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

Nonlinear Optimal Co-Design Based on A Modified Policy Iteration Method

Jiang, Y.; Wang, Y.; Bortoff, S.A.; Jiang, Z.-P.

TR2014-130 January 2015

Abstract

This brief studies the optimal codesign of nonlinear control systems: simultaneous design of physical plants and related optimal control policies. Nonlinearity of the optimal codesign problem could come from either a nonquadratic cost function or the plant. After formulating the optimal codesign into a nonconvex optimization problem, an iterative scheme is proposed in this brief by adding an additional step of system-equivalence-based policy improvement to the conventional policy iteration. We have proved rigorously that the closed-loop system performance can be improved after each step of the proposed policy iteration scheme, and the convergence to a suboptimal solution is guaranteed. It is also shown that under certain conditions, this additional policy improvement step can be conducted by solving a quadratic programming problem. The linear version of the proposed methodology is addressed in the context of linear quadratic regulator. Finally, the effectiveness of the proposed methodology is illustrated through the optimal codesign of a load-positioning system.

IEEE Transactions on Neural Networks and Learning Systems

Optimal Co-Design of Nonlinear Control Systems Based on A Modified Policy Iteration Method

Yu Jiang, Yebin Wang, Scott A. Bortoff, and Zhong-Ping Jiang

***Abstract*—This paper studies the optimal co-design of nonlinear control systems: simultaneous design of physical plants and related optimal control policies. Nonlinearity of the optimal co-design problem could come from either a non-quadratic cost function or the plant. After formulating the optimal co-design into a non-convex optimization problem, an iterative scheme is proposed in this paper by adding an additional step of system-equivalence-based policy improvement to the conventional policy iteration. We have proved rigorously that the closed-loop system performance can be improved after each step of the proposed policy iteration scheme, and the convergence to a suboptimal solution is guaranteed. It is also shown that under certain conditions, this additional policy improvement step can be conducted by solving a quadratic programming problem. The linear version of the proposed methodology is addressed in the context of LQR. Finally, the effectiveness of the proposed methodology is illustrated through the optimal co-design of a load-positioning system.**

***Index Terms*—Nonlinear system, Co-design, Optimal control, Policy iteration.**

I. Introduction

Conventional design of nonlinear control systems decouples the plant and the control design processes, i.e., the plant, also referred as the open-loop system, is given a priori while designing the control policy. Such decoupling is however not necessary due to the fact that both the plant and the control policy jointly affect the closed-loop system performance. Slight adjustments of the plant may result in remarkable improvements of the system performance, as well as robustness. Here, by "co-design", we refer to the simultaneous design of both the plant and the control policy to optimize certain prescribed performance objectives. Similar research work has been carried out under the names of "integrated structure and control design" [1], [2], "optimal redesign" [3], [4], and "simultaneous design" [5], [6], etc. The co-design problem can find a great number of engineering applications, such as aerospace crafts [5], [6], smart buildings [2], [4], and electromechanical devices [7].

In recent years, one popular way to deal with the co-design problem is to formulate it as a nonlinear optimization problem by parameterizing the plant and the control policy [3]. The resultant optimization problem for linear control systems is challenging due to the non-convexity [8]. When nonlinear system dynamics and non-quadratic cost functions are taken into consideration, there is less hope of solving the problem analytically. Indeed, even for fixed system parameters, finding the optimal control policy requires solving the well-known Hamilton-Jacobi-Bellman (HJB) equation of which a closed-form solution is hard to obtain in general cases.

The primary goal of this paper is to develop an iterative methodology for nonlinear co-design problems: co-design of nonlinear control systems. The main idea is to modify the conventional policy iteration technique [9], [10], which has been widely used in neural-network-based online controller designs; see [11]–[16], and references therein. By adding an extra step called *system-equivalence-based policy improvement* to redesign the control policy as well as system parameters simultaneously at each iteration step, we prove that the closed-loop system performance can be improved sequentially until the algorithm converges to a stationary point. We also show that, under certain conditions, the system-equivalence-based policy improvement can be cast to a quadratic programming problem. Inspired from the iterative techniques [2], [3], our novel method has two main advantages. First, it can be applied to a class of nonlinear systems with tunable system parameters. Second, in the linear quadratic setting, our approach has much less computational burden compared with methods in [2], [3].

The remainder of the paper is organized as follows. Section 2 formulates the nonlinear co-design problem. Section 3 presents the modified policy iteration scheme, and investigates when the system-equivalence-based policy improvement can be reduced to a quadratic optimization problem. Section 4 addresses the linear case of the proposed methodology. Application to a load-positioning system is illustrated in Section 5. Finally, concluding remarks are provided in Section 6.

II. Problem formulation

Consider a nonlinear control system

$$\dot{x} = f(x,p) + g(x,p)u \tag{1}$$

where $x \in \Omega \subset \mathbb{R}^n$ is the system state vector, Ω is a compact set containing the origin in its interior, $u \in \mathbb{R}^m$ is the control input, $p \in \mathcal{P} \subset \mathbb{R}^l$ is a vector of constant system parameters to be designed, $f : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ is a vector field satisfying $f(0,p) = 0$ for all $p \in \mathcal{P}$, and $g : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^{n\times m}$. All components of f and g are locally Lipschitz functions in x for each fixed p. The system parameter vector p has $\overline{p} \in \mathbb{R}^l$ and

This work was done while Y. Jiang was an intern with Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA. Y. Jiang is with the Engineering Development Group at The MathWorks, Inc, Natick, MA 01760, USA. yu.jiang@mathworks.com

Y. Wang and S. A. Bortoff are with Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA. yebinwang@ieee.org, bortoff@merl.com

Z.-P. Jiang is with Department of Electrical and Computer Engineering, Polytechnic School of Engineering, New York University, Brooklyn, NY 11201, USA. zjiang@nyu.edu The work of Z.-P. Jiang is supported in part by NSF grants ECCS-1101401 and ECCS-1230040.

