Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

# Control Reconfiguration for Improved Performance via Reverse-engineering and Forward-engineering ⋆

**Han Shu** * **Xuan Zhang** * **Na Li** **
**Antonis Papachristodoulou, *Fellow, IEEE*** ***

* *Tsinghua-Berkeley Shenzhen Institute, Shenzhen, Guangdong 518055, China (e-mail: shu-h18@mails.tsinghua.edu.cn, xuanzhang@sz.tsinghua.edu.cn)*
** *School of Engineering and Applied Sciences, Harvard University, USA (email: nali@seas.harvard.edu)*
*** *Department of Engineering Science, University of Oxford, UK (email: antonis@eng.ox.ac.uk)*

**Abstract:** This paper presents a control redesign approach to improve the performance of a certain class of dynamical systems. Motivated by recent research on re-engineering cyber-physical systems, we propose a three-step control retrofit procedure. First, we reverse-engineer a dynamical system to dig out an optimization problem it actually solves. Second, we apply an augmented Lagrangian or a hat-x method to solve this optimization problem. Finally, by comparing the original and new dynamics, we obtain the implementation of the redesigned part (i.e., the extra dynamics). As a result, the convergence rate/speed or transient behavior of the given system can be improved while the system structure remains. To show the effectiveness of the proposed approach and its potential applications, we present two practical examples including Internet congestion control and distributed proportional-integral (PI) control.

*Keywords:* Control redesign, reverse-engineering, optimization, primal-dual algorithms, augmented Lagrangian.

## 1. INTRODUCTION

Cyber-physical systems (CPSs) integrate, coordinate and monitor the operations of both the physical process and the cyber world (Rajkumar et al. (2010)). They have significant impacts on society, economy, and environment since they are decisive in supporting fundamental infrastructures and smart applications including automotive systems, transportation systems, smart grids, robotic network systems, etc. However, due to historical reasons, some CPSs are still relying on old ways of control and are therefore inefficient and money-/energy-wasted. For example, aging electricity distribution infrastructures are becoming less reliable and less efficient (Kim and Kumar (2013)). Recently, due to the technological advances in sensing, communication and computation, there are growing interests in establishing more advanced control to improve the efficiency, performance as well as robustness to uncertainties of CPSs (Lee (2008)).

Indeed, there are societal and industrial needs for better control of CPSs. For example, with the increasing penetration of renewable energy in power grids, the conventional control architecture may no longer be suitable for the future: it is essential to design better control to quickly attenuate the large fluctuation caused by those energy sources. Furthermore, the automatic vehicles and mobile robots, being operated in an uncertain environment without complete information, require better control for more safety and reliability. In these circumstances, a faster convergence rate or better transient behavior of the controlled system is a necessity to achieve a faster response time or improved reliability. Motivated by these practical needs, in this paper, we present a control redesign approach to systematically improve the performance of a certain class of dynamical CPSs.

The control redesign problem has been studied under various methods for several decades. Usually, improving those existing systems' performance can be achieved from two perspectives. One way is rebuilding a new controller for the whole system. For example, Deveci and Kasnakoğlu (2016) showed that by incorporating numerical modeling and simulation to design the controller for a photo-voltaic system, system performance and robustness could be improved. This method can achieve a better result but with the expense of complexity and high investment. The other way is to redesign the existing controller by adding extra dynamics while maintaining the structure of the existing controller. This can be simply realized, for example, by using additional information produced by newly added sensors, without affecting the structure of the controlled system. For instance, Zhang and Papachristodoulou (2014) proposed a modification approach based on a penalty method for improving the performance and robustness to delays of Internet congestion control protocols, and NešIć and Grüne (2005) redesigned controllers by adding extra terms obtained based on continuous-time systems and Lyapunov functions to enhance stability and robustness.

On the other hand, the idea of redesign using a reverse- and forward-engineering framework for optimality has been introduced for over twienty years (Chiang et al. (2007)).

⋆ This work was supported by Tsinghua-Berkeley Shenzhen Institute Research Start-Up Funding. *Corresponding Author: Xuan Zhang.*

Copyright lies with the authors

From existing protocols designed based on engineering instincts, utility functions are implicitly determined and can be extracted via reverse-engineering. Then based on the insights obtained from reverse-engineering, forward-engineering systematically improves the protocols (Chiang et al. (2007); Low and Lapsley (1999)). Recently, Li et al. (2015) connected automation generation control (AGC) and economic dispatch by reverse-engineering AGC to improve power system economical efficiency, and Zhang et al. (2015, 2018) developed a reverse- and forward-engineering framework to redesign control for improved efficiency, achieving optimal steady-state performance in network systems. Here we use this framework as a tool to study CPSs from an optimization perspective.

Nevertheless, little attention has been paid to improve system performance in terms of convergence rate/speed and transient behaviors. This paper utilizes the reverse- and forward-engineering framework, together with several techniques of changing condition numbers, to improve the performance of one certain class of dynamical CPSs. These systems include but not limited to existing protocols and controlled systems.

The main contribution of this paper is fourfold. Firstly, we propose a control reconfiguration approach to systematically improve the performance of a certain class of systems while maintaining the original control structure, i.e., if the original control structure is centralized/distributed, the redesign control structure will be centralized/distributed as well. This is realized by obtaining extra dynamics and adding it to the original control based on reverse-engineering and several acceleration techniques. Furthermore, we prove the linear convergence rate of the primal-dual gradient descent algorithm in discrete-time and provide bounds for step sizes. Thirdly, we investigate the change of condition numbers under augmented Lagrangian and hat-x methods. This leads to the control reconfiguration. Finally, we demonstrate two practical applications of our theoretical results to existing controlled systems.

The rest of this paper is organized as follows. Section 2 introduces the reverse-engineering technique and presents our target systems, together with two motivating examples. In Section 3, we prove the linear convergence rate of primal-dual gradient algorithm and we investigate the change of condition number using the penalty method and hat-x method. Section 4 presents the reconfiguration steps and the explicit formula of extra dynamics. Section 5 provides simulation results of the motivating applications to illustrate the effectiveness of the redesign framework, and Section 6 concludes the paper.

