Data-driven Dynamic Multi-objective Optimal Control: An Aspiration-satisfying Reinforcement Learning Approach

Majid Mazouchi, Member, IEEE, Yongliang Yang, Member, IEEE, and Hamidreza Modares, Senior Member, IEEE

Abstract—This paper presents an iterative data-driven algorithm for solving dynamic multi-objective (MO) optimal control problems arising in control of nonlinear continuous-time systems. It is first shown that the Hamiltonian functional corresponding to each objective can be leveraged to compare the performance of admissible policies. Hamiltonian-inequalities are then used for which their satisfaction guarantees satisfying the objectives’ aspirations. An aspiration-satisfying dynamic optimization framework is then presented to optimize the main objective while satisfying the aspiration of other objectives. Relation to satisficing (good enough) decision-making framework is shown. An infinite-dimensional linear program (LP) algorithm is developed to solve the formulated aspiration-satisfying MO optimization. To obviate the requirement of complete knowledge of the system dynamics, a data-driven satisficing reinforcement learning approach that uses measured data and a value function formulation with a span of a finite family of basis functions to approximate the solution to the infinite-dimensional LP problem is presented. Finally, two simulation examples are provided to show the effectiveness of the proposed algorithm.

Index Terms—Multi-objective optimization, Reinforcement learning, Satisficing control, Sum-of-squares program.

I. INTRODUCTION

In most of the real-world control applications such as autonomous vehicles, the system designer must account for multiple objectives (such as safety, control effort, transient performance, comfort, etc.) to evaluate candidate control policies. However, since there usually exist conflicts between objectives and the objectives’ preferences might change over time, a control policy is best realized by finding an appropriate context-dependent trade-off among objectives. A multi-objective (MO) optimal control framework that trades-off among objectives and could explicitly account for objectives’ aspirations must be devised to deal with this issue.

While MO optimization has been widely utilized to find a diverse set of efficient solutions (see for example [16], [5], [19] and [20]) there are at least three challenges in control of dynamical systems with multiple objectives that are not well addressed. First, most of the existing MO optimization frameworks assume that the objective functions to be optimized are static. In the control engineering systems, however, several objectives must be optimized over a horizon [14], [18] and performing a sequence of static optimization results in myopic short-sighted decisions that do not possess the capability of proactively responding to uncertainties. Second, to successfully operate in a changing and uncertain environment, systems such as self-driving cars must learn multiple potential solutions for different situational objectives and apply autonomously the one with the appropriate trade-off as the situation becomes apparent. While solving several optimal control problems for a diverse set of preferences using a weighted sum of objectives can produce diverse solutions, however, since different objectives have different physical meanings and units, their scales are incomparable and the weighted-sum approach cannot capture the aspiration level (i.e., level of satisfaction) of each objective function for each context. Moreover, these methods cannot learn control policies in the nonconvex parts of the Pareto optimal set [4], [3]. Finally, the uncertainty of the system’s dynamics must also be taken into account when optimizing multiple objectives. This is mainly ignored in the existing approaches.

Reinforcement Learning (RL) has been widely used to find optimal controllers for systems with uncertain dynamics. Most of existing RL algorithms are presented for single-objective optimal control problems [12], [23], [6], [8], [9], [7]. Recently, there has been a surge of interest in the study of MO reinforcement learning (MORL) problems [11], [14], [3], [15]. Nevertheless, most of existing MORL algorithms assume a given preference and find a single best policy corresponding to it based on the weighted sum of the objective functions. It is, however, desired to learn multiple potential solutions for different situations and decide, without a priori specification of preferences, which policy provides an appropriate trade-off. A novel MO optimal control framework that can satisfy objectives’ aspiration or satisfaction levels must be developed to make the connection between situations and aspiration levels of objectives. A higher level of decision-making can decide on the preferences and the most relevant calculated optimal solution can be used as a warm start to avoid learning from scratch in a novel scenario.

The main motivation of this paper is to develop a novel satisficing reinforcement learning (S-RL) framework to find a diverse set of solutions corresponding to different objectives’ aspirations to a MO optimal control problem without knowing the complete knowledge about the system dynamics. It is first shown that the Hamiltonian functional corresponding to
each objective can be leveraged to compare the performance of admissible policies. It is also shown that the aspiration level of each objective (i.e., the level of the performance at which the objective is satisfactory) can be imposed using a Hamiltonian inequality approach. Using this fact, the MO optimal control problem is formulated as an aspiration-satisfying MO optimization for which the main objective function is optimized subject to Hamiltonian inequalities that capture the aspiration-reaching of other objectives. This formulation can be interpreted as a satisficing MO decision-making framework, for which, instead of optimizing some objective functions, an aspiration level is set for them. A data-driven Sum-of-Squares (SOS)-based iterative algorithm is then developed to find a finite number of solutions of MO optimal control problems using only the information of the system trajectories measured during a time interval online in real-time.

**Notations:** The following notations are needed throughout the paper. Let $\mathbb{R}^n$ and $\mathbb{R}^{n \times m}$ denote the $n$ dimensional real vector space, and the $n \times m$ real matrix space, respectively. Let $\mathbb{Z}^*$ and $\mathbb{R}^+$ denote the sets of all positive integers and real numbers, respectively. The set of all continuously differentiable polynomial functions is denoted by $C^1$. $\mathcal{P}$ denotes the set of all positive definite and proper polynomial functions in $C^1$. Let $0_k \in \mathbb{R}^k$ be the vector with all zeros and $1_k \in \mathbb{R}^k$ the vector with all ones. Assume that $y^1, y^2 \in \mathbb{R}^m$. Then, $y^1 \leq y^2$ denotes weak componentwise order which implies $y^1_k \leq y^2_k$, $k = 1, \ldots, m$. $y^1 \prec y^2$ denotes Pareto order, which implies $y^1_1 \leq y^2_1$, $k = 1, \ldots, m$, $y^1 \neq y^2$, $y^1 \nless y^2$ denotes that $y^1$ is not Pareto dominated by $y^2$. Assume that $d_1, d_2 \in \mathbb{Z}^+$, and $d_2 \geq d_1$, then $\mathbb{R}^{d_1 \times d_2}(x) \in \mathbb{R}^{dm}$ is the arranged in lexicographic order vector of distinct monomials in terms of $x \in \mathbb{R}^n$ with degree $\kappa$ where $\theta := \left( \frac{n + d_2}{d_1} - 1 \right)$ and $d_1 \leq \kappa \leq d_2$. Moreover, the set of all polynomials in $x \in \mathbb{R}^n$ with degree $\kappa$ is denoted by $\mathcal{R}[x]_{d_1, d_2}$.

