2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

Distributed Optimal Steady-state Control Using Reverse- and Forward-engineering

Xuan Zhang, Antonis Papachristodoulou and Na Li

Abstract— **In this paper, we consider the problem of distributed control for linear network systems to achieve optimal steady-state performance. Motivated by recent research on re-engineering cyber-physical systems, we propose a reverse- and forward-engineering framework which consists of two steps. Firstly, we reverse-engineer a dynamic system as a gradient algorithm to solve an optimization problem. Secondly, we use a forward-engineering approach to systematically design distributed control or modify the existing control. As a result, the system can automatically track the optimal solution of a predefined optimization problem and the control scheme can be implemented in a distributed and closed-loop manner. In order to investigate how general this framework is, we establish necessary and sufficient conditions under which a linear dynamic system can be reverse-engineered as a gradient algorithm to solve an optimization problem. Those conditions are characterized using properties of system matrices and relevant linear matrix inequalities. A practical example regarding frequency control in power systems demonstrates the effectiveness of the proposed framework.**

I. INTRODUCTION

In recent years, smart communication, computing, sensing, and actuation technologies have been stimulating the emergence of multi-agent network systems, including the smart grid [1], smart home [2], mobile robots [3], and intelligent transportation systems [4]. The efficient and robust operation of the network systems plays a key role in economic development and environmental sustainability of modern society.

Because those network systems are usually operated under an uncertain environment and with incomplete information, to ensure their economic efficiency and stability, the conventional operation is usually divided into two different time-scales. At a slow time-scale, efficient nominal operating points are determined using optimization methods with predictions of future uncertainties, e.g., disturbances. At a fast time-scale or in real time, the efficient points are tracked and stability of the system is ensured using control techniques. However, as the uncertainties fluctuate faster and by a larger amount, this time-scale separation framework could induce economic inefficiency, poor robustness, and even instability. For example, in power grids, increasing penetration of distributed energy sources introduce fast and large fluctuations in both supply and demand. As a result, conventional frequency control schemes, e.g., automatic generation control, become much less economically efficient [5].

The objective of this paper is to modify/redesign the existing system dynamics and built-in control mechanisms so that the fast time-scale controller can track the efficient operating points automatically. We call this control as *optimal steady-state control*. The idea of using a dynamic system to track an implicitly defined optimal point originated in [6]. Also, this problem was recently studied in [7]–[9], in which the controllers were designed based on a dynamic extension of the Karush-Kuhn-Tucker (KKT) condition, a method of saddle point flows with backstepping, and a dual ϵ-subgradient method respectively.

In this paper, we propose a reverse- and forward-engineering approach to address the problem of optimal steady-state control in network systems. Firstly, we reverse-engineer a dynamic system as a gradient algorithm to solve an optimization problem. Secondly, we use a forward-engineering approach to systematically design control or modify the existing control mechanisms. As a result, the redesigned system can automatically track the optimal solution of a predefined optimization problem. Moreover, under this approach, the resulting controller (i) has a distributed/decentralized structure and can be implemented in a closed-loop manner (i.e., no information of external disturbances is needed); (ii) respects the system operating constraints; (iii) ensures an efficient and reliable network operation. Recent research in frequency control of power grids and Internet congestion control has successfully demonstrated that these systems can be re-engineered using a reverse- and forward-engineering framework [5], [10]–[13].

However, there is not much work investigating the general network systems – how general the reverse- and forward-engineering approach is. Specifically, (i) what kind of systems can be reverse-engineered; (ii) if a system can be reverse-engineered, how to use forward-engineering to do control (re)design. This paper will answer both of these questions, concentrating on Linear Time-Invariant (LTI) systems. In Section II, we provide a formal problem setup on optimal steady-state control and the concept of reverse- and forward-engineering using a saddle point method. In Section III, we present a forward-engineering procedure to modify control schemes for a special class of network systems to achieve optimal steady-state performance. In Section IV, we develop necessary and sufficient conditions under which a system can be reverse-engineered as gradient algorithms to solve optimization problems. Section V presents a practical ex-

The work of X. Zhang, A. Papachristodoulou was supported in part by the Engineering and Physical Sciences Research Council projects EP/J012041/1, EP/I031944/1, EP/J010537/1, EP/M002454/1. The work of N. Li was supported in part by Lincoln Laboratory Projec 06828209-01.

X. Zhang and A. Papachristodoulou are with the Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK (email: {xuan.zhang, antonis}@eng.ox.ac.uk). N. Li is with the School of Engineering and Applied Science, Harvard University, 29 Oxford Street, Cambridge, MA 02138, USA (email: nali@seas.harvard.edu).

ample regarding frequency control in power networks and Section VI concludes the paper.

Notations: $x \in \mathbb{R}^m$ is a column vector in an m-dimensional Euclidean space $\mathbb{R}^m$. $X \in \mathbb{R}^{m\times n}$ is an $m \times n$ real matrix. diag$\{\star\}$ is a diagonal matrix with corresponding entries $\star$ on the main diagonal. x^T (x^H) is the transpose (conjugate transpose) of x. I_m is an identity matrix of size $m \times m$. $\mathbf{0}$ is a matrix of zeros with an appropriate dimension determined by the context. eig(X) is the spectrum of a square matrix X. $X \succeq 0$ ($X \succ 0$) denotes that a square matrix X is positive semi-definite (positive definite). $\nabla_x f(x)$ or $\frac{\partial f}{\partial x}$ is the gradient (as a column vector) of a scalar function $f \in \mathcal{C}^1$ with respect to x, where $\mathcal{C}^n$ is the class of functions that are n times continuously differentiable. $\nabla_x^2 f(x)$ is the Hessian matrix of a scalar function $f \in \mathcal{C}^2$ with respect to x. The positive projection of a function $h(y)$ on a variable x, $(h(y))_x^+$ is:

$$(h(y))_x^+ = \begin{cases} h(y) & \text{if } x > 0 \\ \max(0, h(y)) & \text{if } x = 0 \end{cases}.$$

II. Problem Setup and Preliminaries

A. Problem Setup

Consider a network system consisting of N subsystems

$$\dot{x}_i = \sum_{j\in\mathcal{N}(i)} A_{ij}x_j + B_i u_i + C_i w_i,\ i = 1, \cdots, N \tag{1}$$

and each subsystem is equipped with a built-in controller

$$\dot{u}_i = \sum_{j\in\mathcal{N}(i)} D_{ij}x_j + \sum_{j\in\mathcal{N}(i)} E_{ij}u_j + F_i w_i, i = 1, \cdots, N \tag{2}$$

where $x_i(t) \in \mathbb{R}^{n_i}$ is the state vector of subsystem i, $\mathcal{N}(i)$ is the set of neighbouring subsystems of subsystem i, $A_{ij} \in \mathbb{R}^{n_i\times n_j}$, $B_i \in \mathbb{R}^{n_i\times m_i}$, $u_i(t) \in \mathbb{R}^{m_i}$ is the control input vector to subsystem i, $C_i \in \mathbb{R}^{n_i\times p_i}$, $w_i(t) \in \mathbb{R}^{p_i}$ is the exogenous input vector to subsystem i, e.g., disturbance injection, $D_{ij} \in \mathbb{R}^{m_i\times n_j}$, $E_{ij} \in \mathbb{R}^{m_i\times m_j}$ and $F_i \in \mathbb{R}^{m_i\times p_i}$. Here the dynamic feedback controller (2) is assumed to have a distributed structure. For convenience, let $x = \{x_1, \cdots, x_N\}$, $u = \{u_1, \cdots, u_N\}$ and $w = \{w_1, \cdots, w_N\}$, $n = \sum_{i=1}^N n_i$, $m = \sum_{i=1}^N m_i$ and $p = \sum_{i=1}^N p_i$.