**Notations:** Throughout this paper, we use upper case roman letters to denote matrices, lower case roman letters to denote column vectors and lower case Greek letters to denote scalars. Let diag$\{\star\}$ denote the diagonal matrix with corresponding entries $\star$ on the main diagonal. Let $A \succeq 0$ ($A \succ 0$) denote that a square matrix is positive semi-definite (positive definite). For two symmetric matrices $A_1$ and $A_2$, notation $A_1 \preceq A_2$ implies that $A_2 - A_1$ is positive semi-definite. Let $\langle \cdot, \cdot \rangle$ represent Euclidean inner product, and let $\| \cdot \|$ denote Euclidean norm for vectors and spectrum norm for matrices. Denote by $\mathbb{R}^n$ the $n$-dimensional Euclidean space and by $A \in \mathbb{R}^{m \times n}$ an $m \times n$ real matrix. Let $x^T$, $\nabla f$ and $\nabla^2 f(x)$ denote the transpose of $x$, the gradient (as a column vector) and the Hessian matrix of $f$. Let $\sigma_{\max}(B)$ and $\sigma_{\min}(B)$ denote the maximum and minimum singular values of $B$ respectively. Let $\lambda(A)$ denote the eigenvalues of a square matrix $A$. Denote by $\mathbf{0}$ a matrix of zeros with its dimension determined by the context and by $I_n$ the identity matrix with size $n \times n$.

## 2. PRELIMINARIES

In this section, we introduce one special class of systems that can be reverse-engineered as primal-dual gradient algorithms. In particular, we show what kinds of conditions are required to determine whether or not a system belongs to this class (Zhang et al. (2018)), mainly for the discrete-time linear cases. Also, we present two practical examples in this class as our motivating applications.

### *2.1 Target Systems: Class-$\mathcal{S}$*

Reverse-engineering aims to seek a proper optimization problem inherently from given dynamics. Different from the previous optimization-to-algorithm framework, it generates a reverse flow, i.e., algorithm-to-optimization.

Consider a linear time-invariant (LTI) system

$$x_{k+1} = Ax_k + Cw_k \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times p}$ and $w_k \in \mathbb{R}^p$ is the exogenous input, e.g., disturbances.

In general, any given discrete-time LTI closed-loop system with either static feedback or dynamic feedback can be rearranged to fit (1). This paper focuses on the class of systems that can be reverse-engineered as primal-dual gradient algorithms to solve saddle point problems, denoted as Class-$\mathcal{S}$ [1]. Another special class, Class-$\mathcal{O}$ [2] (as defined in Zhang et al. (2018)) that can be reverse-engineered as gradient descent algorithms to solve unconstrained optimization problems, is a subclass of Class-$\mathcal{S}$. Therefore, the class considered here is more general.

**Class-$\mathcal{S}$**: System (1) belongs to Class-$\mathcal{S}$ if there exists a function $f(x^{(1)}, x^{(2)}) : \mathbb{R}^n \to \mathbb{R}$ and positive definite matrices $P_{x^{(1)}}, P_{x^{(2)}}$ such that $\nabla^2_{x^{(1)}} f \preceq 0$, $\nabla^2_{x^{(2)}} f \succeq 0$, the saddle point set $\{\nabla f(x) = \mathbf{0}\}$ is nonempty, and (1) is a primal-dual gradient algorithm to solve $\max_{x^{(1)}} \min_{x^{(2)}} f$, i.e., $x_{k+1} = x_k + \mathrm{diag}\{P_{x^{(1)}}, -P_{x^{(2)}}\} \nabla f|_{x=x_k}$. Here state $x$ is partitioned into $x^{(1)}$ and $x^{(2)}$.

In the above definition, the partition may not be unique. Accordingly, we rearrange system (1) as follows

$$\underbrace{\begin{bmatrix} {x_{k+1}}^{(1)} \\ {x_{k+1}}^{(2)} \end{bmatrix}}_{x_{k+1}} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A} x_k + Cw \tag{2}$$

where $x_k^{(1)} \in \mathbb{R}^{n_1}$, $x_k^{(2)} \in \mathbb{R}^{n_2}$ and $n_1 + n_2 = n$. For linear systems in Class-$\mathcal{S}$, the associated function $f$ must be a convex quadratic function, i.e.,

$$f = \frac{1}{2} x^T \underbrace{\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}}_{Q} x + x^T R(Cw) + S(w) \tag{3}$$

where $Q_{11} \in \mathbb{R}^{n_1\times n_1}$ is symmetric and negative semi-definite, $Q_{22} \in \mathbb{R}^{n_2\times n_2}$ is symmetric and positive semi-definite (i.e., $f$ is concave in $x^{(1)}$ and convex in $x^{(2)}$), and $Q_{12} \in \mathbb{R}^{n_1\times n_2}$. According to the definition of Class-$\mathcal{S}$, the following theorem is obtained.

**Theorem 1.** *Let $w$ be constant in (2) and the set $\{(A - I)x + Cw = \mathbf{0}\}$ be nonempty. System (2) belongs to Class-$\mathcal{S}$ if and only if the following conditions are satisfied: (i) system (2) is marginally or asymptotically stable; (ii) the eigenvalues of $A_{11} - I_{n_1}$ and $A_{22} - I_{n_2}$ are non-positive real;*

[1] Notation $\mathcal{S}$ stands for *saddle-point algorithms.*
[2] Notation $\mathcal{O}$ stands for *optimization algorithms.*

*(iii)* $A_{11}-I_{n_1}$ *and* $A_{22}-I_{n_2}$ *are diagonalizable with the diagonal canonical forms given by* $A_{11}-I_{n_1}=J_1\Lambda_1 J_1^{-1}, J_1\in\mathbb{R}^{n_1\times n_1}, A_{22}-I_{n_2}=J_2\Lambda_2 J_2^{-1}, J_2\in\mathbb{R}^{n_2\times n_2}$*, and there exist* $V_1$ *and* $V_2$ *such that*

$$\begin{aligned}&(J_1^{-1})^T V_1 J_1^{-1} A_{12} + A_{21}^T (J_2^{-1})^T V_2 J_2^{-1} = \mathbf{0}\\ &V_1\Lambda_1 = \Lambda_1 V_1,\ V_2\Lambda_2 = \Lambda_2 V_2\\ &V_1 \prec 0,\ V_2\prec 0.\end{aligned} \tag{4}$$

Also, there could be multiple optimization problems corresponding to the same system (1). The derivation procedure of those problems can be found in Zhang et al. (2018).

