A Data-driven Methodology for Solving the Control Strategy of Descriptor Systems

Daqing Zhang, Mengmeng Li
Institute of Applied Mathematics,
University of Science and Technology Liaoning,
Anshan, Liaoning Province, 114051, China
E-mail: d.q.zhang@ustl.edu.cn,
2008limengmeng@sina.com

Jinna Li
Shenyang Institute of Automation,
Chinese Academy of Sciences
Shenyang, Liaoning Province, 110016, China
E-mail: lijinna_721@yahoo.com.cn

Abstract—**This paper is concerned with the reinforcement learning methods for the discrete time descriptor systems. An algorithm, as well as its theoretical basis, is presented. The algorithm can generate the optimal controller for the target descriptor system only by the measured input and output data, with no need of the information about the system state and system matrices. The algorithm can work well not only when the system index is equal or less than one, but also can work well when the index is greater than one. Simulation indicates that the presented method can solve the optimal control problem well for descriptor systems when the system model is not exactly known, but the input and output data can be measured.**

I. Introduction

Descriptor systems, which are also named as differential algebra systems, singular systems, or semi-state systems, are found in the area of aeronautics and astronautics, robots, power systems, electrical networks, chemical industry, bioengineerring and economics [1], [2]. After more than three decades researches, many excellent results on the descriptor system have been obtained. Up to now, research on this type of system is still a hot branch in the control theory and control engineering [3]–[8].

Reinforcement learning (RL) is a class of methods used in machine learning to methodically modify the actions of an agent based on observed responses from its environments [9]–[11]. Since RL can solve control problem only relies on the input and output data measured from the system, it provide a data-driven control method.

The data-driven control means that the controller designing depends only on the input/output (I/O) measurement data of the controlled plant, without any explicitly using the system model, but it could be designed using an implicit information of the system dynamics or system structure [12], [13]. In the past decades, many data-driven control approaches could be found in literature, but different authors call them with different names: data-based, data-driven, model-free, iterative learning control (ILC) [14], [15], unfalsified control (UC) [16]–[18], virtual reference feedback tuning (VRFT) [19], iterative feedback tuning (IFT) [20], etc.

In this paper, the main motivation is to establish reinforcement learning methods for discrete time descriptor systems. The presented algorithm solves the control problem of the underlying descriptor system only relies on the input and output data. The results obtained at the rest of the paper are expansion of [9], in which RL methodology was presented for normal systems. For a normal linear discrete system, the causality can always be satisfied. However, in the discrete time descriptor system cases, the causality will be lost when the index of the descriptor system is greater than one. This is the main difficulty in solving the control problems of descriptor system. From the conclusions stated follows we know, if the index is equal or less than one, then the target system is a normal system, and the presented algorithm coincide with the ones presented in [9]. But, if the index is greater than one, the causality of the system will be lost and the method in [9] will fail to generate the desired controller. However, the presented algorithm in this paper can solve the control problem just through the past inputs and past and current outputs of the system.

The rest of the paper is as follows. In section 2, problems are formulated and preliminaries are done. The main results are presented in section 3. In section 4, a descriptor system with index of 2 is studied, and simulation is provided. Finally, the whole paper is summarized in section 5.

II. Problems Formulation and Preliminaries

Consider the following discrete time linear descriptor system

$$\begin{cases} Ex_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{cases} \tag{1}$$

where, $x_k \in R^n$ is the state, $u_k \in R^m$ is the control input and $y_k \in R^p$ is the measured output. E, A, B, C are system matrices with appropriate dimensions. Especially, matrix E is singular, e.g. Rank$(E) < n$ in general. The descriptor system (1) is said to be regular if there is some λ in complex plane such that$\det(\lambda E - A) \neq 0$, causal if $\deg\det(\lambda E - A) =$ Rank(E), and stable if the finite general eigenvalues of E, A are all lie in unit circle of the complex plane [1], [2]. For a given descriptor system (1), there always exist a nonsingular