$\underline{p} \in \mathbb{R}^l$ as its component-wise upper and lower bounds, i.e., the ith components of p is lower and upper bounded by the i-th components of $\underline{p}$ and $\overline{p}$, respectively. For simplicity of notation, we denote the feasible set of p as $\mathcal{P} = \{p \mid \underline{p} \leq p \leq \bar{p}\}$,

The cost associated with system (1) is defined as

$$J(p,u) = \int_0^\infty \left[Q(x) + u^T R u\right] dt, \quad x(0) = x_0 \in \Omega \tag{2}$$

where $Q(x)$ is a positive definite function on Ω, and $R = R^T$ is a positive definite matrix.

Definition 2.1: Consider system (1) and the cost functional (2). A feedback control policy $u(x)$ is *admissible* with respect to the parameter vector $p \in \mathcal{P}$, if

1) the closed-loop system (1) with p and $u(x)$ is asymptotically stable on Ω, and
2) the cost $J(p,u)$ is finite.

In Definition 2.1, the admissible control is assumed to be state feedback. This technical assumption makes the resultant co-design problem exposed to well-established theories including dynamics programming. Defining U_p as the set of all the admissible feedback control policies with respect to p, we assume that there exists a pair $(p_0, u_0(x))$ such that $u_0 \in U_{p_0}$. The nonlinear co-design problem can be formulated as follows.

Problem 2.1 (Nonlinear co-design problem): Given the system (1), find a pair $(p^*, u^*) \in \mathcal{P} \times U_{p^*}$ which minimizes the cost function (2), i.e.

$$(p^*, u^*) = \arg \min_{p \in \mathcal{P}, u \in U_p} J(p,u). \tag{3}$$

Remark 2.1: Concisely formulated, Problem 2.1 is extremely difficult to be solved for at least two reasons. First, this optimization problem is generally non-convex and the difficulty to solve a non-convex constrained optimization problem is well-understood. Second, nonlinearities involved in the problem make it almost impossible to find an analytic solution even for fixed p. To the best of our knowledge, there are no currently available tools for solving general nonlinear co-design problems.

The co-design process seeks an optimal or suboptimal solution which naturally resorts to an optimization problem. For the co-design problem, some work has been devoted to establish the existence and uniqueness of an optimal solution, and most of existing work assume the existence of optimal solutions and study the mathematical characterization of an optimal solution. Work [17] studies the existence of an optimal solution of a system design problem using the Weierstrass Theorem, which requires the compactness of the feasible set. Some researchers have been endeavoring in developing necessary conditions for local optimal solutions [18]–[21]. Nevertheless, their results can only be directly applied to some special cases. In terms of how to compute an optimal solution, one of the earliest studies of Problem 2.1 can be found in [22], where a gradient method was developed to numerically search for the optimal solution. The stability and convergence analysis of this method is however difficult to perform. This paper assumes the existence of optimal solutions and focuses on the iterative method that computes a suboptimal solution to Problem 2.1. In other words, we focus on algorithms of co-design process which take conventional designs as inputs and produce improved designs. The improved design outperforms the conventional design in terms of the system performance.

III. A MODIFIED POLICY ITERATION TECHNIQUE

In this section, we develop an iterative technique based on a modified policy iteration scheme to solve Problem 2.1. We will also provide a parametrization method which gives numerically a suboptimal solution to Problem 2.1.

A. A modified policy iteration algorithm

Suppose the initial vector of the system parameters is $p_0 \in \mathcal{P}$, and assume an associated admissible control policy u_0 is known. The proposed policy iteration can be summarized in the following three steps, with $i = 0, 1, \cdots$.

1) Policy evaluation

Solve for the positive definite function $V_i(x)$, from

$$\begin{aligned} 0 \;=\; & \nabla V_i^T \left[(f(x,p_i) + g(x,p_i)u_i(x)\right] \\ & + Q(x) + u_i^T(x) R u_i(x), \quad \forall x \in \Omega, \end{aligned} \tag{4}$$

where $\nabla V_i^T = \partial V_i / \partial x$.

2) Gradient-based policy improvement

Update the control policy by

$$\mu_i(x) = -\frac{1}{2} R^{-1} g^T(x,p_i) \nabla V_i(x), \quad \forall x \in \Omega. \tag{5}$$

3) System-equivalence-based policy improvement

Simultaneously update the system parameters to p_{i+1} and the control policy to u_{i+1} by solving the following optimization problem:

$$\min_{p_{i+1} \in \mathcal{P}, u_{i+1}} \quad \int_0^\infty u_{i+1}^T(x^{[i]}(t)) R u_{i+1}(x^{[i]}(t)) dt \tag{6}$$

$$s.t. \quad f_c(x, p_{i+1}, u_{i+1}) = f_c(x, p_i, \mu_i), \ \forall x \in \Omega \tag{7}$$

where

$$f_c(x,p,u) := f(x,p) + g(x,p)u(x), \ \forall x \in \Omega, \ \forall p \in \mathcal{P} \tag{8}$$

and $x^{[i]}$ is the solution of the system

$$\dot{x}^{[i]} = f(x^{[i]}, p_i) + g(x^{[i]}, p_i)\mu_i(x), \quad x^{[i]}(0) = x_0. \tag{9}$$

