

Agent-Supervisor Coordination for Decentralized Event-Triggered Optimization

Priyank Srivastava Guido Cavraro Jorge Cortés

Abstract—This paper proposes decentralized resource-aware coordination schemes for solving network optimization problems defined by objective functions that combine locally evaluable costs with network-wide coupling components. These methods are well suited for a group of supervised agents trying to solve an optimization problem under mild coordination requirements. Each agent has information on its local cost and coordinates with the network supervisor for information about the coupling term of the cost. The proposed approach is feedback-based and asynchronous by design, guarantees anytime feasibility, and ensures the asymptotic convergence of the network state to the desired optimizer. Numerical simulations on a power system example illustrate our results.

Index Terms—Event-triggered control; Optimization; Decentralized algorithms.

I. INTRODUCTION

RECENT advancements in digital systems, communication, and sensing technologies have enabled the deployment of large-scale cyber-physical systems in multiple domains. Owing to the intrinsic modular structure of these systems, their applications can often be modeled as optimization problems over networks. In many scenarios, the associated objective functions result from the combination of locally separable costs with global coupling terms whose evaluation requires members of a network layer to interact with an entity at a higher layer. From this arises the need for algorithmic implementations that help individual agents efficiently decide when to resort to such interactions. Motivated by these observations, this paper synthesizes resource-aware agent-supervisor coordination schemes that achieve asymptotic convergence to the desired network optimizer.

Literature Review: The present paper relies on two areas with significant recent activity: distributed optimization and event-triggered control. Because of the impossibility of surveying their vast literature, we provide the introductory refer-

ences [1]–[3] for the first and [4]–[6] for the second. A majority of work in network optimization builds on consensus-based approaches to solve problems where the overall objective is the summation of individual agents’ private cost functions. In such scenarios, each agent maintains and updates an estimate of the complete solution vector, using local computation and peer-to-peer information exchange. An alternative architecture has agents send and receive information from a central entity or a network supervisor. Such architecture is well suited for scenarios where the objective function of the optimization problem is the combination of a component separable among the agents and another one coupling all the agents’ decision variables. Notable examples of applications exhibiting this structure include virtual power plants [7], [8], where a central entity works as an aggregator enabling the participation of a cluster of distributed energy resources in the energy market; HVAC systems in intelligent buildings [9], [10], where a central processor manages the main air supply and helps coordinate a group of thermal zones equipped with thermostats and controllers regulating heated or cooled air input; multi-agent systems with access to the cloud [11], which provides superior processing and storing capabilities; and sensor-actuator networks with a central computation node [12], [13] responsible for computing the control signal with measurements from distributed sensors. In these scenarios, there is a need to structure the interaction between individual agents and the network supervisor to design solutions that scale up. Event-triggered control, see, e.g., [4]–[6] and references therein, offers a framework to prescribe, in a principled way, when to efficiently use the available resources while still guaranteeing a desired quality of service in performing the intended task. Several works [14], [15] have explored the use of event-triggered approaches for achieving network coordination tasks. Of particular relevance to our work are [12], [16], [17], which consider similar network architectures, and [18], which mixes continuous updates, computed with the locally available information, with aperiodic updates, computed with the externally provided information. An issue in event-triggered control is the emergence of an infinite number of triggering times in a finite time interval, a.k.a. *Zeno behavior*. Zeno behavior is addressed in [12], [16] by using *time regularization*, i.e., preventing by design any update before a certain dwell time has elapsed. In general, time regularization requires an offline computation with global information. A final body of work we build on is that of continuous projected dynamical systems for optimization, cf. [19]–[21].

Statement of Contributions: We consider network optimization problems where the objective function is the sum of

This work was authored in part by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract DE-AC36-08GO28308 and supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the paper do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the paper for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

This work was partially supported by NSF under Grant ECCS-1947050.

P. Srivastava is with the Department of Mechanical Engineering, Massachusetts Institute of Technology. psrivast@mit.edu. G. Cavraro is with the National Renewable Energy Laboratory. guido.cavraro@nrel.gov. J. Cortés is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego. cortes@ucsd.edu.

a component given by the summation of local costs and a coupling component whose evaluation requires knowledge of all the agents' decision variables. Individual agents rely on the network supervisor to obtain information about the coupling component and to solve the optimization problem. Our contributions are structured in two blocks, corresponding to unconstrained and constrained problems. For unconstrained systems, we build on the gradient descent dynamics; for constrained systems with separable constraints, we build on globally projected dynamical systems. For both cases, we design novel event-triggered agent-supervisor coordination algorithms where agents continuously employ their local information and resort to opportunistic interactions with the supervisor for information about the coupling cost. The criterion for triggering employed by each agent depends only on locally available information, which makes the proposed approach suitable for applications where, due to privacy concerns, it is preferable not to share the local cost functions. We show the monotonic decrease of the objective function and establish the existence of a minimum inter-event time, thus ruling out Zeno behavior (without the need for any time regularization) and ensuring asymptotic convergence to the optimizer. In the constrained case, we also show that the feasible set is positively invariant, thus guaranteeing anytime feasibility. We illustrate the performance of the proposed algorithms with numerical simulations of a power system application.