### 2.2 Internet Congestion Control

In the following, we propose Internet congestion control as a motivating example that belongs to Class-$\mathcal{S}$. Internet congestion control regulates the data transfer and efficient bandwidth sharing between sources and links. In the literature, numerous Internet congestion control algorithms have been proposed. Here we consider a standard primal-dual congestion control algorithm given by Srikant (2004)

$$x_i(k+1) = x_i(k) + \varepsilon k_{x_i}(U_i'(x_i(k)) - q_i(k)) \tag{5a}$$

$$q_i(k) = \sum_{l=1}^{M} R_{li} p_l(k) \tag{5b}$$

$$p_l(k+1) = p_l(k) + \varepsilon k_{p_l}(y_l(k) - c_l)^+_{p_l(k)} \tag{5c}$$

$$y_l(k) = \sum_{i=1}^{N} R_{li} x_i(k) \tag{5d}$$

where $\varepsilon$ is the step size, $U_i(x_i)$ is the utility function of user $i$, which is assumed to be a continuously differentiable, monotonically increasing, strictly concave function of the transmission rate $x_i$ and $(y_l(k)-c_l)^+_{p_l} = \max\{y_l(k)-c_l, 0\}$ if $p_l = 0$; $(y_l(k)-c_l)^+_{p_l} = y_l(k)-c_l$ if $p_l>0$. $R$ is a routing matrix describing the interconnection, i.e.,

$$R_{li} = \begin{cases}1 & \text{if user } i \text{ uses link } l,\\ 0 & \text{otherwise.}\end{cases}$$

Also, $k_{x_i} > 0$ is the rate gain, $p_l > 0$ is the link price, $k_{p_l} > 0$, $N$ is the number of sources and $M$ is the number of links. The above dynamics can be rearranged in the vector form as

$$x(k+1) = x(k) + \varepsilon \mathrm{diag}\{k_{x_i}\}\left(U'(x(k)) - R^T p(k)\right) \tag{6a}$$

$$p(k+1) = p(k) + \varepsilon \mathrm{diag}\{k_{p_l}\}\left(Rx(k) - c\right)^+_{p(k)} \tag{6b}$$

where $x(k), U'(x(k)) \in \mathbb{R}^N$, $p(k), c\in\mathbb{R}^M$ are the corresponding vector forms of $x_i(k), U_i'(x_i(k)), p_l(k), c_l$.

Suppose $U(x)$ is quadratic, i.e., $U(x) = -\frac{1}{2}x^TQ_1x + Q_2x + s$ where $Q_1$ is a diagonal, positive definite constant matrix, $Q_2$ is a row vector with positive constant elements and $s$ is a constant vector, then (6) becomes

$$\underbrace{\begin{bmatrix}p(k+1)\\ x(k+1)\end{bmatrix}}_{z(k+1)} = \underbrace{\begin{bmatrix} I_M & \varepsilon\mathrm{diag}\{v_l k_{p_l}\}R\\ -\varepsilon \mathrm{diag}\{k_{x_i}\}R^T & I_N - \varepsilon\mathrm{diag}\{k_{x_i}\}Q_1\end{bmatrix}}_{A} \times \underbrace{\begin{bmatrix}p(k)\\ x(k)\end{bmatrix}}_{z(k)} + \underbrace{\begin{bmatrix}d_1\\ d_2\end{bmatrix}}_{Cw} \tag{7}$$

where $v_l = 1$ if $p_l(k) > 0$ or $p_l(k) = 0, y_l(k) > c_l$, otherwise $v_l = 0$, and $d_1, d_2$ denote the remaining constant terms. Compared with (2), it is straightforward: $A_{11} - I_{n_1}$ is $\mathbf{0}$ and $A_{22} - I_{n_2}$ is $-\varepsilon\mathrm{diag}\{k_{x_i}\}Q_1$. They satisfy the conditions listed in Theorem 1 and thus, the above dynamics can be reverse-engineered as a primal-dual algorithm to solve

$$\max_{p\in\mathbb{R}^M, p_l\geq 0}\ \min_{x\in\mathbb{R}^N} f = -U(x) + p^T(Rx - c). \tag{8}$$

In general, the reverse-engineering from (6) to the above problem always works if $U(x)$ is concave. It is of interest to redesign the primal-dual congestion control algorithm (6) for a faster convergence speed and better transient behavior to improve the performance of data transfer.

### 2.3 Distributed PI Control for Single Integrator Dynamics

In the following, we propose distributed PI control for single integrators as a motivating example that belongs to Class-$\mathcal{S}$. Consider $n$ agents with single integrator dynamics

$$\dot{y}_i = d_i + u_i$$

where $d_i$ is a constant disturbance and $u_i$ is the control input given by

$$u_i = -\sum_{j\in\mathcal{N}_i}\left(\rho_2(y_i - y_j) + \rho_1\int_0^t \big(y_i(\tau) - y_j(\tau)\big)d\tau\right) - \delta\left(y_i - y_i(0)\right) \tag{9}$$

where $\rho_1, \rho_2, \delta$ are positive constant parameters, $y_i(0)$ is the initial condition and $\mathcal{N}_i$ is the set of neighbors of agent $i$. This controller drives agents to reach consensus under static disturbances. According to Theorem 6 in Andreasson et al. (2014), this controller maintains stability for any constant disturbance $d_i$ and initial conditions.

Introduce integral action $\dot{z}_i = y_i - y_n$ and rearrange the dynamics after discretizing in the vector form as

$$\underbrace{\begin{bmatrix}z(k+1)\\ y(k+1)\end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix}I_{n-1} & \varepsilon D\\ -\varepsilon\rho_1\tilde{L} & I_n - \varepsilon\rho_2 L - \varepsilon\delta I_n\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}z(k)\\ y(k)\end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix}\mathbf{0}\\ \varepsilon\big(d + \delta y(0)\big)\end{bmatrix}}_{Cw} \tag{10}$$

where $z(k)\in\mathbb{R}^{n-1}$, $y(k), d\in\mathbb{R}^n$, $x(k)\in\mathbb{R}^{2n-1}$, $\varepsilon$ is the step size and $D = [I_{n-1}\quad -\mathbf{1}]$ ($\mathbf{1}$ is a column vector of ones). $L\in\mathbb{R}^{n\times n}$ is the Laplacian of the connected agent network and $\tilde{L}\in\mathbb{R}^{n\times(n-1)}$ is obtained after removing the $n$th column of $L$. Compared with (2), it is straightforward to notice that $A_{11} - I_{n_1}$ is $\mathbf{0}$ and $A_{22} - I_{n_2}$ is $-\varepsilon\rho_2 L - \varepsilon\delta I_n$. They satify the conditions listed in Theorem 1 and therefore, the above dynamics can be reverse-engineered as a primal-dual algorithm to solve

$$\begin{aligned}\max_{z\in\mathbb{R}^{n-1}}\ \min_{y\in\mathbb{R}^n} f &= \frac{1}{2}x^T\underbrace{\begin{bmatrix}\mathbf{0} & \rho_1\tilde{L}^T\\ \rho_1\tilde{L} & \rho_2 L + \delta I_n\end{bmatrix}}_{Q}x - x^T\begin{bmatrix}\mathbf{0}\\ d + \delta y(0)\end{bmatrix}\\ &= \frac{\rho_2}{2}y^TLy + \rho_1 z^T\tilde{L}^Ty + \frac{\delta}{2}y^Ty - y^Td - y^T\delta y(0).\end{aligned} \tag{11}$$

It is of interest to redesign the distributed PI controller (9) to improve system performance.