Remark 3.1: As a standard form of the policy iteration [23], (4) is used to solve a Lyapunov function V_i. As a system of first order nonlinear partial differential equations, the closed-form solution of (4) is difficult to establish. Instead, a good approximate solution is usually of practical interest. Given parameterizations of u_i and V_i, (4) is reduced to algebraic equations, and thus the approximate solution can be readily computed. The three steps (4)-(7) can be repeated until convergence is attained. In the absence of the system-equivalence-based policy improvement step, the algorithm is reduced to the conventional policy iteration [9], [10], [24].

The following theorem shows that the modified policy iteration algorithm can improve the performance of interest.

Theorem 3.1: Consider system (1) and its associated cost (2). Suppose $u_i \in U_{p_i}$, and a positive definite solution of (4) exists, for $i = 0, 1, \cdots$. Then, the following hold:

1) $\mu_i \in U_{p_i}$,
2) $u_{i+1} \in U_{p_{i+1}}$,
3) $J(p_{i+1}, u_{i+1}) \leq J(p_i, \mu_i) \leq J(p_i, u_i)$.

Proof: 1) First, we prove μ_i is stabilizing. Indeed, along the solutions of the system

$$\dot{\xi} = f(\xi, p_i) + g(\xi, p_i)\mu_i(\xi),\ \xi(0) = x \in \Omega \tag{10}$$

we have

$$\begin{aligned}\dot{V}_i &= \nabla V_i^T(\xi)[f(\xi,p_i) + g(\xi,p_i)(u_i(\xi) + \mu_i(\xi) - u_i(\xi))] \\ &= -Q(\xi) - \mu_i^T(\xi)R\mu_i(\xi) \\ &\quad -(\mu_i(\xi) - u_i(\xi))^T R(\mu_i(\xi) - u_i(\xi)) \leq -Q(\xi)\end{aligned}$$

Therefore, V_i is a Lyapunov function for the system (10), and μ_i is stabilizing.

Define $V_i^\mu : \Omega \to R_+$, with $V_i^\mu(0) = 0$, as the solution of

$$0 = (\nabla V_i^\mu)^T [(f(x,p_i) + g(x,p_i)\mu_i] + Q(x) + \mu_i^T R\mu_i. \tag{11}$$

Now, subtracting (4) from (11), similarly as in [10], we obtain

$$\begin{aligned}0 &= \left(\nabla V_i^\mu - \nabla V_i\right)^T [f(x,p_i) + g(x,p_i)\mu_i] \\ &\quad - \left(u_i - \mu_i\right)^T R \left(u_i - \mu_i\right).\end{aligned} \tag{12}$$

Hence, integrating both sides of the above equation along the solutions of system (9), it follows that

$$V_i^\mu(x_0) - V_i(x_0) = -\int_0^\infty (u_i - \mu_i)^T R(u_i - \mu_i)dt \leq 0.$$

As a result, μ_i is admissible with an improved cost compared with u_i. Hence, both 1) and the second inequality in 3) hold. Meanwhile, under the system equivalence condition (7), 2) holds.

Second, subtracting (11) from

$$\begin{aligned}0 &= \nabla V_{i+1}^T [(f(x,p_{i+1}) + g(x,p_{i+1})u_{i+1}(x)] \\ &\quad + Q(x) + u_{i+1}^T(x)Ru_{i+1}(x),\end{aligned} \tag{13}$$

and considering the equality constraint (7), we obtain

$$\begin{aligned}0 &= \left(\nabla V_{i+1} - \nabla V_i^\mu\right)^T f_c(x, p_{i+1}, u_{i+1}) \\ &\quad + u_{i+1}^T R u_{i+1} - \mu_i^T R\mu_i\end{aligned} \tag{14}$$

Integrating (14) along the trajectory of $x^{[i]}$, we have

$$\begin{aligned}&J(p_{i+1}, u_{i+1}) - J(p_i, \mu_i) \\ \leq\ & \int_0^\infty u_{i+1}^T(x^{[i]}(t))Ru_{i+1}(x^{[i]}(t))dt \\ & - \int_0^\infty \mu_i^T(x^{[i]}(t))R\mu_i(x^{[i]}(t))dt\end{aligned} \tag{15}$$

The right hand side of the inequality is apparently less or equal to zero by the definition of u_{i+1}. Hence, the first inequality in 3) is proved. The proof is thus complete. ■

Corollary 3.1: There exists a constant $J^* \geq 0$, such that $\lim_{i\to\infty} J(p_i, u_i) = J^*$.

Proof: By Theorem 3.1, we know the sequence $\{J(p_i, u_i)\}_{i=1}^\infty$ is monotonically decreasing. Also, the sequence is bounded from below because all its elements are non-negative. Therefore, the limit exists. ■

B. Parametrization and neural network approximation

Unfortunately, the proposed policy iteration method is still not directly applicable due to two obstacles. First, a linear partial differential equation needs to be solved in the the policy evaluation step. Second, in the system-equivalence-based policy improvement step, we are facing a nonlinear optimization problem of which the solution is non-trivial in general. To avoid these difficulties, we provide a practical implementation method by parameterizing the control policy.