II. PRELIMINARIES

Here¹ we review the basic concepts on convex analysis, event-triggered control, and constrained optimization.

Event-Triggered Control: The basics of event-triggered control following [4], [5] are presented next. Consider

$$\dot{x} = f(x, u), \quad (1)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$ denote the system state and input, respectively. Assume there exists a control $u = k(x)$, such that the closed-loop dynamics,

$$\dot{x} = f(x, k(x + e)), \quad (2)$$

abbreviated ψ_{cl} , is input-to-state stable (ISS) with respect to the error $e \in \mathbb{R}^n$. Formally, assume there exists a Lyapunov function V such that its Lie derivative along (2) satisfies

$$\mathcal{L}_{\psi_{\text{cl}}} V \leq -\alpha(\|x\|) + \gamma(\|e\|),$$

where α and γ are class \mathcal{K}_∞ functions. The implementation of the closed-loop system (2) requires continuous updates of the actuator, which is not realizable in practice. Instead, event-triggered control seeks to prescribe opportunistic updates of the actuator that ensure the convergence properties of the

closed-loop system are retained. This leads to a sample-and-hold implementation of the controller of the form

$$u(t) = k(x(t_k)) \quad t \in [t_k, t_{k+1}), \quad (3)$$

where $\{t_k\}_{k=0}^\infty$ are the *triggering times* when the control input is updated. To ensure the stability of the nonlinear system under (3) and to prescribe the triggering times, we look at the evolution of the Lyapunov function V . Define the error variable as $e = x - x^k$, where we use the shorthand notation $x^k = x(t_k)$. With $\sigma \in (0, 1)$, if the error satisfies $\gamma(\|e\|) \leq \sigma\alpha(\|x\|)$, then during the time $[t_k, t_{k+1})$, $\mathcal{L}_{\psi_{\text{cl}}} V \leq (\sigma - 1)\alpha(\|x\|)$. Note that at $t = t_k$, the error satisfies $e = 0$; hence, setting $t_0 = 0$, we ensure $\mathcal{L}_{\psi_{\text{cl}}} V < 0$ defining the triggering times as

$$t_{k+1} = \min\{t > t_k \mid \gamma(\|e\|) = \sigma\alpha(\|x\|)\}. \quad (4)$$

Although (4) guarantees the stability of the closed-loop system, Zeno behavior could arise. Hence, for implementation in practice and to conclude asymptotic stability, it is necessary to have a uniform lower bound $\tau > 0$ on the inter-event times, i.e., $t_{k+1} - t_k \geq \tau$ for all k . We refer to τ as the minimum inter-event time (*MIET*). The existence of a MIET is guaranteed with the control law (3) and the triggering condition (4) if the dynamics (1) is linear, and also for certain nonlinear systems under suitable assumptions, cf. [4].

Constrained Optimization via Continuous Projected Dynamical Systems: Here we review the basic concepts on the stability of continuous projected dynamical systems and their application in constrained optimization following [20], [21]. Consider the following optimization problem

$$\min_{x \in \Omega} h(x), \quad (5)$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function and $\Omega \subseteq \mathbb{R}^n$ is a convex set. Problem (5) can be solved using

$$\dot{x} = \Pi_\Omega(x - \lambda \nabla h(x)) - x, \quad (6)$$

where $\lambda > 0$ is a design parameter. Unlike the commonly employed projected gradient dynamics, cf. [22], which is discontinuous at the boundary of Ω , the dynamics (6) is continuous due to the gradual application of the projection operator throughout the constraint set. The following result characterizes its convergence properties.

Theorem II.1. (*Forward invariance of the feasible set and convergence to an optimizer [20], [21]*): Assume that ∇h is locally Lipschitz continuous on an open set containing Ω . Then

- (i) The solution of (6) approaches the set Ω exponentially fast. Moreover, if $x(0) \in \Omega$, then $x(t) \in \Omega$ for all $t > 0$.
- (ii) For all $\lambda > 0$, the dynamics (6) is stable, and for any initial condition $x(0) \in \Omega$, the trajectory of (6) converges to a solution of (5).

III. PROBLEM FORMULATION

Consider a network of $n \in \mathbb{Z}$ agents and a supervisor, collectively seeking to solve

$$\min_{x \in \mathcal{X}} \underbrace{\sum_{i=1}^n f_i(x_i)}_{f(x)} + g(x), \quad (7)$$

