MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

A Gradient-based Approach for Optimal Plant Controller Co-Design

Wang, Y.-S.; Wang, Y.

TR2015-061 July 2015

Abstract

This paper proposes a gradient-based iterative algorithm for optimal co-design of a linear physical plant and a controller. The proposed algorithm does not rely on the common linear parameterization assumption, and thus is applicable to a broader class of problems. The convergence of the algorithm and the verification procedure for a local minimum are given. Numerical examples show that our algorithm is comparable to other complicated algorithms in terms of the performance, but can deal with a more general class of problems.

2015 American Control Conference (ACC)

A Gradient-based Approach for Optimal Plant Controller Co-Design

Yuh-Shyang Wang and Yebin Wang

Abstract— **This paper proposes a gradient-based iterative algorithm for optimal co-design of a linear physical plant and a controller. The proposed algorithm does not rely on the common linear parameterization assumption, and thus is applicable to a broader class of problems. The convergence of the algorithm and the verification procedure for a local minimum are given. Numerical examples show that our algorithm is comparable to other complicated algorithms in terms of the performance, but can deal with a more general class of problems.**

I. Introduction

The process of designing a control system can be roughly divided into two tasks: plant design and controller synthesis. In conventional design process, these two tasks are carried out in a sequential manner, that is, design the plant first and then synthesize the controller for the designed plant model. This separation simplifies the design process at the expense of performance degradation. In fact, the co-design of the plant and the controller can improve the overall system performance and robustness compared to the sequential design procedure. The idea of co-design has been applied to a wide range of areas, including aerospace crafts [1], smart buildings [2], and electric motors [3].

A common way to approach the plant controller co-design problem is solving the optimal control problem and the optimal plant parameter selection problem alternately and iteratively. Although the iterative design procedure does simplify the original co-design problem, tremendous difficulty remains due to the non-convex nature of the optimal plant parameter selection problem. The non-convexity is present even if the plant is linearly dependent on its parameters. Several methods have been proposed to address the non-convex constraints. For instance, work [2], [4] convexify the non-convex problem by adding an extra system-equivalence constraint. Work [5] tries to relax the system-equivalence constraint by establishing a set of convex constraints with the hope to better approximate the original non-convex constraints. The convexification usually leads to a standard semi-definite programming (SDP) problem.

There are two primary limitations for these methods. First, the additional constraint usually leads to a reduced feasible set, and conservative design. Specifically, the plant parameter can converge to an optimal point in a convex subset, but not necessary a stationary point in the original set. Second, the aforementioned convexification-based approaches only work for some specific cases. For example, the SDP method in [5] assumes that the system matrices are affine in the plant parameters, which results in limited applicability. For instance, when transforming a linear second order mass-spring-damper system into a state space form, the system matrices depend on the plant parameter in a nonlinear way. The SDP method will not be applicable in this case unless re-parametrization.

Y.-S. Wang is with the department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA (yswang@caltech.edu). This work was done while Y.-S. Wang was an intern with Mitsubishi Electric Research Laboratories.

Y. Wang is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA (yebinwang@ieee.org).

In order to address the above-mentioned two issues, we propose a gradient-based algorithm which is more general and less conservative. The algorithm can handle nonlinear parametrization, and is applicable for both state and output feedback, continuous-time and discrete-time systems. Numerical examples suggest that our algorithm is less conservative than the system-equivalence-based method, and is comparable to the SDP method on the specialized problem.

The rest of this paper is organized as follows. In Section II, we formulate the optimal plant controller co-design problem as an extension of the optimal control problem. An iterative algorithm is proposed in Section III to approach the co-design problem. In Section IV, we analyze the convergence property of our algorithm, and extend the algorithm to structured controller tuning problems. Numerical examples are provided in Section V to illustrate the effectiveness of our method. Finally, Section VI ends with conclusion and offers some future research directions.

II. Preliminaries and Problem Formulation

In this section, we introduce the common formulation for $\mathcal{H}_2$ optimal control problem, and formulate the plant controller co-design problem. Then, we present the iterative design procedure where the co-design problem is decomposed into an optimal control problem and a plant parameter selection problem, solved alternately.