To begin with, let $\{\phi_j(x)\}_{j=1}^N \in \mathbb{R}$ and $\{\psi_j(x)\}_{j=1}^q \in \mathbb{R}^m$ be two sets of linearly independent, continuously differentiable functions and vector fields, respectively. In addition, we assume that $\phi_j(0) = 0$, $\forall\ 1 \leq j \leq N$ and $\psi_j(0) = 0$, $\forall\ 1 \leq j \leq q$.

Assumption 3.1: Given a pair $(\hat{p}_i, \hat{u}_i)$ such that $\hat{u}_i \in U_{\hat{p}_i}$, and assume $\hat{u}_i(x) \in \text{span}\{\psi_1(x), \psi_1(x), \cdots, \psi_q(x)\}$. Then,

$$\begin{aligned}\hat{V}_i(x) &\in \text{span}\{\phi_1(x), \phi_1(x), \cdots, \phi_N(x)\}, \\ \hat{\mu}_i(x) &\in \text{span}\{\psi_1(x), \psi_1(x), \cdots, \psi_q(x)\}.\end{aligned}$$

where $\hat{V}_i(x)$ and $\hat{\mu}_i(x)$ are obtained from (4) and (5) with u_i replaced by $\hat{u}_i$.

Under Assumption 3.1, we can find three sets of weights $\{w_{i,1}, w_{i,2}, \cdots, w_{i,N}\}$, $\{c_{i,1}, c_{i,2}, \cdots, c_{i,q}\}$, and $\{c_{i,1}^\mu, c_{i,2}^\mu, \cdots, c_{i,q}^\mu\}$, such that $\hat{u}_i(x) = \sum_{j=1}^q c_{i,j}^\mu \psi_j(x)$, $\hat{V}_i(x) = \sum_{j=1}^N w_{i,j}\phi_j(x)$, $\hat{\mu}_i(x) = \sum_{j=q}^l c_{i,j}^\mu \psi_j(x)$.

Remark 3.2: If Assumption 3.1 is not satisfied, these weights can still be numerically obtained based on neural network approximation methods, such as the off-line approximation using Galerkin's method [25]. In addition, for uncertain nonlinear systems, these weights can be trained using approximate-dynamic-programming-based online learning methods [26], [15]. Notice that, when these approximation methods are used, Ω is required to be a compact set to guarantee the boundedness of the approximation error.

Assumption 3.2: There exist matrices of functions $E_j(x) \in \mathbb{R}^{n\times n}$ with $j = 0, 1, \cdots, l$, $A_i(x) \in \mathbb{R}^n$ with $j = 0, 1, \cdots, l$, and $B(x) \in \mathbb{R}^{n\times m}$, such that the following hold for $\forall x \in \mathbb{R}^n$ and $\forall \hat{p}_i \in \mathcal{P}$.

1) The following matrix is invertible

$$E(x, \hat{p}_i) := E_0(x) + \sum_{j=1}^l E_j(x)\hat{p}_{i,j}, \tag{16}$$

where $\hat{p}_{i,j}$ represents the jth component of the vector $\hat{p}$.
2) Matrices $E(x, p')$ and $E(x, p'')$ commute, for $\forall p', p'' \in \mathcal{P}$, i.e.,

$$E(x,p')E(x,p'') = E(x,p'')E(x,p'). \tag{17}$$

3) Functions $f(x, \hat{p}_i)$ and $g(x, \hat{p}_i)$ can be decomposed as

$$f(x, \hat{p}_i) = E^{-1}(x, \hat{p}_i)\left(A_0 + \sum_{j=1}^l A_j \hat{p}_{i,j}\right) \tag{18}$$

$$g(x, \hat{p}_i) = E^{-1}(x, \hat{p}_i)B \tag{19}$$

4) The following rank condition is satisfied

$$\text{rank}\,(\theta_{0,1}, \cdots, \theta_{l,l}, \zeta_{0,1}, \cdots, \zeta_{l,q}) < l + q \tag{20}$$

where $\theta_{i,j} = \text{vec}(E_i A_j), 0 \le i,j \le l$, and $\zeta_{i,j} = \text{vec}(E_i B \psi_j), 0 \le i,j \le q$.

Now, we are ready to replace the system-equivalence-based policy improvement by the following optimization problem.

Problem 3.1: Find the optimal coefficients $c_{i+1,j}$, $1 \le j \le q$ and the vector $\hat{p}_{i+1} \in \mathcal{P}$ from

$$\min_{c_{i+1,j}, 1\le j\le q} \quad \sum_{j=1}^{q}\sum_{k=1}^{q} c_{i+1,j}c_{i+1,k} \times \int_0^\infty \psi_j(x^{[i]})^T R \psi_k(x^{[i]}) dt \tag{21}$$

$$s.t. \quad \gamma_i = \sum_{j=1}^{l} \hat{p}_{i+1,j}\alpha_{i,j} + \sum_{j=1}^{q} c_{i+1,j}\beta_{i,j}, \tag{22}$$

$$\hat{p}_{i+1} \in \mathcal{P}, \tag{23}$$

where

$$\begin{aligned}
\alpha_{i,j} &= \theta_{j,0} - \theta_{0,j} + \hat{p}_{i,k}\sum_{k=1}^{l}\theta_{j,k} \\
&\quad + c^{\mu}_{i,k}\sum_{k=1}^{q}\zeta_{j,k} - \hat{p}_{i,k}\sum_{k=0}^{l}\theta_{k,j}, \\
\beta_{i,j} &= -\zeta_{0,j} - \hat{p}_{i,k}\sum_{k=1}^{l}\zeta_{k,j}, \\
\gamma_i &= \hat{p}_{i,k}\sum_{k=1}^{l}\theta_{k,0} - \hat{p}_{i,k}\sum_{k=1}^{l}\theta_{0,k} - c^{\mu}_{i,k}\sum_{k=1}^{q}\zeta_{0,k}.
\end{aligned}$$

Lemma 3.1: Under Assumptions 3.1 and 3.2, the following hold.