978-1-4673-5769-2/13/$31.00 ©2013 IEEE

matrix H such that $\tilde{x}_k = Hx_k$, and

$$\left\{\begin{array}{rcl} \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{pmatrix} \tilde{x}_{k+1}^{(1)} \\ \tilde{x}_{k+1}^{(2)} \end{pmatrix} & = & \begin{bmatrix} A_1 & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} \tilde{x}_{k}^{(1)} \\ \tilde{x}_{k}^{(2)} \end{pmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u_k \\ y_k & = & [C_1 \quad C_2] \begin{pmatrix} \tilde{x}_{k}^{(1)} \\ \tilde{x}_{k}^{(2)} \end{pmatrix} \end{array}\right. \tag{2}$$

where, $\tilde{x}_k = \left(\tilde{x}_k^{(1)} \quad \tilde{x}_k^{(2)}\right)^T$, $\tilde{x}_k^{(1)} \in R^r, \tilde{x}_k^{(2)} \in R^{n-r}$. N is a nilpotent matrix with $N^{h-1} \neq 0$ and $N^h = 0$. Here, the index h of the nilpotent matrix N is also the index of the descriptor system (1). When $h \leq 1$ the system is causal, otherwise, the system is not a causal system. In the following, we always suppose that the descriptor (1) has already been transformed into the standard form of (2).

Given a stabilizing control policy $u_k = \mu(x_k)$, associate to the system the performance index

$$V^\mu(x_k) = \sum_{i=k}^{\infty} (y_i^T Q y_i + u_i^T R u_i) \tag{3}$$

with weighting matrices $Q = Q^T \geq 0$, $R = R^T > 0$. The optimal control problem is to find the policy $u_k = \mu(x_k)$ that minimizes the cost (3) along the trajectories of the system (1). Due to the special structure of the dynamics and the cost, this is known as the LQR problem.

A difference equation that is equivalent to (3) is given by the Bellman equation

$$V^\mu(x_k) = y_k^T Q y_k + u_k^T R u_k + V^\mu(x_{k+1}) \tag{4}$$

The optimal cost, or value, is given by

$$V^*(x_k) = \min_\mu \sum_{i=k}^{\infty} (y_i^T Q y_i + u_i^T R u_i) \tag{5}$$

and the optimal control is given by

$$\mu^*(x_k) = \underset{u_k}{\operatorname{argmin}}(y_k^T Q y_k + u_k^T R u_k + V^*(x_{k+1})) \tag{6}$$

For the LQR case, any value is quadratic in the state so that the cost associated to any policy $u_k = \mu(x_k)$ is ([2])

$$V^\mu(x_k) = x_k^T (E^{h+1})^T P E^{h+1} x_k \tag{7}$$

for some $n \times n$ matrix P, where, E is the derivative matrix of the system (1), and h is the index of system (1). In this case, the LQR Bellman equation is

$$\begin{aligned} & x_k^T (E^{h+1}) P E^{h+1} x_k \\ & = y_k^T Q y_k + u_k^T R u_k + x_{k+1}^T (E^{h+1})^T P E^{h+1} x_{k+1} \end{aligned} \tag{8}$$

If the policy is a linear state variable feedback so that

$$u_k = \mu(x_k) = -Kx_k \tag{9}$$

then from previous research results [2] we get the Riccati equation as

$$\begin{aligned} & A^T (E^h)^T P E^h A - A^T (E^h)^T P E^h B \times \\ & \qquad [R + B^T (E^h)^T P E^h B]^{-1} (E^h)^T P E^h A - \\ & \qquad (E^{h+1})^T P E^{h+1} \\ & = -(E^{h+1})^T C^T Q C E^{h+1} \end{aligned} \tag{10}$$

and the optimal control input as

$$u_k^* = -[R + B^T (E^h)^T P E^h B]^{-1} B^T (E^h)^T P E^h A x_k^* \tag{11}$$

Equations (10) and (11) give the optimal control strategy for descriptor system (1), when the system model is exactly known. The problem is to be solved in this paper is to establish a RL method to obtain the optimal controller under the condition that the system model is not known. That is to find the optimal controller only through the input and output data with no information about the system states and system matrices E, A, B, C. Before the main results, we make the following assumptions about the target system.