Remark 1. *System (1)-(2) describes a general class of network systems with built-in control mechanisms. Equation (2) represents any distributed dynamic feedback controller with order one or more. Moreover,* $D_{ij} = \mathbf{0}, E_{ij} = \mathbf{0}, F_i = \mathbf{0}, \forall j \in \mathcal{N}(i)$ *means that there is no such controller equipped with subsystem* i*. Note that a zero-order distributed state feedback controller, i.e.,* $u_i = \sum_{j\in\mathcal{N}(i)} K_{ij}x_j$ *where* $K_{ij} \in \mathbb{R}^{m_i\times n_j}$*, can be included in the system dynamics (1) through* A_{ij}*. Lastly,* $F_i = \mathbf{0}$ *indicates that the controller of subsystem* i *does not use any information of disturbance* w_i*.*

Suppose that the exogenous disturbance w is constant, and that the closed-loop system (1)-(2) is stabilized and each trajectory converges to one point in the equilibrium set $\mathcal{X} := \{x | \sum_{j\in\mathcal{N}(i)} A_{ij}x_j + B_i u_i + C_i w_i = \mathbf{0}, \sum_{j\in\mathcal{N}(i)} D_{ij}x_j + \sum_{j\in\mathcal{N}(i)} E_{ij}u_j + F_i w_i = \mathbf{0}, i = 1, \cdots, N\}$. Now consider the following optimization problem associated with the system under a constant disturbance w:

$$\min_{x\in\mathbb{R}^n, u\in\mathbb{R}^m} \sum_{i=1}^N f_i(x_i) + \sum_{i=1}^N g_i(u_i) \tag{3a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}(i)} A_{ij}x_j + B_i u_i + C_i w_i = \mathbf{0} \tag{3b}$$

$$\sum_{j\in\mathcal{N}(i)} D_{ij}x_j + \sum_{j\in\mathcal{N}(i)} E_{ij}u_j + F_i w_i = \mathbf{0} \tag{3c}$$

$$h_i(x_i, u_i) \le 0 \tag{3d}$$

where $i = 1, \cdots, N$, $f_i \in \mathcal{C}^2$: $\mathbb{R}^{n_i} \to \mathbb{R}$, $g_i \in \mathcal{C}^2$: $\mathbb{R}^{m_i} \to \mathbb{R}$, $h_i \in \mathcal{C}^2$: $\mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}$. We assume that (3) is convex (i.e., $\nabla_{x_i}^2 f_i \succeq 0$, $\nabla_{u_i}^2 g_i \succeq 0$ and $\nabla_{x_i,u_i}^2 h_i \succeq 0$ hold), feasible and satisfies Slater's constraint qualification [14]. This optimization problem usually defines a desired operating point for the network system. For instance, in power systems, problem (3) may be an Optimal Power Flow (OPF) problem in which generation cost is minimized and consumption utility is maximized subject to power flow balance constraints and network operating constraints [15]. Another example is the space satellite formation control in which problem (3) is usually formulated as a consensus problem with respect to the positions and velocities of space satellites [16].

Remark 2. *In (3), there can be more inequality constraints for each subsystem, i.e., Equation (3d) can be reformulated in a vector form. For convenience, in this paper, we consider the case in which there is only one inequality constraint for each subsystem, while the result can be immediately extended.*

The problem we study in this paper is to modify the existing controller (2) in a distributed and closed-loop manner so that the system can track the optimal solution of (3) automatically. In the existing literature and practice, e.g., frequency control in power systems, controller (2) is designed to drive the system to a predefined nominal operating point which is derived by solving (3) using a prediction on the future w. Moreover, the optimization problem is usually solved (i) at a much slower time-scale compared with system dynamics (1)-(2) and (ii) by using centralized optimization or distributed optimization methods, both of which require a certain amount of communication and computation. If controller (2) can be modified to track the optimal solution of (3) automatically and be implemented in a distributed and closed-loop manner, then the system itself can adapt to changes of uncertain disturbance w. Furthermore, measurement, communication and computation can be saved.

B. Preliminaries

In this paper, we focus on a special class of systems (1)-(2), which can be interpreted as a primal-dual gradient algorithm [11], [17] to solve an unconstrained quadratic saddle point problem. Given a function $f \in \mathcal{C}^2$: $\mathbb{R}^a \times \mathbb{R}^b \to \mathbb{R}$, $(\tilde{y}, \tilde{z})$ is a saddle point of f if $f(\tilde{y}, z) \le f(\tilde{y}, \tilde{z}) \le f(y, \tilde{z})$ for all y, z. Assume that for all y, z, $\nabla_y^2 f \succeq 0, \nabla_z^2 f \preceq 0$, and the set $\{(y, z) | \nabla_{y,z} f = \mathbf{0}\}$ is nonempty. Then a primal-dual gradient algorithm to solve $\max_z \min_y f(y, z)$ is:

$$\dot{y} = -K_y \frac{\partial f}{\partial y}, \quad \dot{z} = K_z \frac{\partial f}{\partial z} \tag{4}$$

where $K_y \in \mathbb{R}^{a\times a}, K_z \in \mathbb{R}^{b\times b}$ are positive definite constant matrices. We have the following lemma regarding the convergence of (4) which can be inferred from [11], [17].

Lemma 1. *Let $f \in \mathcal{C}^2$: $\mathbb{R}^a \times \mathbb{R}^b \to \mathbb{R}$ satisfy: for all y, z, $\nabla_y^2 f \succeq 0, \nabla_z^2 f \preceq 0$, and the set $\{(y,z)|\nabla_{y,z} f = \mathbf{0}\}$ is nonempty. Then the trajectories of the primal-dual gradient dynamics/saddle point dynamics (4) are bounded. Furthermore, if f is either strictly convex in y or strictly concave in z, each trajectory of (4) asymptotically converges to a saddle point of f.*

Formally speaking, we focus on a network system (1)-(2) which belongs to following class:

Class-$\mathcal{S}'$: System (1)-(2) belongs to Class-$\mathcal{S}'$ means that there exists a function $L_{\text{sys}}(x^{(1)}, x^{(2)}, u) : \mathbb{R}^{m+n} \to \mathbb{R}$ and positive definite matrices $P_{x^{(1)}}, P_{x^{(2)}}$ and $P_{u_i}, i = 1, \cdots, N$ such that $\nabla^2_{x^{(1)}} L_{\text{sys}} \preceq 0$, $\nabla^2_{x^{(2)},u} L_{\text{sys}} \succeq 0$, the saddle point set $\{(x^{(1)}, x^{(2)}, u)|\nabla_{x^{(1)},x^{(2)},u} L_{\text{sys}} = \mathbf{0}\}$ is nonempty, and (1)-(2) is a primal-dual gradient algorithm to solve $\max_{x^{(1)}} \min_{x^{(2)},u} L_{\text{sys}}$, i.e.,

$$\dot{x} = \text{diag}\left\{P_{x^{(1)}}, -P_{x^{(2)}}\right\} \frac{\partial L_{\text{sys}}}{\partial x}$$
$$\dot{u}_i = -P_{u_i} \frac{\partial L_{\text{sys}}}{\partial u_i},\ i = 1, \cdots, N.$$

Remark 3. *In the above definition, (i) state x is partitioned into $x^{(1)}$ and $x^{(2)}$; (ii) u is required as a minimizer; (iii) the gain matrix for $\dot{u}$ is block diagonal consisting of $P_{u_i}, i = 1, \cdots, N$, i.e, $\dot{u}$ has a distributed structure. The definition can be extended to the case where u is required as a maximizer (by adding a minus sign before L_{sys}). For ease of exposition, we only focus on the class in the above definition.*

The motivation to consider this special class includes: (i) recent research has demonstrated that there are many cyber-physical systems, such as power systems and Internet congestion control protocols, belonging to this class [12], [13]; (ii) if a system belongs to Class-$\mathcal{S}'$, then we have a 'forward-engineering' procedure to modify the controller (2) for achieving the optimal steady-state performance described in (3), which is demonstrated in Section III.