A. $\mathcal{H}_2$ Optimal Control Formulation

Consider a generalized continuous time linear time-invariant (LTI) system

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= C_1 x + D_{12} u \\ y &= C_2 x + D_{21} w \end{aligned} \qquad (1)$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $w \in \mathbb{R}^{n_w}$ the external disturbance, $u \in \mathbb{R}^{n_u}$ the control action, $z \in \mathbb{R}^{n_z}$ the regulated output, and $y \in \mathbb{R}^{n_y}$ the measurement. The system (1) can also be written in a compact form as

$$P = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & \mathbf{0} & D_{12} \\ C_2 & D_{21} & \mathbf{0} \end{array}\right] = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

where $P_{ij} = C_i(sI - A)^{-1}B_j + D_{ij}$. Let T_{wz} be the closed-loop transfer function from w to z. The goal of $\mathcal{H}_2$ optimal control is to find a dynamic output feedback control law $u = Ky$ such that the $\mathcal{H}_2$ norm of T_{wz} is minimized. Mathematically, the problem can be written as

$$\begin{array}{ll} \underset{K}{\text{minimize}} & ||T_{wz}(K)||_{\mathcal{H}_2} \\ \text{subject to} & K \text{ stabilizes } P \end{array} \tag{2}$$

in which $T_{wz}(K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}$. The problem (2) can be solved analytically, and the solution can be found in standard textbooks such as [6].

B. Optimal Plant Controller Co-design

We now consider a plant model as follows

$$P(\theta) = \left[\begin{array}{c|cc} A(\theta) & B_1(\theta) & B_2(\theta) \\ \hline C_1(\theta) & \mathbf{0} & D_{12}(\theta) \\ C_2(\theta) & D_{21}(\theta) & \mathbf{0} \end{array}\right]. \tag{3}$$

The plant (3) is parameterized by a vector $\theta \in \mathbb{R}^{n_\theta}$ that can be designed. We assume that all state space matrices in (3) are differentiable with respect to θ, and θ belongs to a convex compact set Θ. In particular, we emphasize the fact that the state space matrices in (3) can be any differentiable nonlinear function of θ. Without losing much generality, we assume $\Theta = \{\theta | \theta_{min} \leq \theta \leq \theta_{max}\}$. We also assume that the pair $[A(\theta), B_2(\theta)]$ is stabilizable and the pair $[A(\theta), C_2(\theta)]$ is detectable for all $\theta \in \Theta$.

The co-design objective is to determine an output feedback control law $u = Ky$ and the plant parameter θ *simultaneously* such that the $\mathcal{H}_2$ norm of the closed-loop transfer function T_{wz} is minimized. Specifically, the optimal co-design problem can be formulated as

$$\begin{array}{ll} \underset{\theta, K}{\text{minimize}} & ||T_{wz}(\theta, K)||_{\mathcal{H}_2} \\ \text{subject to} & \theta \in \Theta, \quad K \text{ stabilizes } P(\theta). \end{array} \tag{4}$$

Although problem (4) looks similar to (2), it is substantially more difficult to solve. Even when the state space matrices in (3) are affine in θ and the controller is state feedback, finding a stabilizing pair (K, θ) is known to be NP-hard [7].

We now give a common engineering example that can be formulated in the form of (4).

Example 1: Consider a second order mass-spring-damper system

$$\begin{aligned} M_s\ddot{x}_s + D_s\dot{x}_s + K_s x_s &= Bu \\ y &= Cx_s \end{aligned} \tag{5}$$

During the system design process, not only the control law $u = Ky$ but also the physical plant parameters (M_s, D_s, K_s) can be chosen within some range in order to meet the performance specification. One simple case is defining the system parameter vector θ by the collection of all entries of the matrices (M_s, D_s, K_s), that is, $\theta = [\{M_s\}_{ij}, \{D_s\}_{ij}, \{K_s\}_{ij}] \in \Theta$. This corresponds to the case that all the entries of (M_s, D_s, K_s) are independently tunable. Equation (5) can be written in state space form by introducing a new state vector $x = [x_s^\top \quad \dot{x}_s^\top]^\top$ as

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \mathbf{0} & I \\ -M_s^{-1}K_s & -M_s^{-1}D_s \end{bmatrix} x + \begin{bmatrix} \mathbf{0} \\ M_s^{-1}B \end{bmatrix} u \\ &= A(\theta)x + B_2(\theta)u \\ y &= \begin{bmatrix} C & \mathbf{0} \end{bmatrix} x \\ &= C_2 x. \end{aligned} \tag{6}$$

Equation (6) shows that the plant model is parameterized by the system parameter θ. Note that due to the inverse operation M_s^{-1} in (6), the matrices $A(\theta)$ and $B_2(\theta)$ usually depend on the parameter θ in a nonlinear way. The co-design of the mass-spring-damper system and the controller can then be formulated in the form of (4).

C. Iterative Design Procedure

A heuristic to deal with (4) is solving the optimal control problem and the optimal parameter selection problem alternately and iteratively. Specifically, we solve (2) for a fixed plant parameter θ to get a controller K, and then fix K and solve the plant parameter selection problem as

$$\begin{array}{ll} \underset{\theta}{\text{minimize}} & ||T_{wz}(\theta, K)||_{\mathcal{H}_2} \\ \text{subject to} & \theta \in \Theta, \quad K \text{ stabilizes } P(\theta). \end{array} \tag{7}$$

We solve (2) and (7) repetitively until the iteration converges.