## 3. SOLUTION ALGORITHMS

For CPSs (1) in Class-$\mathcal{S}$, they can be reverse-engineered as primal-dual gradient algorithms to solve saddle point problems. Then it is natural to apply faster or slower algorithms to solve this problem, which can result in a redesigned system formula with improved performance. In this section, we introduce the standard primal-dual gradient algorithm and prove its linear convergence rate, which is related to the condition number. Then methods that change the condition number including augmented Lagrangian and hat-x methods are introduced. By combining these methods with the reverse-engineering technique, a well-structured redesign approach naturally appears.

Consider a convex-concave saddle point problem given by

$$\max_{\lambda\in\mathbb{R}^m}\min_{x\in\mathbb{R}^n}\mathcal{L}(x,\lambda)=f(x)+\lambda^T(Bx-b) \tag{12}$$

where $x\in\mathbb{R}^n$, $\lambda\in\mathbb{R}^m$ is the Lagrangian multiplier vector (dual variable vector), $B\in\mathbb{R}^{m\times n}$ and $b\in\mathbb{R}^m$.

**Assumption 1.** *The objective function $f\in\mathscr{S}_{\mu,L}$, i.e., for any $x,y\in\mathbb{R}^n$,*

$$\mu\left\|x-y\right\|\leq\left\|\nabla f(x)-\nabla f(y)\right\|\leq L\left\|x-y\right\| \tag{13}$$

$$\frac{\mu}{2}\left\|y-x\right\|^2\leq f(y)-f(x)-\nabla f(x)^T(y-x)\leq\frac{L}{2}\left\|y-x\right\|^2. \tag{14}$$

**Assumption 2.** *Matrix $B$ is full row rank.*

**Remark 1.** *Parameters $L$ and $\mu$ can be obtained by:*

$$\mu\leq\frac{\left\|\nabla f(x)-\nabla f(y)\right\|}{\left\|x-y\right\|}\leq L. \tag{15}$$

*If the objective is twice differentiable, $\nabla^2 f$ is bounded by*

$$\mu I_n\preceq\nabla^2 f(x)\preceq LI_n. \tag{16}$$

The corresponding primal problem of (12) is an equality constrained convex optimization problem given by

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t. } Bx=b. \tag{17}$$

This is a very important type of optimization problems due to the wide applications in engineering and control systems. *For problem in this form, $A_{11}=I_{n1}$ in (2), which is the target case considered in this paper.* Equation (12) can be further expressed as

$$\min_{\lambda\in\mathbb{R}^m} f^*(-B^T\lambda)+\lambda^T b \tag{18}$$

where $f^*$ denotes the conjugate of $f$ and is defined as $f^*(y)=\sup_{x\in\mathbb{R}^n}\{\langle x,y\rangle-f(x)\}$ for all $y\in\mathbb{R}^n$.

### *3.1 Primal-Dual Gradient Descent*

Our target systems (1), i.e., belonging to Class-$\mathcal{S}$ with $A_{11}=I_{n1}$, can be reverse-engineered as a primal-dual gradient descent (PDGD) algorithm to solve problem (12).

---

**Algorithm 1** First-order primal-dual gradient method

**Setting:** Choose appropriate positive step sizes $\varepsilon_1,\varepsilon_2$, and let $x_0$ and $\lambda_0$ be arbitrary initial conditions.

$$x_{k+1}=x_k-\varepsilon_1(\nabla f(x_k)+B^T\lambda_k) \tag{19a}$$
$$\lambda_{k+1}=\lambda_k+\varepsilon_2(Bx_k-b) \tag{19b}$$

---

Let $(x^*,\lambda^*)$ denote the saddle point of $\mathcal{L}$ and it satisfies the optimality conditions

$$\nabla_x\mathcal{L}=\nabla f(x^*)+B^T\lambda^*=0 \tag{20a}$$
$$\nabla_\lambda\mathcal{L}=Bx^*-b=0. \tag{20b}$$

Since the primal problem is strongly convex and the constraint is affine, strong duality holds. Therefore, $x^*$ is unique and is an optimal solution of the primal problem (17). When Assumption 2 holds, $\lambda^*$ is unique.

**Lemma 1.** *Assume $\tilde{f}(x)=f(x)+x^TB^T\lambda_k$ and $\tilde{x}^*=\arg\min_{x\in\mathbb{R}^n}\tilde{f}(x)$, then $\tilde{x}^*=\nabla f^*(-B^T\lambda_k)$ and as $k\to\infty$, $\tilde{x}^*$ tends to $x^*$.*

*Proof.* For fixed $\lambda_k$, the update rule of $x_k$ (19a): $x_{k+1}=x_k-\varepsilon_1(\nabla f(x_k)+B^T\lambda_k)$ is a gradient descent step for unconstrained problem $\min_{x\in\mathbb{R}^n}\tilde{f}(x)$. Function $\tilde{f}(x)$ has the same function parameters as $f(x)$, i.e., $\tilde{f}(x)$ is also $\mu$-strongly convex and has Lipschitz continuous gradient $L$. According to the optimality condition, we have $\nabla\tilde{f}(\tilde{x}^*)=\nabla f(\tilde{x}^*)+B^T\lambda_k=0$. Since the gradient $\nabla f^*$ is the inverse of $\nabla f$ (Rockafellar (1970)), we have $\tilde{x}^*=\nabla f^{-1}(-B^T\lambda_k)=\nabla f^*(-B^T\lambda_k)$. Similarly, according to (20a), we have $x^*=\nabla f^*(-B^T\lambda^*)$. Since the sequence $\{(x_k,\lambda_k)\}_{k\geq 0}$ generated by Algorithm 1 converges to $(x^*,\lambda^*)$ (Bertsekas (1997)), $\tilde{x}^*$ tends to $x^*$. □

