



Conference Paper

Optimal Parameter Tuning of Feedback Controllers with Application to Biomolecular Antithetic Integral Control

Author(s):

Filo, Maurice; Khammash, Mustafa Hani

Publication Date:

2019

Permanent Link:

<https://doi.org/10.3929/ethz-b-000393857> →

Originally published in:

<http://doi.org/10.1109/CDC40024.2019.9029430> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Optimal Parameter Tuning of Feedback Controllers with Application to Biomolecular Antithetic Integral Control

Maurice Filo and Mustafa Khammash

Abstract—We consider the deterministic setting of a general nonlinear plant in feedback with a nonlinear controller that is parameterized by a finite number of unknown constants to be tuned. This setting is particularly useful in applications where the architecture of the feedback controller is constrained to have a specific structure, but the controller parameters can be tuned to optimize a given performance measure (e.g. biomolecular controllers, PID controllers, etc.). We first cast the tuning problem as a dynamically constrained optimization problem, then we convert the latter to an unconstrained one by introducing a suitable nonlinear operator. It is shown that the necessary conditions of optimality can be written as a parameter-dependent Two-Point Boundary Value Problem (TPBVP) that is difficult to solve analytically. Hence, we derive and compare two (first order) numerical methods to solve the optimization problem based on the Gradient Descent (GD) and the Conjugate Gradient Descent (CGD) algorithms. Finally, we apply the developed algorithms to tune a biomolecular antithetic integral controller. Tuning this controller has the advantages of shaping the dynamic response of the plant and minimizing the effect of dilution of the controller species.

I. INTRODUCTION

Feedback control has proven to be a very important tool on which numerous industrial applications rely. These applications span a broad range of engineering disciplines such as mechanical, electrical and chemical engineering. More recently, feedback control has found its way to molecular biology ([1], [2], [3] among others). In fact, feedback control mechanisms proved to be indispensable for engineering biomolecular systems that are capable of rejecting disturbances or achieving robust perfect adaptation [4]. Furthermore, it was shown that feedback control mechanisms exist naturally in the living cells and their role is to robustly regulate various cellular behaviors ([5], [6], [7]).

In general, to design and implement a suitable feedback controller that regulates a particular plant, two layers of challenges must be overcome. The first layer is to *design a suitable control law that respects a particular architecture* (i.e. design the functions g and κ in Figure 1). In many applications, one does not have the luxury of freely designing the structure (or architecture) of the feedback controllers. For example, one cannot implement a biomolecular controller that has the simple structure of a linear PID [4]. This is opposed to digital controllers where a PID implementation is trivial. Particularly, in biomolecular systems, feedback

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 766840 (COSY-BIO project).

Maurice Filo and Mustafa Khammash are with the Department of Biosystems Science and Engineering at ETH-Zürich.

maurice.filo@bsse.ethz.ch, mustafa.khammash@bsse.ethz.ch

controllers should respect the structure of chemical reaction networks [4]. The second challenge is to *make sure that the controller is physically implementable*. That is, find the suitable mechanical, electrical or biological parts that dynamically interact with each other according to the designed control law. The first challenge is purely mathematical and is typically overcome by resorting to theoretical tools such as control theory, chemical reaction network theory, etc. However, the second challenge is less mathematical but more practical, and it heavily depends on the application at hand.

In digital control systems, where the controller is implemented on a computer, the second challenge is practically nonexistent (assuming that the computational power is not a limitation). However, in biomolecular applications for example, both challenges are equally important. In these applications, the designed feedback controllers are limited to have a particular structure (g and κ in Figure 1) to (a) be biologically implementable and (b) guarantee certain robust properties such as disturbance rejection or robust perfect adaptation [4] (see Section VII). Yet, the controller may possess a set of tunable parameters (Θ in Figure 1). As a result, tuning these parameters is important to get the best out of that particular feedback control architecture and to shape the dynamic response of the plant. Parameter tuning is most common to PID controllers (eg. [8], [9] among others) and more general linear controllers with Iterative Feedback Tuning (IFT) [10]. Furthermore, tuning of nonlinear controllers using a data-driven gradient descent method is developed in [11].

In this paper, we cast the parameter tuning problem as a model-based optimization problem, and adopt an operator approach to derive the necessary conditions of optimality. Then we develop two first order numerical methods based on the Gradient Descent (GD) and Conjugate Gradient Descent (CGD) algorithms. We apply the GD and CGD algorithms to a nonlinear biomolecular controller called the Antithetic Integral Controller [3], and demonstrate that the CGD outperforms GD without adding a significant computational load.

The paper is organized as follows. We first introduce some useful notation that will be exploited throughout the paper. In Section III, we describe the problem statement where the tuning problem is cast as an optimization problem in a finite dimensional space. The necessary conditions of optimality are presented in Section IV leaving the detailed mathematical derivations to Section VI. Two numerical methods that solve the optimization problem are shown in Section V. Finally, before concluding, we apply the algorithms to a biomolecular controller in Section VII and discuss the biological implica-

tions of the tuning.

II. PRELIMINARIES AND NOTATION

This section is devoted to introduce some useful notation that will be adopted throughout the paper.

1) Function Space & Inner Product: Let $\mathbb{L}_n^2[0, T]$ denote the set of n -vector valued functions, whose entries are all real-valued, square-integrable functions over the finite time horizon $[0, T]$. Furthermore, let $\langle \cdot, \cdot \rangle$ (resp. $\langle \cdot, \cdot \rangle_{\mathbb{R}}$) denote the inner product in $\mathbb{L}^2[0, T]$ (resp. \mathbb{R}) with the appropriate dimension, i.e. if $x, y \in \mathbb{L}_n^2[0, T]$ and $v, w \in \mathbb{R}^n$ then

$$\langle x, y \rangle := \int_0^T x^T(t)y(t)dt \quad \text{and} \quad \langle v, w \rangle_{\mathbb{R}^n} := v^T w,$$

where x^T denotes the transpose of x .

2) Partial Derivatives and Jacobians: Suppose that $f(x, y)$ is a differentiable function in both vectors x and y . Then the partial derivatives of f with respect to x and y , evaluated at some trajectory (\bar{x}, \bar{y}) , is denoted by $\partial_x f(\bar{x}, \bar{y})$ and $\partial_y f(\bar{x}, \bar{y})$, respectively. If the index of ∂ is dropped, then the derivative is understood to be with respect to all the variables, i.e. $\partial f(\bar{x}, \bar{y}) := [\partial_x f(\bar{x}, \bar{y}) \quad \partial_y f(\bar{x}, \bar{y})]$.