¹Throughout the paper, we employ the following notation. Let \mathbb{R} , $\mathbb{R}_{\geq 0}$, and \mathbb{Z} represent the set of real, nonnegative real, and integer numbers, respectively. $|\cdot|$ and $\|\cdot\|$ denote the absolute value of a scalar and the 2-norm of a vector, respectively. $(\cdot)^T$ denotes the transpose of a vector or a matrix. A vector or a matrix of appropriate dimensions of all zero entries is denoted by $\mathbf{0}$. $A \succ \mathbf{0}$ means that the matrix A is positive definite. A continuous function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class \mathcal{K}_∞ if it is strictly increasing, $\alpha(0) = 0$, and $\alpha(x) \rightarrow \infty$ as $x \rightarrow \infty$. For a convex set Ω , $\Pi_\Omega(y)$ denotes the projection of a point $y \in \mathbb{R}^n$ on Ω , i.e., $\Pi_\Omega(y) = \operatorname{argmin}_{x \in \Omega} \|x - y\|$.

where, for all $i \in \{1, \dots, n\}$, $f_i : \mathbb{R} \rightarrow \mathbb{R}$ is the local cost function of agent i , $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function coupling all the agents' states, $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$ is the constraint set, and \mathcal{X}_i is agent i 's constraint set. Each agent $i \in \{1, \dots, n\}$ has knowledge of its local state $x_i \in \mathbb{R}$, constraint set \mathcal{X}_i and cost f_i , and relies on the network supervisor to obtain information pertaining the coupling cost g . For simplicity of exposition, we assume that (7) has a unique solution x^* , albeit the results of the paper can be extended with minor modifications to the case of multiple optimizers. We make the following assumptions on the cost functions and the constraints.

Assumption 1. (*Convexity and Lipschitz gradients*): The functions $\{f_i\}_{i=1}^n$ and g are convex; the functions $\{f_i\}_{i=1}^n$ are twice continuously differentiable, and g is continuously differentiable with locally Lipschitz gradient; and the sets $\{\mathcal{X}_i\}_{i=1}^n$ are compact and convex.

Our goal is to design a decentralized algorithm that allows the agents to collectively solve (7). We want the algorithm to be anytime, meaning that if the network state starts feasible, it remains so during the algorithm's execution. This anytime nature is desirable in applications where the optimization problem is not stand-alone and its solution serves as an input to another layer in the control design. In such cases, the algorithm should yield a feasible solution even if terminated in finite time. Note that, without the presence of the coupling function g , (7) could be solved easily by having each agent i solve a local optimization problem with the function f_i over the constraint set \mathcal{X}_i . Instead, the presence of g couples the agents' decisions. Since information about g is not available at all times, we seek to endow the individual agents with a criterion that allows them to determine when to query the supervisor in an opportunistic fashion – this is what corresponds to the event-triggered component of the algorithmic solution. Based on the application at hand and the supervisor's capabilities, the coordination between the supervisor and the agents could be feedback-based or computation-based:

(i) *Sensing-based*: each agent $i \in \{1, \dots, n\}$ can evaluate $\nabla_{x_i} g$ with its local information and the one broadcast from the supervisor, when an agent asks for an update. This is because the supervisor has access from its own measurements to enough knowledge about g . This is common in cyber-physical scenarios where access to field measurements of the physical layer provides global information about the network state (for instance, in power systems, see e.g. [23], local measurements of the frequency deviation provide information about the overall network mismatch in meeting the prescribed load). In this case, the states of the agents remain private (the virtual power plant in [8] is an example falling in this case).

(ii) *Computation-based*: the supervisor knows the functional form of the cost. Whenever an agent asks for an update, the supervisor gathers the state of all the agents, evaluates ∇g , and broadcasts it to the agents (scheduling links and channels for transmission of information in wireless networks is an example scenario for this case, see, e.g., [13]).

The forthcoming design and the ensuing analysis can be applied to both scenarios.

Remark 1. (Synchronicity of the updates): The coordination between the agents and the supervisor described above requires the broadcast of information to all the agents whenever any agent asks for an update. As a result, $\{\nabla_{x_i} g(x)\}_{i=1}^n$ are updated synchronously. From a practical viewpoint, this is reasonable in the sensing-based scenario because once the supervisor has the required information available, it can broadcast it to all the agents. Instead, in the computation-based scenario, the supervisor might not have up to date information about all the agents. In this case, we assume that the gathering of information by the supervisor can be done simultaneously from all the agents whenever there is an update request. •

IV. EVENT-TRIGGERED COORDINATION FOR UNCONSTRAINED PROBLEMS

Here, an event-triggered decentralized algorithm to solve (7) when $\mathcal{X} = \mathbb{R}^n$ is provided. Consider the standard gradient-descent dynamics

$$\dot{x} = -\lambda(\nabla f(x) + \nabla g(x)),$$

where $\lambda > 0$ is a design parameter. For agent $i \in \{1, \dots, n\}$, this takes the form

$$\dot{x}_i = -\lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x)). \quad (8)$$

From an implementation viewpoint, the first term in (8) can be evaluated locally by each agent, the second term, however, entails continuous communication with the supervisor. We avoid this by designing an event-triggered scheme that has the supervisor broadcast the information needed to compute $\nabla g(x)$ in an opportunistic fashion. With the shorthand notation $x^k = x(t_k)$, consider the dynamics

$$\dot{x} = -\lambda(\nabla f(x) + \nabla g(x^k)) \quad t_k \leq t < t_{k+1}. \quad (9)$$

To implement (9), the network supervisor needs to broadcast the information required to compute $\nabla g(x)$ only at some specified times $\{t_k\}_{k=0}^\infty$. Here, $\nabla g(x^k)$ is the equivalent of the input in the standard event-triggered control, cf. Section II. When convenient, we refer to the dynamics (9) as ψ_{ev} . The next result identifies a decentralized condition on the triggering times $\{t_k\}_{k=0}^\infty$ that ensures that the dynamics (9) is stable. By decentralized, we mean that each agent $i \in \{1, \dots, n\}$ can identify the triggering criterion locally without knowing the states of the other agents or the coupling function.