**II. PROBLEM FORMULATION**

Consider the following continuous-time nonlinear system

$$
\dot{x} = f(x) + g(x)u
$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and control input of the system, respectively. In this work, we assume that $f(.) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g(.) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are polynomial mappings and $f(0) = 0$.

For simplicity, throughout the paper, we assume the system has only two objectives. The proposed approach, however, can be readily extended to more than two objectives. The two cost or objective functions associated with the system (1) are defined as

$$
J_i(x,u) := \int r_i(x(t), u(t)) dt, i = 1,2,
$$

(2)

where $r_i(x,u) = Q_i(x) + u^T R_i u$, with $Q_i(x) \geq 0$ as the penalty on the states, and $R_i \in \mathbb{R}^{n \times n}$ as a symmetric positive definite matrix.

**Definition 1.** A control policy $u(x)$ is said to be admissible with respect to the cost functions $J_i(\cdot), i = 1,2$, if it is continuous, $u(0) = 0$, and it globally stabilizes the dynamics $\Phi(x)$ and makes $J_i(\cdot), i = 1,2$ finite. The set of admissible policies is denoted by $\Phi$ in this paper.

Define the value function for a control policy $u \in \Phi$ as

$$
V_i(x(t)) = \int r_i(x(\tau), u) d\tau, \quad i = 1,2,
$$

(3)

where $V_i(x(\infty)) = 0$.

Next, for an associated admissible policy $u \in \Phi$, define the Hamiltonian functionals corresponding to the value functions (3) as

$$
\mathcal{H}_i(x, u, V_i) = Q_i(x) + u^T R_i u + \nabla V_i^T (f(x) + g(x)u),
$$

(4)

for $i = 1,2$, where $\nabla V_i$ is the gradient of $V_i$.

**Definition 2.** For the system (1) with two objectives given by (2), a control policy $u^1 \in \Phi$, is said to dominate a control policy $u^2, u^2 \in \Phi$, in a Pareto sense, if and only if $V_i(u^1) \leq V_i(u^2), \forall i \in \{1,2\}$ and $V_i(u^1) < V_i(u^2)$, for some $i \in \{1,2\}$.

**Problem 1.** Consider the nonlinear system (1). Design an admissible control policy $u(x) \in \Phi$ that minimizes the cost functions (2) in a Pareto sense.

Approximately minimizing each cost function independently while ignoring the other cost functions can be performed using standard optimal control techniques (12). However, for dynamic MO optimal control, it is rarely possible to design a controller that optimizes all objective functions simultaneously and independently. Therefore, normally, utopian point, i.e., $J_{\text{stopian}}^{\text{topian}} := [J_{\text{stopian}}^{1 \text{topian}}, J_{\text{stopian}}^{2 \text{topian}}]^T$ where $J_{\text{stopian}} \leq J_i(x(0), u), \forall x \in \mathbb{R}^n$, $\forall u \in \mathbb{R}^m$, $\forall i \in \{1,2\}$, is unattainable. However, it is of great importance to find solutions that are as close as possible to a utopian point. Such solutions are called Pareto optimal solutions.

**III. A HAMILTONIAN-DRIVEN SATISFYING MO OPTIMAL CONTROL FRAMEWORK**

In this section, it is shown that the Hamiltonian functional corresponding to each objective can serve as a comparison function to compare the performance of admissible policies in a Pareto sense. The following theorem shows that minimizing one objective function while converting the other objective as a constraint resembles the satisficing (good enough) decision making framework for which the constraint bound is an indication of the aspiration level (the level of satisfaction) of the other objective function.

**Theorem 1:** Let $u^j(\cdot), j = 1,2$ be two different admissible policies, with their value function vectors given as $V^j(x) = [V_1^j(x), V_2^j(x)]^T, j = 1,2$, where $V_i^j(\cdot), i = 1,2$ being the solution to $\mathcal{H}_i(u^j, V_i^j) = 0, i = 1,2$. Consider the following aspiration-satisfying dynamic optimization problem

$$
\bar{u}^j := \arg \min_{u^j} \mathcal{H}_1(x, u(x), V_1^j) \quad \text{s.t.} \quad -\delta^j \leq \mathcal{H}_2(x, u(x), V_2^j) \leq 0
$$

(5)

(6)

with $\delta^j > 0$ as the aspiration for objective 2. Let also $\mathcal{H}_{\text{min}}^j := [\mathcal{H}_1^j \mathcal{H}_2^j]^T$ where $\mathcal{H}_1^j := \mathcal{H}_1(x, \bar{u}^j(x), V_1^j)$ and $\mathcal{H}_2^j := \mathcal{H}_2(x, \bar{u}^j(x), V_2^j)$. Then, the following properties hold, $\forall x \in \mathbb{R}^n$. 
1) $H_{\text{min}}^1 \leq H_j^1 = 0, j = 1, 2$.
2) If $-\delta^2 \leq H_2^j \leq 0, j = 1, 2$, and $H_1^j < H_2^j$, then $V_1^j < V_2^j$ and consequently $V_1^j < V_2^j$ and $V_2^j < V_1^j$.
3) For sufficiently small $\delta$ and $\delta^2$, if $\delta^2 < \delta^1$ and $H_1^j < H_2^j$, then $V_1^j < V_2^j$ and $V_2^j < V_1^j$.

**Proof.** The proof has three parts. It follows from (5)-(6) that $-\delta^j \leq H_2^j \leq H_2 = H_2(x, \bar{u}(x), V_2^j) \leq 0$ and $H_2 = H_1(x, \bar{u}(x), V_1^j) \leq H_1(x, u^j, V_2^j) = 0, \forall j = 1, 2$.

