Computational Adaptive Optimal Control with an Application to Blood Glucose Regulation in Type 1 Diabetics

Yu Jiang [1], Zhong-Ping Jiang[1]

1. Department of Electrical and Computer Engineering, Polytechnic Institute of New York University, Brooklyn, NY 11201, USA
E-mail: yjiang06@students.poly.edu, zjiang@poly.edu

Abstract: This paper presents an online policy iteration approach for finding optimal controllers for continuous-time linear systems with completely unknown dynamics, via adaptive/approximate dynamic programming. Using the proposed scheme, the a priori knowledge of the system matrices is not required, and all iterations can be conducted by using repeatedly the same state and input information collected on some fixed finite time-intervals. A practical computational adaptive optimal control algorithm is developed in this paper, and is applied to the optimal blood glucose regulation problem in type 1 diabetics.

Key Words: Optimal adaptive control, linear quadratic regulator (LQR), blood glucose regulation.

1 INTRODUCTION

Reinforcement learning [29] and approximate/adaptive dynamic programming (ADP) [38] theories have been broadly applied for solving optimal control problems in recent years (see, for example, [1, 7, 13, 21, 31, 32, 34, 41]). Compared with traditional adaptive control techniques (for example, [12, 30]), the ADP method directly approximates the optimal control policy without identifying the unknown system parameters. For discrete-time systems, the action-dependent heuristic dynamic programming (ADHDP) [39] (or Q-learning [36]) is an online iterative scheme that does not depend on the model to be controlled, and it has been applied to many different areas (see [1, 7, 22, 37]). However, due to the different structures of the algebraic Riccati equations between discrete-time and continuous-time systems, these results cannot be directly applied for solving continuous-time problems. Hence, in the setting of continuous-time systems, it is commonly assumed that partial knowledge of the system dynamics is exactly known in order to apply ADP-based algorithms.

The primary objective of this paper is to remove this assumption on partial knowledge of the system dynamics, and thus to develop a truly model-free ADP algorithm. More specifically, we propose a novel computational adaptive optimal control methodology that employs the approximate/adaptive dynamic programming technique to iteratively solve the algebraic Riccati equation using the online information of state and input, without requiring the a priori knowledge of the system matrices. In addition, all iterations can be conducted by using repeatedly the same state and input information on some fixed time intervals. It should be noticed that our approach may serve as a computational tool to study ADP-related problems for continuous-time systems.

As one of the most widespread diseases, diabetes mellitus is characterized by the inability of the pancreas to control blood glucose concentration, and type 1 diabetes is caused by destruction of beta cells in the pancreas by the immune system [5]. With the advent of new technologies in glucose and insulin sensing (see [10, 11, 35]) as well as insulin infusion, a variety of methods have been proposed for the closed-loop glucose regulation to improve insulin therapy (see, for example, [5, 6, 17, 24, 25, 26]). However, one obstacle in designing such controllers is that physiological parameters are not only difficult to be obtained, but also different from individuals to individuals. As a result, it is desired that the controller has certain adaptability to achieve optimal blood glucose regulation for different patients. In this paper, we will show that this problem can be well handled using the proposed ADP algorithm.

The remainder of this paper is organized as follows: In Section 2, we briefly introduce the standard linear optimal control problem for continuous-time systems and the policy iteration technique. In Section 3, we develop our computational adaptive optimal control method and show its convergence. A practical online algorithm will be proposed. In Section 4, we apply the proposed online scheme to the optimal blood glucose regulation problem for different patients. Concluding remarks as well as potential future extensions are contained in Section 5.

Throughout this paper, we use $\mathbb{R}$ and $\mathbb{Z}_+$ to denote the sets of real numbers and non-negative integers, respectively. Vertical bars $\|\cdot\|$ represent the Euclidean norm for vectors, or the induced matrix norm for matrices. We use $\otimes$ to indicate Kronecker product, and $\mathrm{vec}(A)$ is defined to be the mn-vector formed by stacking the columns of $A \in \mathbb{R}^{n\times m}$ on top of one another, i.e., $\mathrm{vec}(A) = [a_1^T\ a_2^T\ \cdots\ a_m^T]^T$, where $a_i \in \mathbb{R}^n$ are the columns of A. A control law is also called a *policy*, and it is said to be stabilizing if under the policy, the closed-loop system is asymptotically stable at the origin.