Proposition IV.1. (*Decentralized trigger*): Let $x^k \neq x^*$ be the state when the trigger was last implemented, $\sigma \in (0, 1)$, and define $\mathcal{F}_k = \{x \in \mathbb{R}^n \mid f(x) + g(x) \leq f(x^k) + g(x^k)\}$. Then for all $\lambda > 0$, the dynamics (9) is stable and the value of the objective function $f + g$ is nonincreasing if the triggering times are updated according to

$$t_{k+1} = \min_{i \in \{1, \dots, n\}} \min\{t > t_k \mid L_g |x_i - x_i^k| = \sigma |\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k)| \neq 0\}, \quad (10)$$

where L_g is the Lipschitz constant of ∇g over \mathcal{F}_0 .

Proof: Consider the Lyapunov function $V : \mathbb{R}^n \rightarrow \mathbb{R}$

$$V(x) = f(x) + g(x) - f(x^*) - g(x^*), \quad (11)$$

whose Lie derivative is

$$\mathcal{L}_{\psi_{\text{ev}}} V(x) = -\lambda(z+e)^\top z \leq -\lambda\|z\|^2 \left(1 - \frac{\|e\|}{\|z\|}\right),$$

where $z = [z_1 \dots z_n]^\top$, $z_i = \nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k)$, and $e = \nabla g(x) - \nabla g(x^k)$. At $t = t_k$, we have $e = 0$; then the error starts increasing as x^k becomes obsolete. However, $\mathcal{L}_{\psi_{\text{ev}}} V \leq 0$ if we ensure that $\|e\| \leq \sigma\|z\|$. The direct evaluation of the latter condition requires complete information about the network state. However, note that $\|e\| \leq L_g^k \|x - x^k\|$, where L_g^k is the Lipschitz constant of ∇g over \mathcal{F}_k . Hence, we can guarantee the stability of (9) if

$$L_g^k \|x - x^k\| \leq \sigma\|z\|. \quad (12)$$

The triggering rule (10) ensures that (12) is satisfied noting that the set \mathcal{F}_k is forward invariant, $\mathcal{F}_{k+1} \subseteq \mathcal{F}_k$, and hence $L_g \geq L_g^k$ for all k . ■

From Proposition IV.1, it is clear that if the agents have knowledge of (an upper bound on) L_g , they can check (10) locally and request the supervisor for an update accordingly. Although (10) guarantees that the dynamics (9) is stable, we still need to establish the convergence to x^* and whether the proposed event-triggered scheme is Zeno-free. We prove both facts in the next result.

Proposition IV.2. (Non-Zeno behavior and convergence to the optimizer): *With the notation of Proposition IV.1, if the triggering times are updated as (10), then for all $\lambda > 0$, the MIET is lower bounded by $\tau = \frac{1}{\lambda H} \log(\sigma\lambda H/L_g + 1) > 0$, where $H = \max_{i \in \{1, \dots, n\}} \max_{x \in \mathcal{F}_0} \nabla_{x_i}^2 f_i(x_i)$. Moreover, any trajectory of (9) converges asymptotically to x^* .*

Proof: If $x^k = x^*$, then the result is immediate. Assume then that $x^k \neq x^*$. Let $\mathcal{I} = \{i \mid z_i(t_k) \neq 0\}$. Since for all $i \in \{1, \dots, n\}$, $\dot{z}_i = -\lambda \nabla_{x_i}^2 f_i(x_i) z_i$, we deduce that $z_i = 0$ for all $t \in [t_k, t_{k+1})$ if $i \notin \mathcal{I}$. For $i \in \mathcal{I}$, we examine the evolution of $|x_i - x_i^k|/|z_i|$,

$$\begin{aligned} \frac{d}{dt} \frac{|x_i - x_i^k|}{|z_i|} &= \frac{(x_i - x_i^k) z_i}{\sqrt{(x_i - x_i^k)^2} \sqrt{z_i^2}} - \frac{z_i \dot{z}_i \sqrt{(x_i - x_i^k)^2}}{|z_i|^3} \\ &\leq 1 + \frac{|\dot{z}_i| |x_i - x_i^k|}{|z_i| |z_i|} \leq 1 + \lambda H \frac{|x_i - x_i^k|}{|z_i|}. \end{aligned} \quad (13)$$