3) Directional Derivatives and Adjoints: Suppose that $\mathcal{M} : \mathbb{R}^p \rightarrow \mathbb{L}_m^2[0, T]$ is an operator that acts on a constant vector $\Theta \in \mathbb{R}^p$ to yield a vector-valued function $y = \mathcal{M}(\Theta) \in \mathbb{L}_m^2[0, T]$. Then the directional derivative of \mathcal{M} evaluated at Θ is denoted by $\partial \mathcal{M}_{\Theta}$. In fact, the directional derivative $\partial \mathcal{M}_{\Theta} : \mathbb{R}^p \rightarrow \mathbb{L}_m^2[0, T]$ is a linear operator whose action on some $\tilde{\Theta} \in \mathbb{R}^p$ is defined as

$$\partial \mathcal{M}_{\Theta}(\tilde{\Theta}) := \lim_{\epsilon \rightarrow 0} \frac{\mathcal{M}(\Theta + \epsilon \tilde{\Theta}) - \mathcal{M}(\Theta)}{\epsilon}.$$

Furthermore, the adjoint of $\partial \mathcal{M}_{\Theta}$ is denoted by a $\partial \mathcal{M}_{\Theta}^* : \mathbb{L}_m^2[0, T] \rightarrow \mathbb{R}^p$ and is defined as

$$\langle y, \partial \mathcal{M}_{\Theta}(\tilde{\Theta}) \rangle = \langle \partial \mathcal{M}_{\Theta}^*(y), \tilde{\Theta} \rangle_{\mathbb{R}^p},$$

for all $y \in \mathbb{L}_m^2[0, T]$ and $\tilde{\Theta} \in \mathbb{R}^p$.

4) Gradients: Suppose that $\mathcal{J} : \mathbb{L}_m^2[0, T] \times \mathbb{R}^p \rightarrow \mathbb{R}$ is a (sufficiently smooth) nonlinear functional that takes $y \in \mathbb{L}_m^2[0, T]$ and $\Theta \in \mathbb{R}^p$ as inputs to yield a scalar real number $\mathcal{J}(y; \Theta)$. The partial directional derivatives of \mathcal{J} with respect to y and Θ , evaluated at $(\bar{y}; \bar{\Theta})$, are denoted by $\partial_y \mathcal{J}_{(\bar{y}; \bar{\Theta})} : \mathbb{L}_m^2[0, T] \rightarrow \mathbb{R}$ and $\partial_{\Theta} \mathcal{J}_{(\bar{y}; \bar{\Theta})} : \mathbb{R}^p \rightarrow \mathbb{R}$, respectively. These derivatives are, in fact, linear functionals whose actions can be expressed using the appropriate inner product, that is

$$\begin{aligned} \partial_y \mathcal{J}_{(\bar{y}; \bar{\Theta})}(\tilde{y}) &:= \lim_{\epsilon \rightarrow 0} \frac{\mathcal{J}(\bar{y} + \epsilon \tilde{y}; \bar{\Theta}) - \mathcal{J}(\bar{y}; \bar{\Theta})}{\epsilon} =: \langle \nabla_y \mathcal{J}_{(\bar{y}; \bar{\Theta})}, \tilde{y} \rangle \\ \partial_{\Theta} \mathcal{J}_{(\bar{y}; \bar{\Theta})}(\tilde{\Theta}) &:= \lim_{\epsilon \rightarrow 0} \frac{\mathcal{J}(\bar{y}; \bar{\Theta} + \epsilon \tilde{\Theta}) - \mathcal{J}(\bar{y}; \bar{\Theta})}{\epsilon} =: \langle \nabla_{\Theta} \mathcal{J}_{(\bar{y}; \bar{\Theta})}, \tilde{\Theta} \rangle_{\mathbb{R}^p}, \end{aligned}$$

where $\nabla_y \mathcal{J}_{(\bar{y}; \bar{\Theta})}$ and $\nabla_{\Theta} \mathcal{J}_{(\bar{y}; \bar{\Theta})}$ denote the gradients of \mathcal{J} at $(\bar{y}; \bar{\Theta})$ with respect to y and Θ , respectively.

III. PROBLEM STATEMENT

Consider the feedback setting depicted in the block diagram of Figure 1. Both, the plant and the feedback controller are represented as nonlinear dynamical systems described by a state differential equation and an output algebraic equation. We assume that the output y of the plant does not depend explicitly on the inputs u and w (i.e. no feedthrough term). We also assume that all of the functions f, h, g, κ and \mathcal{J} are sufficiently smooth. The main objective of this paper

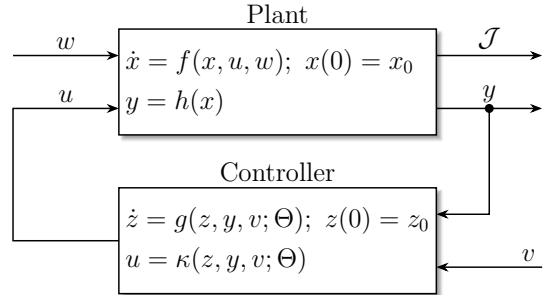


Fig. 1. Nonlinear Feedback Control System. The plant considered here is represented as a general nonlinear dynamical system whose inputs are denoted by $u \in \mathbb{R}^{n_u}$ and $w \in \mathbb{R}^{n_w}$, where w may represent an exogenous input to the plant and/or plant parameters (that is known). The plant's output, that is “sensed” by the controller, is denoted by $y \in \mathbb{R}^m$, and \mathcal{J} represents some performance measure (a cost functional). The plant is in feedback with a nonlinear controller whose architecture is assumed to be known and fixed; however, the controller is parameterized by a tunable constant vector $\Theta \in \mathbb{R}^p$. Furthermore, let $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^l$ denote the internal states of the plant and the controller, respectively. The objective of this paper is to tune Θ to optimize the performance measure \mathcal{J} . Note that $v \in \mathbb{R}^{n_v}$ may play the role of an exogenous input to the controller or a known fixed controller parameter (for example a reference set-point).

is to tune the controller parameters Θ to optimize some performance measure \mathcal{J} . More precisely, the goal is to solve the following optimization problem

$$\begin{aligned} &\underset{\Theta}{\text{minimize}} \quad \mathcal{J}(y; \Theta) \\ &\text{subject to} \quad \begin{cases} \dot{x} = f(x, u, w); & x(0) = x_0 \\ \dot{z} = g(z, y, v; \Theta); & z(0) = z_0 \\ u = \kappa(z, y, v; \Theta) \\ y = h(x), \end{cases} \end{aligned} \quad (1)$$

where the cost functional $\mathcal{J} : \mathbb{L}_m^2[0, T] \times \mathbb{R}^p \rightarrow \mathbb{R}$ is taken to be

$$\mathcal{J}(y; \Theta) = \frac{1}{2} \int_0^T (y(t) - y_r(t))^T Q(t) (y(t) - y_r(t)) dt + b(\Theta), \quad (2)$$

such that y_r is some desired output-tracking trajectory, $Q \in \mathbb{R}^{m \times m}$ is a symmetric matrix-valued signal, and $b : \mathbb{R}^p \rightarrow \mathbb{R}$ is a differentiable functional. Note that the first term in \mathcal{J} penalizes the output-tracking error, while the second term (if present) may represent a penalty term on Θ or a barrier function that replaces possible inequality constraints (see [12, chapter 17]) that arise due to feasibility constraints on Θ . The matrix-valued signal Q is an optimization design parameter that allows us to penalize particular performance measures related to the transient and/or steady-state response such as the rise-time, overshoot, steady-state error, etc.