Assumption:

- The system is linear and R-observable, that is the matrix $[C_1^T, (C_1 A_1)^T, \cdots, (C_1 A_1)^T]^T$ has full column rank.
- There is no noise in the measured input and output data.
- The index h of the system is known.

Remark 1: The main difference of discrete time descriptor to the normal system is its index may be greater than one. If the index is unknown, then the normal system model is appreciated. However, if there are some algebra constraints are known before the controller designing, then the descriptor system model has to be adopted. In this case, the information of the index may be obtained by the known algebra constraints of the systems.

III. Main Results

This section presents the new results of this paper. Suppose that the index of the system is h. Then LQR Bellman equation is replaced as

$$V^\mu(x_k) = y_k^T Q y_k + u_{k,k+h}^T R u_{k,k+h} + V^\mu(x_{k+1}) \tag{12}$$

and

$$\begin{aligned} & x_k^T (E^{h+1}) P E^{h+1} x_k \\ & = y_k^T Q y_k + u_{k,k+h}^T R u_{k,k+h} + x_{k+1}^T (E^{h+1})^T P E^{h+1} x_{k+1} \end{aligned} \tag{13}$$

where, $u_{k,k+h} = [u_k, u_{k+1}, \cdots, u_{k+h}]^T$ is the inputs of current time k and future inputs $k+1, \cdots, k+h$.

The Bellman error for LQR is quadratic in the state. This can be used in a policy iteration (PI) or value iteration (VI) algorithm for online learning of optimal controls as long as full measurements of state $x_k \in R^n$ are available [9].

$$x_k = \begin{bmatrix} A_1^L & \\ & I \end{bmatrix} \begin{pmatrix} x_{k-L}^{(1)} \\ 0 \end{pmatrix} + \begin{bmatrix} A_1^{L-1}B_1 & A_1^{L-2}B_2 & \cdots & B_1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -B_2 & -NB_2 & \cdots & -N^{h-1}B_2 \end{bmatrix} \begin{bmatrix} u_{k-L} \\ u_{k-L+1} \\ \vdots \\ u_{k-1} \\ u_k \\ u_{k+1} \\ \vdots \\ u_{k+h-1} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} y_{k-L} \\ y_{k-L+1} \\ \vdots \\ y_{k-2} \\ y_{k-1} \end{bmatrix} = \begin{bmatrix} C_1 & v_1 \\ C_1A_1 & v_2 \\ \vdots & \vdots \\ C_1A_1^{L-2} & v_{L-1} \\ C_1A_1^{L-1} & v_L \end{bmatrix} \begin{pmatrix} x_{k-L}^{(1)} \\ 0 \end{pmatrix} +$$

$$\begin{bmatrix} -C_2B_2 & -C_2NB_2 & \cdots & -C_2N^{h-1}B_2 & 0 & 0 & \cdots & 0 & 0 \\ C_1B_1 & -C_2B_2 & \cdots & -C_2N^{h-2}B_2 & -C_2N^{h-1}B_2 & 0 & \cdots & 0 & 0 \\ C_1A_1B_1 & C_1B_1 & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & -C_2B_2 & -C_2NB_2 & -C_2N^2B_2 & \cdots & 0 & 0 \\ C_1A_1^{L-2}B_1 & C_1A_1^{L-3}B_1 & \cdots & C_1B_1 & -C_2B_2 & -C_2NB_2 & \cdots & -C_2N^{h-1}B_2 & 0 \end{bmatrix} \begin{bmatrix} u_{k-L} \\ u_{k-L+1} \\ \vdots \\ u_{k-1} \\ u_k \\ u_{k+1} \\ \vdots \\ u_{k+h-1} \end{bmatrix} \tag{15}$$

Consider again the deterministic linear discrete time invariable system (1). Given the current time k, the dynamics can be written on a time horizon $[k-L, k+h-1]$ as the expanded state equation as (14) and (15) at the top of the next page, where, $v_1, \cdots, v_L$ are matrices that make the matrix

$$V_L = \begin{bmatrix} C_1^T & (C_1A_1)^T & \cdots & (C_1A_1^{L-2})^T & (C_1A_1^{L-1})^T \\ v_1^T & v_2^T & \cdots & v_{L-1}^T & v_L^T \end{bmatrix}^T$$

to have full column rank. Note that, the system is supposed to be R-observable, there always exist matrices $v_1, \cdots, v_L$ such that the matrix V_L has full column rank.