As mentioned in Section I, (7) is non-convex even when the system matrices are θ-affine. The common strategy in both [4] and [5] is adding some conservative constraints to make the modified problem convex. Then a sub-optimal solution can be obtained by standard convex optimization technique. This strategy however suffers two primary limitations: conservative design, and limited range of applicability.

Our key observation is that there is no need to solve either (7) or its analogs exactly at each iteration. Instead, we propose to update a new plant parameter θ_{new} as long as it results in significant performance improvement. With this in mind, we develop a plant parameter update algorithm based on gradient computation to resolve the above-mentioned two limitations. Note that the gradient-based algorithm is applicable to a more general class of problem, as long as the system matrices are differentiable with respect to the plant parameter. Other methods using Hessian or its approximations, for instance Newton or Gaussian-Newton, can also be applied to update the plant parameter.

III. Co-design Algorithm

A. System Parameter Update

We begin with a given plant parameter θ and a fixed stabilizing controller K. Our goal is to find a θ_{new} such that the closed-loop system performance has a *sufficient improvement*. Denote the objective function as $f(\theta) = ||T_{wz}(\theta, K)||^2_{\mathcal{H}_2}$ for all θ. The plant parameter update is given by the rule

$$\theta_{new} = \mathbb{P}_\Theta(\theta + \alpha p) \tag{8}$$

where α is a positive scalar representing the step length, p the search direction, and $\mathbb{P}_\Theta(.)$ the projection operator on the set Θ. In particular, we use the steepest descent direction $p = -\nabla f(\theta)$ based on the gradient of the cost function. Unlike the usual line search methods for nonlinear optimization, we only perform the update (8) *once*, and then turn back to solve the optimal control problem (2) with the updated plant parameter θ_{new}.

We now discuss how to find an appropriate step length α with sufficient improvement. A common way to choose the step length α is by the Wolfe conditions [8] given by

$$\begin{aligned} f(\theta_{new}) &\leq f(\theta) + c_1\alpha\nabla f(\theta)^\top p \quad (9)\\ \nabla f(\theta_{new})^\top p &\geq c_2 \nabla f(\theta)^\top p \quad (10)\end{aligned}$$

for some $0 < c_1 < c_2 < 1$. Equation (9) ensures that the search gives an improvement when p is a descent direction. Note that (9) will be satisfied for any small α, in which case the improvement may be limited. Therefore, the curvature condition (10) is added to ensure that the chosen step length is not too conservative. In practice, we can use the backtracking technique to dispense the condition (10) [8]. The idea is to set α to a large value initially, and decrease the value until (9) is satisfied. When $\Theta = \{\theta | \theta_{min} \leq \theta \leq \theta_{max}\}$ and $p \neq 0$, there exists a finite $\bar{\alpha}$ such that $\mathbb{P}_\Theta(\theta + \alpha p) = \mathbb{P}_\Theta(\theta + \bar{\alpha} p)$ for all $\alpha \geq \bar{\alpha}$. The backtracking technique can start with this initial value. The complete system parameter update algorithm with backtracking is presented as follows.

Algorithm 1: System Parameter Update

```
Given θ, K, f(θ) = ||T_wz(θ,K)||²_H2 ;
Compute the gradient ∇f(θ) and set p = −∇f(θ) ;
if p = 0 then
    θ_new = θ ;
else
    Compute ᾱ and set α = ᾱ ;
    Choose ρ ∈ (0,1), c1 ∈ (0,1) ;
    Set θ_new = P_Θ(θ + αp) ;
    while f(θ_new) > f(θ) + c1 α ∇f(θ)^T p do
        α = ρα ;
        θ_new = P_Θ(θ + αp) ;
```

B. Gradient Computation

We now present an efficient way to compute the gradient $\nabla f(\theta)$ in Algorithm 1. We begin with a state space representation of the $\mathcal{H}_2$ optimal controller

$$K = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right]. \quad (11)$$

The closed loop transfer function T_{wz} can be written in state space form as

$$T_{wz}(\theta, K) = \left[\begin{array}{c|c} \mathcal{A}(\theta,K) & \mathcal{B}(\theta,K) \\ \hline \mathcal{C}(\theta,K) & \mathcal{D}(\theta,K) \end{array}\right] \quad (12)$$

where

$$\begin{aligned}\mathcal{A}(\theta,K) &= \begin{bmatrix} A(\theta) + B_2(\theta)D_KC_2(\theta) & B_2(\theta)C_K \\ B_KC_2(\theta) & A_K \end{bmatrix}\\ \mathcal{B}(\theta,K) &= \begin{bmatrix} B_1(\theta) + B_2(\theta)D_KD_{21}(\theta) \\ B_KD_{21}(\theta) \end{bmatrix}\\ \mathcal{C}(\theta,K) &= \begin{bmatrix} C_1(\theta) + D_{12}(\theta)D_KC_2(\theta) & D_{12}(\theta)C_K \end{bmatrix}\\ \mathcal{D}(\theta,K) &= D_{12}(\theta)D_KD_{21}(\theta).\end{aligned}$$