Now consider the differential equation $\dot{y} = 1 + \lambda H y$, $t_k \leq t < t_{k+1}$, with initial condition $y(t_k) = 0$, whose closed-form solution is given by

$$y = \frac{1}{\lambda H} (e^{\lambda H(t-t_k)} - 1), \quad t_k \leq t < t_{k+1}.$$

By the Comparison Principle, cf. [24, Lemma 3.4], we have

$$\frac{|x_i - x_i^k|}{|z_i|} \leq \frac{1}{\lambda H} (e^{\lambda H(t-t_k)} - 1), \quad t_k \leq t < t_{k+1}.$$

Equating the right-hand side of the above inequality with σ/L_g implies that the inter-event time is lower bounded by τ provided $z_i \neq 0$ for all $t \in [t_k, t_{k+1})$ and each $i \in \mathcal{I}$. We reason by contradiction to prove this. Since the ratio $|x_i - x_i^k|/|z_i|$ is bounded, $z_i = 0$ only if $x_i - x_i^k = 0$. Let $\bar{t} = \min\{t > t_k \mid x_i - x_i^k = 0\}$. Since $x_i - x_i^k = 0$ and $z_i \neq 0$

at $t = t_k$, this means that the sign of z_i has to change before \bar{t} , and from the continuity of the dynamics, there exists $\hat{t} < \bar{t}$ such $z_i(\hat{t}) = 0$, which contradicts $z_i \neq 0$ for all $t \in [t_k, \bar{t})$. To prove the attractivity part, note that from Proposition IV.1, $\mathcal{L}_{\psi_{\text{ev}}} V \leq 0$, and hence, $\mathcal{L}_{\psi_{\text{ev}}} V(x) < 0$ for all $x \neq x^*$ as $z_i \neq 0$ for all $i \in \mathcal{I}$ and all $t \in [t_k, t_{k+1})$. ■

Remark 2. (Differentiability of the local objective functions): Note that ruling out Zeno behavior in Proposition IV.2 relies on the functions $\{f\}_{i=1}^n$ being twice continuously differentiable, whereas the dynamics (9) and the triggering condition (10) involve only first-order derivatives. We believe, although we do not pursue it here for space reasons, that Proposition IV.2 can be extended for the case when the separable component of the objective function is just continuously differentiable, using tools from nonsmooth analysis, e.g., [25], [26]. •

Remark 3. (Self-triggered implementation): In the absence of errors in the solution of the differential equations by the individual agents, the criterion (10) can also be implemented in a self-triggered fashion. In fact, we can write it as

$$\begin{aligned} t_{k+1}^i &= \min\{t > t_k \mid L_g |x_i - x_i^k| = \\ &\quad \sigma |\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k)| \neq 0\}, \\ t_{k+1} &= \min_{i \in \{1, \dots, n\}} t_{k+1}^i. \end{aligned}$$

This means that, with the information provided at time t_k , each agent $i \in \{1, \dots, n\}$ can compute t_{k+1}^i by solving its differential equation, and convey it to the supervisor, which can then schedule the next triggering event at t_{k+1} . •

V. EVENT-TRIGGERED COORDINATION FOR CONSTRAINED PROBLEMS

To deal with constrained problems, we build on the continuous projected dynamics (6), which takes the form

$$\dot{x} = \Pi_{\mathcal{X}}(x - \lambda(\nabla f(x) + \nabla g(x))) - x,$$

where $\lambda > 0$. Its event-triggered counterpart is

$$\dot{x} = \Pi_{\mathcal{X}}(x - \lambda(\nabla f(x) + \nabla g(x^k))) - x, \quad (14)$$

for $t \in [t_k, t_{k+1})$. When convenient, we refer to the dynamics (14) as ψ_{evco} . The following result identifies a decentralized condition on the triggering times $\{t_k\}_{k=0}^\infty$ that ensures the stability of (14).

Proposition V.1. (Decentralized trigger for constrained problems): *Let $x^k \neq x^*$ be the state when the trigger was last implemented, and $\sigma \in (0, 1)$. If $x(0) \in \mathcal{X}$, then for all $\lambda > 0$, $x(t) \in \mathcal{X}$ for all $t > 0$, the dynamics (14) is stable and the value of the objective function $f + g$ is non-increasing if the triggering times are updated as*

$$\begin{aligned} t_{k+1} &= \min_{i \in \{1, \dots, n\}} \min\{t > t_k \mid \lambda \bar{L}_g |x_i - x_i^k| = \\ &\quad \sigma |\Pi_{\mathcal{X}_i}(x_i - \lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k))) - x_i| \neq 0\}, \end{aligned} \quad (15)$$

where \bar{L}_g is the Lipschitz constant of ∇g over \mathcal{X} .