The optimization problem presented in (1) is fundamentally different from the classical open-loop optimal control problem. In the latter, we search for optimal control laws in function space; whereas in (1) we search for a constant parameter in \mathbb{R}^p , after fixing a particular feedback control architecture.

IV. NECESSARY CONDITIONS OF OPTIMALITY

In this section, we present the necessary conditions of optimality of (1). The operator approach adopted in this paper is similar in spirit to that adopted in [13] and [14]. This approach has the advantage of making the optimization procedure conceptually transparent by hiding the details of the calculations. In fact, by introducing a new operator (we call it the parameter-to-output operator, see (5)) that maps the controller parameters Θ to the output signal y , we convert the constrained optimization problem given in (1) into an unconstrained one. As a result, one can easily write down the (abstract) necessary conditions of optimality and treat the detailed calculations separately.

Define the augmented (plant-controller) state variable χ , the augmented exogenous input variable ω , and the functions F and H as

$$\begin{aligned}\chi &= \begin{bmatrix} x \\ z \end{bmatrix}, \quad \chi_0 = \begin{bmatrix} x_0 \\ z_0 \end{bmatrix}, \quad \omega = \begin{bmatrix} w \\ v \end{bmatrix}, \quad H(\chi) = h(x), \\ F(\chi, \omega; \Theta) &= \begin{bmatrix} f(x, \kappa(z, h(x), v; \Theta), w) \\ g(z, h(x), v; \Theta) \end{bmatrix}.\end{aligned}\quad (3)$$

With the augmented variables and functions at hand, one can rewrite the optimization problem (1), more compactly, as

$$\begin{aligned}&\underset{\Theta}{\text{minimize}} \quad \mathcal{J}(y; \Theta) \\&\text{subject to} \quad \begin{cases} \dot{\chi} = F(\chi, \omega; \Theta); \quad \chi(0) = \chi_0 \\ y = H(\chi). \end{cases}\end{aligned}\quad (4)$$

This is a constrained optimization problem that can be converted to an unconstrained one by exploiting the parameter-to-output operator $\mathcal{M} : \mathbb{R}^p \rightarrow \mathbb{L}_m^2[0, T]$ defined as

$$y = \mathcal{M}(\Theta) \iff \begin{cases} \dot{\chi} = F(\chi, \omega; \Theta); \quad \chi(0) = \chi_0 \\ y = H(\chi). \end{cases}\quad (5)$$

This operator takes the parameter Θ as an input to produce the output y (for a given exogenous input ω). Hence, by substituting $y = \mathcal{M}(\Theta)$ in the cost functional $\mathcal{J}(y; \Theta)$, we arrive at an unconstrained optimization problem that is equivalent to (1) and (4)

$$\underset{\Theta}{\text{minimize}} \quad J(\Theta) := \mathcal{J}(\mathcal{M}(\Theta); \Theta).\quad (6)$$

This is an unconstrained optimization problem in \mathbb{R}^p , where the new cost functional $J : \mathbb{R}^p \rightarrow \mathbb{R}$ is now written abstractly in terms of Θ only. It is now straight forward to write down the *abstract* necessary condition of optimality which is obtained by simply setting the gradient of J to zero, i.e. $\nabla J_{\bar{\Theta}} = 0$. The gradient is calculated separately in Section VI and the result is shown in (15). In fact, by setting the gradient

in (15) to zero, we obtain the following necessary conditions of optimality

$$\begin{aligned}\dot{\xi} &= - \begin{bmatrix} \bar{B} \bar{C}^{\Theta} \\ \bar{B}^{\Theta} \end{bmatrix}^T \bar{\lambda}; \quad \bar{\xi}(0) = -\nabla b_{\bar{\Theta}}, \quad \bar{\xi}(T) = 0 \\ \dot{\bar{\lambda}} &= - \begin{bmatrix} \bar{A} + \bar{B} \bar{D}^c \bar{C} & \bar{B} \bar{C}^c \\ \bar{B}^c \bar{C} & \bar{A}^c \end{bmatrix}^T \bar{\lambda} - \begin{bmatrix} \bar{C}^T Q(\bar{y} - y_r) \\ 0 \end{bmatrix}; \quad \bar{\lambda}(T) = 0 \\ \dot{\bar{x}} &= f(\bar{x}, \bar{u}, w); \quad \bar{x}(0) = x_0 \\ \bar{y} &= h(\bar{x}) \\ \dot{\bar{z}} &= g(\bar{z}, \bar{y}, v; \bar{\Theta}); \quad \bar{z}(0) = z_0 \\ \bar{u} &= \kappa(\bar{z}, \bar{y}, v; \bar{\Theta}),\end{aligned}\quad (7)$$

where $\bar{A}, \bar{B}, \bar{C}, \bar{A}^c, \bar{B}^c, \bar{C}^c, \bar{D}^c, \bar{B}^{\Theta}$ and \bar{C}^{Θ} are time-varying matrices defined in (13), and $\nabla b_{\bar{\Theta}}$ is the gradient of the barrier function b evaluated at $\bar{\Theta}$. The set of equations in (7) represent a nonlinear two-point boundary value problem with an unknown parameter $\bar{\Theta}$ [15]. Therefore, solving $\nabla J_{\bar{\Theta}} = 0$ for the optimal parameter $\bar{\Theta}$ is equivalent to solving (7) for $\bar{\Theta}$. In general this is not analytically tractable, and one has to resort to numerical methods.

V. NUMERICAL METHODS

Equipped with the gradient of the cost functional J (15), one can easily devise a first order numerical method to solve the unconstrained optimization problem given in (6). Furthermore, the Hessian of J can also be calculated to devise a second order (Newton) method. In this paper, we only treat first order methods, namely Gradient Descent (GD) and Conjugate Gradient Descent (CGD), and leave second order methods for future work.