By appropriate definition of variables, equations (14) and (15) can be rewritten as

$$x_k = A^L \begin{bmatrix} x_{k-L}^{(1)} \\ 0 \end{bmatrix} + U_L u_{k-L,k+h-1} \tag{16}$$

$$y_{k-L,k-1} = V_L \begin{bmatrix} x_{k-L}^{(1)} \\ 0 \end{bmatrix} + T_L u_{k-L,k+h-1} \tag{17}$$

Since V_L has full column rank n, it has left inverse as

$$V_L^+ = (V_L^T V_L)^{-1} V_L^T \tag{18}$$

and there exists a matrix $M \in R^{pL}$ such that

$$A^L = M V_L \tag{19}$$

so that

$$M = A^N V_L^+ + Z(I - V_L V_L^+) = M_0 + M_1 \tag{20}$$

for any matrix Z, with M_0 denoting the minimum norm operator and $P(R^{\perp}(V_L)) = I - V_L V_L^+$ being the projection onto a range perpendicular to V_N. This was used in [9].

Lemma 1: Let the matrix V_L have full column rank. Then, the system state is given uniquely in terms of the measured output $y_{k-L,k-1}$, the measured input $u_{k-L,k-1}$, and the input $u_{k,k+h-1}$ sequences by

$$\begin{aligned} x_k &= M_0 y_{k-L,k-1} + (U_L - M_0 T_L) u_{k-L,k+h-1} \\ &= M_y y_{k-L,k-1} + M_u u_{k-L,k+h-1} \end{aligned} \tag{21}$$

or

$$x_k = [M_u \quad M_y] \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} \tag{22}$$

where $M_y = M_0$ and $M_u = U_L - M_0 T_L$, with $M_0 = A^L V_L^+$, $V_L^+ = (V_L^T V_L)^{-1} V_L^T$, being the left inverse of the matrix V_L.

Proof: According to equations (16), (17) and (19), we have

$$\begin{aligned} x_k &= A^L \begin{pmatrix} x_{k-L}^{(1)} \\ 0 \end{pmatrix} + U_L u_{k-L,k+h-1} \\ &= M V_L \begin{pmatrix} x_{k-L}^{(1)} \\ 0 \end{pmatrix} + U_L u_{k-L,k+h-1} \\ &= M(y_{k-L,k-1} - T_L u_{k-L,k+h-1}) + U_L u_{k-L,k+h-1} \\ &= (U_L - A^L V_L^+ T_L) u_{k-L,k+h-1} + A^L V_L^+ y_{k-L,k+h-1} \\ &= M_u u_{k-L,k+h-1} + M_y y_{k-L,k+h-1} \\ &= [M_u \quad M_y] \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} \end{aligned}$$

■

Consider the optimal value $V^\mu(x_k)$, which can be rewritten in terms of the input and output data as

$$\begin{aligned} V^\mu(x_k) = x_k^T (E^{h+1})^T P E^{h+1} x_k &= \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix}^T \times \\ \begin{bmatrix} M_u^T \\ M_y^T \end{bmatrix} (E^{h+1})^T P E^{h+1} [M_u \quad M_y] & \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} \\ = \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix}^T \bar{P} & \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} \end{aligned} \tag{23}$$

And the Bellmans equation can be rewritten in terms of the input and output data as

$$\begin{aligned} \begin{bmatrix} u_{k-N,k+h-1} \\ y_{k-N,k-1} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} = y_k^T Q y_k + \\ u_{k,k+h}^T R u_{k,k+h} + \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix} \end{aligned} \tag{24}$$

Based on equation (24), the temporal difference error can be written in terms of the inputs and outputs as

$$\begin{aligned} e_k = - \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} + y_k^T Q y_k + \\ u_{k,k+h}^T R u_{k,k+h} + \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix} \end{aligned} \tag{25}$$

Using this temporal difference error, the policy evaluation step of any form of reinforcement learning based on the Bellman temporal difference error can be equivalently performed using only the measured data, not the state.