This proves part 1. We now prove part 2. Let $V_2^j(x) = V_1^j(x) + \lambda(x)$. Based on the Hamiltonian (4) for $V_1^j(.)$ and the stationary condition [13], one has

$$H_2^j = Q_1(x) + \nabla V_2^j(x) f(x) - \frac{1}{2} \nabla V_2^j(x) g(x) R_1^{-1} g(x) \nabla V_2^j(x)$$

$$= H_1^j + \frac{d\lambda}{dt} - \frac{1}{2} \nabla \lambda(x) g(x) R_1^{-1} g(x) \nabla \lambda(x)$$

(7)

After some manipulation, (7) can be rewritten as

$$\frac{d\lambda}{dt} = H_2^j - H_1^j + \frac{1}{2} \nabla \lambda(x) g(x) R_1^{-1} g(x) \nabla \lambda(x)$$

(8)

If $H_2^j - H_1^j \geq 0$, (8) implies that $d\lambda/dt \geq 0$. Based on (3), $\Lambda(x(\infty)) = 0$, so (8) implies that $\Lambda(x) \leq 0, \forall x \in \mathbb{R}^n$. Thus, $H_1^j < H_2^j$ implies that $V_2^j < V_1^j$ and consequently $V_1^j < V_2^j$, $V_2^j < V_1^j$. This completes the proof of part 2. To prove part 3, considering the inequality condition (6), the Lagrangian is $L' = H_1(x, \bar{u}(x), V_1^j) + \lambda_{12}[H_2(x, \bar{u}(x), V_2^j(x)) - \delta^j]$ where $\lambda_{12}$ is Lagrange multiplier. Provided that $\delta^1$ and $\delta^2$ are sufficiently small, from the Kuhn-Tucker condition [13], one can see that constraint (6) will be active, i.e., $H_2(x, \bar{u}(x), V_2^j(x)) = \delta^j, \lambda_{12} > 0$. Moreover, $\lambda_{12} = -\partial H_1(x, \bar{u}(x), V_1^j(x))/\partial H_2(x, \bar{u}(x), V_2^j(x))$ which based on property 2 indicates that an improvement in $H_1(x, \bar{u}(x), V_1^j(x))$ may only be obtained at the cost of degradation in $H_2(x, \bar{u}(x), V_2^j(x))$. Therefore, the inequality condition (6) is active, i.e., $H_2^j = H_2(x, \bar{u}(x), V_2^j(x)) = -\delta^j$ for $j = 1, 2$. Thus, using property 2, $\delta^2 < \delta^1$ implies that $H_2^j > H_1^j$ which implies that $V_2^j < V_1^j$ and consequently $V_2^j < V_1^j, \forall x \in \mathbb{R}^n$. Moreover, from property 1, one has $H_1^j < H_2^j$ implies that $V_2^j < V_1^j$ and consequently $V_1^j < V_2^j$. This completes the proof. □

**Remark 1.** Theorem 1 implies that active constraint correspond to Pareto optimal solutions. Therefore, by tightening or loosening the aspiration level, i.e., $\delta^j$, one can find different Pareto optimal solutions on the Pareto front, each corresponding on different demands on the other objective function. The desired aspiration level might depend on the circumstance the system is encountering. Using this sense, in the next section, the problem in hand will be formulated as an aspiration-satisfying optimization problem with HJB inequality as constraints.

**Remark 2.** Fig. 1 provides us with an intuition that one can compare between different admissible policies by using corresponding Hamiltonian as a measure. Based on Theorem 1, since $H_{\text{min}}^1(u_1)$ dominates $H_{\text{min}}^2(u_2)$ in Pareto sense, i.e., $H_{\text{min}}^2(u_2) < H_{\text{min}}^1(u_1)$, $V_1^j$ dominates $V_2^j$ in Pareto sense, i.e., $V_2^j < V_1^j$, and consequently, admissible policy $u^j$ gives us better solution in terms of optimality. However, since $H_{\text{min}}^1(u_3)$ is not dominate $H_{\text{min}}^1(u_1)$ and $H_{\text{min}}^2(u_2)$ in Pareto sense, and vice versa, admissible policies $u^k, k = 1, 2$ and $u^3$ are indifferent to each other.

**IV. MULTI-OBJECTIVE SUBOPTIMAL CONTROL WITH HJB INEQUALITY**

In this section, we formulate Problem 1 as an aspiration-satisfying optimization problem with HJB inequalities as constraints. To this end, MO optimal control Problem 1 can be reformulated as the following aspiration-satisfying optimization problem.

**Problem 2.** Consider the nonlinear system (1) associated with the cost functions (2). Design the control policy $u(x)$, to solve the following aspiration-satisfying problem (9)-(12).

$$\min_{V_1, u(V_1)} \int_{\Omega} V_1(x) dx$$

s.t. $H_1(x, u(V_1), V_1) \leq 0$ (10)

$$\delta \leq H_2(x, u(V_1), V_2(x)) \leq 0$$ (11)

$$V_i(.) \in \mathcal{P}, i = 1, 2$$ (12)

where $\delta > 0$ implicitly indicates the aspiration on optimizing objective $V_2$. Moreover, $\mathcal{P} \in \mathbb{R}^n$ is an arbitrary closed compact set containing the origin that describes the region in which the objective function $V_1(x)$ is expected to be minimized the most.

**Remark 3.** Based on (4), (10) implies that the closed-loop system (1) converges to the origin. Moreover, based on Theorem 1, (9)-(11) are equivalent to (10)-(12) which indicates that the cost functions (2) are minimized in a Pareto sense.

**Assumption 1.** Consider the nonlinear system (1). There exist feedback control policy $u_1(.)$ and functions $V_{01}(u_1(.) ) \in \mathcal{P}$ and $V_{02}(u_1(.) ) \in \mathcal{P}$, and a $\delta > 0$ such that

$$0 \leq \mathcal{L}_1(V_{01}(u_1(.) ), u_1(.) )$$

$$0 \leq \mathcal{L}_2(V_{02}(u_1(.) ), u_1(.) ) \leq \delta, \forall x \in \mathbb{R}^n$$ (13)

where, for any $V_i \in C^1$ and $u \in \Phi$

$$\mathcal{L}_i(V_i, u) = -\nabla V_i^T(x) f(x) + g(x) u - r_i(x, u), i = 1, 2$$

(14)

**Remark 4.** Assumption 1 implies that the control policy $u_1(.)$ makes the closed-loop system (1) asymptotically stable at the origin.
Theorem 2: Let $V_{01} \in \mathcal{P}$ and its corresponding control policy $u_{01}$ be the solution to $H_1(u_{01}, V_{01}) = 0$. Let Assumption 1 hold for the cost function $V_{01}(u_{01}(\cdot)) \in \mathcal{P}$ and $V_{02}(u_{01}(\cdot)) \in \mathcal{P}$, and control policy $u_{01}$. For a fixed $\delta > 0$, the following hold:

1) The aspiration-satisfying optimization Problem 2 has a nonempty feasible set.

2) Let $V\bar{1}(\bar{u}_1(\cdot)) \in \mathcal{P}$ and $V\bar{2}(\bar{u}_1(\cdot)) \in \mathcal{P}$ be a feasible solution to the constrained optimization Problem 2. Then, the control policy $\bar{u}_1(\cdot)$ is globally stabilizing.