1) Let $c_{i+1,j}$ and $\hat{p}_{i+1}$ be the solution of Problem 3.1, and define the control policy.

$$\hat{u}_{i+1}(x) = \sum_{j=1}^{q} c_{i+1,j}\psi_j(x). \tag{24}$$

Then, the system equivalence condition (7) holds, i.e.,

$$f_c(x, \hat{p}_{i+1}, \hat{u}_{i+1}) = f_c(x, \hat{p}_i, \hat{\mu}_i). \tag{25}$$

2) Problem 3.1 is equivalent to a quadratic optimization problem with linear constraints.

Proof: 1) Under Assumptions 3.1 and 3.2, the equivalence condition (7) becomes

$$\begin{aligned}
& f(x,\hat{p}_{i+1}) + g(x,\hat{p}_{i+1})\hat{u}_{i+1}(x) \\
&= f(x,\hat{p}_i) + g(x,\hat{p}_i)\hat{\mu}_i(x) \\
\Leftrightarrow \quad & E(x,\hat{p}_i)\left(A_0 + \hat{p}_{i+1,j}\sum_{j=1}^{l}A_j + c_{i+1,j}\sum_{j=1}^{q}B_j\psi_{i+1}\right) \\
&= E(x,\hat{p}_{i+1})\left(A_0 + \hat{p}_{i,j}\sum_{j=1}^{l}A_j + c^{\mu}_{i,j}\sum_{j=1}^{q}B_j\psi_{i+1}\right) \\
\Leftrightarrow \quad & \gamma_i = \sum_{j=1}^{l}\hat{p}_{i+1,j}\alpha_{i,j} + \sum_{j=1}^{q}c_{i+1,j}\beta_{i,j}.
\end{aligned}$$

2) Under Assumption 3.2 4), define $\mathcal{V}$ as the null space of

$$\text{span}\left(\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,l}, \beta_{i,1}, \beta_{i,2}, \cdots, \beta_{i,q}, \gamma_i\right).$$

Then, $\mathcal{V}$ is a linear space with a non-zero dimension, and (7) is equivalent to

$$[\hat{p}_{i+1,1}, \hat{p}_{i+1,2}, \cdots, \hat{p}_{i+1,l}, c_{i+1,1}, c_{i+1,2}, \cdots, c_{i+1,q}, -1] \in \mathcal{V}$$

which is a linear equality constraint. ■

The following proposition summarizes the performance improvement. The proof is omitted here because it is nearly identical to the proof of Theorem 3.1.

Proposition 3.1: Under Assumptions 3.1 and 3.2,

1) $\hat{u}_{i+1}$ is admissible with respect to p_{i+1},
2) $J(\hat{p}_{i+1}, \hat{u}_{i+1}) \le J(\hat{p}_i, \hat{\mu}_i) \le J(\hat{p}_i, \hat{u}_i)$, and
3) there exists $\hat{J}^* \ge 0$, such that $\lim_{i\to\infty} J(\hat{p}_i, \hat{u}_i) = \hat{J}^*$.

IV. The LQR case: Specialization to linear systems

In this section, we study the co-design of a linear tim-invariant (LTI) control system where the control policy is chosen to be the linear quadratic regulator (LQR). The co-design of the LTI plant and the LQR is a special yet important case where the proposed methodology can be applied.

To this end, consider the LTI control system

$$E(p)\dot{x} = A(p)x + Bu \tag{26}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ the control input, $p \in \mathbb{R}^l \subset \mathcal{P}$ is the vector of system parameters. Given any $p' \in \mathcal{P}$, it is assumed that the matrix $E(p')$ is invertible and the constant pair of matrices $(A(p'), B)$ is stabilizable. In addition, we further assume that there exist constant matrices E_0, E_1, $\cdots$, E_q, and A_0, A_1, $\cdots$, A_q, such that

$$\begin{aligned}
E(p_i) &= E_0 + \sum_{j=1}^{q} E_j p_{i,j}, \\
A(p_i) &= A_0 + \sum_{j=1}^{q} A_j p_{i,j}, \quad \forall p_i \in \mathcal{P}.
\end{aligned}$$

The cost associated with (26) is defined as

$$J(p,u) = \int_0^\infty \left(x^T Q x + u^T R u\right) dt, \quad x(0) = x_0. \tag{27}$$

where $Q = Q^T \ge 0$, $R = R^T > 0$, and the pair $(A(p), Q^{1/2})$ is assumed detectable for any $p \in \mathcal{P}$.