A general numerical method can be stated as follows: given the current approximation Θ_i of the optimal parameter $\bar{\Theta}$, a new approximation Θ_{i+1} is given by the following iterative equation

$$\Theta_{i+1} = \Theta_i + \alpha_i s_i, \quad (8)$$

where α_i and s_i are the step size and update direction at iteration i , respectively. The choice of the update direction s_i depends on the particular numerical method. For the GD method, s_i is chosen to be the negative of the gradient, evaluated at the current estimate Θ_i , i.e. $s_i := -\nabla J_{\Theta_i}$. For the CGD method, s_i is chosen as follows

$$s_i = \begin{cases} -\nabla J_{\Theta_i} & i = 0 \\ -\nabla J_{\Theta_i} + \frac{\|\nabla J_{\Theta_i}\|^2}{\|\nabla J_{\Theta_{i-1}}\|^2} s_{i-1} & i > 0, \end{cases}$$

where $\|\nabla J_{\Theta_i}\|^2 := \nabla J_{\Theta_i}^T \nabla J_{\Theta_i}$. As for the choice of the step size, theoretically one can pick the optimal step size α_i by solving the following scalar optimization problem at each iteration

$$\alpha_i = \underset{\alpha}{\operatorname{argmin}} \quad J(\Theta_i + \alpha s_i).$$

However, in practice, the step size α_i can be chosen to be some constant, or it can be chosen by using other rules. For the numerical results in this paper, the Armijo rule [16] is employed. Finally, our two proposed numerical methods

(GD and CGD) to solve the optimization problem (1) are given in Algorithm 1. The algorithm can be applied when the following criteria are available

- The closed loop dynamics: $f, g, h, \kappa, v, w, x_0$ and z_0 .
- A desired tracking output signal: $y_r(t)$.
- Penalty matrix and barrier function: $Q(t)$ and b .
- An initial guess Θ_0 .

The algorithm produces a local minimum $\bar{\Theta}$ of the cost functional $J(\Theta)$. The usual procedure to avoid local minima for this class of algorithms is to start from several different initial guesses Θ_0 .

Algorithm 1 (Conjugate) Gradient Descent Algorithm

- 1: Start with an initial guess $\Theta_0 \in \mathbb{R}^p$ and set $i = 0$.
- 2: Compute the gradient at Θ_i , ∇J_{Θ_i} :

(a) Simulate the closed-loop dynamics with $\Theta = \Theta_i$:

$$\begin{aligned}\dot{x}_i &= f(x_i, u_i, w); & x_i(0) &= x_0 \\ \dot{z}_i &= g(z_i, y_i, v; \Theta_i); & z_i(0) &= z_0 \\ u_i &= \kappa(z_i, y_i, v; \Theta_i) \\ y_i &= h(x_i).\end{aligned}$$

(b) Compute the time-varying Jacobians:

$$\begin{aligned}A_i &= \partial_x f_{(x_i, u_i, w)}, & B_i &= \partial_u f_{(x_i, u_i, w)} \\ C_i &= \partial h_{x_i}, & A_i^c &= \partial_z g_{(z_i, y_i, v; \Theta_i)} \\ B_i^c &= \partial_y g_{(z_i, y_i, v; \Theta_i)}, & C_i^c &= \partial_z \kappa_{(z_i, y_i, v; \Theta_i)} \\ D_i^c &= \partial_y \kappa_{(z_i, y_i, v; \Theta_i)} & B_i^\Theta &= \partial_\Theta g_{(z_i, y_i, v; \Theta_i)} \\ C_i^\Theta &= \partial_\Theta \kappa_{(z_i, y_i, v; \Theta_i)}.\end{aligned}$$

(c) Solve for $\lambda_i(t)$, with $\lambda_i(T) = 0$:

$$\dot{\lambda}_i = - \begin{bmatrix} A_i + B_i D_i^c C_i & B_i C_i^c \\ B_i^c C_i & A_i^c \end{bmatrix}^T \lambda_i - \begin{bmatrix} C_i^T Q \\ 0 \end{bmatrix} (y_i - y_r).$$

(d) Compute $\xi_i(0)$:

$$\dot{\xi}_i = - \begin{bmatrix} B_i C_i^\Theta \\ B_i^\Theta \end{bmatrix}^T \lambda_i; \quad \xi_i(T) = 0.$$

(e) $\nabla J_{\Theta_i} = \xi_i(0) + \nabla b_{\Theta_i}$.

- 3: Compute the update direction s_i :
- (a) For a Gradient Descent Method: $s_i = -\nabla J_{\Theta_i}$.
- (b) For a Conjugate Gradient Descent Method:

$$s_i = \begin{cases} -\nabla J_{\Theta_i} & i = 0 \\ -\nabla J_{\Theta_i} + \frac{\|\nabla J_{\Theta_i}\|^2}{\|\nabla J_{\Theta_{i-1}}\|^2} s_{i-1} & i > 0. \end{cases}$$

4: Pick a step size: $\alpha_i = \underset{\alpha}{\operatorname{argmin}} J(\Theta_i + \alpha s_i)$.

5: Update the estimate: $\Theta_{i+1} = \Theta_i + \alpha_i s_i$.

6: Set $i = i + 1$ and go back to step 2. Repeat until convergence.

\mathcal{M} are given in (2) and (5), respectively. Let $\partial J_{\bar{\Theta}}$ denote the directional derivative of J evaluated at $\bar{\Theta}$. Then $\partial J_{\bar{\Theta}} : \mathbb{R}^p \rightarrow \mathbb{R}$ is a linear functional whose action on some $\tilde{\Theta} \in \mathbb{R}^p$ can be calculated by invoking the chain rule

$$\partial J_{\bar{\Theta}}(\tilde{\Theta}) = \partial_y \mathcal{J}_{(\bar{y}; \bar{\Theta})} \left(\partial \mathcal{M}_{\bar{\Theta}}(\tilde{\Theta}) \right) + \partial_\Theta \mathcal{J}_{(\bar{y}; \bar{\Theta})}(\tilde{\Theta}), \quad (9)$$

where $\bar{y} := \mathcal{M}(\bar{\Theta})$ and $\partial \mathcal{M}_{\bar{\Theta}}$ is the directional derivative of \mathcal{M} evaluated at $\bar{\Theta}$ (refer to Section II-3). Let us now calculate the partial directional derivatives $\partial_y \mathcal{J}$ and $\partial_\Theta \mathcal{J}$.