3) For sufficiently large $\delta > 0$, $V_{01}(u_{01}(\cdot))$ and $V_{02}(u_{01}(\cdot))$ is a optimal solution to constrained optimization Problem 2.

Proof. The proof has two parts.

1) Under Assumption 1 and Theorem 1, it is straightforward that the feasible set is not empty.

2) For a feasible solution $V\bar{1}(\bar{u}_1(\cdot)) \in \mathcal{P}$ and $V\bar{2}(\bar{u}_1(\cdot)) \in \mathcal{P}$, the inequality equations (10)-(11) are satisfied. It follows from (14) that

$$
\dot{V}\bar{1} = \nabla V\bar{1}(\bar{u}_1(\cdot))(f(x) + g(x)\bar{u}_1(\cdot)) = -L_{ij}(\bar{V}\bar{1}, \bar{u}_1(\cdot)) - r_i(x, \bar{u}_1(\cdot))
$$

which implies that if $L_1(\bar{V}\bar{1}, \bar{u}_1(\cdot)) \geq 0$ and $L_2(\bar{V}\bar{2}, \bar{u}_1(\cdot)) \geq 0$, then $\bar{V}\bar{1}$ and $\bar{V}\bar{2}$ are well-defined Lyapunov functions for the closed-loop system composed of the dynamical system (1) and control input $\bar{u}_1(\cdot)$.

3) It follows from (9)-(11) that for sufficiently large $\delta > 0$, $V_{01} \in \mathcal{P}$ and its corresponding control policy $u_{01}$ are the solution to constrained optimization Problem 2. Now, it is remaining to show that this solution is the optimal solution of optimization Problem 2. Using (1) and (15), one has

$$
\dot{\bar{V}}_1 = -L_{ij}(\bar{V}_1, \bar{u}_1(\cdot)) - r_i(x, \bar{u}_1(\cdot))
$$

$$
\bar{V}_1(x_0) = \int_0^T L_1(\bar{V}_1, \bar{u}_1(\cdot)) + r_i(x, \bar{u}_1(\cdot)) dt + \bar{V}_1(x(T))
$$

which implies that for any other feasible solution of Problem 2, i.e., $V_1(x)$, one has $\bar{V}_1(x) \geq V_1(x)$ and consequently $\int \dot{\bar{V}}_1(x) dx \geq \int \bar{V}_1(x) dx$. This completes the proof.

V. LINEAR PROGRAM SOS-BASED MULTI-OBJECTIVE CONTROL

In this section, a novel iterative method is developed to find the solution of Problem 2 and accordingly Problems 1 based on the Sum-of-Squares (SOS)-based methods (11). To do so, the following definition is needed.

Definition 3. A polynomial $p(x)$ is an SOS polynomial, i.e., $p(x) \in \mathcal{P}^{SOS}$ where $\mathcal{P}^{SOS}$ is a set of SOS polynomial, if $p(x) = \sum_{i=1}^m p_i(x)$ where $p_i(x) \in \mathcal{P}$, $i = 1, ..., m$.

Let $V_i(x) = \sum_{j=1}^N c_{ij}m_{ij}(x) = C_i^T \bar{m}_i^{(2,2d)}(x)$, $i = 1, 2$ where $m_{ij}(x)$, $i = 1, 2$ are predefined monomials in $x$ and $c_{ij}$, $i = 1, 2$ are coefficients to be determined. Denote $V_i^k(x) := C_i^T \bar{m}_i^{(2,2d)}(x)$, $i = 1, 2$.

Assumption 2. For system (1), there exist polynomial functions $V_{01}$ and $V_{02}$, control policy $u_{01}(\cdot)$, and aspiration level $\delta(x) \in \mathcal{P}^{SOS}$ such that $V_{01} \in \mathcal{P}^{SOS}$, $V_{02} \in \mathcal{P}^{SOS}$, $\mathcal{L}_i(V_{01}(u), u) \in \mathcal{P}^{SOS}$, and $\delta(x) - \mathcal{L}_2(V_{02}(u), u) \in \mathcal{P}^{SOS}$, $i = 1, 2$.

Motivated by the work in [6] Algorithm 1 is given to find the solution of Problem 2.

Algorithm 1: LP-MO-SOS based algorithm.

1: procedure
2: Start with $\{V_{01}(\cdot), V_{02}(\cdot), u^{(0)}, \delta^{(0)}(\cdot)\}$ that satisfy Assumption 2.
3: If there is a feasible solution then solve the following SOS program:

$$
\min_{C_1, \mathcal{K}_e} \left( \int \bar{m}_i^{(2,2d)}(x) dx \right)^T C_1
$$

s.t. $\mathcal{L}_1(u(V_1), V_1(\cdot)) \in \mathcal{P}^{SOS}$, $i = 1, 2$ \hspace{1cm} (19)

$\delta^{(1)}(x) - \mathcal{L}_2(u(V_1), V_2(u(V_1))) \in \mathcal{P}^{SOS}$ \hspace{1cm} (20)

$V_{1}^{k-1} - V_{1} \in \mathcal{P}^{SOS}$ \hspace{1cm} (21)

$V_{1} \in \mathcal{P}^{SOS}$ \hspace{1cm} , $i = 1, 2$ \hspace{1cm} (22)

4: If $\|C_1 - C_1^{(k-1)}\| \leq \gamma$, where $\gamma$ is a predefined threshold, or if there is no more feasible solution $u^{*} = u(V_1)$, $U^{*} = U^{*} \cup \{u^{*}\}$ where $U^{*}$ is the set of efficient control policies and go to Step 5 else go back to Step 2 with $k = k + 1$.
5: Set $\bar{r} = \bar{r} + 1$, if $\delta^{(k+1)}(x) = \delta^{(1)}(x)$, where $0 < \nu < 1$ is predefined design parameter go to Step 2.
6: end procedure

Theorem 3: Assume that Assumptions 1-2 hold. Then, for a fixed aspiration level $\delta(x)$, the following properties hold:

1) The SOS program (18)-(22) has at least one feasible solution;

2) The control policy $u^{(k+1)}(\cdot)$ is globally asymptotically stabilizing the system (1) at the origin;

3) $0 \leq V_{1}^{k+1} \leq V_{1}^{k}, \forall k$, where $V_{1}^{k} \in \mathcal{P}^{SOS}$.

Proof. The proof has three parts.

1) Under Assumption 2 and Theorems 1-2, it is straightforward that SOS program (18)-(22) has at least one feasible solution.