As K stabilizes $P(\theta)$, T_{wz} is always a stable transfer function and $\mathcal{A}$ in (12) is Hurwitz. Denote θ_i the i-th component of θ and $\hat{\theta}_i$ the unit vector in the direction of θ_i. The gradient can be computed as

$$\begin{aligned}\nabla f(\theta) &= \sum_{i=1}^{n_\theta} \frac{\partial < T_{wz}, T_{wz} >}{\partial \theta_i}\hat{\theta}_i \\ &= \sum_{i=1}^{n_\theta} 2 < T_{wz}, \frac{\partial T_{wz}}{\partial \theta_i} > \hat{\theta}_i. \quad (13)\end{aligned}$$

where $< .,. >$ is the inner product defined on the $\mathcal{H}_2$ space. The partial derivative in (13) can be calculated by

$$\begin{aligned}\frac{\partial T_{wz}}{\partial \theta_i} =& \frac{\partial \mathcal{C}(sI-\mathcal{A})^{-1}\mathcal{B} + \mathcal{D}}{\partial \theta_i}\\ =& \frac{\partial \mathcal{C}}{\partial \theta_i}(sI-\mathcal{A})^{-1}\mathcal{B} + \mathcal{C}(sI-\mathcal{A})^{-1}\frac{\partial \mathcal{B}}{\partial \theta_i} \quad (14)\\ &+ \frac{\partial \mathcal{D}}{\partial \theta_i} + \mathcal{C}(sI-\mathcal{A})^{-1}\frac{\partial \mathcal{A}}{\partial \theta_i}(sI-\mathcal{A})^{-1}\mathcal{B}\end{aligned}$$

where the last term in (14) is derived by considering the following identity [9]

$$\frac{\partial Y^{-1}}{\partial x} = -Y^{-1}\frac{\partial Y}{\partial x}Y^{-1}. \quad (15)$$

Note that each component in (14) can be computed from the state space realization of T_{wz} by simple state space manipulation. For instance, the state space representation for the last term of (14) is given by

$$\mathcal{C}(sI-\mathcal{A})^{-1}\frac{\partial \mathcal{A}}{\partial \theta_i}(sI-\mathcal{A})^{-1}\mathcal{B} = \left[\begin{array}{cc|c} \mathcal{A} & \frac{\partial \mathcal{A}}{\partial \theta_i} & \mathbf{0} \\ \mathbf{0} & \mathcal{A} & \mathcal{B} \\ \hline \mathcal{C} & \mathbf{0} & \mathbf{0} \end{array}\right]. \quad (16)$$

We now exemplify how to compute the gradient with a second order mass-spring-damper system.

Example 2: Consider the system (6). When $\mathcal{H}_2$ optimal controller is used, $D_K = 0$. The state space representation for T_{wz} can be simplified into

$$\left[\begin{array}{c|c} \mathcal{A} & \mathcal{B} \\ \hline \mathcal{C} & \mathcal{D} \end{array}\right] = \left[\begin{array}{cc|c} A(\theta) & B_2(\theta)C_K & B_1 \\ B_KC_2 & A_K & B_KD_{21} \\ \hline C_1 & D_{12}C_K & \mathbf{0} \end{array}\right].$$

Note that $\mathcal{A}$ is the only term that depends on θ, so the partial derivative $\frac{\partial T_{wz}}{\partial \theta_i}$ is just (16), in which case we only need to compute $\frac{\partial \mathcal{A}}{\partial \theta_i}$. This can be carried out by computing the partial derivative for A and B_2 in (6) directly. For example, we have

$$\frac{\partial B_2(\theta)}{\partial \theta_i} = \begin{bmatrix} \mathbf{0} \\ M_s^{-1}\frac{\partial M_s}{\partial \theta_i}M_s^{-1}B \end{bmatrix}$$

where the identity (15) is applied to M_s^{-1}. The partial derivative for A can be computed in a similar way.