Directional Derivatives of \mathcal{J} :

Observe that \mathcal{J} given in (2) can be equivalently rewritten in terms of the inner product as

$$\mathcal{J}(y; \Theta) = \frac{1}{2} \langle Q(y - y_r), y - y_r \rangle + b(\Theta), \quad (10)$$

where the inner product is defined in Section II-1. Then, the directional derivative of \mathcal{J} with respect to y can be calculated using the definition in Section II-4 to obtain

$$\begin{aligned}\partial_y \mathcal{J}_{(\bar{y}; \bar{\Theta})}(\tilde{y}) &= \lim_{\epsilon \rightarrow 0} \frac{1}{2\epsilon} \left\{ \left\langle Q(\bar{y} + \epsilon \tilde{y} - y_r), \bar{y} + \epsilon \tilde{y} - y_r \right\rangle \right. \\ &\quad \left. - \left\langle Q(\bar{y} - y_r), \bar{y} - y_r \right\rangle \right\} = \langle Q(\bar{y} - y_r), \tilde{y} \rangle,\end{aligned}$$

where the last equality is obtained by exploiting the symmetry of the matrix-valued signal Q and by neglecting second orders of ϵ in the limit as $\epsilon \rightarrow 0$. The directional derivative of \mathcal{J} with respect to Θ is written in terms of the gradient of the barrier function $\nabla b_{\bar{\Theta}}$ as

$$\partial_\Theta \mathcal{J}_{(\bar{y}; \bar{\Theta})}(\tilde{\Theta}) = \langle \nabla b_{\bar{\Theta}}, \tilde{\Theta} \rangle_{\mathbb{R}^p}.$$

By recalling that $\bar{y} := \mathcal{M}(\bar{\Theta})$ and substituting for $\partial_y \mathcal{J}$ and $\partial_\Theta \mathcal{J}$ in (9), we arrive at the following expression for the directional derivative of the unconstrained cost functional J

$$\partial J_{\bar{\Theta}}(\tilde{\Theta}) = \langle Q(\mathcal{M}(\bar{\Theta}) - y_r), \partial \mathcal{M}_{\bar{\Theta}}(\tilde{\Theta}) \rangle + \langle \nabla b_{\bar{\Theta}}, \tilde{\Theta} \rangle_{\mathbb{R}^p}.$$

Finally, to explicitly calculate the gradient $\nabla J_{\bar{\Theta}}$, we move the operator $\partial \mathcal{M}_{\bar{\Theta}}$ to the other side of the inner product by taking its adjoint

$$\begin{aligned}\partial J_{\bar{\Theta}}(\tilde{\Theta}) &= \left\langle \partial \mathcal{M}_{\bar{\Theta}}^* \left(Q(\mathcal{M}(\bar{\Theta}) - y_r) \right) + \nabla b_{\bar{\Theta}}, \tilde{\Theta} \right\rangle_{\mathbb{R}^p} \\ &=: \langle \nabla J_{\bar{\Theta}}, \tilde{\Theta} \rangle_{\mathbb{R}^p},\end{aligned} \quad (11)$$

where $\partial \mathcal{M}_{\bar{\Theta}}^*$ is the adjoint of $\partial \mathcal{M}_{\bar{\Theta}}$ defined in Section II-3, and $\nabla J_{\bar{\Theta}} := \partial \mathcal{M}_{\bar{\Theta}}^* \left(Q(\mathcal{M}(\bar{\Theta}) - y_r) \right) + \nabla b_{\bar{\Theta}}$ is a column vector that represents the gradient of J at $\bar{\Theta}$. The directional derivative $\partial \mathcal{M}_{\bar{\Theta}}$ and the adjoint $\partial \mathcal{M}_{\bar{\Theta}}^*$ are both linear operators whose actions can be calculated (the derivations are omitted in this paper for lack of space) as shown in Table I.

VI. MATHEMATICAL DERIVATION OF THE GRADIENT

The objective of this section is to calculate the gradient of the cost functional $J(\Theta) := \mathcal{J}(\mathcal{M}(\Theta); \Theta)$ where \mathcal{J} and

The gradient can now be calculated by exploiting the expression of the adjoint operator $\partial\mathcal{M}_{\bar{\Theta}}^*$ in Table I. Hence,

$$\nabla J_{\bar{\Theta}} = \bar{\xi}(0) + \nabla b_{\bar{\Theta}}$$

with $\begin{cases} \dot{\bar{\xi}} = -\partial_{\Theta} F_{(\bar{x}, v; \bar{\Theta})}^T \bar{\lambda}; & \bar{\xi}(T) = 0 \\ \dot{\bar{\lambda}} = -\partial_x F_{(\bar{x}, v; \bar{\Theta})}^T \bar{\lambda} - \partial H_{\bar{x}}^T Q(\bar{y} - y_r); & \bar{\lambda}(T) = 0 \end{cases}$

and $\begin{cases} \dot{\bar{x}} = F(\bar{x}, v; \bar{\Theta}); & \bar{x}(0) = x_0 \\ \bar{y} = H(\bar{x}). \end{cases}$

(12)

Finally, to write the gradient $\nabla J_{\bar{\Theta}}$ in terms of the original controller and state variables (x, z) , we express the Jacobians $\partial_x F, \partial_{\Theta} F$ and ∂H in terms of the various Jacobians of f, g, h and κ using (3). For notational convenience, define the following time-varying matrices

$$\begin{aligned} \bar{A} &= \partial_x f_{(\bar{x}, \bar{u}, w)}, & \bar{B} &= \partial_u f_{(\bar{x}, \bar{u}, w)}, & \bar{C} &= \partial_h \bar{x}, \\ \bar{A}^c &= \partial_z g_{(\bar{z}, \bar{y}, v; \bar{\Theta})}, & \bar{B}^c &= \partial_y g_{(\bar{z}, \bar{y}, v; \bar{\Theta})}, \\ \bar{C}^c &= \partial_z \kappa_{(\bar{z}, \bar{y}, v; \bar{\Theta})}, & \bar{D}^c &= \partial_y \kappa_{(\bar{z}, \bar{y}, v; \bar{\Theta})}, \\ \bar{B}^{\Theta} &= \partial_{\Theta} g_{(\bar{z}, \bar{y}, v; \bar{\Theta})}, & \bar{C}^{\Theta} &= \partial_{\Theta} \kappa_{(\bar{z}, \bar{y}, v; \bar{\Theta})}. \end{aligned}$$
(13)

Hence, by recalling that $\chi := [x^T \ z^T]^T$ and invoking the chain rule, we have

$$\begin{aligned} \partial_x F_{(\bar{x}, v; \bar{\Theta})} &= \begin{bmatrix} \bar{A} + \bar{B} \bar{D}^c \bar{C} & \bar{B} \bar{C}^c \\ \bar{B}^c \bar{C} & \bar{A}^c \end{bmatrix}, \\ \partial_{\Theta} F_{(\bar{x}, v; \bar{\Theta})} &= \begin{bmatrix} \bar{B} \bar{C}^{\Theta} \\ \bar{B}^{\Theta} \end{bmatrix}, \quad \partial H_{\bar{x}} = [\bar{C} \ 0], \end{aligned}$$
(14)

where $\bar{u} := \kappa(\bar{z}, \bar{y}, v; \bar{\Theta})$ and $\bar{y} := h(\bar{x})$. Therefore, by substituting (14) in (12), we arrive at the final expression for the gradient