Note that, the matrix $\bar{P}$ depends on E, A, B, C of the descriptor system (1). However, the following presented method allows one to learn $\bar{P}$ online without the system matrices.

Theorem 1: Let the discrete time descriptor system (1) be R-observable, then the optimal control strategy can be obtained by an ARMA model of the measured input and output data as

$$u_{k,k+h} = -(R + p_0)^{-1} (p_u u_{k-L+1,k} + p_y y_{k-L+1,k}) \tag{26}$$

where, matrices p_o, p_u and p_y are defined as in (28) in the following.

Proof: With the equation (23), the policy improvement step may be written in terms of the input and output data as

$$\begin{aligned} \mu(x_k) &= \underset{u_{k,k+h}}{\operatorname{argmin}} (y_k^T Q y_k^T + u_{k,k+h}^T R u_{k,k+h} + x_{k+1}^T P x_{k+1}) \\ &= \underset{u_{k,k+h}}{\arg\min} (y_k^T Q y_k^T + u_{k,k+h}^T R u_{k,k+h} + \\ & \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}) \end{aligned} \tag{27}$$

Partition $\begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}$ as

$$\begin{aligned} \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P} \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix} = \\ \begin{bmatrix} u_{k,k+h} \\ u_{k-L+1,k-1} \\ y_{k-L+1,k} \end{bmatrix}^T \begin{bmatrix} p_0 & p_u & p_y \\ p_u^T & p_{22} & p_{23} \\ p_y & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} u_{k,k+h} \\ u_{k-L+1,k-1} \\ y_{k-L+1,k} \end{bmatrix} \end{aligned} \tag{28}$$

Then, differentiating with respect to $u_{k,k+h}$ to perform the minimization in (27) yields

$$0 = R u_{k,k+h} + p_0 u_{k,k+h} + p_u u_{k-L+1,k-1} + p_y y_{k-L+1,k} \tag{29}$$

or

$$u_{k,k+h} = -(R + p_0)^{-1} (p_u u_{k-L+1,k} + p_y y_{k-L+1,k}) \tag{30}$$

■

Remark 2: The optimal controller given by Theorem 1 relies only on the past measured input and output data. It is an

ARMA model, and does not need the information about the system state and system matrices.

Remark 3: If the system index is equal or less than one, the conclusion in Theorem is same with the ones given in [9].

Algorithm 1 (Value Iteration (VI) Algorithm): Select any initial control policy $u_k^0 = \mu^0$. Then for $j = 0, 1, 2, \cdots$, perform until convergence:

1. **Policy Evaluation:** Solve for $\bar{P}^{j+1}$ such that

$$\begin{aligned} &\begin{bmatrix} u_{k-N,k+h-1} \\ y_{k-N,k-1} \end{bmatrix}^T \bar{P}^{j+1} \begin{bmatrix} u_{k-L,k+h-1} \\ y_{k-L,k-1} \end{bmatrix} \\ &= y_k^T Q y_k + u_{k,k+h}^T R u_{k,k+h} + \\ &\begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P}^j \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix} \end{aligned} \tag{31}$$

2. **Policy Improvement:** Partition $\bar{P}$ as in (28). Then define the updated policy by

$$\begin{aligned} u_{k,k+h}^{j+1} = &-(R + p_0^{j+1})^{-1} \times \\ &(p_u^{j+1} u_{k-L+1,k-1} + p_y^{j+1} y_{k-L+1,k}) \end{aligned} \tag{32}$$