The last thing to do is computing the inner product in (13). From (14) and the fact that $\mathcal{A}$ is Hurwitz, we know that $\frac{\partial T_{wz}}{\partial \theta_i}$ is stable. Assume that $\mathcal{D} = 0$, the inner product can be computed in the state space form by forming a Lyapunov-like equation, similar to the state space method to compute the $\mathcal{H}_2$ norm in [6]. Specifically, let

$$\frac{\partial T_{wz}}{\partial \theta_i} = \left[\begin{array}{c|c} \mathcal{A}_i & \mathcal{B}_i \\ \hline \mathcal{C}_i & \mathbf{0} \end{array} \right].$$

The inner product can be computed by

$$< T_{wz}, \frac{\partial T_{wz}}{\partial \theta_i} >= \text{trace}(\mathcal{C}L\mathcal{C}_i^\top)$$

where L is the solution of the equation

$$\mathcal{A}L + L\mathcal{A}_i^\top + \mathcal{B}\mathcal{B}_i^\top = 0. \tag{17}$$

Equation (17) can be solved algebraically by the technique of vectorization. When the plant and controller are discrete time systems, instead of (17), we solve a discrete time Lyapunov-like equation given by

$$\mathcal{A}L\mathcal{A}_i^\top - L + \mathcal{B}\mathcal{B}_i^\top = 0.$$

C. Iterative Co-design Algorithm

We now complete the iterative co-design algorithm by combining optimal control problem and the parameter update algorithm. From now on, we use $\theta^{(k)}$ to represent the vector of the system parameter in the k-th iteration, and $K^{(k)}$ the optimal controller for the system $P(\theta^{(k)})$. For a given pair $(\theta^{(k)}, K^{(k)})$, we can use Algorithm 1 to compute $\theta^{(k+1)} = \theta_{new}$ for a fixed $K^{(k)}$. Denote the objective function as $f(\theta, K) = ||T_{wz}(\theta, K)||^2_{\mathcal{H}_2}$. We assume the initial parameter $\theta^{(0)}$ is given and the initial cost $f(\theta^{(0)}, K^{(0)})$ is finite. The stopping criteria is based on the relative improvement ratio $[f(\theta^{(k)}, K^{(k)}) - f(\theta^{(k+1)}, K^{(k+1)})]/f(\theta^{(0)}, K^{(0)})$. For other purpose, the stopping criteria can also be based on other measure such as $||\theta^{(k)} - \theta^{(k+1)}||$. The proposed iterative co-design method is summarized in Algorithm 2.

Algorithm 2: Co-design Algorithm

Choose a threshold $\epsilon \in (0, 1)$;
Initialize $\theta^{(0)} = \theta_0$, $K^{(0)}$ the optimal controller for $P(\theta^{(0)})$;
Set $r = 1$, $k = 0$;
while $r > \epsilon$ **do**
  Calculate $\theta^{(k+1)}$ from $(\theta^{(k)}, K^{(k)})$ by Algorithm 1 ;
  Compute the optimal controller $K^{(k+1)}$ for $P(\theta^{(k+1)})$ by (2);
  Compute $r = \frac{f(\theta^{(k)}, K^{(k)}) - f(\theta^{(k+1)}, K^{(k+1)})}{f(\theta^{(0)}, K^{(0)})}$;
  Set $k = k + 1$;

IV. Analysis and Extension

In this section, we analyze the property of Algorithm 2. In particular, we show some convergence results for the algorithm. We then outline the procedure to check whether a stationary point is a local minimum. Finally, we explain how to use this algorithm to tune a structured controller.

A. Convergence Analysis

The following lemma states that the cost function is monotonically decreasing at each step in Algorithm 2.

Lemma 1: We have

$$0 \le f(\theta^{(k+1)}, K^{(k+1)}) \le f(\theta^{(k+1)}, K^{(k)}) \le f(\theta^{(k)}, K^{(k)}) \tag{18}$$

for all non-negative integer k.

Proof: There are totally three inequalities in (18). The first inequality comes from the fact that the objective (norm) is lower bounded by 0. The second inequality holds since $K^{(k+1)}$ is the optimal controller for $P(\theta^{(k+1)})$. The third inequality comes from the condition (9) on plant parameter update. ■

Next, we show that the cost sequence $\{f(\theta^{(k)}, K^{(k)})\}_{k=0}^{\infty}$ converges.

Lemma 2: The sequence $\{f(\theta^{(k)}, K^{(k)})\}_{k=0}^{\infty}$ converges to a value f^*.