**Theorem 2.** *Suppose Assumptions 1 and 2 hold and step sizes $\varepsilon_1,\varepsilon_2$ satisfy $0<\varepsilon_1\leq\frac{2}{L+\mu}$, $0<\varepsilon_2\leq\frac{2}{\sigma_{\min}^2(B)/L+\sigma_{\max}^2(B)/\mu}$ and $c<1$, then Algorithm 1 converges to the unique saddle point $(x^*,\lambda^*)$ exponentially. Let $a_k=\left\|x_k-\nabla f^*(-B^T\lambda_k)\right\|$, $b_k=\left\|\lambda_k-\lambda^*\right\|$ and define a potential function $V_k=\gamma a_k+b_k$, then for some constants $c,\gamma$ that depend on $\varepsilon_1,\varepsilon_2$, we have*

$$V_{k+1}\leq cV_k \tag{21}$$

*where $\gamma>0$ and $c=\max\{c_1,c_2\}<1$ with $c_1=1-\mu\varepsilon_1+\frac{\varepsilon_2\sigma_{\max}^2(B)}{\mu}+\frac{\varepsilon_2\sigma_{\max}(B)}{\gamma}$ and $c_2=1-\frac{\varepsilon_2\sigma_{\min}^2(B)}{L}+\frac{\varepsilon_2\gamma\sigma_{\max}^3(B)}{\mu^2}$.*

*Proof.* See the Appendix in Shu et al. (2020). □

Note that the potential function $V_k$ decreases at a geometric rate and the error of $\|x_k-x^*\|$ and $\|\lambda_k-\lambda^*\|$ are bounded by $V_k$: $\|\lambda_k-\lambda^*\|\leq V_k$ and $\|x_k-x^*\|\leq\left\|x_k-\nabla f^*(-B^T\lambda_k)\right\|+\left\|\nabla f^*(-B^T\lambda_k)-x^*\right\|\leq a_k+\frac{\sigma_{\max}(B)}{\mu}b_k\leq\max\left\{\frac{1}{\gamma},\frac{\sigma_{\max}(B)}{\mu}\right\}V_k$. Therefore, as $V_k$ approaches zero, $(x_k,\lambda_k)$ approaches the saddle point $(x^*,\lambda^*)$.

**Corollary 1.** *When the step sizes $\varepsilon_1=\frac{2}{L+\mu}$ and $\varepsilon_2=\left(\frac{\sigma_{\max}^2(B)}{\mu}+\frac{\sigma_{\max}(B)}{\gamma}+\frac{\sigma_{\min}^2(B)}{L}-\frac{\gamma\sigma_{\max}^3(B)}{\mu^2}\right)^{-1}\frac{2\mu}{L+\mu}$, the optimal rate is achieved.*

*Proof.* Since the geometric factor is determined by $c$, the convergence rate is optimal when $c$ is minimized. For a specific problem, parameters including $\mu,L,\gamma,\sigma_{\max}(B)$, $\sigma_{\min}(B)$ are fixed and we are able to adjust step sizes only. It is straightforward to notice that $c_1=1-\mu\varepsilon_1+\frac{\varepsilon_2\sigma_{\max}^2(B)}{\mu}+\frac{\varepsilon_2\sigma_{\max}(B)}{\gamma}$ is monotonically decreasing in $\varepsilon_1$ and increasing in $\varepsilon_2$ while $c_2=1-\frac{\varepsilon_2\sigma_{\min}^2(B)}{L}+\frac{\varepsilon_2\gamma\sigma_{\max}^3(B)}{\mu^2}$ is monotonically decreasing in $\varepsilon_2$ since $\gamma<\frac{\mu^2\sigma_{\min}^2(B)}{L\sigma_{\max}^3(B)}$. Therefore, $c$ is minimized when $\varepsilon_1$ takes its upper limit $\frac{2}{L+\mu}$ and $c_1=c_2$, from which we obtain the value of $\varepsilon_2$. □

Define condition number $\kappa=\frac{L}{\mu}$. Problems with small $\kappa$ are called good-conditioned (Nesterov (2018)). Qualitatively speaking, smaller $\kappa$ leads to a faster convergence rate in Algorithm 1, since $c_1$ and $c_2$ decrease when $\mu$ increases and $L$ decreases. By changing $\kappa$ of objectives, we can change the convergence rate when applying Algorithm 1.

*3.2 Augmented Lagrangian*

Adding the square of equality constrains as penalty terms can change the condition number of the original problem while the optimal solution stays unchanged. This method is known as the Augmented Lagrangian (AL) method or method of multipliers. The corresponding augmented Lagrangian function for problem (17) is

$$\mathcal{L}_a = f(x) + \lambda^T(Bx - b) + \frac{\alpha}{2}\|Bx - b\|^2 \tag{22}$$

where $\alpha$ is a scalar. Equation (22) is the Lagrangian for

$$\min_{x\in\mathbb{R}^n} f(x) + \frac{\alpha}{2}\|Bx - b\|^2 \quad \text{s.t.} \;\; Bx = b \tag{23}$$

which has the same minimum and optimal solution as the original problem (17) (Bertsekas (1997)).

Let $h(x) = f(x) + \frac{\alpha}{2}\|Bx - b\|^2$ and $H = \nabla^2 h(x)$, then $H = \nabla^2 f(x) + \alpha B^T B$ and $\lambda_{\min}(H) I_n \preceq \nabla^2 h(x) \preceq \lambda_{\max}(H) I_n$, and its condition number, denoted as $\kappa_a$, is $\frac{\lambda_{\max}(H)}{\lambda_{\min}(H)}$. Denote by $\kappa_0$ the condition number of the standard primal-dual and $\kappa_0 = L/\mu$. For a specific problem, we are able to obtain the numerical form of matrix $H$. Under this circumstance, we can calculate $\kappa_a$ and $\kappa_0$ precisely. If $\kappa_a < \kappa_0$, AL is able to improve the convergence rate; otherwise, $\alpha$ should be set to zero to avoid the influence of penalty terms. Note that matrix $B$ has a significant influence on the effectiveness of this method.

Another benefit of using augmented Lagrangian method is the convexification when $f$ is not strongly convex in $\mathbb{R}^n$.

**Proposition 1.** *Assume that $\nabla^2 f(x)$ is positive definite on the nullspace of $B^T B$, i.e., $y^T \nabla^2 f(x) y > 0$ for all $y \neq 0$ with $y^T B^T B y = 0$. Then there exists a scalar $\bar{\alpha}$ such that for $\alpha > \bar{\alpha}$, we have (See Theorem 4.2 in Anstreicher et al. (2000) for the proof)*

$$\nabla^2 f(x) + \alpha B^T B \succ 0. \tag{24}$$

Therefore, according to Proposition 1, we can achieve convexification via augmented Lagrangian.