2 PROBLEM FORMULATION AND PRELIMINARIES

We consider continuous-time linear systems described by

$$\dot{x} = Ax + Bu \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state fully available for feedback control design, $u \in \mathbb{R}^m$ is the control input, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are unknown constant matrices. In addition, the system is assumed to be stabilizable.

The design objective is to find an optimal linear control

This work has been supported in part by the US National Science Foundation, under grants DMS-0906659 and ECCS-1101401.

Z. P. Jiang is also with College of Engineering, Beijing University, P.R. China.

law in the form of

$$u = -Kx \tag{2}$$

which minimizes the following performance index

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{3}$$

where $Q = Q^T \geq 0$, $R = R^T > 0$, with $(A, Q^{1/2})$ observable.

By linear optimal control theory [20], when both A and B are accurately known, solution to this problem can be found by solving the following well-known algebraic Riccati equation (ARE)

$$A^T P + PA + Q - PBR^{-1}B^T P = 0. \tag{4}$$

By assumptions, (4) has a unique symmetric positive definite solution P^*. The optimal feedback gain matrix K^* in (2) can thus be determined by

$$K^* = R^{-1}B^T P^*. \tag{5}$$

Since (4) is nonlinear in P, it is usually difficult to directly solve P^* from (4), especially for large-size matrices. Nevertheless, many efficient algorithms have been developed to numerically approximate the solution of (4). One of such algorithms was developed in [18], and is introduced in the following.

Theorem 2.1 ([18]) *Let $K_0 \in \mathbb{R}^{m\times n}$ be any constant matrix such that $A - BK_0$ is stable, and let $\{P_k\}$, with $k \in \mathbb{Z}_+$, be a sequence of real symmetric matrices, such that each P_k is the positive definite solution of the Lyapunov equation*

$$(A-BK_k)^T P_k + P_k (A-BK_k) + Q + K_k^T R K_k = 0, \tag{6}$$

where K_k, with $k = 1, 2, \cdots$, are defined recursively by:

$$K_k = R^{-1}B^T P_{k-1}. \tag{7}$$

Then, the following properties hold:

1. *$A - BK_k$ is Hurwitz,*
2. *$P^* \leq P_{k+1} \leq P_k$,*
3. *$\lim_{k\to\infty} K_k = K^*$, $\lim_{k\to\infty} P_k = P^*$.*

Therefore, by iteratively solving the Lyapunov equation (6), which is linear in P_k, and updating K_k by (7), solution to the nonlinear equation (4) is numerically approximated.

For the purpose of solving (6) without the knowledge of A, in [32], (6) was online implemented by

$$\begin{aligned} & x^T(t)P_k x(t) - x^T(t+\delta t)P_k x(t+\delta t) \\ = & \int_t^{t+\delta t} \left(x^T Q x + u_k^T R u_k\right) d\tau \end{aligned} \tag{8}$$

where $u_k = -K_k x$ is the control input of the system on the time interval $[t, t+\delta t]$.

Since both x and u_k can be measured online, a symmetric solution P_k can be uniquely determined under certain persistent excitation (PE) condition [32]. However, as we can see from (7), the exact knowledge of system matrix B is still required for the iterations. Also, to guarantee the PE condition, the state may need to be reset at each iteration step, but this may cause technical problems for stability analysis of the closed loop system [32]. An alternative way is to add exploration noise [1, 4, 14, 31] such that $u_k = -K_k x + e$, with e the exploration noise, is used as the true control input in (8). As a result, P_k solved from (8) and the one obtained from solving (6) are not exactly the same. In addition, after each time the control policy is updated, information of the state and input must be re-collected for the next iteration. This may slow down the learning process, especially for high-dimensional systems.

3 COMPUTATIONAL ADAPTIVE OPTIMAL CONTROL: A NEW DESIGN

In this section, we will present our new online learning strategy that does not rely on either A or B. First, we assume a stabilizing K_0 is known. Then, for each $k \in \mathbb{Z}_+$, we seek to solve a symmetric positive definite matrix P_k satisfying (6), and obtain a feedback gain matrix $K_{k+1} \in \mathbb{R}^{m\times n}$ using $K_{k+1} = R^{-1}B^T P_k$.