Although some recent work has shown that power system frequency control and Internet congestion control belong to Class-$\mathcal{S}'$, fundamental questions remain. For example, how general this class is? Do we have a rigorous and general characterization of this class in terms of the property of system matrices? These questions lead to Section IV in which we study reverse-engineering for general LTI systems.

III. Forward-engineering for Control Modification

In this section we will provide a procedure to modify the controller of system (1)-(2) so that the closed-loop system can track the optimal solution of (3) automatically, under the premise that system (1)-(2) belongs to Class-$\mathcal{S}'$.

Step 1): Introduce auxiliary decision variables
Modify problem (3) by replacing x (x_i) with an auxiliary decision vector $y \in \mathbb{R}^n$ ($y_i \in \mathbb{R}^{n_i}$):

$$\min_{y\in\mathbb{R}^n, u\in\mathbb{R}^m} \sum_{i=1}^N f_i(y_i) + \sum_{i=1}^N g_i(u_i) \tag{5a}$$
$$\text{subject to} \quad \sum_{j\in\mathcal{N}(i)} A_{ij} y_j + B_i u_i + C_i w_i = \mathbf{0} \tag{5b}$$
$$\sum_{j\in\mathcal{N}(i)} D_{ij} y_j + \sum_{j\in\mathcal{N}(i)} E_{ij} u_j + F_i w_i = \mathbf{0} \tag{5c}$$
$$h_i(y_i, u_i) \le 0, \text{ where } i = 1, \cdots, N. \tag{5d}$$

Step 2): Merge objective functions
Since problem (5) is convex and strong duality holds, derive a saddle point problem corresponding to (5), given by

$$\begin{aligned}\max_{\zeta_i\in\mathbb{R}^{n_i}, \lambda_i\in\mathbb{R}^{m_i}, \mu_i\ge 0}\ \min_{y\in\mathbb{R}^n, u\in\mathbb{R}^m} L_{\text{op}} = \sum_{i=1}^N f_i(y_i) + \sum_{i=1}^N g_i(u_i) \\ - \sum_{i=1}^N \zeta_i^T \left(\sum_{j\in\mathcal{N}(i)} A_{ij} y_j + B_i u_i + C_i w_i \right) + \sum_{i=1}^N \mu_i h_i(y_i, u_i) \\ - \sum_{i=1}^N \lambda_i^T \left(\sum_{j\in\mathcal{N}(i)} D_{ij} y_j + \sum_{j\in\mathcal{N}(i)} E_{ij} u_j + F_i w_i \right)\end{aligned} \tag{6}$$

where $\zeta_i \in \mathbb{R}^{n_i}, \lambda_i \in \mathbb{R}^{m_i}, \mu_i \ge 0$ are Lagrangian multipliers (dual variables) for the constraints in (5). By adding L_{sys}, we obtain an augmented saddle point problem:

$$\max_{\zeta_i\in\mathbb{R}^{n_i}, \lambda_i\in\mathbb{R}^{m_i}, \mu_i\ge 0, x^{(1)}\in\mathbb{R}^{n^{(1)}}}\ \min_{y\in\mathbb{R}^n, u\in\mathbb{R}^m, x^{(2)}\in\mathbb{R}^{n^{(2)}}} L_{\text{au}} = L_{\text{sys}} + \gamma L_{\text{op}} \tag{7}$$

where $\gamma > 0$. The next lemma shows the properties of L_{au}.

Lemma 2. *$\nabla^2_{y,u,x^{(2)}} L_{au} \succeq 0$ and $\nabla^2_{\zeta_i,\lambda_i,\mu_i,x^{(1)}} L_{au} \preceq 0$ hold. Moreover, if A is invertible in (1) (here we rewrite (1) as $\dot{x} = Ax + Bu + Cw$, where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{n\times p}$), then $(y, u, x, \zeta_i, \lambda_i, \mu_i)$ is a saddle point of L_{au} if and only if $(y, u, \zeta_i, \lambda_i, \mu_i)$ is a saddle point of L_{op} and (x, u) is a saddle point of L_{sys}.*

Proof. See the Appendix. □

Step 3): Forward-engineering
With the new saddle point function L_{au}, derive the following saddle point dynamics:

$$\dot{x}_i = \sum_{j\in\mathcal{N}(i)} A_{ij} x_j + B_i u_i + C_i w_i \tag{8a}$$
$$\begin{aligned}\dot{u}_i = \sum_{j\in\mathcal{N}(i)} D_{ij} x_j + \sum_{j\in\mathcal{N}(i)} E_{ij} u_j + F_i w_i - \gamma P_{u_i} \Bigg(\frac{\partial g_i}{\partial u_i} \\ - B_i^T \zeta_i - \sum_{j\in\mathcal{N}(i)} E_{ji}^T \lambda_j + \mu_i \frac{\partial h_i}{\partial u_i} + K_{eu_i}(u_i - \hat{u}_i) \Bigg)\end{aligned} \tag{8b}$$
$$\dot{\hat{u}}_i = \hat{K}_{eu_i}(u_i - \hat{u}_i) \tag{8c}$$

$$\dot{y}_i = -K_{y_i}\left(\frac{\partial f_i}{\partial y_i} - \sum_{j\in\mathcal{N}(i)} A_{ji}^T\zeta_j - \sum_{j\in\mathcal{N}(i)} D_{ji}^T\lambda_j + \mu_i\frac{\partial h_i}{\partial y_i} + K_{ey_i}(y_i - \hat{y}_i)\right) \tag{8d}$$

$$\dot{\hat{y}}_i = \hat{K}_{ey_i}(y_i - \hat{y}_i) \tag{8e}$$

$$\dot{\zeta}_i = -K_{\zeta_i}\left(\sum_{j\in\mathcal{N}(i)} A_{ij}y_j + B_iu_i + C_iw_i + K_{e\zeta_i}(\zeta_i - \hat{\zeta}_i)\right) \tag{8f}$$

$$\dot{\hat{\zeta}}_i = \hat{K}_{e\zeta_i}(\zeta_i - \hat{\zeta}_i) \tag{8g}$$

$$\dot{\lambda}_i = -K_{\lambda_i}\left(\sum_{j\in\mathcal{N}(i)} D_{ij}y_j + \sum_{j\in\mathcal{N}(i)} E_{ij}u_j + F_iw_i + K_{e\lambda_i}(\lambda_i - \hat{\lambda}_i)\right) \tag{8h}$$

$$\dot{\hat{\lambda}}_i = \hat{K}_{e\lambda_i}(\lambda_i - \hat{\lambda}_i) \tag{8i}$$

$$\dot{\mu}_i = k_{\mu_i}(h_i(y_i, u_i))^+_{\mu_i} \tag{8j}$$

where $K_{eu_i}, \hat{K}_{eu_i}, K_{\lambda_i}, K_{e\lambda_i}, \hat{K}_{e\lambda_i} \in \mathbb{R}^{m_i\times m_i}$, $K_{y_i}, K_{ey_i}, \hat{K}_{ey_i}, K_{\zeta_i}, K_{e\zeta_i}, \hat{K}_{e\zeta_i} \in \mathbb{R}^{n_i\times n_i}$ are positive definite constant diagonal matrices, $k_{\mu_i} > 0$, and $i = 1, \cdots, N$.