The equations in Algorithm 1 are solved online by standard techniques using methods such as batch least-squares (LS). One solves (31) in the form of

$$\begin{aligned} &\left(\begin{bmatrix} u_{k-N,k+h-1} \\ y_{k-N,k-1} \end{bmatrix} \otimes \begin{bmatrix} u_{k-N,k+h-1} \\ y_{k-N,k-1} \end{bmatrix} \right)^T \mathrm{Vec}(\bar{P}^{j+1}) \\ &= y_k^T Q y_k + u_{k,k+h}^T R u_{k,k+h} + \\ &\begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix}^T \bar{P}^j \begin{bmatrix} u_{k-L+1,k+h} \\ y_{k-L+1,k} \end{bmatrix} \end{aligned} \tag{33}$$

Since the kernel matrix $\bar{P}^{j+1}$ in (31) is symmetric and with dimensions as

$$[(L+h)m + Lp] \times [(L+h)m + Lp]$$

there are

$$[(L+h)m + Lp] \times [(L+h)m + Lp + 1]/2$$

unknown variables to be determined. Hence,

$$[(L+h)m + Lp] \times [(L+h)m + Lp + 1]/2$$

outputs and

$$[(L+h)m + Lp] \times [(L+h)m + Lp + 1]/2 + h$$

inputs must be known before the algorithm be performed.

IV. Simulations

Consider the following descriptor system

$$\begin{cases} Ex_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{cases} \tag{34}$$

where,

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ A = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 2 & -4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & -1 & 1 \end{bmatrix}$$

This system is already in the form of (2). Since

$$E = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}$$

and $N^2 = 0$, the index of the system is $h = 2$. Set the length of time horizon as $L = 4$ to insure the observability matrix to have full column rank. In this case, the kernel matrix $\bar{P}$ in policy evaluation step is with the dimensions of 10×10. It is a symmetric matrix, thus there are 55 unknown variables to be determined.

Input the system initial input data as

$$u_k = 1,\ (k = 0, 1, \cdots, 55 + h)$$

Suppose that the initial state is $x_0^{(1)} = (-1, 2)^T$ and x_0^2 be comfortable with the inputs. Then, measured the relevant output y_k, $k = 1, 2, \cdots, 55$ can be measured. Let matrices Q, R in (31) be identity matrix with appropriate dimensions and initial the kernel matrix $\bar{P}^0$ as an identity matrix $I_{10\times10}$. Then by algorithm 1, we can get

$$p_0 = \begin{bmatrix} 0.1127 & 0.2253 & 0.2253 \\ 0.2253 & 0.1127 & 0.2253 \\ 0.2253 & 0.2253 & 0.1127 \end{bmatrix}$$

$$p_u = \begin{bmatrix} 0.2253 & 0.2253 & 0.2253 \\ 0.2253 & 0.2253 & 0.2253 \\ 0.2253 & 0.2253 & 0.2253 \end{bmatrix}$$

$$p_y = \begin{bmatrix} -0.0146 & -0.1333 & 0.2977 & 0.3052 \\ -0.0146 & -0.1333 & 0.2977 & 0.3052 \\ -0.0146 & -0.1333 & 0.2977 & 0.3052 \end{bmatrix}$$

With matrices p_0, p_u, p_y, one can compute the input data step by step through

$$u_{k,k+h} = -(R + p_0)^{-1}(p_u u_{k-N+1,k-1} + p_y y_{k-N+1,k}) \tag{35}$$

Figure 1 shows the response of the system outputs and under the computed control strategy given by Algorithm 1.

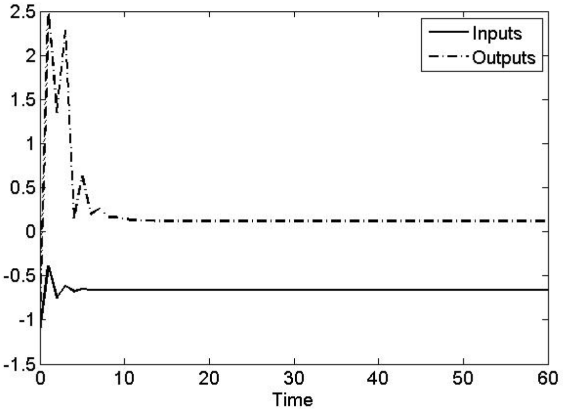

Fig. 1. The inputs and the outputs of the discrete-time descriptor system (34).