To begin with, we rewrite the original system (1) as

$$\dot{x} = A_k x + B(K_k x + u) \tag{9}$$

where $A_k = A - BK_k$.

Then, along the solutions of (9), it follows that

$$\begin{aligned} & x(t+\delta t)^T P_k x(t+\delta t) - x(t)^T P_k x(t) \\ = & \int_t^{t+\delta t} \left[x^T (A_k^T P_k + P_k A_k) x + 2(u + K_k x)^T B^T P_k x\right] d\tau \\ = & \int_t^{t+\delta t} \left[-x^T Q_k x + 2(u + K_k x)^T R K_{k+1} x\right] d\tau \end{aligned} \tag{10}$$

where $Q_k = Q + K_k^T R K_k$.

Remark 3.1 *Notice that in* (10)*, the term $x^T(A_k^T P_k + P_k A_k)x$ depending on the unknown matrices A and B is replaced by $-x^T Q_k x$, which can be obtained by measuring the state online. Also, the term $B^T P_k$ involving B is replaced by RK_{k+1}, in which K_{k+1} is treated as another unknown matrix to be solved together with P_k. Therefore,* (10) *plays an important role in separating the system dynamics from the iterative process. As a result, the requirement of the system matrices in (6) and (7) can be replaced by the state and input information measured online.*

Remark 3.2 *It is also noteworthy that in* (10) *we always have exact equality if P_k, K_{k+1} satisfy (6), (7), and x is the solution of system (9) with arbitrary control input u. This fact enables us to employ $u = -K_0 x + e$, with e the exploration noise, as the input signal for learning, without affecting the convergence of the learning process.*

Next, we will give a condition under which (10) has unique solutions for all $k \in \mathbb{Z}_+$. To this end, we define the following two operators

$$\begin{aligned} P \in \mathbb{R}^{n\times n} & \rightarrow \hat{P} \in \mathbb{R}^{\frac{1}{2}n(n+1)}, \text{ and} \\ x \in \mathbb{R}^n & \rightarrow \bar{x} \in \mathbb{R}^{\frac{1}{2}n(n+1)} \end{aligned}$$

where

$$\begin{aligned} \hat{P} &= [p_{11}, 2p_{12}, \cdots, 2p_{1n}, p_{22}, 2p_{23}, \cdots, 2p_{n-1,n}, p_{nn}]^T, \\ \bar{x} &= [x_1^2, x_1x_2, \cdots, x_1x_n, x_2^2, x_2x_3, \cdots, x_{n-1}x_n, x_n^2]^T. \end{aligned}$$

In addition, by Kronecker product representation, we have

$$\int_t^{t+\delta t} x^T Q_k x \, d\tau = \left[\int_t^{t+\delta t} x \otimes x d\tau\right]^T \mathrm{vec}(Q_k),$$

and

$$\begin{aligned}&\int_t^{t+\delta t} (u+K_k x)^T R K_{k+1} x \, d\tau \\ = &\int_t^{t+\delta t} \left[(x^T \otimes x^T)(I_n \otimes K_k^T R)\right] d\tau \, \mathrm{vec}(K_{k+1}) \\ &+ \int_t^{t+\delta t} \left[(x^T \otimes u^T)(I_n \otimes R)\right] d\tau \, \mathrm{vec}(K_{k+1})\end{aligned}$$

Further, for any positive integer l, we define

$$\begin{aligned}\delta_{xx} &= \left[\begin{array}{cccc} \bar{x}|_{t_0}^{t_1} & \bar{x}|_{t_1}^{t_2} & \cdots & \bar{x}|_{t_{l-1}}^{t_l} \end{array}\right]^T \in \mathbb{R}^{l\times \frac{1}{2}n(n+1)}, \\ I_{xx} &= \left[\int_{t_0}^{t_1} x\otimes x d\tau \ \int_{t_1}^{t_2} x\otimes x d\tau \cdots \int_{t_{l-1}}^{t_l} x\otimes x d\tau\right]^T \in \mathbb{R}^{l\times n^2}, \\ I_{xu} &= \left[\int_{t_0}^{t_1} x\otimes u d\tau \ \int_{t_1}^{t_2} x\otimes u d\tau \cdots \int_{t_{l-1}}^{t_l} x\otimes u d\tau\right]^T \in \mathbb{R}^{l\times mn}\end{aligned}$$

where $0 \le t_0 < t_1 < \cdots < t_l$ are arbitrary constants.