2) It follows from (19) that $\mathcal{L}_1(u(V_1), V_1(\cdot)) \in \mathcal{P}^{SOS}$ and $\mathcal{L}_2(u(V_1), V_2(\cdot)) \in \mathcal{P}^{SOS}$. Therefore, $\mathcal{L}_1(u(V_1), V_1(\cdot)) \geq 0$ and $\mathcal{L}_2(u(V_1), V_2(\cdot)) \geq 0$. It follows from (14) that

$$
\dot{V}_1 = \nabla V_1^T(k(\cdot))(f(x) + g(x)k(\cdot)) = -L_{ij}(\bar{V}_1, k(\cdot)) - r_i(x, k(\cdot))
$$

which implies that $V_1$ and $V_2$ are well-defined Lyapunov functions for the closed-loop system composed of the dynamical
system \([1]\) and control input \(u^k(\cdot)\). Therefore, the control policy \(u^k(x)\) makes the system \([1]\) globally asymptotically stabilizing at the origin.

3) Constraints \([21]\) and \([22]\) imply \(0 \leq V_{k+1}^1 \leq V_k^1 \) and \(V_k^1 \in P_{SOS}, \forall k\). This completes the proof. \(\square\)

**Remark 5.** The proposed Algorithm 1 requires the perfect knowledge of \(f \) and \(g\). In practice, precise system knowledge may be difficult to obtain. Hence, in the next section, we develop an online learning method based on the idea of approximate dynamic programming (ADP) to implement the iterative scheme with real-time data, instead of identifying first the system dynamics.

**VI. DATA-DRIVEN REINFORCEMENT LEARNING IMPLEMENTATION**

In this section, a data-driven satisficing reinforcement learning algorithm is developed to implement Algorithm 2 without having the full knowledge of the system dynamics. This algorithm uses measured data and a value function formulation with a span of a finite family of basis functions to approximate the solution of the infinite-dimensional LP problem 2.

Now, consider the system \([1]\), after adding an exploratory probing noise, one has

\[
\dot{x} = f + g(u^{k+1} + \epsilon) \quad (24)
\]

where \(u^{k+1}\) is a control policy at iteration \(k + 1\) and \(\epsilon\) is an added bounded exploration probing noise.

In the infinite-dimensional linear program (LP)-MO SOS-based Algorithm 1, under Assumption 2, one has \(\forall k, r, \mathcal{L}_i(u^k, V_i(\cdot)) \in \mathcal{R}[x]_{2,2 \delta r} \), \(i = 1, 2, \delta r(x) = \mathcal{L}_2(u^k, V_2(\cdot)) \in \mathcal{R}[x]_{2,2 \delta r}\), where \(\delta r(x)\), if the integer \(\delta t\) satisfies

\[
delta t \geq \frac{1}{2} \max(\text{deg}(f(\cdot)) + 2d - 1, 2 \text{deg}(g(\cdot)) + 2(2d - 1), \text{deg}(Q_1(\cdot)) + \text{deg}(Q_2(\cdot)), \text{deg}(\delta r(\cdot)))
\]

(25)

where \(\text{deg}(\cdot)\) represents the degree of the polynomial which is the highest degree of any of the terms. Also, \(u^{k+1}\) obtained from the proposed LP-MO SOS-based Algorithm 1 satisfies \(u^{k+1} \in \mathcal{R}[x]_{1, \delta t}, \forall k, r\).

Hence, there exits a constant matrix \(K_{C_1}^{k+1} \in \mathcal{R}^{m \times n_{2\delta t}}\), with \(n_{2\delta t} = \left(\frac{n + 2 \delta t}{2d}\right) - 1\), such that \(u^{k+1} = K_{C_1}^{k+1} m_1^{1, (\delta t)}\). Also, suppose there exist constant vectors \(C_1 \in \mathcal{R}^{n_{2\delta t}}\) and \(C_2 \in \mathcal{R}^{n_{2\delta t}}\), with \(n_{2\delta t} = \left(\frac{n + 2 \delta t}{2d}\right) - n - 1\), such that \(V_i(x) := C_1^T m_1^{1, (2\delta t)}(x)\) and \(V_2(x) := C_2^T m_2^{1, (2\delta t)}(x)\). It follows then from \((24)\) that

\[
\dot{V}_1 = -r_1(x, u^k) - \mathcal{L}_1(u^k, V_1(\cdot)) + (R_1^{-1} g^T \nabla V_1) R_1 e
\]

(26)

\[
\dot{V}_2 = -r_2(x, u^k) - \mathcal{L}_2(u^{k+1}, V_2(\cdot)) + \nabla V_2^T \mathcal{R}_1^1 g^T \nabla V_1 R_1 e
\]

(27)

Notice that the terms \(\mathcal{L}_1(u^k, V_1(\cdot)), \mathcal{L}_2(u^k, V_2(\cdot)), R_1^{-1} g^T \nabla V_1, \) and \(\nabla V_2^T \mathcal{R}_1^1 g^T \nabla V_1) R_1^T \) depend on the dynamic of the system. Also, note that constant vectors and matrix \(l_{C_1} \in \mathcal{R}^{n_{2\delta t}}\) and \(l_{C_2} \in \mathcal{R}^{n_{2\delta t}}\), and \(K_{C_1} \in \mathcal{R}^{m \times n_{2\delta t}}\) with \(\delta t = \left(\frac{n + 2 \delta t}{2d}\right) - \delta t - 1\) for the tuple \((V_1, V_2, u^{k+1})\) can be chosen such that:

\[
\mathcal{L}_i(u^k, V_i(\cdot)) = l_{C_i}^T m_1^{1, (2\delta t)}(x), \quad i = 1, 2
\]

(28)

\[
\frac{1}{2} R_1^{-1} g^T \nabla V_1 = K_{C_1} m_1^{1, (\delta t)}
\]

(29)

Therefore, calculating \(\mathcal{L}_i(u^k, V_i(\cdot)), \quad i = 1, 2\) and \(R_1^{-1} g^T \nabla V_1\) amounts to find \(l_{C_1}, l_{C_2}, \) and \(K_{C_1}\).