The above saddle point algorithm is not exactly the primal-dual gradient algorithm (4). It is a modified saddle point algorithm given by the following lemma.

Lemma 3. *Let $f \in \mathcal{C}^2$: $\mathbb{R}^a \times \mathbb{R}^b \to \mathbb{R}$ satisfy: for all y, z, $\nabla_y^2 f \succeq 0, \nabla_z^2 f \preceq 0$, and the set $\{(y,z)|\nabla_{y,z}f = \mathbf{0}\}$ is nonempty. Then each trajectory of the modified saddle point dynamics given by*

$$\dot{y} = -K_y\left(\frac{\partial f}{\partial y} + K_{ey}(y - \hat{y})\right) \tag{9a}$$

$$\dot{\hat{y}} = \hat{K}_{ey}(y - \hat{y}) \tag{9b}$$

$$\dot{z} = K_z\left(\frac{\partial f}{\partial z} - K_{ez}(z - \hat{z})\right) \tag{9c}$$

$$\dot{\hat{z}} = \hat{K}_{ez}(z - \hat{z}) \tag{9d}$$

asymptotically converges to an equilibrium point at which (y, z) is a saddle point of f. Here $\hat{y}(t) \in \mathbb{R}^a, \hat{z}(t) \in \mathbb{R}^b$ are auxiliary state vectors, and $K_{ey}, \hat{K}_{ey} \in \mathbb{R}^{a\times a}, K_{ez}, \hat{K}_{ez} \in \mathbb{R}^{b\times b}$ are positive definite constant diagonal matrices.

Proof. See the Appendix. □

Remark 4. *Note that system (8) is slightly different from the formulation (9), in which only $u_i, y_i, \zeta_i, \lambda_i$ are equipped with auxiliary decision variables. Also, the positive projection in (8j) does not affect the convergence property of the dynamics [17]. Similar examples are shown in [18], [19].*

Step 4): Closed-loop implementation

Let $\tilde{\zeta}_i = x_i + K_{\zeta_i}^{-1}\zeta_i$ and $\tilde{\lambda}_i = u_i + K_{\lambda_i}^{-1}\lambda_i$. Rewrite Equations (8f)-(8i) as (also substitute $\zeta_i = K_{\zeta_i}(\tilde{\zeta}_i - x_i), \lambda_i = K_{\lambda_i}(\tilde{\lambda}_i - u_i)$ to Equations (8b) and (8d))

$$\dot{\tilde{\zeta}}_i = \sum_{j\in\mathcal{N}(i)} A_{ij}(x_j - y_j) - K_{e\zeta_i}(K_{\zeta_i}(\tilde{\zeta}_i - x_i) - \hat{\zeta}_i) \tag{10a}$$

$$\dot{\hat{\zeta}}_i = \hat{K}_{e\zeta_i}(K_{\zeta_i}(\tilde{\zeta}_i - x_i) - \hat{\zeta}_i) \tag{10b}$$

$$\dot{\tilde{\lambda}}_i = \sum_{j\in\mathcal{N}(i)} D_{ij}(x_j - y_j) - \gamma P_{u_i}\left(\frac{\partial g_i}{\partial u_i} - B_i^T K_{\zeta_i}(\tilde{\zeta}_i - x_i) - \sum_{j\in\mathcal{N}(i)} E_{ji}^T K_{\lambda_j}(\tilde{\lambda}_j - u_j) + \mu_i\frac{\partial h_i}{\partial u_i} + K_{eu_i} \times (u_i - \hat{u}_i)\right) - K_{e\lambda_i}(K_{\lambda_i}(\tilde{\lambda}_i - u_i) - \hat{\lambda}_i) \tag{10c}$$

$$\dot{\hat{\lambda}}_i = \hat{K}_{e\lambda_i}(K_{\lambda_i}(\tilde{\lambda}_i - u_i) - \hat{\lambda}_i) \tag{10d}$$

so that the extra states $\tilde{\zeta}_i, \tilde{\lambda}_i$ are independent of w.

For the optimality and stability of system (8), we have the following theorem.

Theorem 1. *If (1)-(2) belongs to Class-$\mathcal{S}'$ and A is Hurwitz*[1]*, each trajectory of (8) asymptotically converges to an equilibrium point at which (x, u) is an optimal solution of (3).*

Proof. See the Appendix. □

Going from the original closed-loop system (1)-(2) to the modified one (8), we have introduced extra dynamics while the structure of the original dynamic feedback controller is preserved as shown in Equation (8b). The benefit of this forward-engineering modification is summarized as follows. First, the modification allows us to embed different types of steady-state convex optimization problems. Second, as long as the steady-state optimization problem is distributed, e.g., with separable objective functions and local constraints as shown in problem (3), the resulting extra dynamics are completely distributed as given in system (8). Third, the modification ensures that the closed-loop system can achieve optimal steady-state performance without any information on the constant disturbance w, i.e., the system itself can adapt to changes in the objective optimization problem (3). The case when w is dynamically changing is left for future research.

IV. LTI Systems as Gradient Algorithms for Quadratic Optimization

In the previous section, a forward-engineering framework is presented to modify an existing control scheme for solving optimal steady-state control problems. Class-$\mathcal{S}'$ is considered as a prerequisite, which implies that the original closed-loop system (1)-(2) can be reverse-engineered as a primal-dual gradient algorithm to solve an unconstrained quadratic saddle point problem. In this section, we investigate Class-$\mathcal{S}'$ by studying what kind of systems belongs to it. Generally speaking, a system should be at least marginally stable [20] if it can be interpreted as a primal-dual gradient algorithm for an unconstrained quadratic saddle point problem, since system trajectories are bounded in this case (recall Lemma 1). Also, the equilibrium set of the system should be equivalent to the saddle point set of the resulting problem. To limit the scope of discussion, we only focus on reverse-engineering continuous-time LTI systems that are in accord with (1)-(2).

[1] Recall Lemma 2 for the definition of A. This assumption is also used in [7], Chapter 5.2. On the other hand, if A is not Hurwitz, a static feedback controller $u = Kx, K \in \mathbb{R}^{n\times n}$ can be designed to pre-stabilize (1) first.

A. LTI Autonomous Systems

We begin our investigation by studying a simple class of systems, i.e., LTI autonomous systems. This allows us to understand the insight behind the mathematical derivation better. Consider an LTI autonomous system:

$$\dot{x} = Ax \tag{11}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and $A \in \mathbb{R}^{n\times n}$. First, let us study the following class of autonomous systems which can be reverse-engineered as gradient algorithms for solving an unconstrained convex quadratic programming problem.

Class-$\mathcal{O}$: System (11) belongs to Class-$\mathcal{O}$ means that there exists a function $L(x) : \mathbb{R}^n \to \mathbb{R}$ and a positive definite matrix $P \in \mathbb{R}^{n\times n}$ such that $\nabla_x^2 L \preceq 0$, $\{x|\nabla_x L = \mathbf{0}\}$ is nonempty, and (11) is a gradient algorithm to solve $\max_x L$, i.e., $\dot{x} = P\frac{\partial L}{\partial x}$.

Since (11) is linear, if (11) belongs to Class-$\mathcal{O}$, then the associated L must be a convex quadratic function, i.e., $L = \frac{1}{2}x^T Qx$ for some $Q \preceq 0$. Therefore, system (11) belongs to Class-$\mathcal{O}$ if and only if there exist matrices $P \succeq 0$ and $Q \preceq 0$ such that $A = PQ$ holds. This leads to our first result regarding reverse-engineering.