In the LQR co-design problem, we seek a linear state-feedback control policy in the form of $u = Kx$, and a set of system parameters such that the cost (27) can be minimized. Therefore, the problem can also be formulated as:

Problem 4.1 (LQR co-design problem):

$$\min_{p_{i+1}\in\mathcal{P},\ K_{i+1}} J_q(p_{i+1}, K_{i+1}) = \int_0^\infty x_0^T e^{A_{i+1}^T t} \times \left(Q + K_{i+1}^T R K_{i+1}\right) \times e^{A_{i+1}t} x_0 dt \tag{28}$$

where, for $i = 0, 1, \cdots$,

$$A_i := E^{-1}(p_i)[A(p_i) + BK_i], \ \forall p_i \in \mathcal{P}, \ \forall K_i \in \mathbb{R}^{m\times n}$$

Again, Problem 4.1 is non-convex, and finding a global minimum is not practical in general cases. Here, we give the

LQR version of the policy iteration methodology proposed in the previous section.

Let $p_0 \in \mathcal{P}$ and assume $K_0 \in \mathbb{R}^{m\times n}$ is such that $E^{-1}(p_0)\left[A(p_0)+B(p_0)K_0\right]$ is Hurwitz. Then, conduct the following steps iteratively until convergence is attained.

1) Policy evaluation

Solve for $P_i = P_i^T > 0$ from the Lyapunov equation

$$0 = A_i^T P_i + P_i A_i + Q + K_i^T R K_i. \tag{29}$$

2) Gradient-based policy improvement

Update the feedback gain matrix

$$K_i^\mu = -R^{-1}B^T E^{-T}(p_i)P_i. \tag{30}$$

3) System-equivalence-based policy improvement

Simultaneously update the system parameters to p_{i+1} and the feedback gain matrix to K_{i+1}, by solving the following optimization problem:

$$\min_{p\in\mathcal{P},K_{i+1}} \quad \int_0^\infty x_0^T e^{A_i^{\mu T}t} K_{i+1}^T R K_{i+1} e^{A_i^\mu t} x_0 dt \tag{31}$$

$$s.t. \quad A(p_{i+1}) + BK_{i+1} = E(p_{i+1})A_i^\mu \tag{32}$$

where $A_i^\mu = E^{-1}(p_i)\left[A(p_i)+BK_i^\mu\right]$.

Remark 4.1: The optimization problem described in (31)-(32) is a quadratic programming problem. Also, similar assumptions as 2) 4) in Assumption 3.2 can be imposed to guarantee there are non-trivial feasible solutions to the problem other than (p_i, K_i^μ).

The following corollary is a direct result from Theorem 3.1.

Corollary 4.1: Consider system (1) and its associated cost (2). Suppose u_i is admissible with respect to $p_i \in \mathcal{P}$. Then, the following hold:

1) $A_{i+1} = A_i^\mu$ is Hurwitz,
2) $J_q(p_{i+1}, K_{i+1}) \le J_q(p_i, K_i^\mu) \le J_q(p_i, K_i)$, and
3) there exists $J_q^* \ge 0$, such that $\lim_{i\to\infty} J_q(p_i, K_i) = J_q^*$.

Remark 4.2: In the absence of Step 3) and with fixed system parameters, the two steps described by (29) and (30) become the algorithm derived in [27]. Clearly, these two steps involve much less computational burden compared with the co-design methods for linear systems proposed in [3], [2], in which a linear matrix inequality with fixed system parameters needs to be solved at each iteration step.

V. An application to a load-positioning system

Consider the load-positioning system

$$\begin{aligned}\ddot{x}_L &= (u - d_L\dot{x}_L)(\frac{1}{m_L}+\frac{1}{m_B}) + \frac{k_B}{m_B}x_B \\ &\quad + \frac{k_{Bn}}{m_B}x_B^3 + \frac{d_B}{m_B}\dot{x}_B \end{aligned}\tag{33}$$

$$\begin{aligned}\ddot{x}_B &= (d_L\dot{x}_L - u)\frac{1}{m_B} - \frac{k_B}{m_B}x_B \\ &\quad - \frac{k_{Bn}}{m_B}x_B^3 - \frac{d_B}{m_B}\dot{x}_B \end{aligned}\tag{34}$$

where x_L is the relative displacement of the load with respect to the platform, x_B is the absolute displacement of the platform, and d_L, m_B, m_L, k_B, k_{Bn} and d_B are constant system parameters. Notice that the system (33)-(34), with $k_{Bn} = 0$, is studied in [28].

The control objective is to track a step command. For this purpose, we define $x_1 = x_L - y_d$, with y_d the desired constant output, $x_2 = \dot{x}_L + \dot{x}_B$, $x_3 = x_B$, and $x_4 = \dot{x}_B$. Then, the system is converted to

$$\dot{x}_1 = x_2 \tag{35}$$

$$\begin{aligned}\dot{x}_2 &= -\left(\frac{1}{m_L}+\frac{1}{m_B}\right)d_L x_2 + \frac{k_B}{m_B}x_3 \\ &\quad + \frac{k_{Bn}}{m_B}x_3^3 + \frac{d_B}{m_B}x_4 + \left(\frac{1}{m_L}+\frac{1}{m_B}\right)u \end{aligned}\tag{36}$$

$$\dot{x}_3 = x_4 \tag{37}$$

$$\begin{aligned}\dot{x}_2 &= \frac{d_L}{m_B}x_2 - \frac{k_B}{m_B}x_3 - \frac{k_{Bn}}{m_B}x_3^3 \\ &\quad - \frac{d_B}{m_B}x_4 - \frac{1}{m_B}u \end{aligned}\tag{38}$$

The cost to be minimizes is chosen as

$$J(p,u) = \int_0^\infty \left(1000x_1^2 + x_2^2 + x_3^2 + x_4^2 + u^2\right)dt \tag{39}$$

where $p = [m_L, m_B, d_L, k_B, d_B]^T$, and their bounds are shown in the second and the third columns in Table I.