Substituting \((28)\) and \((29)\) in \((26)-(27)\), we have

\[
\dot{V}_1 = -r_1(x, u^k) - l_{C_1}^T m_1^{1, (2\delta t)}(x) - 2(m_1^{1, (\delta t)})^T K_{C_1} R_1 e
\]

(30)

\[
\dot{V}_2 = -r_2(x, u^k) - l_{C_2}^T m_2^{1, (2\delta t)}(x) - 2((V_1^{(2\delta t)}(x(t)))^T X_2 m_1^{1, (\delta t)})^T C_1^T K_{C_1} R_1 e
\]

(31)

Integrating both sides of \((30)-(31)\) on the interval \([t, t + \delta t]\) yields the following off-policy integral RL Bellman equations

\[
C_1^T (m_1^{1, (2\delta t)}(x(t)) - m_1^{1, (\delta t)}(x(t + \delta t))) =
\]

\[
\int_t^{t + \delta t} \left(r_1(x, u^k) + l_{C_1}^T m_1^{1, (2\delta t)}(x) + 2(m_1^{1, (\delta t)})^T K_{C_1} R_1 e\right) d\tau
\]

(32)

\[
C_2^T (m_2^{1, (2\delta t)}(x(t)) - m_2^{1, (\delta t)}(x(t + \delta t))) =
\]

\[
\int_t^{t + \delta t} \left(r_2(x, u^k) + l_{C_2}^T m_2^{1, (2\delta t)}(x) + 2((V_1^{(2\delta t)}(x(t)))^T X_2 m_1^{1, (\delta t)})^T C_1^T K_{C_1} R_1 e\right) d\tau
\]

(33)

It follows from \((32)-(33)\) that \(l_{C_1}, l_{C_2}, \) and \(K_{C_1}\) can be found by using only the information of the system trajectories measured during a time interval, without requiring any system dynamic information. To this end, we define the following matrices:

\[
\sigma^k_1 = [\begin{array}{c} m_1^{1, (2\delta t)} \\ 2(m_1^{1, (\delta t)})^T \end{array}] \times e^T R_1, \quad \sigma^k_2 = [\begin{array}{c} m_2^{1, (2\delta t)} \\ 2((V_1^{(2\delta t)}(x(t)))^T C_1^T (m_1^{1, (\delta t)})^T) \end{array}] \times e^T R_1
\]

(34)

\[
\phi^k_1 = [\begin{array}{c} t_{k+1, k+1}^1 \\ \vdots \\ t_{0, k+1}^1 \\ \vdots \\ t_{0, 0}^1 \end{array}]^T
\]

(35)

\[
\Xi^k_1 = [\begin{array}{c} r_1(x, u^k) \\ \vdots \\ r_1(x, u^k) \end{array}]^T
\]

(36)

\[
\theta^k_1 = [\begin{array}{c} m_1^{1, (2\delta t)} \\ m_1^{1, (\delta t)} \end{array}] \times e^T R_1
\]

(37)

\[
\phi^k_2 = [\begin{array}{c} t_{k+1, k+1}^2 \\ \vdots \\ t_{0, k+1}^2 \\ \vdots \\ t_{0, 0}^2 \end{array}]^T
\]

(38)

for \(i = 1, 2\), where \(\phi^k_1 \in \mathcal{R}^{k+1 \times (n_{2\delta t} + m_{2\delta t})}\) and \(\Xi^k_1 \in \mathcal{R}^{k+1 \times 1}\).

It follows from \((32)-(33)\) that

\[
\phi^k_1 + \frac{l_{C_1}^T}{V_{cc}(C_1)} \Xi^k_1 + \theta^k_1 = C_1^T K_{C_1} e
\]

(39)

\[
\phi^k_2 + \frac{l_{C_2}^T}{V_{cc}(C_1)} \Xi^k_1 + \theta^k_1 = C_1^T K_{C_1} e
\]

(40)
Assumption 3. At each iteration \( k \), there exists a lower-bound \( q^k \in \mathbb{Z}^+ \) such that if \( q_{1}^k \geq q_0^k \), \( q_{2}^k \geq q_0^k \), \( q_{1}^k \) and \( q_{2}^k \) are dimensional vectors of \( \Xi_1^k \) and \( \Xi_2^k \), respectively, then \( \text{rank}(\phi_{1}^k) = n_2^k \) and \( \text{rank}(\phi_{1}^k) = n_2^k \).

Now, assume that \( q_{1}^k + q_{2}^k \geq q_0^k \), \( k \in \mathbb{Z}^+ \). It follows from (39)-(40) that the values of \( l_{C_1} \in \mathbb{R}^{n_2} \), \( l_{C_2} \in \mathbb{R}^{n_2} \), and \( K_{C_i} \in \mathbb{R}^{n_2 \times n_2} \) are determined as follows:

\[
\begin{bmatrix}
    l_{C_1} \\
    V_{C_1}(K_{C_1}) \\
    l_{C_2} \\
    V_{C_2}(K_{C_2})
\end{bmatrix} = \begin{bmatrix}
    (\phi_{1}^k)^T\phi_{1}^k - 1(\phi_{1}^k)^T(\Xi_{1}^k + \phi_{1}^k C_1) \\
    (\phi_{1}^k)^T\phi_{1}^k - 1(\phi_{1}^k)^T(\Xi_{2}^k + \phi_{1}^k C_2)
\end{bmatrix}
\]

(41)

So, an iterative LP-SOS based data-driven learning algorithm is proposed in Algorithm 2 for online implementation of Algorithm 1.

Algorithm 2: Data-driven LP-MO-SOS based algorithm.

1: procedure
2: Find the tuple \( (V_1^0, V_2^0, u^0) \) such that Assumption 2 be satisfied. Choose \( C_1^0 \) and \( C_2^0 \) such that \( V_1^0(x) := (C_1^0)^Tm_1^{(2,2)}(x) \) and \( V_2^0(x) := (C_2^0)^Tm_2^{(2,2)}(x) \).
3: Employ \( u = u^0 + e \) as the input to the system \( \lambda \), where \( e \) is the probing noise and calculate and construct \( \Xi_1, \Xi_2, \theta_1, \) and \( \theta_2 \) as (33)-(38), until \( \phi_1, \phi_2 \) be of full column rank.
4: Solve the following SOS program to find an optimal solution \( \{C_1^k, C_2^k, K_{C_1}^k\} \):

\[
\min_{C_1^k, C_2^k} \left( \int_\Omega m_1^{(2,2)}(x)(dx)^T C_1 \right)
\]

s.t. \[
\begin{bmatrix}
    l_{C_1} \\
    V_{C_1}(K_{C_1}) \\
    l_{C_2} \\
    V_{C_2}(K_{C_2})
\end{bmatrix} = \begin{bmatrix}
    (\phi_{1}^k)^T\phi_{1}^k - 1(\phi_{1}^k)^T(\Xi_{1}^k + \phi_{1}^k C_1), \\
    (\phi_{1}^k)^T\phi_{1}^k - 1(\phi_{1}^k)^T(\Xi_{2}^k + \phi_{1}^k C_2),
\end{bmatrix}
\]

(42)

5: Update the value functions and control policy as follows:

\[
V_i^k(x) := C_i^k m_1^{(2,2)}(x), i = 1, 2
\]

(43)

\[
u_i^{(k+1)}(x) = K_{C_i}^k m_1^{(1,1)}(x)
\]

(44)

6: If \( \|C_1^k - C_1^{k-1}\| \leq \gamma \), where \( \gamma \) is a predefined threshold, or if there is no more feasible solution \( u_i^* = u_i^{(k+1)}(x) \) and go to Step 7; else go back to Step 2 with \( k = k + 1 \).