$$\nabla J_{\bar{\Theta}} = \bar{\xi}(0) + \nabla b_{\bar{\Theta}}$$

with $\begin{cases} \dot{\bar{\xi}} = -\begin{bmatrix} \bar{B} \bar{C}^{\Theta} \\ \bar{B}^{\Theta} \end{bmatrix}^T \bar{\lambda} \\ \dot{\bar{\lambda}} = -\begin{bmatrix} \bar{A} + \bar{B} \bar{D}^c \bar{C} & \bar{B} \bar{C}^c \\ \bar{B}^c \bar{C} & \bar{A}^c \end{bmatrix}^T \bar{\lambda} - \begin{bmatrix} \bar{C}^T Q \\ 0 \end{bmatrix} (\bar{y} - y_r) \\ \bar{\xi}(T) = 0; \quad \bar{\lambda}(T) = 0 \end{cases}$

and $\begin{cases} \dot{\bar{x}} = f(\bar{x}, \bar{u}); & \bar{x}(0) = x_0 \\ \bar{y} = h(\bar{x}) \\ \dot{\bar{z}} = g(\bar{z}, \bar{y}, v; \bar{\Theta}); & \bar{z}(0) = z_0 \\ \bar{u} = \kappa(\bar{z}, \bar{y}, v; \bar{\Theta}), \end{cases}$

(15)

where all the time-varying matrices are given in (13). Equation (15) gives a recipe to compute the gradient at $\bar{\Theta}$.

VII. OPTIMAL TUNING OF THE ANTITHETIC INTEGRAL CONTROLLER

Recently, several biomolecular feedback controllers have been designed (e.g. [3], [17], [18], [19], [20]). Particularly, the Antithetic Integral Feedback Controller [3] overcomes both challenges discussed in Section I and succeeds, under certain assumptions, in achieving robust perfect adaptation.

That is, a plant species is regulated to have a zero steady-state error in the presence of plant uncertainties and disturbances. However, in the presence of controller species dilution ($\gamma \neq 0$ in Figure 2), the steady-state error increases [21], [22]. Furthermore, most of the biomolecular controllers developed so far only address steady-state behavior, although dynamic transient behavior is equally important (e.g. [21]). In this section, we consider the deterministic setting of the Antithetic Integral Feedback Controller introduced in [3]. We show that by optimally tuning the controller parameters (using Algorithm 1), we minimize the effect of controller species dilution and shape the transient response. We also compare the performance of the GD and CGD algorithms.

Consider the reaction network diagram depicted in Figure 2 which shows the antithetic integral controller sensing and actuating a particular plant taken here as an example. It is fairly straight forward to write down the differential equations using mass-action kinetics and show that the closed loop system fits the general setting of Figure 1 with

$$\begin{aligned} x &:= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & z &:= \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} & f(x, u) &:= \begin{bmatrix} u - \delta x_1 \\ kx_1 - \delta x_2 \end{bmatrix} & h(x) &:= x_2 \\ g(z, y; \Theta) &:= \begin{bmatrix} \mu - \eta z_1 z_2 - \gamma z_1 \\ \theta y - \eta z_1 z_2 - \gamma z_2 \end{bmatrix} & \kappa(z, y; \Theta) &:= \nu z_1, \end{aligned}$$
(16)

where Θ can be any combination of the controller parameters μ, θ, η and ν . Note that we set the exogenous inputs v and w to zero in this section. For simplicity, the cost functional considered here is given by

$$\mathcal{J} = \frac{1}{2} \int_0^T (y(t) - y_r(t))^2 dt,$$
(17)

where $y_r(t)$ is shown in Figure 2. Of course for the parameters to be biologically feasible, one should add inequality constraints to the optimization problem to force the parameters to be positive. In general one can design a barrier function $b(\Theta)$ to incorporate the inequality constraints in \mathcal{J} . However, in the following numerical examples, the optima are all positive, and therefore, there is no need to add inequality constraints. We consider three scenarios. In the first scenario, we assume that the dilution (or degradation) of the controller species Z_1 and Z_2 is negligible, i.e. $\gamma = 0$. In this case, it is easily verified that the controller has an integral feedback action that achieves robust perfect adaptation [3], [22]. The integral feedback action can be seen by using (16) to write $z_1 - z_2 = \mu - \theta y$. Then we have

$$z_1(t) - z_2(t) = \int_0^t (\mu - \theta y(\tau)) d\tau.$$

On the other hand, robust perfection adaptation can be seen by examining the steady state at which the time derivatives of z_1 and z_2 are set to zero. This implies that

$$\lim_{t \rightarrow \infty} y(t) = \frac{\mu}{\theta} =: y_{ss}.$$
(18)

This means that, as long as the closed-loop system is stable, the concentration of X_2 , denoted by $y = x_2$, goes to μ/θ at steady state no matter what the plant is. However, during the

Operator	Differential Equations
$\mathcal{M} : \mathbb{R}^p \rightarrow \mathbb{L}_m^2[0, T]$	$\bar{y} = \mathcal{M}(\bar{\Theta}) \iff \begin{cases} \dot{\bar{x}} = F(\bar{x}, \omega; \bar{\Theta}); & \bar{x}(0) = x_0 \\ \bar{y} = H(\bar{x}). \end{cases}$
$\partial \mathcal{M}_{\bar{\Theta}} : \mathbb{R}^p \rightarrow \mathbb{L}_m^2[0, T]$	$\bar{y} = \partial \mathcal{M}_{\bar{\Theta}}(\bar{\Theta}) \iff \begin{cases} \dot{\bar{x}} = \partial_x F_{(\bar{x}, \omega; \bar{\Theta})} \bar{x} + \partial_{\Theta} F_{(\bar{x}, \omega; \bar{\Theta})} \bar{\Theta}; & \bar{x}(0) = 0 \\ \bar{y} = \partial H_{\bar{x}} \bar{x} \end{cases}$
$\partial \mathcal{M}_{\bar{\Theta}}^* : \mathbb{L}_m^2[0, T] \rightarrow \mathbb{R}^p$	$\hat{\Theta} = \partial \mathcal{M}_{\bar{\Theta}}^*(\hat{y}) \iff \hat{\Theta} = \bar{\xi}(0) \text{ where } \begin{cases} \dot{\bar{\xi}} = -\partial_{\Theta} F_{(\bar{x}, \omega; \bar{\Theta})}^T \bar{\lambda}; & \bar{\xi}(T) = 0 \\ \dot{\bar{\lambda}} = -\partial_x F_{(\bar{x}, \omega; \bar{\Theta})}^T \bar{\lambda} - \partial H_{\bar{x}}^T \hat{y}; & \bar{\lambda}(T) = 0 \end{cases}$