*3.3 Hat-x*

Another way is to introduce a free variable $\hat{x} \in \mathbb{R}^n$ to prevent the dramatic change of $x$. This leads to

$$\begin{aligned} &\min_{x,\hat{x}\in\mathbb{R}^n} f(x) + \frac{\alpha}{2}\|x - \hat{x}\|^2 \\ &\text{s.t.} \quad Bx = b \end{aligned} \tag{25}$$

where $\alpha$ is a scalar. This problem is equivalent to

$$\min_{z\in\mathbb{R}^{2n}} h(z) \qquad \text{s.t.} \quad \bar{B}z = b$$

where $z = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$, $h(z) = f(x) + \frac{\alpha}{2} z^T \begin{bmatrix} I_n & -I_n \\ -I_n & I_n \end{bmatrix} z$, $\bar{B} = [\,B\ \mathbf{0}\,] \in \mathbb{R}^{m\times 2n}$. Suppose Assumption 1 holds, then $\bar{B}$ is also full row rank and its maximal and minimal singular values of $\bar{B}$ are the same as $B$. The Lagrangian is

$$\mathcal{L}_h = f(x) + \lambda^T(Bx - b) + \frac{\alpha}{2}\|x - \hat{x}\|^2. \tag{26}$$

Applying an optimality condition, we have

$$\nabla_x \mathcal{L}_h = \nabla f(x^*) + B^T\lambda^* + \alpha(x^* - \hat{x}^*) = 0 \tag{27a}$$
$$\nabla_{\hat{x}} \mathcal{L}_h = \alpha(\hat{x}^* - x^*) = 0 \tag{27b}$$
$$\nabla_\lambda \mathcal{L}_h = Bx^* - b = 0. \tag{27c}$$

Then $\hat{x}^* = x^*$ and (27) is equivalent to (20). Therefore, the value of $x^*, \lambda^*$ in the optimal solution $(x^*, \hat{x}^*, \lambda^*)$ in (25) is the same as that in (17). Denote the condition number as $\kappa_h$. Let $H = \nabla^2 h(z)$, then $\kappa_h = \frac{\lambda_{\max}(H)}{\lambda_{\min}(H)}$.

**Lemma 2.** *Assume $A, B \in \mathbb{R}^{n\times n}$, if $A \preceq B$, then $\lambda_{\max}(A) \le \lambda_{\max}(B)$ and $\lambda_{\min}(A) \le \lambda_{\min}(B)$ hold.*

*Proof.* For any $x \in \mathbb{R}^n$, we have $x^T A x \le x^T B x$. Assume $x^* = \arg\max_{\|x\|=1} x^T A x$. Then $\lambda_{\max}(A) = x^{*T} A x^* \le x^{*T} B x^* \le \max_{\|x\|=1} x^T B x = \lambda_{\max}(B)$. On the other hand, assume $\bar{x} = \arg\min_{\|x\|=1} x^T B x$. Then $\lambda_{\min}(B) = \bar{x}^T B \bar{x} \ge \bar{x}^T A \bar{x} \ge \min_{\|x\|=1} x^T A x = \lambda_{\min}(A)$. □

**Proposition 2.** *Assume Assumption 1 holds, then* $\frac{2\alpha+\mu+\sqrt{\mu^2+4\alpha^2}}{2\alpha+L-\sqrt{L^2+4\alpha^2}} \le \kappa_h \le \frac{2\alpha+L+\sqrt{L^2+4\alpha^2}}{2\alpha+\mu-\sqrt{\mu^2+4\alpha^2}}$.

*Proof.* Since $f \in \mathscr{S}_{\mu,L}$, we have $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ and the Hessian of $h(z)$ is $H = \begin{bmatrix} \nabla^2 f(x) + \alpha I_n & -\alpha I_n \\ -\alpha I_n & \alpha I_n \end{bmatrix}$. Then $\underline{H} \preceq H \preceq \overline{H}$, where $\underline{H} = \begin{bmatrix} (\mu+\alpha) I_n & -\alpha I_n \\ -\alpha I_n & \alpha I_n \end{bmatrix}$ and $\overline{H} = \begin{bmatrix} (L+\alpha) I_n & -\alpha I_n \\ -\alpha I_n & \alpha I_n \end{bmatrix}$. Their eigenvalues are $\lambda(\underline{H}) = \alpha + \frac{\mu}{2} \pm \frac{1}{2}\sqrt{\mu^2 + 4\alpha^2}$ and $\lambda(\overline{H}) = \alpha + \frac{L}{2} \pm \frac{1}{2}\sqrt{L^2 + 4\alpha^2}$. By applying Lemma 2, we have $\lambda_{\min}(\underline{H}) \le \lambda_{\min}(H) \le \lambda_{\min}(\overline{H})$ and $\lambda_{\max}(\underline{H}) \le \lambda_{\max}(H) \le \lambda_{\max}(\overline{H})$. According to the definition of $\kappa_h$, we obtain the range of $\kappa_h$. □

Next we compare $\kappa_h$ with $\kappa_0$. For the lower bound of $\kappa_h$, let $Z = \frac{2\alpha+\mu+\sqrt{\mu^2+4\alpha^2}}{2\alpha+L-\sqrt{L^2+4\alpha^2}} - \kappa_0$, $x = \frac{\mu}{L}$ and $y = \frac{\alpha}{L}$ with $0 < x \le 1$ and $y > 0$, then

$$Z = \frac{2y + x + \sqrt{x^2 + 4y^2}}{2y + 1 - \sqrt{1 + 4y^2}} - \frac{1}{x}.$$

When $x$ tends to 0, $Z < 0$; otherwise, $Z > 0$. For the higher bound of $\kappa_h$, we have

$$\begin{aligned} &\frac{2\alpha + L + \sqrt{L^2 + 4\alpha^2}}{2\alpha + \mu - \sqrt{\mu^2 + 4\alpha^2}} - \frac{L}{\mu} \\ =&\frac{(2\alpha + L + \sqrt{L^2 + 4\alpha^2})(2\alpha + \mu + \sqrt{\mu^2 + 4\alpha^2}) - 4\alpha L}{4\alpha\mu} \\ =&\frac{1}{4\alpha\mu}\big[(2\alpha + \sqrt{L^2 + 4\alpha^2})(2\alpha + \mu + \sqrt{\mu^2 + 4\alpha^2}) \\ &+ L(\mu + \sqrt{\mu^2 + 4\alpha^2} - 2\alpha)\big]. \end{aligned}$$

Since $\mu + \sqrt{\mu^2 + 4\alpha^2} > 2\alpha$, $\frac{2\alpha+L+\sqrt{L^2+4\alpha^2}}{2\alpha+\mu-\sqrt{\mu^2+4\alpha^2}} > \frac{L}{\mu}$. This implies that $\kappa_h$ can be bigger than $\kappa_0$. In fact, hat-x method increases the dimension of the original problems and works like a low-pass filter. This method slows down system dynamics but smooths the trajectories.