7: end procedure

Theorem 4: Assume that Assumptions 1-3 hold. Then, for a fixed \( \delta^*(x) \), the following properties hold:
1) There exists at least one feasible solution for the SOS program (42)-(47);
2) The control policy \( u_i^{(k+1)}(x) \) is globally asymptotically stabilizing the system (1) at the origin;
3) \( 0 \leq V_i^{k+1}(x) \leq V_i^1(x) \), \( \forall x \in \Omega \), where \( V_1^1(x) \) is given in (48).

Proof. Provided that \( \{C_1^0, C_2^0\} \) is a feasible solution to the LP-MO-SOS program (15)-(22), one can find the corresponding matrix \( K_{C_i}^0 \in \mathbb{R}^{n_2 \times n_2} \) such that the tuple \( \{C_1^0, C_2^0, K_{C_i}^0\} \) be a feasible solution to the data-driven LP-MO-SOS program (42)-(47) and (48)-(49), which imply that property 1 holds. Moreover, since the tuple \( \{C_1^0, C_2^0, K_{C_i}^0\} \) is a feasible solution to the data-driven LP-MO-SOS program (42)-(47) and (48)-(49) and the tuple \( \{C_1^k, C_2^k\} \) is a feasible solution to the LP-MO-SOS program (15)-(22) and Algorithms 1 and 2 have the equal objective function, \( K_{C_i}^k m_1^{(1,1)} \) is an optimal solution to the LP-MO-SOS program (15)-(22) and consequently the results of Theorem 3 are further extended to Theorem 4. This completes the proof.

VII. SIMULATION

In this section, the effectiveness of the proposed scheme is verified by two simulation examples.

Example 1. Consider the linearized double inverted pendulum in a cart, with dynamics given by (15)

\[
A = \begin{bmatrix}
    0 & 0 & 0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

(50)

where \( x_1, x_2, \) and \( x_3 \) are the position of the cart and angles of both pendulums, respectively; \( x_4 \) and \( x_5 \) are the velocities. The quadratic cost functions chosen as

\[
J_i(x_0, u) = \int_0^{\infty} (x^T Q_i x + u^T R_i u) dt, i = 1, 2
\]

(51)

With \( Q_1 = 1 _6 \), \( Q_2 = 200 \times 1_6 \), and \( R_1 = R_2 = 1 \), in the set \( \Omega = \{x \mid x \in \mathbb{R}^6 and |x| \leq 1.7 \} \). After the implementation of Algorithm 2 with three different aspiration levels as \( \delta = \delta^*(0.2x^2_2 x_3 + 0.1x^2_2 x_5 + 0.25x_2^2 x_6 + 0.5x_5 x_6 + 0.7x^2_2 x_4 + 0.2x^2_2 x_6 + 0.5x_4 x_5 x_6 + 0.2 x_2 x_3 x_4) \), \( i = 1, 2, 3 \) with \( \delta^* \in \{0.01, 0.14, 2\} \), three suboptimal control policies are obtained. Fig. 2 shows the evolution of the system states after applying the obtained policies. It can be seen in Fig. 2 that by changing the aspiration level on second objective the obtained control policies and corresponding system states are changed. That is, the trade-off between regulation error and control effort are changed by changing the aspiration level on second objective. Moreover, it can be seen in Fig. 2 that all the state trajectories are stabilized by the controller, which shows that the MO control problem 1 is solved in this case.

Example 2. Consider the quarter-suspension system depicted in Fig. 3 with dynamics given by (6) and (24)

\[
\dot{x}_1 = x_2 \\
\dot{x}_2 = -k_1(x_1-x_s) + k_2(x_1-x_s) - b_s(x_2-x_d)-u \\
\dot{x}_3 = x_4 \\
\dot{x}_4 = k_2(x_3-x_s)+k_3(x_3-x_s) + b_4(x_4-x_d)+k_4 x_3 -u
\]

(52)

where \( x_1, x_2, x_3, \) and \( x_4 \) represent the position and velocity of the car and the position and velocity of the wheel assembly, respectively; \( m_b \) and \( m_w \) denote the mass of the car and the mass of the wheel assembly; \( k_s, k_s, k_s, \) and \( b_s \) denote
tyre stiffness, the linear suspension stiffness, the nonlinear suspension stiffness, and the damping rate of the suspension. Moreover, $u$ is the control force from the hydraulic actuator.

Let $m_b = 300\,\text{kg}$, $m_o = 60\,\text{kg}$, $k_l = 190000\,\text{N/m}$, $k_s = 16000\,\text{N/m}$, $k_n = 1600$, and $b_s = 1000\,\text{N/m/s}$. We use the proposed online learning algorithm to design an active suspension control system which simultaneously reduces the following cost functions

$$J_1(x_0, u) = \int_0^\infty \left( \sum_{i=1}^2 10x_i^2 + u^2 \right) dt \tag{53}$$

After the implementation of Algorithm 2 with two different aspiration levels, two suboptimal control policies are obtained.
after seven and four iterations, as follows

\[ u^{(7)}_{i2}(x) = -11.94x_1^2 - 17.05x_2x_3 + 5.44x_3^3 + 0.302x_4^2 + 0.00055x_1^2 - 11.06x_2x_3 + 11.95x_1x_2x_3 - 0.62x_1x_2x_4 + 0.00014x_2x_3 - 33.95x_1x_3^2 + 3.37x_1x_2x_4 - 0.00024x_1x_3^2 - 2.80x_1x_3 - 0.000046x_3x_4 - 16.84x_1 - 3.36x_3 + 8.51x_2^2x_3 - 1.34x_2x_3 + 0.00059x_2^2 + 4.52x_2x_3^2 + 1.52x_2x_3x_4 - 0.00014x_2x_3^2 - 1.61x_4^2x_3 - 0.00082x_2x_4 - 27.66x_2 + 41.72x_2^3 + 0.34x_3x_4 - 0.00021x_3^2 + 4.47x_3x_4^2 + 0.00010x_3x_4^3 + 12.73x_3 - 0.014x_3^2 - 0.000083x_4^3 + 0.31x_4 \]