TABLE I
THE PARAMETER-TO-OUTPUT OPERATOR \mathcal{M}

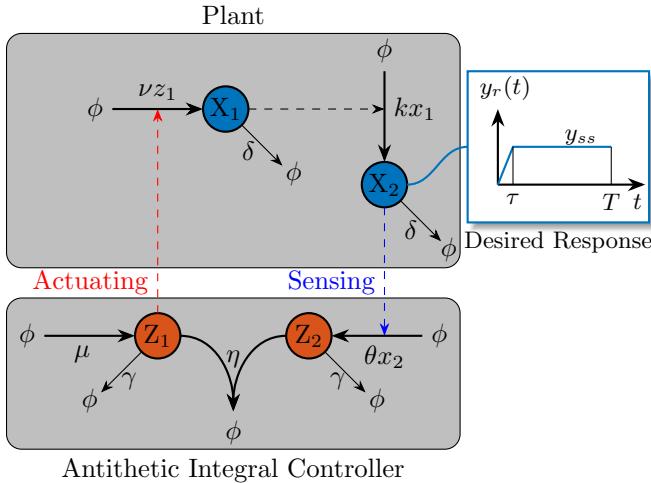


Fig. 2. Reaction network diagram of the antithetic integral controller interacting with a given plant. The output of the plant is the concentration of the species X_2 , i.e. $y(t) := x_2(t)$. The goal here is to tune the accessible parameters of the controller to obtain a dynamic response that is as close as possible to the desired reference signal $y_r(t)$ shown on the right. For all of the numerical experiments, we let $k = \delta = 1$, $\eta = 100$, $\tau = 5$, $T = 100$ and set all the initial conditions to zero.

transient phase several undesirable features may occur such as overshoots, oscillations, sluggish response, etc. In the first scenario, we fix the steady-state value y_{ss} to some desired constant and tune θ to shape the transient dynamics without affecting the steady state of $y = x_2$. More precisely, let the first scenario be denoted by S_1 such that

$$S_1 : \gamma = 0, \nu = 1, y_{ss} = 10, \mu = y_{ss}\theta, \Theta = \theta.$$

Observe from (18) that the steady state of y is fixed to y_{ss} for any value of θ . Therefore, tuning θ in S_1 guarantees that the controller (1) maintains robust perfect adaptation and (2) produces a desired transient response.

In the second scenario, we allow the controller species to have nonzero dilution rates. We also assume that we have access to tune θ only and all other parameters are fixed, i.e.

$$S_2 : \gamma = 0.5, \nu = 1, \mu = 10, \Theta = \theta.$$

Note that the nonzero dilution rate in scenario S_2 gives rise to a steady-state error [22]. The optimization problems for both scenarios S_1 and S_2 are only one dimensional, i.e. we tune only θ to minimize (17). This minimization penalizes both steady-state error and undesired transient response. In

one dimension, both GD and CGD algorithms are the same and thus only GD is used for S_1 and S_2 . The results for both scenarios are illustrated in Figure 3, where $\bar{\theta}$ denotes the optimal value of θ . A Brute-force search is carried out to illustrate the effectiveness of our optimization algorithm.

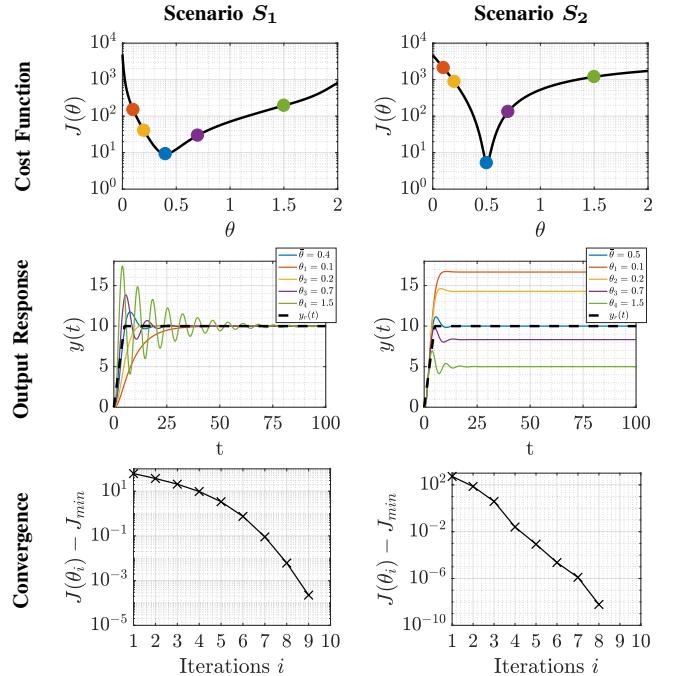


Fig. 3. Parameter Optimization under Scenarios S_1 and S_2 . The parameter spaces in both scenarios are one dimensional: only θ is tuned. In scenario S_1 , the dilution rate γ of the controller species is set to zero, unlike in scenario S_2 where $\gamma = 0.5$. Furthermore, in scenario S_1 , μ is automatically tuned with θ ($\mu = y_{ss}\theta$) to maintain the same steady-state value, unlike scenario S_2 where $\mu = 10$ and is fixed. A brute-force search was carried out over the range $[0, 2]$ for θ to plot the cost functional (17). Example values of θ are taken to show the corresponding output responses. In scenario S_1 , the optimal parameter is $\bar{\theta} = 0.4$. As θ deviates from $\bar{\theta}$, the response either becomes slow or oscillatory while maintaining the same steady-state value $y_{ss} = 10$. On the other hand, in scenario S_2 , the optimal parameter is $\bar{\theta} = 0.5$. As θ deviates from $\bar{\theta}$, the response exhibits a larger steady-state error. The bottom plots show the convergence of Algorithm 1 toward the optimal parameter in both scenarios. The convergence is achieved in less than ten iterations. This figure demonstrates that optimal tuning of the Antithetic Integral Controller is capable of shaping the transient dynamics and/or minimizing the effect of dilution of the controller species.

Finally, we consider a two dimensional optimization problem where we have access to tune both θ and ν , i.e.

$$S_3 : \gamma = 0.5, \mu = 10, \Theta = [\theta \nu]^T.$$

The results are illustrated in Figure 4. A brute-force search is also carried out here to illustrate the effectiveness of our algorithm. Clearly, CGD outperforms GD because the latter exhibits the “zigzag” effect resulting in slower convergence.

Figures 3 and 4 show that by optimally tuning the controller parameters, we achieve the desired output response. The advantage of scenario S_1 over S_2 and S_3 is that the controller is structurally robust to plant uncertainties. That is, the control architecture guarantees a zero steady-state error; whereas the parameter tuning shapes the transient response without affecting the steady state of the output. This is feasible when the controller species have zero or negligible degradation rate (e.g. in silico control [23]). We should note that Algorithm 1 can be applied to any number of parameters. In this paper, we limit the number of parameters to two so that the cost functionals can be visualized.