Theorem 2. *System (11) belongs to Class-$\mathcal{O}$ if and only if (11) is marginally or asymptotically stable, eig$(A) \in \mathbb{R}$ and A is diagonalizable*[2].

Proof. Necessity. Suppose that system (11) is a gradient algorithm to solve an unconstrained convex quadratic programming problem given by

$$\max_{x\in\mathbb{R}^n} L = \frac{1}{2}x^T Qx \tag{12}$$

where $Q \in \mathbb{R}^{n\times n}$ satisfies $Q = Q^T \preceq 0$. Then the trajectories of (11) are bounded under Lemma 1, and there exists a matrix $P = P^T \in \mathbb{R}^{n\times n}$ and $P \succ 0$, so that $\dot{x} = P\frac{\partial L}{\partial x} = PQx = Ax$ is always true. i.e., $A = PQ$ holds. This leads to $P^{-1}A = A^T P^{-1}$ which is equivalent to eig$(A) \in \mathbb{R}$ and A is diagonalizable, based on Lemma 4 in the Appendix. Also, system (11) is marginally or asymptotically stable.

Sufficiency. Because system (11) is marginally or asymptotically stable, eig$(A) \in \mathbb{R}$ and A is diagonalizable, A can be written as a diagonal canonical form $A = J\Lambda J^{-1}$ where $J \in \mathbb{R}^{n\times n}$ and $\Lambda \preceq 0$. Under Lemma 4, there exists a positive definite matrix $V \in \mathbb{R}^{n\times n}$ so that $V\Lambda = \Lambda V$ holds. Based on Lemma 1 in [21], we have $V\Lambda \preceq 0$. Define a matrix $P = JV^{-1}J^T$. Then $P^{-1}A = A^T P^{-1} \preceq 0$ holds. Considering the unconstrained convex quadratic programming problem (12) where $Q = P^{-1}A$, we conclude that system (11) belongs to Class-$\mathcal{O}$. □

Theorem 2 proposes a necessary and sufficient condition to reverse-engineer system (11) as a gradient algorithm to solve a related unconstrained convex quadratic programming problem. We now study the case where we can interpret system (11) as a primal-dual gradient algorithm to solve an unconstrained quadratic saddle point problem. Consider the following class of systems.

Class-$\mathcal{S}$: System (11) belongs to Class-$\mathcal{S}$ means that there exists a function $L(x^{(1)}, x^{(2)}) : \mathbb{R}^n \to \mathbb{R}$ and positive definite matrices $P_{x^{(1)}}, P_{x^{(2)}}$ such that $\nabla^2_{x^{(1)}} L \preceq 0$, $\nabla^2_{x^{(2)}} L \succeq 0$, the saddle point set $\{x|\nabla_x L = \mathbf{0}\}$ is nonempty, and (11) is a primal-dual gradient algorithm to solve $\max_{x^{(1)}} \min_{x^{(2)}} L$, i.e., $\dot{x} = \text{diag}\{P_{x^{(1)}}, -P_{x^{(2)}}\}\frac{\partial L}{\partial x}$.

In the above definition, state x is partitioned into $x^{(1)}$ and $x^{(2)}$. Similarly, (11) is rearranged in the following form[3]

$$\underbrace{\begin{bmatrix} \dot{x}^{(1)} \\ \dot{x}^{(2)} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A} x \tag{13}$$

where $x^{(1)}(t) \in \mathbb{R}^{n_1}$, $x^{(2)}(t) \in \mathbb{R}^{n_2}$, and $n_1 + n_2 = n$.

Using similar arguments as before, we only need to focus on a function L in a quadratic form, i.e.,

$$L = \frac{1}{2}x^T \underbrace{\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}}_{Q} x \tag{14}$$

where $Q_{11} \in \mathbb{R}^{n_1\times n_1}$ satisfies $Q_{11} = Q_{11}^T \preceq 0$ (i.e., L is concave in $x^{(1)}$), $Q_{22} \in \mathbb{R}^{n_2\times n_2}$ satisfies $Q_{22} = Q_{22}^T \succeq 0$ (i.e., L is convex in $x^{(2)}$), and $Q_{12} \in \mathbb{R}^{n_1\times n_2}$. Based on the definition of Class-$\mathcal{S}$, we have the following theorem.

Theorem 3. *System (13) belongs to Class-$\mathcal{S}$ if and only if the following conditions are satisfied: (i) (13) is marginally or asymptotically stable; (ii) the eigenvalues of both A_{11} and A_{22} are non-positive real; (iii) both A_{11} and A_{22} are diagonalizable with the diagonal canonical forms given by $A_{11} = J_1\Lambda_1 J_1^{-1}, J_1 \in \mathbb{R}^{n_1\times n_1}, A_{22} = J_2\Lambda_2 J_2^{-1}, J_2 \in \mathbb{R}^{n_2\times n_2}$, and there exist V_1 and V_2 such that*

$$\begin{aligned} &(J_1^{-1})^T V_1 J_1^{-1} A_{12} + A_{21}^T (J_2^{-1})^T V_2 J_2^{-1} = \mathbf{0} \\ &V_1\Lambda_1 = \Lambda_1 V_1,\ V_2\Lambda_2 = \Lambda_2 V_2 \\ &V_1 \succ 0,\ V_2 \succ 0. \end{aligned} \tag{15}$$

Proof. Necessity. Suppose that system (13) is a primal-dual gradient algorithm to solve $\max_{x^{(1)}} \min_{x^{(2)}} L$ where L is given by (14). Then the trajectories of (13) are bounded under Lemma 1, and there exist matrices $P_{x^{(1)}} \in \mathbb{R}^{n_1\times n_1}$ and $P_{x^{(2)}} \in \mathbb{R}^{n_2\times n_2}$ satisfying $P_{x^{(1)}} = P_{x^{(1)}}^T \succ 0$ and $P_{x^{(2)}} = P_{x^{(2)}}^T \succ 0$, so that $Q = \text{diag}\{P_{x^{(1)}}, -P_{x^{(2)}}\}^{-1}A = A^T\text{diag}\{P_{x^{(1)}}, -P_{x^{(2)}}\}^{-1}$ holds. This equation leads to

$$P_{x^{(1)}}^{-1} A_{11} = A_{11}^T P_{x^{(1)}}^{-1} = Q_{11} \preceq 0 \tag{16a}$$

$$P_{x^{(2)}}^{-1} A_{22} = A_{22}^T P_{x^{(2)}}^{-1} = -Q_{22} \preceq 0 \tag{16b}$$

$$P_{x^{(1)}}^{-1} A_{12} + A_{21}^T P_{x^{(2)}}^{-1} = \mathbf{0}. \tag{16c}$$

Based on Lemma 4 in the Appendix, (16a)-(16b) are equivalent to eig$(A_{11}) \in \mathbb{R}$, A_{11} is diagonalizable, eig$(A_{22}) \in \mathbb{R}$, A_{22} is diagonalizable, and moreover, the eigenvalues of both

[2]We use eig$(A) \in \mathbb{R}$ to indicate that all eigenvalues of A are real numbers, although eig(A) is a set.

[3]In this section, we abuse notation e.g., $A_{ij}, n_i, \lambda_i, B_i, C_i$, whose meaning should be clear from the context.