Proof: From Lemma 1, $\{f(\theta^{(k)}, K^{(k)})\}_{k=0}^{\infty}$ is a monotonically decreasing sequence lower bounded by 0. Thus this sequence converges to some value f^*. ■

If the initial cost $f(\theta^{(0)}, K^{(0)})$ is nonzero, then Lemma 2 implies that the sequence of relative improvement ratio $\{[f(\theta^{(k)}, K^{(k)}) - f(\theta^{(k+1)}, K^{(k+1)})]/f(\theta^{(0)}, K^{(0)})\}_{k=0}^{\infty}$ converges to 0. We can then infer that Algorithm 2 terminate in a *finite* number of steps for any $\epsilon > 0$.

The sequence of the gradient have the following property.

Lemma 3: $\sum_{k=0}^{\infty} \alpha^{(k)} ||\nabla f(\theta^{(k)})||^2 < \infty$

Proof: Since $p^{(k)} = -\nabla f(\theta^{(k)})$, we can rearrange (9) and get

$$\begin{aligned} c_1 \alpha^{(k)} ||\nabla f(\theta^{(k)})||^2 &\le f(\theta^{(k)}, K^{(k)}) - f(\theta^{(k+1)}, K^{(k)}) \\ &\le f(\theta^{(k)}, K^{(k)}) - f(\theta^{(k+1)}, K^{(k+1)}). \end{aligned} \tag{19}$$

Summing over all positive integer k, the right-hand-side of (19) is equal to $f(\theta^{(0)}, K^{(0)}) - f^*$, which is finite. As c_1 is nonzero, we have $\sum_{k=0}^{\infty} \alpha^{(k)} ||\nabla f(\theta^{(k)})||^2 < \infty$. ■

Lastly, we consider the sequence of the plant parameter $\{\theta^{(k)}\}_{k=0}^{\infty}$.

Lemma 4: $\sum_{k=0}^{\infty} ||\theta^{(k+1)} - \theta^{(k)}||^2 < \infty$

Proof: We have

$$\begin{aligned} ||\theta^{(k+1)} - \theta^{(k)}||^2 &= ||P_\Theta(\theta^{(k)} + \alpha^{(k)} p^{(k)}) - \theta^{(k)}||^2 \\ &\le ||\theta^{(k)} + \alpha^{(k)} p^{(k)} - \theta^{(k)}||^2 \\ &= ||\alpha^{(k)} \nabla f(\theta^{(k)})||^2 \end{aligned} \tag{20}$$

where the inequality holds since P_Θ is a projection on the set $\Theta = \{\theta | \theta_{min} \le \theta \le \theta_{max}\}$. As the upper bound for all $\alpha^{(k)}$ is finite, we can prove this lemma by summing over all k on both sides of (20) and applying Lemma 3. ■

B. Second Order Condition

In this subsection, we assume the existence of a stationary point (θ^*, K^*) such that $\nabla f(\theta^*) = 0$ for a fixed K^*, where K^* is the optimal controller for $P(\theta^*)$. We now outline the procedure to check whether the stationary point is a local minimum.

To check whether $(\theta^{(*)}, K^{(*)})$ is a *co-design* local minimum, we need to treat the state space matrices of K^* as parameters as well. Specifically, we define the augmented parameter vector $\bar{\theta} = [\theta \quad \{A_K\}_{ij}\{B_K\}_{ij}\{C_K\}_{ij}\{D_K\}_{ij}]$. Then, we compute the Hessian of the cost function $\nabla^2_{\bar{\theta}} f(\theta^*, K^*)$. If the stationary point θ^* is on the boundary of the set Θ, we compute the bordered Hessian matrix for constrained optimization instead. If the Hessian or the bordered Hessian is a positive definite matrix, then this stationary point is a local minimum. The procedure of computing the Hessian is similar to the gradient computation in Section III-B.

C. Structured Controller Tuning

The parameter update algorithm can also be applied to structured controller tuning problems by treating the entries of (A_K, B_K, C_K, D_K) in (12) as plant parameters. Additionally, we can impose structured constraints on the state space matrices, including sparsity constraints or lower and upper bounds.

Consider the static output feedback control problem. This corresponds to the case that $A_K = 0$, $B_K = 0$, and $C_K = 0$. It is known that finding a stabilizing static output feedback controller for a fixed plant is NP-hard [10]. Assuming that the static output feedback control problem has a stabilizing solution, we can apply our algorithm to tune the stabilizing controller and improve the system performance by treating the entries of D_K as the plant parameter.

For a proportional-integral (PI) controller, we have $A_K = 0$. For a completely decentralized controller, we impose the appropriate sparsity patterns on (A_K, B_K, C_K, D_K). The tuning process is similar to that for the static output feedback.

V. Numerical Examples

Two numerical examples are performed to illustrate the effectiveness of our algorithm. The first one is a load-positioning system [11]. We show that our algorithm have comparable performance to the one in [5], which is developed specifically for state feedback with affine plant parameterization. The second experiment is a real engineering problem about chip mounter plant controller co-design.

