity of TCLs with cycling constraints," *arXiv preprint arXiv:1909.11497*, 2019.

[21] A. R. Coffman, Z. Guo, and P. Barooah, "Characterizing capacity of flexible loads for providing grid support," *arXiv preprint arXiv:2005.01591*, 2020.

[22] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Aggregate flexibility of thermostatically controlled loads," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 189–198, Jan 2015.

[23] A. A. Goldstein, "Convex programming in hilbert space," *Bull. Amer. Math. Soc.*, vol. 70, no. 5, pp. 709–710, 09 1964. [Online]. Available: https://projecteuclid.org:443/euclid.bams/1183526263

[24] A. Y. Popkov, "Gradient methods for nonstationary unconstrained optimization problems," *Automation and Remote Control*, vol. 66, no. 6, pp. 883–891, Jun 2005.

[25] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2009.

[26] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[27] J. W. Daniel, "Stability of the solution of definite quadratic programs," *Mathematical Programming*, vol. 5, no. 1, pp. 41–53, Dec 1973.

[28] H. K. Khalil, *Nonlinear systems, 3rd ed.* Upper Saddle River, NJ, USA: Prentice Hall, 1996.

[29] C. Zhao, U. Topcu, and S. Low, "Frequency-based load control in power systems," in *American Control Conference*, 2012, pp. 4423–4430.

[30] A. R. Coffman, M. Hale, and P. Barooah, "Predictive resource allocation for flexible loads with local QoS," *arXiv preprint arXiv:2006.16044*, 2020.

[31] R. S. Varga, *Geršgorin and his circles*. Springer Science & Business Media, 2010, vol. 36.

APPENDIX

A. Proof of Lemma 2

Proceeding directly, by the triangle inequality we have

$$\frac{1}{N}\|Pz_{t-1}^* - z_t^*\| \le \frac{1}{N}\|z_{t-1}^* - z_t^*\| + \frac{1}{N}\|Pz_{t-1}^* - z_{t-1}^*\|,$$
$$\le \frac{\bar{u}_{t,t-1}^*}{\lambda_{\min}(\nabla^2\eta)} + \frac{2}{N}\|z_{t-1}^*\|.$$

We can bound $\|z_{t-1}^*\|$ by using Lemma 1 where z_t^* will be zero when $u_t^* = u_t^{*,0}$, yielding

$$\frac{1}{N}\|Pz_{t-1}^* - z_t^*\| \le \frac{\bar{u}_{t,t-1}^*}{\lambda_{\min}(\nabla^2\eta)} + \frac{2\tilde{u}_t^*}{\lambda_{\min}(\nabla^2\eta)} = \frac{\bar{g}_t^*}{\lambda_{\min}(\nabla^2\eta)}.$$

B. Proof of Theorem 1

A more detailed proof can be found in [30]. We start with the developed vectorized notation,

$$\|z_t - z_t^*\| = \|\Pi_\mathcal{D}(P\psi_{t-1}) - \Pi_\mathcal{D}(\psi_t^*)\| \le \|P\psi_{t-1} - \psi_t^*\|$$
$$= \|Pz_{t-1} - z_t^* - \alpha\big(P\nabla\eta(z_{t-1}) - \nabla\eta(z_t^*)\big)\|,$$

where we used Proposition 2 and the non-expansive property of the projection operator. From Proposition 3 and 4 it can be shown that,

$$\|z_t - z_t^*\| \le M(\alpha)\|Pz_{t-1} - z_t^*\| + \alpha\|Pu_{t-1} - u_t^*\|,$$

where $M(\alpha) = \|I - \alpha\nabla^2\eta\|$. We iterate this backwards a total of t times to reach $t = 0$ whilst using Lemmas 1 and 2, yielding:

$$\|z_t - z_t^*\| \le M^t(\alpha)\|z_0 - z_0^*\|$$
$$+ \alpha\sum_{\ell=1}^t M^{t-\ell}(\alpha)\bigg(\|Pu_{t-\ell} - u_t^*\| + \frac{N\bar{g}_t^*}{\alpha\lambda_{\min}(\nabla^2\eta)}\bigg).$$

Now, from our assumptions we can bound the quantity in parentheses in the summation by $\bar{u}$ yielding,

$$\|z_t - z_t^*\| \le M^t(\alpha)\|z_0 - z_0^*\| + \frac{\alpha\bar{u}}{1 - M(\alpha)}$$

which will give the desired result as long as $M(\alpha) < 1$, which we ensure next. Denote $\lambda_i(\nabla^2\eta)$, the ith eigenvalue of $\nabla^2\eta$. To guarantee $M(\alpha) < 1$ it is sufficient to require

$$0 < \alpha\lambda_{\max}(\nabla^2\eta) < 1.$$

Applying the Gershgorin circle theorem [31] then yields,

$$\alpha \in \left(0, \frac{1}{\zeta^{\max} + N}\right),$$

for $M(\alpha) < 1$.