Proof: We start by noting that from Theorem II.1, for $t \in [t_k, t_{k+1})$, positive invariance of the feasible set \mathcal{X} under (14) can be established by taking $h(x) \equiv f(x) + \nabla g(x^k)^\top x$. To

prove stability, consider again the Lyapunov function candidate V defined in (11), whose Lie derivative is now given by

$$\mathcal{L}_{\psi_{\text{veco}}} V(x) = (\nabla f(x) + \nabla g(x^k) + e)^\top \bar{z},$$

where $\bar{z} = [\bar{z}_1 \dots \bar{z}_n]^\top$, $\bar{z}_i = \Pi_{\mathcal{X}_i}(x_i - \lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k))) - x_i$, and $e = \nabla g(x) - \nabla g(x^k)$. It is well known, cf. [21], that for a convex set Ω

$$(u - \Pi_\Omega(u))^\top (\Pi_\Omega(u) - v) \geq 0,$$

for all $v \in \Omega$ and all $u \in \mathbb{R}^n$. With $\Omega = \mathcal{X}$, $v = x$, and $u = x - \lambda(\nabla f(x) + \nabla g(x^k))$, this implies that

$$(\lambda \nabla f(x) + \lambda \nabla g(x^k) + \bar{z})^\top \bar{z} \leq 0.$$

Using this, we upper bound the Lie derivative as

$$\mathcal{L}_{\psi_{\text{veco}}} V(x) \leq -\frac{1}{\lambda} \bar{z}^\top \bar{z} + e^\top \bar{z} \leq -\|\bar{z}\|^2 \left(\frac{1}{\lambda} - \frac{\|e\|}{\|\bar{z}\|} \right).$$

This expression is analogous to that of $\mathcal{L}_{\psi_{\text{ev}}} V$ in the proof of Proposition IV.1, and a similar argument concludes the proof. ■

As in the unconstrained case, without excluding Zeno behavior, Proposition V.1 is not enough to conclude the asymptotic convergence to x^* .

Proposition V.2. (*Non-Zeno behavior and convergence to the optimizer in the constrained case*): *With the notation of Proposition V.1, if the triggering times are updated as (15), then for all $\lambda < \bar{\lambda} = 1/\bar{H}$, the MIET is lower bounded by $\bar{\tau} = \log(\sigma/\lambda \bar{L}_g + 1) > 0$, where $\bar{H} = \max_{i \in \{1, \dots, n\}} \max_{x_i \in \mathcal{X}_i} \nabla_{x_i}^2 f_i(x_i)$. Moreover, any trajectory of (14) with $x(0) \in \mathcal{X}$ converges asymptotically to x^* .*

Proof: Since $\{\mathcal{X}_i\}_{i=1}^n$ are compact and convex, without loss of generality let $\mathcal{X}_i = \{x_i \in \mathbb{R} \mid \underline{x}_i \leq x_i \leq \bar{x}_i\}$ for all i . For each agent $i \in \{1, \dots, n\}$, define $u_i : \mathcal{X}_i \rightarrow \mathbb{R}$ as $u_i(x_i) = x_i - \lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k))$. The derivative of u_i w.r.t x_i is given by

$$\frac{du_i}{dx_i} = 1 - \lambda \nabla_{x_i}^2 f_i(x_i).$$

For a given $i \in \{1, \dots, n\}$, the sign of du_i/dx_i at any $x_i \in \mathcal{X}_i$ depends on the value of λ and $\nabla_{x_i}^2 f_i(x_i)$. If $\lambda < \bar{\lambda}$, then $du_i/dx_i > 0$ for all $x_i \in \mathcal{X}_i$. This means that if there is a point $\hat{x}_i \in \mathcal{X}_i$ such that $\Pi_{\mathcal{X}_i}(u_i(\hat{x}_i)) = \bar{x}_i$, then $\Pi_{\mathcal{X}_i}(u_i(x_i)) = \bar{x}_i$ for all $x_i > \hat{x}_i$. Similarly, if there is a point $\tilde{x}_i \in \mathcal{X}_i$ such that $\Pi_{\mathcal{X}_i}(u_i(\tilde{x}_i)) = \underline{x}_i$, then $\Pi_{\mathcal{X}_i}(u_i(x_i)) = \underline{x}_i$ for all $x_i < \tilde{x}_i$. Therefore, \dot{z}_i can be represented as a set-valued map, cf. [26],

$$|\dot{z}_i| = \begin{cases} |\lambda \nabla_{x_i}^2 f_i(x_i) \bar{z}_i| & \tilde{x}_i < x_i < \hat{x}_i, \\ [|\lambda \nabla_{x_i}^2 f_i(x_i) \bar{z}_i|, |\bar{z}_i|] & x_i = \tilde{x}_i, \hat{x}_i, \\ |\bar{z}_i| & \underline{x}_i \leq x_i < \tilde{x}_i, \hat{x}_i < x_i \leq \bar{x}_i. \end{cases}$$

Since $\lambda < \bar{\lambda}$, we have an expression similar to (13) for $\frac{d}{dt}|x_i - x_i^k|/|\bar{z}_i|$ for all i with $\bar{z}_i(t_k) \neq 0$, with λH replaced by 1. The remainder of the argument follows analogously to the proof of Proposition IV.2. ■

The upper bound on λ in Proposition V.2 is conservative in general. In fact, the dynamics (14) with the triggering rule (15) may be Zeno-free even if this condition is not satisfied, something that we have observed in simulation.