## 4. REDESIGN METHODOLOGY

In this section, we propose the redesign approach and derive the explicit form of the extra dynamics when applying augmented Lagrangian and hat-x methods. In particular, we focus on linear dynamical systems in Class-$\mathcal{S}$ with $A_{11} = I_{n1}$, as many existing systems fall into that category. For convenience, we only consider discrete-time cases here, and continuous-time cases will be discussed in a future paper.

### 4.1 Reconfiguration Steps

The reconfiguration steps are as follows:

(1) **Reverse-engineering:** For a given system (1) with $A_{11} = I_{n1}$, apply Theorem 1 to reverse-engineer it as a primal-dual gradient algorithm (19) for solving certain convex optimization problem (12) (i.e., system dynamics (1) can be rewritten as the form (19)).
(2) **Acceleration:** Apply augmented Lagrangian or hat-x method to solve the problem obtained via reverse-engineering, which results in a redesigned system.
(3) **Implementation:** Rearrange the redesigned system and compare it with the original system (19) to obtain the implementation of the extra dynamics.

### 4.2 Extra Dynamics Derivation

We derive the explicit form of extra dynamics in step (3). For augmented Lagrangian, it is straightforward to identify an extra part compared with the original dynamics:

$$x_{k+1} = x_k - \varepsilon_1 \left(\nabla f(x_k) + B^T\lambda_k\right) \underbrace{-\varepsilon_1 \alpha B^T(Bx_k - b)}_{\Delta u_k} \tag{28}$$

where $\Delta u_k = -\varepsilon_1\alpha B^T(Bx_k - b)$. Similarly, for hat-x method, the iteration of primal variables are

$$x_{k+1} = x_k - \varepsilon_1 \left(\nabla f(x_k) + A^T\lambda_k\right) \underbrace{-\varepsilon_1 \alpha (x_k - \hat{x}_k)}_{\Delta u_k} \tag{29a}$$

$$\hat{x}_{k+1} = \hat{x}_k + \varepsilon_1\alpha(x_k - \hat{x}_k) \tag{29b}$$

where $\Delta u_k = -\varepsilon_1\alpha(x_k - \hat{x}_k)$. Note that in both methods, the extra part is only added to primal variables.

## 5. EXAMPLE REVISITED

In this section, we revisit two practical examples from Section 2.2 and 2.3 in order to show the effectiveness of the proposed retrofit framework.

### 5.1 Internet Congestion Control

Problem (8) obtained via reverse-engineering is equal to

$$\min_{x\in\mathbb{R}^N} f = -U(x) \qquad \text{s.t.} \quad Rx \le c. \tag{30}$$

When $U(x)$ is quadratic as expressed in Section 2.2, $\nabla^2 f = Q_1 \succ 0$, $f \in \mathscr{S}_{L,u}$. Usually, the optimal solution is obtained when constrains are active, i.e., $Rx = c$. Following the redesign procedure, applying augmented Lagrangian and hat-x methods, we obtain the same form of redesigned dynamics but with different $\Delta u(k)$ given by

$$x(k+1) = x(k) + \varepsilon_1 \mathrm{diag}\{k_{x_i}\}\left(U'(x(k)) - R^T p(k)\right) + \Delta u(k)$$
$$p(k+1) = p(k) + \varepsilon_2 \mathrm{diag}\{k_{p_l}\}\left(Rx(k) - c\right)^+_{p(k)}.$$

For augmented Lagrangian method-based redesign,

$$\Delta u(k) = -\varepsilon_1\alpha R^T(Rx(k) - c)$$

while for hat-x-based redesign,

$$\Delta u(k) = -\varepsilon_1\alpha(x(k) - \hat{x}(k))$$

where $\hat{x}(k+1) = \hat{x}(k) + \varepsilon_1\alpha(x(k) - \hat{x}(k))$. With the extra dynamics added to the primal variable, primal-dual congestion control algorithm can achieve a faster convergence speed and better transient behavior.

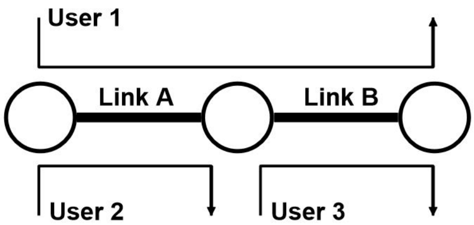


Fig. 1. A 2-link network shared by 3 users.

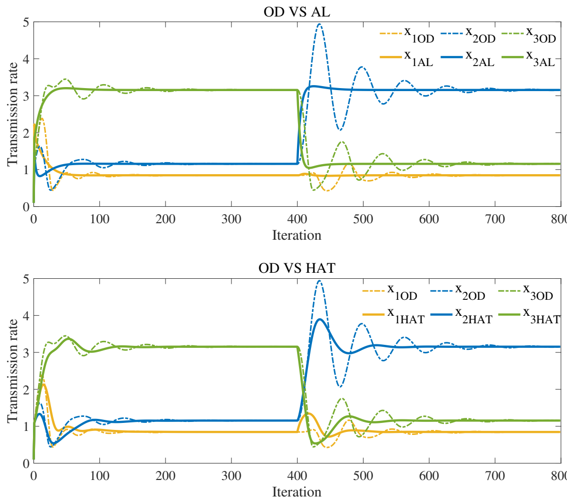


Fig. 2. Simulation results of Internet congestion control algorithms ("OD" stands for original dynamics; "AL" and "HAT" stand for dynamics via AL and hat-x).

Consider a simple 2-link network shared by 3 users as shown in Fig. 1. Initially, the capacities of links A and B are 2, 4 respectively and they change to 4, 2 after some time. Let $k_{x_i} = k_{p_l} = 1$ and utility function $U_i(x_i) = \log x_i$. The simulation results of the transmission rates for the three users are shown in Fig. 2. Both the original and the redesigned dynamics (AL and HAT) converge to their optimal values while the redesigned dynamics are faster.