A_{11} and A_{22} are non-positive (otherwise, for example, there would exist a positive eigenvalue of A_{11} and a corresponding non-zero eigenvector, denoted by λ_1 and x_{λ_1}, for which $x_{\lambda_1}^T P_{x^{(1)}}^{-1} A_{11} x_{\lambda_1} = \lambda_1 x_{\lambda_1}^T P_{x^{(1)}}^{-1} x_{\lambda_1} > 0$, i.e., a contradiction to (16a)). By defining $V_1 = J_1^T P_{x^{(1)}}^{-1} J_1$, $V_2 = J_2^T P_{x^{(2)}}^{-1} J_2$, condition (iii) holds, which completes the proof of necessity.

Sufficiency. Let conditions (i)-(iii) be true. Consider the following unconstrained quadratic saddle point problem:

$$\max_{x^{(1)} \in \mathbb{R}^{n_1}} \min_{x^{(2)} \in \mathbb{R}^{n_2}} L = \frac{1}{2} x^T P^{-1} A x \tag{17}$$

where $P^{-1} = \text{diag}\{(J_1^{-1})^T V_1 J_1^{-1}, -(J_2^{-1})^T V_2 J_2^{-1}\}$. Due to $V_1 \Lambda_1 \preceq 0, V_2 \Lambda_2 \preceq 0$, L is concave in $x^{(1)}$ and convex in $x^{(2)}$. Define matrices $P_{x^{(1)}} = J_1 V_1^{-1} J_1^T \succ 0$, $P_{x^{(2)}} = J_2 V_2^{-1} J_2^T \succ 0$. Under Lemma 1, the trajectories of the primal-dual gradient algorithm given by $\dot{x} = \text{diag}\{P_{x^{(1)}}, -P_{x^{(2)}}\} \frac{\partial L}{\partial x}$ are bounded, which is the same as (13). So we conclude that (13) belongs to Class-$\mathcal{S}$. □

Theorem 3 establishes a necessary and sufficient condition to reverse-engineer (13) as a primal-dual gradient algorithm to solve an unconstrained quadratic saddle point problem. One of the conditions is to check the feasibility of (15). This problem is actually a Semi-Definite Programming (SDP) problem which can be solved using SDP solvers, e.g., SeDuMi [22]. Similar to Remark 3, adding a minus sign before L leads to an alternative formulation of (17) in which $x^{(1)}$ appears a minimizer and $x^{(2)}$ appears a maximizer.

B. LTI Systems with Inputs

In this subsection, we extend the results in Theorems 2-3 to LTI systems with inputs. Consider the following system

$$\dot{x} = Ax + Bu + Cw \tag{18}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $u(t) \in \mathbb{R}^m$ is the control input vector, $C \in \mathbb{R}^{n \times p}$, and $w(t) \in \mathbb{R}^p$ is the exogenous input vector, e.g., disturbance injection.

Remark 5. *A given LTI closed-loop system with either static feedback or dynamic feedback can be rearranged to fit (18). More specifically, system (1)-(2) can be rearranged as:*

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{bmatrix} A & B \\ D & E \end{bmatrix}}_{\tilde{A}} \tilde{x} + \underbrace{\begin{bmatrix} C \\ F \end{bmatrix}}_{\tilde{C}} w$$

where u is included in the augmented state vector $\tilde{x}$. □

The following result follows from Theorem 2.

Theorem 4. *Let u, w be constant in system (18) and the set $\{x | Ax + Bu + Cw = \mathbf{0}\}$ be nonempty. System (18) belongs to Class-$\mathcal{O}$ if and only if (18) is marginally or asymptotically stable, eig$(A) \in \mathbb{R}$ and A is diagonalizable.*

Proof. The proof is similar to that of Theorem 2, by replacing problem (12) with the following unconstrained convex quadratic programming problem:

$$\max_{x \in \mathbb{R}^n} L = \frac{1}{2} x^T P^{-1} A x + x^T P^{-1} (Bu + Cw)$$

where P is positive definite satisfying $P^{-1} A = A^T P^{-1}$. □

In the above theorem, we have extended the definition of Class-$\mathcal{O}$ to include linear systems with inputs, which is straightforward. Similar to (13), let us partition system (18) in the form

$$\underbrace{\begin{bmatrix} \dot{x}^{(1)} \\ \dot{x}^{(2)} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A} x + Bu + Cw \tag{19}$$

where $x^{(1)}(t) \in \mathbb{R}^{n_1}$, $x^{(2)}(t) \in \mathbb{R}^{n_2}$ and $n_1 + n_2 = n$. We then obtain the following theorem.

Theorem 5. *Let u, w be constant in (19) and the set $\{x | Ax + Bu + Cw = \mathbf{0}\}$ be nonempty. System (19) belongs to Class-$\mathcal{S}$ if and only if the following conditions are satisfied: (i) (19) is marginally or asymptotically stable; (ii) the eigenvalues of both A_{11} and A_{22} are non-positive real; (iii) both A_{11} and A_{22} are diagonalizable with the diagonal canonical forms $A_{11} = J_1 \Lambda_1 J_1^{-1}, J_1 \in \mathbb{R}^{n_1 \times n_1}, A_{22} = J_2 \Lambda_2 J_2^{-1}, J_2 \in \mathbb{R}^{n_2 \times n_2}$, and there exist V_1 and V_2 such that (15) holds.*

Proof. The proof is similar to that of Theorem 3, by replacing problem (17) with the following unconstrained quadratic saddle point problem:

$$\max_{x^{(1)} \in \mathbb{R}^{n_1}} \min_{x^{(2)} \in \mathbb{R}^{n_2}} L = \frac{1}{2} x^T P^{-1} A x + x^T P^{-1} (Bu + Cw)$$

where $P^{-1} = \text{diag}\{(J_1^{-1})^T V_1 J_1^{-1}, -(J_2^{-1})^T V_2 J_2^{-1}\}$. □

In the above theorem, we have extended the definition of Class-$\mathcal{S}$ to include linear systems with inputs. Because Class-$\mathcal{S}'$ is a subset of Class-$\mathcal{S}$, we can apply the results to system (1)-(2) and Class-$\mathcal{S}'$. Partition (1) in the form (19) and rearrange (1)-(2) as

$$\underbrace{\begin{bmatrix} \dot{x}^{(1)} \\ \dot{x}^{(2)} \\ \dot{u} \end{bmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{bmatrix} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ D_1 & D_2 & E \end{bmatrix}}_{\tilde{A}} \tilde{x} + \underbrace{\begin{bmatrix} C_1 \\ C_2 \\ F \end{bmatrix}}_{\tilde{C}} w \tag{20}$$

where $x^{(1)}(t) \in \mathbb{R}^{n_1}$, $x^{(2)}(t) \in \mathbb{R}^{n_2}$, $n_1 + n_2 = n$. Then the following corollary is immediate.