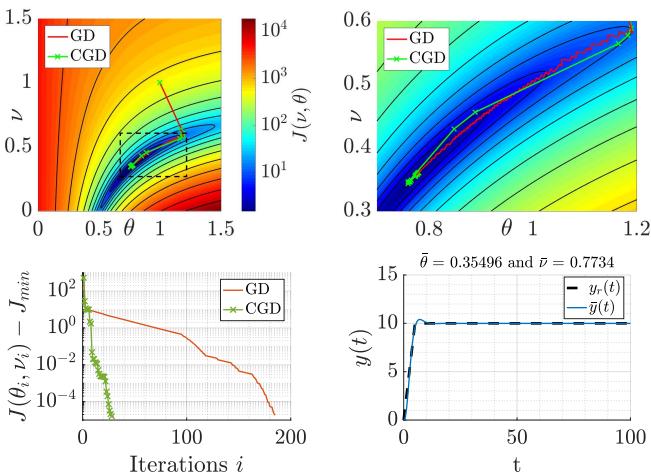


Fig. 4. Parameter Optimization under Scenario S_3 where the dilution rate of the controller species is nonzero. The parameter space is two dimensional: θ and ν are both tuned. A brute-force search was carried out over the range $(\theta, \nu) \in [0, 1.5] \times [0, 1.5]$ to plot the two dimensional cost functional (17). The two intensity plots on the top correspond to the same cost functional where the one on the right is zoomed in to demonstrate the undesirable “zig-zag” behavior of the Gradient Descent Algorithm. Both algorithms converge to the optimal parameters $\bar{\theta} = 0.36$ and $\bar{\nu} = 0.77$, but the GD algorithm converges much slower (≈ 180 iterations) as compared to the Conjugate GD algorithm (≈ 30 iterations) as illustrated by the convergence plot to the lower left. Finally, the optimal output trajectory is shown in the lower right plot which demonstrates that the desired response is achieved.

VIII. CONCLUSION

This paper develops two numerical methods to optimally tune feedback controllers, based on the GD and CGD algorithms. A nonlinear operator is introduced to cast the tuning problem as an abstract unconstrained optimization problem. This makes the derivations of the necessary conditions of optimality and the numerical methods conceptually transparent. The algorithm is tested on a biomolecular controller, and it is shown that optimally tuning the controller parameters can shape the dynamic and steady-state response of the output.

REFERENCES

- [1] D. Del Vecchio, A. J. Dy, and Y. Qian, “Control theory meets synthetic biology,” *Journal of The Royal Society Interface*, vol. 13, no. 120, p. 20160380, 2016.
- [2] V. Hsiao, A. Swaminathan, and R. M. Murray, “Control theory for synthetic biology: Recent advances in system characterization, control design, and controller implementation for synthetic biology,” *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 32–62, 2018.
- [3] C. Briat, A. Gupta, and M. Khammash, “Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks,” *Cell systems*, vol. 2, no. 1, pp. 15–26, 2016.
- [4] F. Xiao and J. C. Doyle, “Robust perfect adaptation in biomolecular reaction networks,” in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4345–4352, IEEE, 2018.
- [5] T.-M. Yi, Y. Huang, M. I. Simon, and J. Doyle, “Robust perfect adaptation in bacterial chemotaxis through integral feedback control,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 9, pp. 4649–4653, 2000.
- [6] D. Muzzey, C. A. Gómez-Uribe, J. T. Mettetal, and A. van Oudeaarden, “A systems-level analysis of perfect adaptation in yeast osmoregulation,” *Cell*, vol. 138, no. 1, pp. 160–171, 2009.
- [7] H. El-Samad, J. Goff, and M. Khammash, “Calcium homeostasis and parturient hypocalcemia: an integral feedback perspective,” *Journal of theoretical biology*, vol. 214, no. 1, pp. 17–29, 2002.
- [8] J.-X. Xu, D. Huang, and S. Pindi, “Optimal tuning of pid parameters using iterative learning approach,” *SICE Journal of Control, Measurement, and System Integration*, vol. 1, no. 2, pp. 143–154, 2008.
- [9] F.-S. Wang, W.-S. Juang, and C.-T. Chan, “Optimal tuning of pid controllers for single and cascade control loops,” *Chemical Engineering Communications*, vol. 132, no. 1, pp. 15–34, 1995.
- [10] H. Hjalmarsson, “Iterative feedback tuning—an overview,” *International journal of adaptive control and signal processing*, vol. 16, no. 5, pp. 373–395, 2002.
- [11] F. De Bruyne, B. D. Anderson, M. Gevers, and N. Linard, “Iterative controller optimization for nonlinear systems,” in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 4, pp. 3749–3754, IEEE, 1997.
- [12] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.
- [13] M. Filo and B. Bamieh, “Function space approach for gradient descent in optimal control,” in *2018 Annual American Control Conference (ACC)*, pp. 3447–3453, IEEE, 2018.
- [14] M. G. Filo, *Topics in Stochastic Stability, Optimal Control and Estimation Theory*. PhD thesis, University of California, Santa Barbara, 2018.
- [15] L. F. Shampine, J. Kierzenka, and M. W. Reichelt, “Solving boundary value problems for ordinary differential equations in matlab with bvp4c,” *Tutorial notes*, vol. 2000, pp. 1–27, 2000.
- [16] L. Armijo, “Minimization of functions having lipschitz continuous first partial derivatives,” *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.
- [17] C. Briat, C. Zechner, and M. Khammash, “Design of a synthetic integral feedback circuit: dynamic analysis and dna implementation,” *ACS synthetic biology*, vol. 5, no. 10, pp. 1108–1116, 2016.
- [18] C. C. Samaniego and E. Franco, “An ultrasensitive biomolecular network for robust feedback control,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10950–10956, 2017.
- [19] H.-H. Huang, Y. Qian, and D. Del Vecchio, “A quasi-integral controller for adaptation of genetic modules to variable ribosome demand,” *Nature communications*, vol. 9, no. 1, p. 5415, 2018.
- [20] D. K. Agrawal, R. Marshall, V. Noireaux, and E. D. Sontag, “In vitro implementation of robust gene regulation in a synthetic biomolecular integral controller,” *BioRxiv*, p. 525279, 2019.
- [21] N. Olsman, A.-A. Baetica, F. Xiao, Y. P. Leong, J. Doyle, and R. Murray, “Hard limits and performance tradeoffs in a class of sequestration feedback systems,” *BioRxiv*, p. 222042, 2018.
- [22] Y. Qian and D. Del Vecchio, “Realizing ‘integral control’ in living cells: how to overcome leaky integration due to dilution?,” *Journal of The Royal Society Interface*, vol. 15, no. 139, p. 20170902, 2018.
- [23] A. Milius-Argeitis, M. Rullan, S. K. Aoki, P. Buchmann, and M. Khammash, “Automated optogenetic feedback control for precise and robust regulation of gene expression and cell growth,” *Nature communications*, vol. 7, p. 12546, 2016.