A. Load-Positioning System

The dynamics of the load-positioning system are given in [11] and recited as

$$\begin{aligned} \ddot{x}_L &= (u - d_L \dot{x}_L)\left(\frac{1}{m_L} + \frac{1}{m_B}\right) + \frac{k_B}{m_B} x_B + \frac{d_B}{m_B} \dot{x}_B \\ \ddot{x}_B &= (d_L \dot{x}_L - u)\frac{1}{m_B} - \frac{k_B}{m_B} x_B - \frac{d_B}{m_B} \dot{x}_B \end{aligned} \tag{21}$$

where x_L is the relative displacement of the load with respect to the platform, x_B the displacement of the platform, and d_B, d_L, m_B, m_L, k_B are system parameters that can be designed. The co-design objective is to improve the track performance during load positioning tasks. Let y_d be the desired constant output. We define $x_1 = x_L - y_d$, $x_2 = \dot{x}_L + \dot{x}_B$, $x_3 = x_B$, $x_4 = \dot{x}_B$, and introduce the state vector $x = [x_1, x_2, x_3, x_4]^\top$. To make fair comparison, we take $d_L = 10$ and define the system parameter $\theta = [\frac{1}{m_L}, \frac{1}{m_B}, \frac{k_B}{m_B}, \frac{d_B}{m_B}]^\top$. The range of θ is given by the set $\Theta = \{\theta | \theta_{min} \leq \theta \leq \theta_{max}\}$ with $\theta_{min} = [0.3333, 0.04, 0.4, 0.004]^\top$ and $\theta_{max} = [1, 0.0667, 1.3333, 0.0667]^\top$. Choosing the same cost function as the one in [5], system (21) can be written in the form of (3) as

$$\left[\begin{array}{c|cc} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10(\theta_1+\theta_2) & \theta_3 & \theta_4 \\ 0 & 0 & 0 & 1 \\ 0 & 10\theta_2 & -\theta_3 & -\theta_4 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \theta_1+\theta_2 \\ 0 \\ -\theta_2 \end{bmatrix} \\ \hline \begin{bmatrix} \sqrt{1000} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{0} & \begin{bmatrix} 0 \\ \sqrt{0.1} \end{bmatrix} \\ I_4 & \mathbf{0} & \mathbf{0} \end{array}\right]$$

We choose $\rho = \frac{1}{2}$ and $c_1 = 10^{-4}$ in Algorithm 1, while these values do not affect the performance of our algorithm much. The threshold value ϵ in Algorithm 2 is chosen to be 10^{-4} and 10^{-8} for two different trials. The proposed algorithm is compared to the system-equivalence-based and the SDP methods in [2], [5]. We summarize the optimal point, total cost, and number of iterations of each method in Table I. It can be seen that the performance of our gradient algorithm is comparable to the SDP method, and is much better than the system-equivalence-based algorithm. In addition, our algorithm converges faster than the SDP one. In particular, when $\epsilon = 10^{-4}$, the algorithm converges in only 2 steps within a second. Finally, it should be noted that the SDP method is developed specifically for state feedback with affine plant parameterization, while our gradient algorithm can handle a much broader class of problems.

TABLE I
Comparison between Different Algorithms

	Original	S.E.-based	SDP	$\epsilon = 10^{-4}$	$\epsilon = 10^{-8}$
θ_1	0.5	0.6667	1	1	1
θ_2	0.05	0.0667	0.0667	0.0667	0.0667
θ_3	0.75	0.75	0.4393	0.4	0.4
θ_4	0.025	0.025	0.0369	0.004	0.009
Cost	215.3952	193.1239	169.5836	169.5756	169.5733
Iter.	-	-	24	2	8

B. Chip Mounter Machine

The proposed algorithm is applied to solve a chip mounter co-design problem as shown in Fig. 1. The control objective is to move the head and beam of the mounter to track a reference trajectory while keeping the vibration of the base small. Thus, we define the objective function as a quadratic function weighting tracking error, base vibration, and control effort. The system dynamics are given in the form of (5),

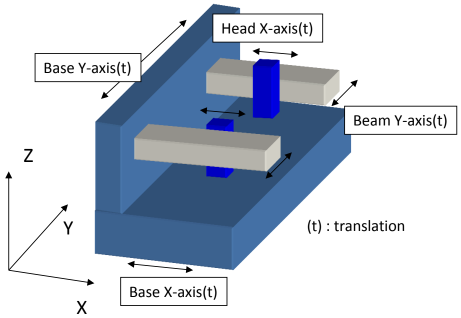

Fig. 1. Chip mounter

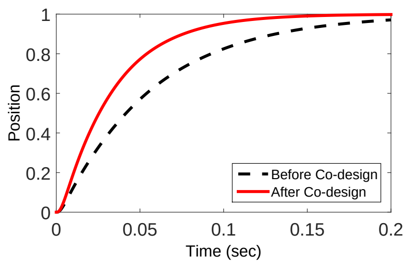

Fig. 2. Tracking performance with respect to a step signal.