To perform the policy evaluation and the gradient policy improvement, $\{\phi_j\}_{j=1}^{20}$ are selected to be second and fourth order polynomials of x. Also, $\{\psi_j\}_{j=1}^{20} = \{x_1,x_2,x_3,x_4,x_1^3,x_2^3,x_3^3,x_4^3,x_1x_2^2,x_1x_3^2,x_1x_4^2,x_2x_1^2,x_2x_3^2,x_2x_4^2, x_3x_1^2,x_3x_2^2,x_3x_4^2, \ x_2x_1^2,x_2x_1^2,x_2x_4^2\}$. The initial control policy is set to be $u_0 = -x_1$. The constant weights are computed using Galerkin approximation [25], with $\Omega = \{x||x_1| \le 1.5, |x_2| \le 1.5, |x_3| \le 1.5, |x_4| \le 2\}$. From (32), 24 equality constraints with 25 variables can be derived.

To make a fair comparison between the conventional policy iteration without co-design and the modified policy iteration with co-design, steps (4)-(5) of these two cases are implemented exactly the same. For the modified policy iteration case, the additional quadratic programming problem at the system-equivalence-based policy improvement step is solved by invoking the MATLAB function *quadprog* with medium-scale option on. Simulation is performed on a desktop with Windows 7 and an Intel Core 2@3.0GHz processor, and the results are summarized by Figures 1-3. From Figure 1, we see that the modified policy iteration algorithm converges within three iterations and gives a system cost $J = 409.5055$. As a comparison, the conventional policy iteration algorithm converges within six iterations and gives a system cost $J = 501.62$. Clearly, by applying the proposed co-design technique, the system performance has been improved by 27.085%. As for the computation time, the conventional and modified policy iterations take 7.66sec and 10.88sec, respectively to run ten iterations. In Figure 2, it can be found that a shorter settling time can be achieved while less control energy is required after co-design, compared with conventional optimal control design. Finally, value functions obtain with and without co-design are compared on the (x_1, x_3) plane and are shown in Figure 3.

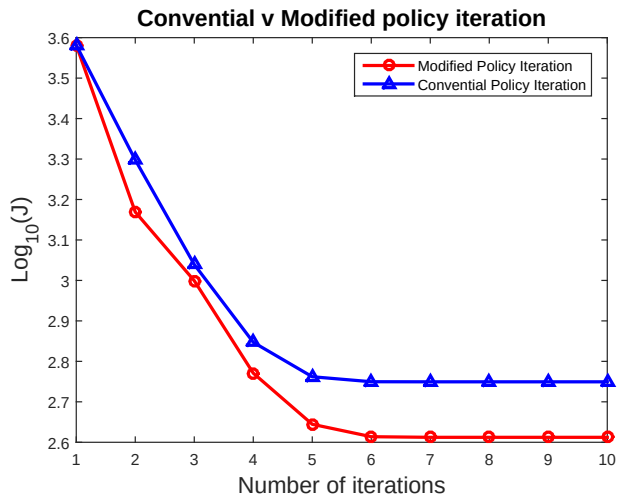

Fig. 1. Illustration of the convergence property of the modified policy iteration technique.

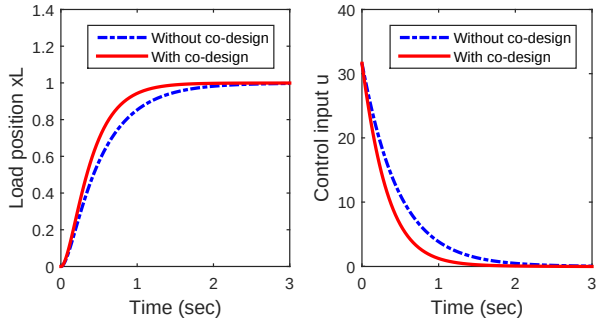

Fig. 2. Tracking performance to a step command.

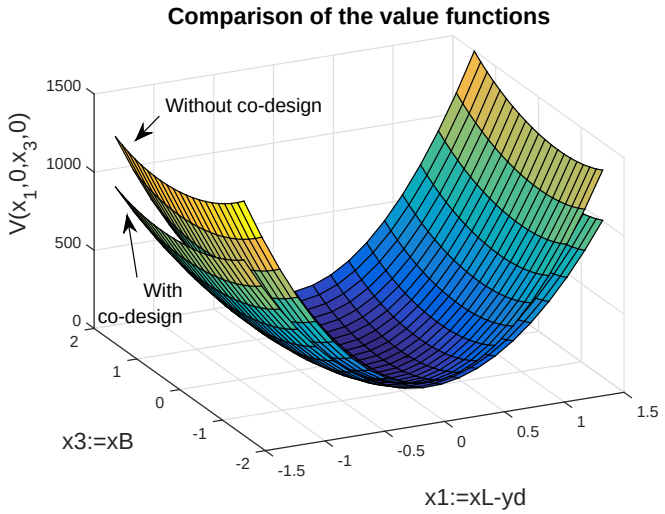

Fig. 3. Comparison of the value functions.