(57)

\[ u^{(4)}_{i2}(x) = -0.51x_1^3 - 0.13x_1^2x_2 + 0.23x_1^2x_3 + 0.026x_1x_2x_3^2 + 0.000000017x_2^2 - 0.014x_1x_2^2 + 0.079x_1x_2x_3 + 0.0135x_1x_2x_4 + 0.0000000029x_1x_2 - 0.502x_1x_3^2 - 0.048x_1x_3x_4 - 0.00000000131x_1x_3 - 0.00508x_1x_3 - 0.00000000995x_1x_4 - 0.361x_1 + 0.00077x_3^2 + 0.00192x_2x_3 - 0.00031x_2x_4 + 0.00000000035x_2^2 - 0.0077x_2x_3^2 + 0.00143x_2x_3x_4 - 0.00000000038x_2x_3 - 0.000039x_2x_4 - 0.00000000016x_2x_4 - 0.114x_2 + 0.775x_3^2 + 0.114x_3^2x_1 + 0.0000000030x_3^2 + 0.0253x_3x_4 + 0.00000000031x_3x_4 + 0.38x_3 + 0.000109x_4^2 + 0.000000000092x_4^2 + 0.00030 \]

(58)

To test the learned controllers, a disturbance as a single pulse bump with the magnitude of 10 is simulated at \( t = 0 \sim 0.001 \) sec such that the states deviate from the origin. The trajectories of the states after applying \( u^{(7)}_{i2}(x) \) and \( u^{(4)}_{i2} \) are given in Fig. The aspiration level \( \delta(x) \), the level of optimality on the second performance objective is not tight, therefore, we learn controller that has a better performance on the first performance objective, i.e., \( x_1 \) and \( x_2 \). For the aspiration level \( \delta^2(x) \), however, the level of optimality on the first performance objective is tighter. Therefore, the learned control policy has a better performance on the second performance objective.

VIII. CONCLUSION

This paper has developed an iterative data-driven adaptive dynamic programming algorithm for dynamic MO optimal control problem for nonlinear continues-time polynomial systems. The MO optimal control problem was, first, formulated as a aspiration-satisfying optimization problem with HJB inequalities as constraints. To deal with this problem, then, a LP-SOS based iterative algorithm was presented to find some Pareto optimal solutions of MO optimal control problem with HJB inequalities. This LP-SOS based iterative algorithm required the knowledge of the system dynamic. To obviate the requirement of complete knowledge of the system dynamics, an online data-driven reinforcement learning method was proposed for online implementation of the proposed LP-SOS based algorithm. Finally, two simulation examples were provided to show the effectiveness of the proposed algorithm.

REFERENCES

[1] Ahmadi, A. (2018). Sum of Squares (SOS) Techniques: An Introduction. 1-9.
[2] Carmichael, D. G. (1980). Computation of Pareto optima in structural design. Int. J. Numer. Methods Eng., 15 (6), 925929.
[3] Caramia, M., and Dell’Olmo, P. (2008). Multi-objective optimization. in Multi-Objective Management in Freight Logistics. Increasing Capacity, Service Level and Safety with Optimization Algorithms. London, U.K. Springer.
[4] Das, I., and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct. Optim., 14 (1), 6369.
[5] Gambier, A. and Jipp, M. (2011). Multi-objective optimal control: An introduction. ASCC 2011 - 8th Asian Control Conf. - Final Progr. Proc., 10841089.
[6] Jiang, Y. and Jiang, Z. (2015). Global Adaptive Dynamic Programming for Continuous-Time Nonlinear Systems. in IEEE Transactions on Automatic Control, 60(11), 2917-2929.
[7] Tzanakakis, A. and Lygeros, J. (2020). Data-Driven Control of Unknown Systems: A Linear Programming Approach. ArXiv ID:2003.00779.
[8] Wang, Y., O’Donoghue, B., and Boyd, S. (2014). Approximate dynamic programming via iterated Bellman inequalities. International Journal of Robust and Nonlinear Control, 25(10), 14721496.
[9] Beuchat, P. N., Georghiou, A., and Lygeros, J. (2020). Performance Guarantees for Model-Based Approximate Dynamic Programming in Continuous Spaces. IEEE Transactions on Automatic Control, 65(1), 143158.
[10] Nalabapurkar, R., Walters, P., Rosenfeld, J., and Dixon, W. (2018). Reinforcement Learning for Optimal Feedback Control. Cham: Springer International Publishing.
[11] Kang, D. O., and Bien, Z. (2004). Multi-objective control problems by reinforcement learning. in Handbook of Learning and Approximate Dynamic Programming, 433461.
[12] Lewis, Frank L., and Draguna Vrabie. (2009). Adaptive Dynamic Programming for Feedback Control. Proceedings of 2009 7th Asian Control Conference, ASCC 2009, 14029.
[13] Lewis, F. L., Vrabie, D. L., and Syrmos, V. L. (2012). Optimal Control: Third Edition. Hoboken, NJ, USA: John Wiley and Sons, Inc.
[14] Logist, F., Sager, S., Kirches, C., and Van Impe, J. F. (2010). Efficient multiple objective optimal control of dynamic systems with integer controllers. J. Process Control, 20 (7), 810822.
[15] Lofberg, J. (2004). CVX: Matlab Software for Disciplined Convex Programming. 26 (6), 306395.
[16] Modares, H. Lewis, F. L. and Jiang, Z. P. (2016). Optimal Output-Feedback Control of Unknown Continuous-Time Linear Systems Using Off-policy Reinforcement Learning. IEEE Trans. Cybern., 46(11), 24012140.
[17] Ober-Blocha, S., Ringkamp, M., and Zum Felde, G. (2012). Solving multiobjective Optimal Control problems in space mission design using Discrete Mechanics and reference point techniques. Proc. IEEE Conf. Decis. Control, 57115716.
[18] Peitz, S. and Dellnitz, M. (2018). A Survey of Recent Trends in Multi-objective Optimal ControlSurrogate Models, Feedback Control and Objective Reduction. Math. Comput. Appl., 23(2), 30.
[19] Røpke, D. M., Vanpoucke, P., Dazeley, R., Whiteson, S. and Dazeley, R. (2013). A survey of multi-objective sequential decision-making. J. Artif. Intell. Res., 48, 67113.
[20] Toivonen, H. T. (1986). A primaldual method for linearquadratic gaus-