Corollary 1. *Let w be constant in (20) and the set $\{\tilde{x} | \tilde{A}\tilde{x} + \tilde{C}w = \mathbf{0}\}$ be nonempty. System (20) belongs to Class-$\mathcal{S}'$ if and only if the following conditions are satisfied: (i) (20) is marginally or asymptotically stable; (ii) the eigenvalues of A_{11}, A_{22}, E, $E_{ii}, i = 1, \cdots, N$ and $\begin{bmatrix} A_{22} & B_2 \\ D_2 & E \end{bmatrix}$ are non-positive real, and these matrices are diagonalizable; (iii) let the diagonal canonical forms of A_{11}, A_{22}, $E_{ii}, i = 1, \cdots, N$ be $A_{11} = J_1 \Lambda_1 J_1^{-1}, J_1 \in \mathbb{R}^{n_1 \times n_1}$, $A_{22} = J_2 \Lambda_2 J_2^{-1}, J_2 \in \mathbb{R}^{n_2 \times n_2}$, $E_{ii} = J_{E_i} \Lambda_{E_i} J_{E_i}^{-1}, J_{E_i} \in \mathbb{R}^{m_i \times m_i}$, there exist V_1, V_2, V_{E_i} such that*

$$(J_1^{-1})^T V_1 J_1^{-1} A_{12} + A_{21}^T (J_2^{-1})^T V_2 J_2^{-1} = \mathbf{0}$$
$$(J_1^{-1})^T V_1 J_1^{-1} B_1 + D_1^T \textit{diag}\{(J_{E_i}^{-1})^T V_{E_i} J_{E_i}^{-1}\} = \mathbf{0}$$
$$(J_2^{-1})^T V_2 J_2^{-1} B_2 - D_2^T \textit{diag}\{(J_{E_i}^{-1})^T V_{E_i} J_{E_i}^{-1}\} = \mathbf{0}$$
$$(J_{E_i}^{-1})^T V_{E_i} J_{E_i}^{-1} E_{ij} - E_{ji}^T (J_{E_j}^{-1})^T V_{E_j} J_{E_j}^{-1} = \mathbf{0}, \ j \in \mathcal{N}(i)$$

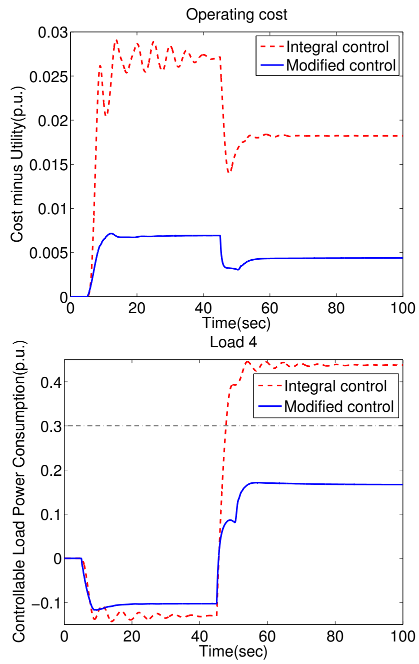

Fig. 1: System response under two control schemes. The black dashdot line in the bottom figure is the capacity upper bound of Load 4 in the DC OPF problem.

$V_1\Lambda_1 = \Lambda_1 V_1,\ V_2\Lambda_2 = \Lambda_2 V_2,\ V_{E_i}\Lambda_{E_i} = \Lambda_{E_i} V_{E_i}$ *and* $V_1 \succ 0,\ V_2 \succ 0,\ V_{E_i} \succ 0$ *hold, where* $i = 1, \cdots, N$.

V. A Practical Example

In this section, we provide a practical example to demonstrate the use of reverse- and forward-engineering for control modification, where we consider frequency control in power networks. In [23], a decentralized frequency integral control scheme is investigated to restore system frequency in the presence of disturbances. Regarding this scheme as a built-in control mechanism, we apply the proposed procedure to redesign the closed-loop system that belongs to Class-$\mathcal{S}'$. The objective is not only to restore the frequency, but also to result in a better supply-demand balance respecting system operating constraints, which can be described as a DC OPF problem given in [12]. To illustrate the effective performance of the modified control scheme, we use the IEEE 14-bus network as an example. The parameters of the network and the scenario are provided in Section IV in [12]. The simulation results are shown in Figure 1. Compared with the case of using only integral control, the modified control scheme leads to less cost and satisfies operating constraints. Due to space limitation, we refer the details to [12].

VI. Conclusion and Future Work

In this paper, we have studied distributed control for a class of linear network systems to achieve optimal steady-state performance using the framework of reverse- and forward-engineering. This framework consists of two steps: firstly, seek an appropriate optimization problem that the system dynamics implicitly solve (*reverse-engineering*); secondly, modify the resulting optimization problem by incorporating a predefined optimization problem and derive control mechanisms to solve the augmented optimization problem (*forward-engineering*). In order to investigate how general this framework is, we have developed necessary and sufficient conditions under which an LTI system can be reverse-engineered as a gradient algorithm to solve either (i) an unconstrained convex optimization problem or (ii) an unconstrained saddle point problem. These conditions are characterized using properties of system matrices and relevant linear matrix inequalities.

This paper serves as our initial step towards developing a reverse- and forward-engineering framework for system control (re)design. In the future, we will focus on more general systems with more complexity. First, we will extend our result to discrete-time LTI systems, and then focus on linear time-varying systems. In addition, we will consider developing necessary and sufficient conditions to reverse-engineer nonlinear dynamic systems. Another research direction is to develop matrix partition methods for Theorems 3 and 5 in Section IV. Last but not least, besides power networks, we are interested in applying our result to do control (re)design for other practical systems, e.g., Hamiltonian systems.

References

[1] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.

[2] W. Shi, N. Li, X. Xie, C.-C. Chu, and R. Gadh, "Optimal residential demand response in distribution network," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 7, pp. 1441–1450, 2014.

[3] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[4] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010.

[5] N. Li, L. Chen, C. Zhao, and S. H. Low, "Connecting automatic generation control and economic dispatch from an optimization view," in *Proc. of 2014 American Control Conference*, 2014, pp. 735–740.

[6] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming*. Stanford University Press, 1958.

[7] A. Jokic, "Price-based optimal control of electrical power systems," *PhD dissertation, Department of Electrical Engineering, Eindhoven University of Technology*, 2007.

[8] F. D. Brunner, H.-B. Dürr, and C. Ebenbauer, "Feedback design for multi-agent systems: A saddle point approach," in *Proc. 51th IEEE Conference on Decision and Control*, 2012, pp. 3783–3789.

[9] E. Dall'Anese, S. Dhople, and G. B. Giannakis, "Regulation of dynamical systems to optimal solutions of semidefinite programs: Algorithms and applications to AC optimal power flow," in *Proc. of 2015 American Control Conference*, 2015, pp. 2087–2092.

[10] 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, 2014.

[11] S. You and L. Chen, "Reverse and forward engineering of frequency control in power networks," in *Proc. 53rd IEEE Conference on Decision and Control*, 2014, pp. 191–198.

[12] X. Zhang, N. Li, and A. Papachristodoulou, "Achieving real-time economic dispatch in power networks via a saddle point design approach," in *Proc. IEEE PES Power and Energy Society General Meeting*, 2015, to appear.

[13] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.

[16] M. Andreasson, D. V. Dimarogonas, H. Sandberg, and K. H. Johansson, "Distributed control of networked dynamical systems: Static feedback, integral action and consensus," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1750–1764, 2014.

[17] D. Feijer and F. Paganini, "Stability of primal-dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, no. 12, pp. 1974–1981, 2010.

[18] X. Zhang and A. Papachristodoulou, "A distributed PID controller for network congestion control problems," in *Proc. of 2014 American Control Conference*, 2014, pp. 5453–5458.

[19] ——, "Improving the performance of network congestion control algorithms," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 522–527, 2015.

[20] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford University Press, Inc. New York, 1999.

[21] G. Vinnicombe, "On the stability of end-to-end congestion control for the internet," in *Technical Report, University of Cambridge, CUED/F-INFENG/TR.398*, 2000.

[22] J. F. Sturm, "Using SEDUMI 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 12, pp. 625–653, 1999.

[23] C. Zhao, E. Mallada, and F. Dörfler, "Distributed frequency control for stability and economic dispatch in power systems," in *Proc. of 2015 American Control Conference*, 2015, pp. 2359–2364.

[24] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, Upper Saddle River, New Jersey, 2002.