VI. SIMULATIONS

Here we test our coordination approach in a power distribution scenario, where n generators managed by a distribution system operator that acts as the network supervisor seek to minimize the total power generation cost. The active power output of generator $i = \{1, \dots, n\}$ is x_i and the power that flows from the external grid to the distribution grid through the substation, labeled as node 0, is denoted as x_0 . Assuming the power line losses are negligible, x_0 can be linearized as

$$x_0(x) \simeq -\sum_{i=1}^n x_i + c, \quad (16)$$

where c depends on the grid load, see, e.g., [8]. We choose polynomial costs for power injection at each node $i = \{0, 1, \dots, n\}$. Each function f_i is known only by generator $i \in \{1, \dots, n\}$, and the network supervisor has access only to f_0 . By substituting (16) into $f_0(x_0)$, the problem of minimizing the total power cost is equivalent to solving (7) with $g(x) = f_0(-\sum_{i=1}^n x_i + c)$. From the chain rule, $\nabla_{x_i} g(x) = -\nabla_{x_0} f_0(x_0)$, for all $i \in \{1, \dots, n\}$. Since the supervisor can measure x_0 , whenever there is an update request, it can evaluate ∇g directly and broadcast it to the generators. This corresponds to the feedback-based scenario (cf. Section III).

We test the algorithms resulting from the triggering criteria (10) and (15) on a single-phase equivalent of the IEEE 37-bus test feeder, reported in Figure 1(a). The network has five generators. The load buses are a mixture of constant-current, constant-impedance, and constant-power loads [27]. The initial active and reactive power demands are 2 MW and 1 MVAR, respectively. At $t = 40$ seconds, the active power demand increases by 1 MW. The algorithms are simulated using the nonlinear exact AC power flow solver MATPOWER [28].

We provide two sets of simulations based on whether the generation capacities of the generators are constrained or unconstrained. For the constrained case, for each $i \in \{1, \dots, 5\}$, $\mathcal{X}_i = \{x_i \mid 0 \leq x_i \leq \bar{x}_i\}$, where $\{\bar{x}_i\}_{i=1}^5$ are taken as 0.7 MW, 1 MW, 0.8 MW, 0.5 MW and 0.3 MW, respectively. Figure 1(b)-(c) show the evolution of the injected active powers of the agents using the proposed event-triggered dynamics and the continuous dynamics for the unconstrained and constrained case, respectively. Note that depending on the operating region, different agents request for the information update non-uniformly. For a fixed load, we simulate 10 different initial conditions for 60 seconds (sec) and observe an average MIET of 3.3×10^{-1} sec for the unconstrained and 2.0×10^{-1} sec for the constrained case, with respective standard deviations of 6.7×10^{-3} sec and 23.1×10^{-3} sec, and 177 (unconstrained) and 190 (constrained) average updates per resource-aware execution.

VII. CONCLUSIONS AND FUTURE WORK

We have designed decentralized event-triggered coordination mechanisms to solve network optimization problems whose objective function is a combination of a separable component among the agents and a non-separable coupling term. The proposed coordination mechanisms prescribe opportunistic requests for information from the agents to the

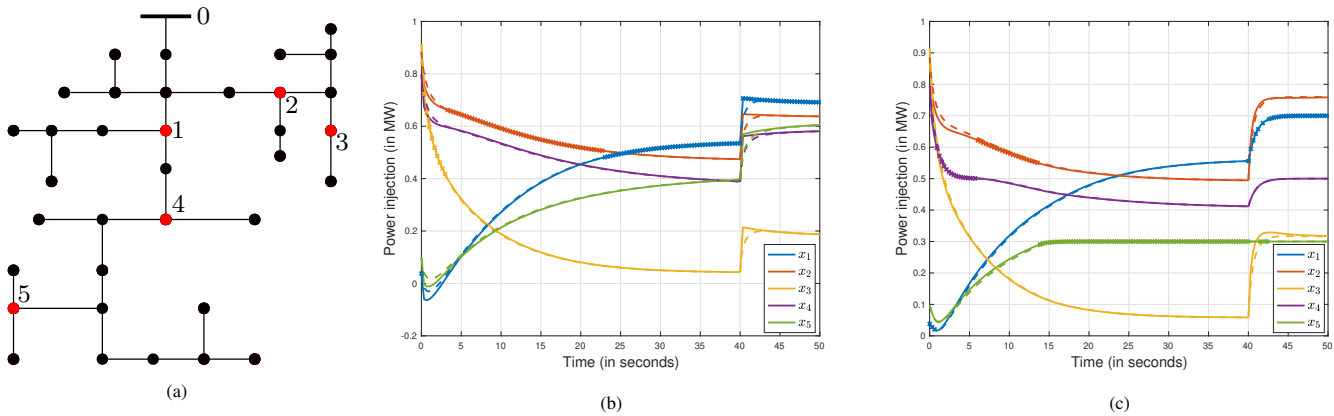


Fig. 1: Performance of proposed event-triggered coordination algorithms on the power distribution network. (a) shows the IEEE 37-bus test feeder, where node 0 represents the supervisor, red nodes (1-5) represent the generators, and black nodes represent the loads; edges represent the electrical connection between the nodes. The total load increases by 1MW after 40 seconds. The values of λ and γ are taken as 0.2 and 0.9, respectively and stepsize is taken as 10^{-2} . (b) and (c) compare the state evolution using the proposed event-triggered mechanisms (solid lines) using the triggering rules (10) and (15) with their respective continuous update cases. \times markers denote the triggering instances for the corresponding agent.