### 5.2 Distributed PI Control for Single Integrator Dynamics

Problem (11) obtained via reverse-engineering is equal to

$$\min_{y\in\mathbb{R}^n} f = \frac{\rho_2}{2}y^T Ly + \frac{\delta}{2}y^T y - y^T d - \delta y^T y(0) \tag{31}$$
$$\text{s.t.} \quad \rho_1 \tilde{L}^T y = \mathbf{0}.$$

Since $\nabla^2 f = \rho_2 L + \delta I_n \succ 0$, $f \in \mathscr{S}_{L,u}$. Following the redesign procedure, after applying augmented Lagrangian and hat-x method, we obtain the same form of redesigned dynamics but with different $\Delta u(k)$ given by

$$y(k+1) = y(k) - \varepsilon_1\big(\rho_2 Ly(k) + \delta y(k) - d - \delta y(0) + \rho_1 \tilde{L}z(k)\big) + \Delta u(k)$$
$$z(k+1) = z(k) + \varepsilon_2 Dy(k).$$

For augmented Lagrangian method-based redesign,

$$\Delta u(k) = -\varepsilon_1\alpha\tilde{L}\tilde{L}^T y(k)$$

while for hat-x-based redesign

$$\Delta u(k) = -\varepsilon_1\alpha\big(y(k) - \hat{y}(k)\big)$$

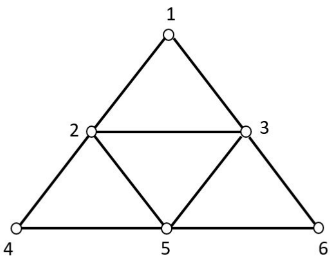


Fig. 3. A regular network of 6 agents.

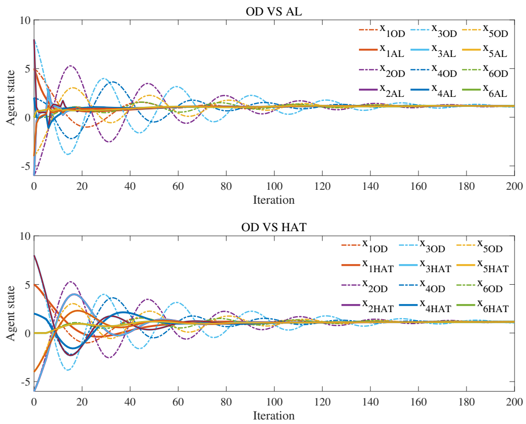


Fig. 4. Simulation results of distributed PI control ("OD" stands for original dynamics; "AL" and "HAT" stand for redesigned dynamics via AL and hat-x).

where $\hat{y}(k+1) = \hat{y}(k) + \varepsilon_1 \alpha\big(y(k) - \hat{y}(k)\big)$. With the extra dynamics added to the primal variable, the distributed PI controller could reach consensus faster.

Consider a network of 6 agents with its topology as shown Fig. 3. Let the constant disturbance $d = [0, 2, 0, 0, 0, 0]^T$, the initial condition $x(0) = [5, -6, 8, 2, -4, 0]^T$, the integral gain $\rho_1 = 10$, the static gain $\rho_2 = 0.5$, $\delta = 1$. Fig. 4 shows the simulation results of both the original dynamics and redesigned dynamics. They all converge to the optimal values while redesigned dynamics is faster via AL and smoother via hat-x.

## 6. CONCLUSION AND FUTURE WORK

This paper has proposed a control redesign approach to improve the performance of one certain class of dynamical systems. This approach is to firstly reverse-engineer a given dynamical system as a primal-dual method to solve a certain convex optimization problem. Then, by utilizing the augmented Lagrangian or hat-x method, the extra control term is obtained and added to the original control structure. Under this retrofit procedure, system performance could be improved, as demonstrated by both theoretical results and practical applications.

## REFERENCES

Alghunaim, S.A. and Sayed, A.H. (2019). Linear convergence of primal-dual gradient methods and their performance in distributed optimization. *arXiv e-prints arXiv:1904.0119*.

Andreasson, M., Dimarogonas, D.V., Sandberg, H., and Johansson, K.H. (2014). Distributed control of networked dynamical systems: Static feedback, integral action and consensus. *IEEE Transactions on Automatic Control*, 59(7), 1750–1764.

Anstreicher, Kurt M. and Wright, Margaret H. (2000). A note on the augmented Hessian when the reduced Hessian is semidefinite. *SIAM Journal on Optimization*, 11(1), 243–253.

Bertsekas, D.P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3), 334–334.

Chiang, M., Low, S.H., Calderbank, A.R., and Doyle, J.C. (2007). Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1), 255–312.

Deveci, O. and Kasnakoğlu, C. (2016). Performance improvement of a photovoltaic system using a controller redesign based on numerical modeling. *International Journal of Hydrogen Energy*, 41(29), 12634–12649.

Du, S.S. and Hu, W. (2019). Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. arXiv preprint arXiv:1802.01504.

Kakade, S., Shalev-Shwartz, S., and Tewari, A. (2009). On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, 2(1).

Kim, K.D. and Kumar, P.R. (2013). An overview and some challenges in cyber-physical systems. *Journal of the Indian Institute of Science*, 93(3), 341–352.

Lee, E.A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 363–369. IEEE.

Li, N., Zhao, C., and Chen, L. (2015). Connecting automatic generation control and economic dispatch from an optimization view. *IEEE Transactions on Control of Network Systems*, 3(3), 254–264.

Low, S.H. and Lapsley, D.E. (1999). Optimal flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6), 861–874.

Nešić, D. and Grüne, L. (2005). Lyapunov-based continuous-time nonlinear controller redesign for sampled-data implementation. *Automatica*, 41(7), 1143–1156.

Nesterov, Y. (2018). *Lectures on convex optimization*, volume 137. Springer.

Shu, H., Zhang, X., Li, N., and Papachristodoulou, A. (2020). Control Reconfiguration of Dynamical Systems for Improved Performance via Reverse-engineering and Forward-engineering. *arXiv e-prints*, arXiv:2003.09279.

Srikant, R. (2004). *The Mathematics of Internet Congestion Control*. Springer Science & Business Media.

Rajkumar, R., Lee, I., Sha, L., and Stankovic, J. (2010). Cyber-physical systems: the next computing revolution. In *Design Automation Conference*, 731–736. IEEE.

Rockafellar, R.T. (1970). *Convex analysis*, volume 28. Princeton university press.

Zhang, X. and Papachristodoulou, A. (2014). Improving the performance of network congestion control algorithms. *IEEE Transactions on Automatic Control*, 60(2), 522–527.

Zhang, X., Papachristodoulou, A., and Li, N. (2015). Distributed optimal steady-state control using reverse- and forward-engineering. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 5257–5264. IEEE.

Zhang, X., Papachristodoulou, A., and Li, N. (2018). Distributed control for reaching optimal steady state in network systems: An optimization approach. *IEEE Transactions on Automatic Control*, 63(3), 864–871.