V. Conclusion

A reinforcement learning method has been established for the discrete time descriptor systems. The presented algorithm can generate optimal controller for the underlying descriptor system only from the measured input and output data. The controller design process need no information about the system state and system matrices. However, the system index must be known before the computation start. The system index is a special value about the descriptor systems. If the system index is equal or less than one, then it is no need to model the target system by descriptor system model. However, if the prior information about the target system indicates that the system index is greater than one, then the descriptor system model has to be applied. This case may occur when there are some algebra constraints in the system variables.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61273011, 61273003 and 61104093) and the Research Project of Liaoning Province of China under Grant L2012141, L2011064.

References

[1] L. Dai, *Singular control systems*, ser. Lecture Notes in Control and Information sciences. Springer- Verlag, 1989.

[2] D. M. Yang, Q. L. Zhang, and B. Yao, *Descriptor systems*. Beijing: Science press, 2004.

[3] J. H. Kim, "Delay-dependent robust H-infinity filtering for uncertain discrete-time singular systems with interval time-varying delay," *Automatica*, vol. 46, no. 3, pp. 591–576, 2010.

[4] Z. Zuo, D. W. C. Ho, and Y. Wang, "Fault tolerant control for singular systems with actuator saturation and nonlinear perturbation," *Automatica*, vol. 46, no. 3, pp. 569–576, 2010.

[5] L. Wu, P. Shi, and H. Gao, "State estimation and sliding-mode control of markovian jump singular systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1213–1219, 2010.

[6] W. J. Mao and J. Chu, "Regularization and stabilization of linear discrete-time descriptor systems," *IET Control Theory and Applications*, vol. 4, no. 10, pp. 2205–2211, 2010.

[7] L. M. Sun and Y. Z. Wang, "H_∞ control of a class nonlinear hamiltonian descriptor systems," *SCIENCE CHINA: Information Sciences*, vol. 53, no. 11, pp. 2195–2204, 2010.

[8] Z. H. Jiang, W. H. Gui, Y. F. Xie, and C. H. Yang, "Memory state feedback control for singular systems with multiple internal incommensurate constant point delays," *ACTA Automatica Sinca*, vol. 35, no. 2, pp. 174–179, 2009.

[9] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, vol. 41, no. 1, pp. 14–25, 2011.

[10] B. Luo and H. N. Wu, "Online policy iteration algorithm for optimal control of linear hyperbolic pde systems," *Journal of Process Control*, vol. 22, no. 7, pp. 1161–1170, 2012.

[11] H. N. Wu and B. Luo, "Simultaneous policy update algorithms for learning the solution of linear continuous-time H_∞ state feedback control," *Information Sciences*, vol. 222, pp. 471–485, 2013.

[12] Z. S. Hou and S. T. Jin, "Data-driven model-free adaptive control for a class of mimo nonlinear discrete-time systems," *IEEE Transations on Neural Networks*, vol. 22, no. 12, pp. 2173–2188, 2011.

[13] H. G. Zhang, L. L. Cui, and Y. H. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[14] H. S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1211, 2007.

[15] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 546–551, 2012.

[16] J. V. Helvoort and B. D. Jager, "Direct data-driven recursive controller unfalsification with analytic update," *Automatica*, vol. 43, no. 12, pp. 2034–2046, 2007.

[17] J. V. Helvoort, B. D. Jager, and M. Steinbuch, "Data-driven multivariable controller design using ellipsoidal unfalsified control," *Systems Control Letters*, vol. 57, no. 9, pp. 759–722, 2008.

[18] A. Dehghani, B. D. O. Anderson, and A. Lanzon, "Unfalsified adaptive control: A new controller implementation and some remarks," in *Proc. Eur. Control Conf*, 2007, pp. 709–716.

[19] M. C. Campi and S. M. Savaresi, "Direct nonlinear control design: The virtual reference feedback tuning (VRFT) approach," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 14–27, 2006.

[20] H. Hjalmarsson, "From experiment design to closed-loop control," *Automatica*, vol. 41, no. 3, pp. 393–438, 2005.