network supervisor, are anytime, and guarantee asymptotic convergence to the desired optimizer. Future work will focus on synthesizing novel adaptive triggering criteria ensuring uniform participation of all the agents and minimizing the total number of updates, consider more general interaction topologies, optimization problems with locally-coupled constraints, and asynchronous updates in computation-based scenarios, where the network supervisor receives only fresh state information from the agent triggering the update and as a result evaluates the information about the coupling term with outdated knowledge of the overall network state.

REFERENCES

- [1] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [2] P. Wan and M. D. Lemmon, "Event-triggered distributed optimization in sensor networks," in *Symposium on Information Processing of Sensor Networks*, San Francisco, CA, 2009, pp. 49–60.
- [3] A. Nedić, "Distributed optimization," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. New York: Springer, 2015.
- [4] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
- [5] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. on Decision and Control*, Maui, HI, 2012, pp. 3270–3285.
- [6] L. Hetel, C. Fiter, H. Omran, A. Seuret, E. Fridman, J. P. Richard, and S. I. Niculescu, "Recent developments on the stability of systems with aperiodic sampling: An overview," *Automatica*, vol. 76, pp. 309–335, 2017.
- [7] H. Saboori, M. Mohammadi, and R. Taghe, "Virtual power plant (VPP), definition, concept, components and types," in *Asia-Pacific Power and Energy Engineering Conference*, Wuhan, China, 2011, pp. 1–4.
- [8] G. Cavararo, A. Bernstein, R. Carli, and S. Zampieri, "Distributed minimization of the power generation cost in prosumer-based distribution networks," in *American Control Conference*, Denver, CO, 2020, pp. 2370–2375.
- [9] S. Koehler and F. Borrelli, "Building temperature distributed control via explicit MPC and "Trim and Respond" methods," in *European Control Conference*, Zurich, Switzerland, 2013, pp. 4334–4339.
- [10] J. T. Wen and S. Mishra, *Intelligent Building Control Systems*, ser. Advances in Industrial Control. Springer, 2018.
- [11] M. T. Hale, A. Nedić, and M. Egerstedt, "Asynchronous multiagent primal-dual optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4421–4435, 2017.
- [12] M. Mazo Jr. and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456–2461, 2011.
- [13] H. Zhang, P. Soldati, and M. Johansson, "Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Seoul, South Korea, 2009, pp. 1–8.
- [14] S. S. Kia, J. Cortés, and S. Martinez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.
- [15] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered control and communication of networked systems for multi-agent consensus," *Automatica*, vol. 105, pp. 1–27, 2019.
- [16] S. L. Bowman, C. Nowzari, and G. J. Pappas, "Coordination of multi-agent systems via asynchronous cloud communication," in *IEEE Conf. on Decision and Control*, Las Vegas, NV, 2016, pp. 2215–2220.
- [17] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, "Control of multi-agent systems with event-triggered cloud access," in *European Control Conference*, Linz, Austria, 2015, pp. 954–961.
- [18] P. Ong and J. Cortés, "Opportunistic robot control for interactive multiobjective optimization under human performance limitations," *Automatica*, vol. 123, p. 109263, 2021.
- [19] T. L. Friesz, D. H. Bernstein, N. J. Mehta, R. L. Tobin, and S. Ganjizadeh, "Day-to-day dynamic network disequilibria and idealized traveler information systems," *Operations Research*, vol. 42, no. 6, pp. 1120–1136, 1994.
- [20] Y. S. Xia and J. Wang, "On the stability of globally projected dynamical systems," *Journal of Optimization Theory and Applications*, vol. 106, no. 1, pp. 129–150, 2000.
- [21] X.-B. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 426–431, 2003.
- [22] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. Springer, 1994.
- [23] M. Colombino, E. Dall'Anese, and A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 422–432, 2020.
- [24] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
- [25] F. H. Clarke, *Optimization and Nonsmooth Analysis*, ser. Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.
- [26] J. Cortés, "Discontinuous dynamical systems – a tutorial on solutions, nonsmooth analysis, and stability," *IEEE Control Systems*, vol. 28, no. 3, pp. 36–73, 2008.
- [27] W. H. Kersting, "Radial distribution test feeders," in *IEEE Power Engineering Society Winter Meeting*, vol. 2, Columbus, OH, Jan. 2001, pp. 908–912.
- [28] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.