APPENDIX

Proof of Lemma 2. Due to $\nabla^2_{y,u,x^{(2)}} L_{\text{au}} = \nabla^2_{y,u,x^{(2)}} L_{\text{sys}} + \gamma \nabla^2_{y,u,x^{(2)}} L_{\text{op}}$, $\nabla^2_{y,u,x^{(2)}} L_{\text{sys}} \succeq 0$ and $\nabla^2_{y,u,x^{(2)}} L_{\text{op}} \succeq 0$, we have $\nabla^2_{y,u,x^{(2)}} L_{\text{au}} \succeq 0$. Similarly, $\nabla^2_{\zeta_i,\lambda_i,\mu_i,x^{(1)}} L_{\text{au}} \preceq 0$ holds. If A is invertible in (1), based on the KKT conditions [14], the following two sets are equivalent:

$\{(y,u,x,\zeta_i,\lambda_i,\mu_i)|(y,u,x,\zeta_i,\lambda_i,\mu_i)$ is a saddle point of $L_{\text{au}}\} \Leftrightarrow \{(y,u,x,\zeta_i,\lambda_i,\mu_i)|(x,u)$ is a saddle point of L_{sys}, $(y,u,\zeta_i,\lambda_i,\mu_i)$ is a saddle point of $L_{\text{op}}\}$

and furthermore, $y = x$ holds. Thus, $(y,u,x,\zeta_i,\lambda_i,\mu_i)$ is a saddle point of L_{au} if and only if $(y,u,\zeta_i,\lambda_i,\mu_i)$ is a saddle point of L_{op} and (x,u) is a saddle point of L_{sys}. □

Proof of Lemma 3. Let (y^*, z^*) be a saddle point of f. Define a candidate Lyapunov function for (9) as

$$U = \frac{1}{2}\Big((y-y^*)^T K_y^{-1}(y-y^*) + (z-z^*)^T K_z^{-1}(z-z^*) + (\hat{y}-y^*)^T K_{ey}\hat{K}_{ey}^{-1}(\hat{y}-y^*) + (\hat{z}-z^*)^T K_{ez}\hat{K}_{ez}^{-1}(\hat{z}-z^*)\Big)$$

which is radially unbounded and positive definite with respect to (y^*, z^*, y^*, z^*) (note that at any equilibrium of (9), $y^* = \hat{y}^*, z^* = \hat{z}^*$ hold). The derivative of U with respect to time along the trajectory of system (9) is given by

$$\begin{aligned}\dot{U} = &-\left(\frac{\partial f}{\partial y}\right)^T (y-y^*) + \left(\frac{\partial f}{\partial z}\right)^T (z-z^*) \\ &- (y-\hat{y})^T K_{ey}(y-\hat{y}) - (z-\hat{z})^T K_{ez}(z-\hat{z}) \\ \le &- f(y,z) + f(y^*,z) + f(y,z) - f(y,z^*) \\ &- (y-\hat{y})^T K_{ey}(y-\hat{y}) - (z-\hat{z})^T K_{ez}(z-\hat{z}) \\ = &f(y^*,z) - f(y^*,z^*) + f(y^*,z^*) - f(y,z^*) \\ &- (y-\hat{y})^T K_{ey}(y-\hat{y}) - (z-\hat{z})^T K_{ez}(z-\hat{z}) \le 0\end{aligned}$$

where the first inequality comes from the fact that f is convex in y and concave in z, and the last inequality follows that (y^*, z^*) is a saddle point of f. When $\dot{U} = 0$, we have $y = \hat{y}, z = \hat{z}$, leading to $\dot{\hat{y}} = \dot{y} = \mathbf{0}, \dot{\hat{z}} = \dot{z} = \mathbf{0}$. So $\frac{\partial f}{\partial y} = \mathbf{0}$ and $\frac{\partial f}{\partial z} = \mathbf{0}$ hold, indicating that (y,z) is a saddle point of f. From LaSalle's invariance principle [24], we conclude that each trajectory of system (9) asymptotically converges to an equilibrium point at which (y,z) is a saddle point of f. □

Proof of Theorem 1. Since A is Hurwitz in (1), under Lemma 2, at any equilibrium point of (8), (x,u) is an optimal solution of problem (3). The proof of convergence is similar to that in Lemma 3, by constructing a quadratic Lyapunov function and showing that its derivative with respect to time is non-increasing along the trajectory of system (8). When this derivative is 0, $u_i = \hat{u}_i$, $y_i = \hat{y}_i$, $\zeta_i = \hat{\zeta}_i$, $\lambda_i = \hat{\lambda}_i$ hold, which leads to $\dot{u}_i = \mathbf{0}, \dot{y}_i = \mathbf{0}, \dot{\zeta}_i = \mathbf{0}, \dot{\lambda}_i = \mathbf{0}$. Given constant u and w, system (1) eventually converges to an equilibrium point at which $x = y$. Then $\dot{\mu}_i = 0, i = 1, \cdots, N$ are true. From LaSalle's invariance principle [24], each trajectory of (8) asymptotically converges to an equilibrium point at which (x,u) is an optimal solution of problem (3). □

Lemma 4. *Given $A \in \mathbb{R}^{n\times n}$, the following statements are equivalent:*

(i) There exists a matrix $P = P^T \in \mathbb{R}^{n\times n}$ and $P \succ 0$, so that $PA = A^T P$ holds;

(ii) eig$(A) \in \mathbb{R}$ and A is diagonalizable.

Proof. (i)⇒(ii). Write A in its Jordan canonical form $A = J\Lambda J^{-1}$, where $\Lambda = \text{diag}\{\Lambda_1, \cdots, \Lambda_a\}$ and each Λ_i is a Jordan block. Without loss of generality, let

$$\Lambda_1 = \begin{bmatrix} \lambda_1 & 1 & \mathbf{0} \\ 0 & \lambda_1 & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots \end{bmatrix}$$

and the dimension of Λ_1 is at least 2×2. Then (i) leads to $PJ\Lambda J^{-1} = (J^{-1})^H \Lambda^H J^H P$ which is equivalent to

$$J^H PJ\Lambda = \Lambda^H J^H PJ. \tag{21}$$

Note that $J^H PJ$ is a Hermitian matrix, and $J^H PJ \succ 0$ since $P \succ 0$ and J is invertible. So we obtain that all the diagonal entries of $J^H PJ$ are positive real. Let

$$J^H PJ = \begin{bmatrix} s_{11} & s_{12} & \cdots \\ s_{12}^H & s_{22} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}$$

where $s_{11}, s_{22} > 0$. From (21), we have $\lambda_1 s_{11} = \lambda_1^H s_{11}$ and $s_{11} + s_{12}\lambda_1 = \lambda_1^H s_{12}$ which result in $\lambda_1 = \lambda_1^H$ and $s_{11} = 0$, i.e., a contradiction. Therefore, each Jordan block should have the dimension 1×1 and a real diagonal entry, i.e., A is diagonalizable and eig$(A) \in \mathbb{R}$ are true.

(ii)⇒(i). Based on (ii), write A as its diagonal canonical form $A = J\Lambda J^{-1}$, where Λ is a diagonal matrix with real entries on the diagonal and $J \in \mathbb{R}^{n\times n}$. Define a matrix $V \in \mathbb{R}^{n\times n}$ so that $V = V^T \succ 0$ and $V\Lambda = \Lambda V$ hold. Such a matrix V always exists and the simplest choice is $V = I_n$. We can then find a matrix $P = P^T = (J^{-1})^T V J^{-1} \in \mathbb{R}^{n\times n}$ so that $P \succ 0$ and $PA = A^T P$ hold, i.e., (i) is true. □