TABLE I
SYSTEM PARAMETERS

Variable	Min value	Max value	Initial value	Optimized value
m_L	1	3	2	1.13
m_B	15	25	20	15
d_L	10	20	15	10
k_B	10	20	15	11.25
d_B	0.1	1	0.5	0.375

VI. CONCLUSIONS

In this paper, a novel iterative technique has been proposed to solve the nonlinear co-design problem. The key idea is to modify conventional policy iteration by adding an additional step of system-equivalence-based policy improvement, and rigorous mathematical proofs were given. The proposed co-design methodology has been illustrated through the application to a load-positioning system. It will be interesting to extend the proposed method for the co-design of dynamic systems with static uncertainties via adaptive dynamics programming [29]–[37] and robust adaptive dynamic programming [26], [38]–[40].

REFERENCES

[1] M. M. da Silva, O. Brüls, W. Desmet, and H. Van Brussel, "Integrated structure and control design for mechatronic systems with configuration-dependent dynamics," *Mechatronics*, vol. 19, no. 6, pp. 1016–1025, 2009.

[2] J. Lu and R. E. Skelton, "Integrating structure and control design to achieve mixed H_2/H_∞ performance," *International Journal of Control*, vol. 73, no. 16, pp. 1449–1462, 2000.

[3] K. M. Grigoriadis, M. J. Carpenter, G. Zhu, and R. E. Skelton, "Optimal redesign of linear systems," in *Proceedings of the American Control Conference*, San Francisco, CA, 1993, pp. 2680–2684.

[4] R. E. Skelton and J. H. Kim, "The optimal mix of structure redesign and active dynamic controllers," in *Proceedings of the American Control Conference*, Chicago, IL, 1992, pp. 2775–2779.

[5] A. L. Hale, W. Dahl, and J. Lisowski, "Optimal simultaneous structural and control design of maneuvering flexible spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 1, pp. 86–93, 1985.

[6] A. Messac, "Control-structure integrated design with closed-form design metrics using physical programming," *AIAA Journal*, vol. 36, no. 5, pp. 855–864, 1998.

[7] 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.

[8] 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.

[9] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press, 1960.

[10] G. N. Saridis and C.-S. G. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 3, pp. 152–159, 1979.

[11] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Neurodynamic programming and zero-sum games for constrained control systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1243–1252, 2008.

[12] Z. Chen and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 90–106, 2008.

[13] T. Cheng, F. L. Lewis, and M. Abu-Khalaf, "Fixed-final-time-constrained optimal control of nonlinear systems using neural network hjb approach," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1725–1736, 2007.

[14] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 631–647, 2007.

[15] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.

[16] X. Xu, D. Hu, and X. Lu, "Kernel-based least squares policy iteration for reinforcement learning," *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 973–992, 2007.

[17] P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design*. UK: Cambridge University Press, 2000.

[18] S. F. Alyaqout, P. Y. Papalambros, and A. G. Ulsoy, "Coupling in design and robust control optimization," in *Proceedings of the European Control Conference*, Kos, Greece, 2007.

[19] H. K. Fathy, J. A. Reyer, P. Y. Papalambros, and G. Ulsoy, "On the coupling between the plant and the controller optimization problems," in *Proceedings of the American Control Conferences*, Arlington, VA, 2001, pp. 1864–1869.

[20] R. Patil, Z. Filipi, and H. Fathy, "Computationally efficient combined design and control optimization using a coupling measure," *ASME Journal of Mechanical Design*, vol. 134, no. 7, p. 071008, 2012.

[21] D. L. Peters, P. Y. Papalambros, and A. G. Ulsoy, "Control proxy functions for sequential design and control optimization," *Journal of Mechanical Design*, vol. 133, p. 091007, 2011.

[22] M. Salama, J. Garba, L. Demsetz, and F. Udwadia, "Simultaneous optimization of controlled structures," *Computational Mechanics*, vol. 3, no. 4, pp. 275–282, 1988.

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, 1995.

[24] R. J. Leake and R.-W. Liu, "Construction of suboptimal control sequences," *SIAM Journal on Control*, vol. 5, no. 1, pp. 54–63, 1967.

[25] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[26] Z.-P. Jiang and Y. Jiang, "Robust adaptive dynamic programming for linear and nonlinear systems: An overview," *European Journal of Control*, vol. 19, no. 5, pp. 417–425, 2013.

[27] D. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 114–115, 1968.

[28] V. Shilpiekandula, S. A. Bortoff, J. C. Barnwell, and K. El-Rifai, "Load positioning in the presence of base vibrations," in *Proceedings of the American Control Conference*, Montreal, Canada, 2012, pp. 6282–6287.

[29] D. Liu, H. Li, and D. Wang, "Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 8, pp. 1015–1027, Aug 2014.

[30] D. Liu, D. Wang, and H. Li, "Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 418–428, Feb 2014.

[31] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 621–634, March 2014.

[32] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, pp. 628–634, July 2012.

[33] Q. Wei, D. Liu, and G. Shi, "A novel dual iterative Q-learning method for optimal battery management in smart residential environments," *IEEE Transactions on Industrial Electronics, article in press*, 2014.

[34] Q. Wei, F. Wang, D. Liu, and X. Yang, "Finite-approximation-error-based discrete-time iterative adaptive dynamic programming," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2820–2833, Dec 2014.

[35] Q. Wei and D. Liu, "Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6399–6408, Nov 2014.

[36] ——, "A novel iterative $theta$-adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1176–1190, Oct 2014.

[37] ——, "Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1020–1036, Oct 2014.

[38] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624 – 2632, 2014.

[39] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming with an application to power systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, pp. 1150–1156, 2013.

[40] ——, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 882–893, May 2014.