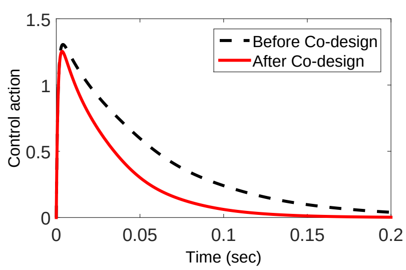

Fig. 3. Control action for step tracking.

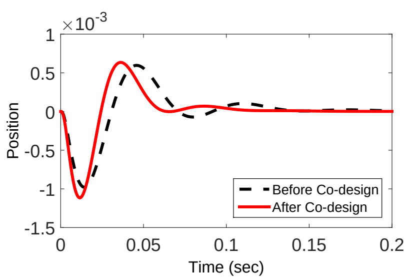

Fig. 4. Base vibration when performing step tracking.

which can be further rewritten in state space form as (6). The mass, damping ratio, coefficient of friction, and the stiffness of the system are free to adjust within $\pm 25\%$ from its nominal value. As in (6), the state space matrices $A(\theta)$ and $B_2(\theta)$ depend on the plant parameter in a nonlinear way. Details of the system model are proprietary information and thus omitted.

Figures 2-4 show the comparison results before and after co-design. It can be seen from Fig. 2 that the tracking performance after co-design improves remarkably. The settling time is shortened to almost half of the original one. In addition, as indicated in Fig. 3, the control effort to achieve the desired tracking performance is reduced as well. The maximum overshoot of the base vibration is however slightly increase after the co-design. This is understood as the natural outcome of the trade-off between tracking performance and base vibration suppression. Basically, the system tends to increase the base mass and stiffness when base vibration suppression is emphasized.

The degree of performance improvement and the parameter selection results highly depend on the relative penalty on tracking error, base vibration, and control effort. This means the cost function shall be carefully chosen so that the co-design yields meaningful results. Roughly speaking, we can treat plant parameter selection as passive control design. When the active control is expensive, we depend more on passive control design. So the plant parameter selection will have a significant effect on system performance. On the other hand, if active control is cheap, then there is not much difference before and after co-design.

VI. Conclusion

In this paper, we proposed a plant controller co-design algorithm for a general linear time invariant systems. The algorithm has a wide range of applications, including state and output feedback control designs, continuous time and discrete time systems, and with linear or nonlinear plant parameterizations. We shown some convergence properties of the algorithm, and outlined a test to check whether a stationary point is a local optimum. In addition to the generality of our algorithm, simulation also verified that our algorithm results in comparable system performance to the SDP-based method.

References

[1] A. L. Hale, W. Dahl, and J. Lisowski, "Optimal simultaneous structural and control design of maneuvering flexible spacecraft," *Journal of Guidance, Control, and Dynamics*, 1985.

[2] J. Lu and R. E. Skelton, "Integrating structure and control design to achieve mixed H_2/H_∞ performance," *International Journal of Control*, 2000.

[3] J. A. Reyer and P. Y. Papalambros, "Combined optimal design and control with application to an electric DC motor," *Journal of Mechanical Design*, vol. 124, no. 2, pp. 183–191, 2002.

[4] K. M. Grigoriadis, M. J. Carpenter, G. Zhu, and R. E. Skelton, "Optimal redesign of linear systems," in *1993 IEEE American Control Conference (ACC)*, 1993.

[5] Y. Jiang, Y. Wang, S. A. Bortoff, and Z.-P. Jiang, "An iterative approach to the optimal co-design of linear control systems," *International Journal of Control*, 2014, submitted.

[6] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice Hall New Jersey, 1996.

[7] O. Toker and H. Özbay, "On the np-hardness of solving bilinear matrix inequalities and simulataneous stabilization with static output feedback," in *1995 IEEE American Control Conference (ACC)*, Jun. 1995.

[8] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[9] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," 2012. [Online]. Available: http://www.mit.edu/~wingated/stuff_i_use/matrix_cookbook.pdf

[10] V. Blondel and J. N. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM J. Control Optim.*, vol. 35, no. 6, pp. 2118–2127, 1997.

[11] V. Shilpiekandula, S. A. Bortoff, J. C. Barnwell, and K. El-Rifai, "Load positioning in the presence of base vibrations," in *2012 IEEE American Control Conference (ACC)*, Jun. 2012.