Then, for any given stabilizing gain matrix K_k, (10) implies the following matrix form of linear equations

$$\Theta_k \left[\begin{array}{c} \hat{P}_k \\ \mathrm{vec}\,(K_{k+1}) \end{array}\right] = \Xi_k \tag{11}$$

where $\Theta_k \in \mathbb{R}^{l\times\left[\frac{1}{2}n(n+1)+mn\right]}$ and $\Xi_k \in \mathbb{R}^l$ are defined as:

$$\begin{aligned}\Theta_k &= \left[\delta_{xx}, -2I_{xx}(I_n \otimes K_k^T R) - 2I_{xu}(I_n \otimes R)\right], \\ \Xi_k &= -I_{xx}\,\mathrm{vec}(Q_k).\end{aligned}$$

The following lemma studies the uniqueness of the solution of (11).

Lemma 3.1 *If there exists an integer $l_0 > 0$, such that, for all $l \ge l_0$,*

$$\mathrm{rank}\left(\left[\begin{array}{cc} I_{xx}, & I_{xu} \end{array}\right]\right) = \frac{n(n+1)}{2} + mn. \tag{12}$$

then Θ_k has full column rank, for all $k \in \mathbb{Z}_+$.

Proof: See the Appendix. □

Under the rank condition (12), P_k and K_{k+1} can be directly solved from

$$\left[\begin{array}{c} \hat{P}_k \\ \mathrm{vec}\,(K_{k+1}) \end{array}\right] = (\Theta_k^T\Theta_k)^{-1}\Theta_k^T\Xi_k. \tag{13}$$

Theorem 3.1 *Under the rank condition* (12)*, starting from $K_0 \in \mathbb{R}^{m\times n}$ such that $A - BK_0$ is Hurwitz, the sequences $\{P_i\}_{i=0}^{\infty}$ and $\{K_j\}_{j=1}^{\infty}$ obtained from solving (13) converge to the optimal values P^* and K^*, respectively, where P^* is the solution of the ARE (4) and K^* is determined by (5).*

Proof: If $P_k = P_k^T$ is the solution of (6), K_{k+1} is uniquely determined by $R^{-1}B^TP_k$. By (10), we know that P_k and K_{k+1} satisfy (13).

On the other hand, let $P = P^T \in \mathbb{R}^{n\times n}$ and $K \in \mathbb{R}^{m\times n}$, such that

$$\left[\begin{array}{c} \hat{P} \\ \mathrm{vec}(K) \end{array}\right] = (\Theta_k^T\Theta_k)^{-1}\Theta_k^T\Xi_k.$$

Then, we immediately have $\hat{P} = \hat{P}_k$ and $\mathrm{vec}(K) = \mathrm{vec}(K_{k+1})$. By Lemma 3.1, under the rank condition (12), $P = P^T$ and K are unique. In addition, by the definitions of $\hat{P}$ and $\mathrm{vec}(K)$, $P_k = P_k^T = P$ and $K_{k+1} = K$ are uniquely determined.

Therefore, under the rank condition (12), the policy iteration (13) is equivalent to (6) and (7). By Theorem 2.1, the convergence is thus proved. □

Now we are able to give the following computational adaptive optimal control algorithm for practical online implementation:

Algorithm 3.1 *(Computational adaptive optimal control algorithm)*

1. *Employ $u = -K_0x + e$, with e the exploration noise, as the input. Record δ_{xx}, I_{xx} and I_{xu} until the rank condition* (12) *is satisfied. Let $k \leftarrow 0$.*
2. *Solve P_k and K_{k+1} from (13).*
3. *Let $k \leftarrow k+1$, and repeat Step 2 until $\|P_k - P_{k-1}\| \le \epsilon$ for $k \ge 1$, where the constant $\epsilon > 0$ can be any predefined small threshold.*
4. *Use $u = -K_kx$ as the approximated optimal control policy.*

Remark 3.3 *It can be seen that Algorithm 3.1 contains two separated phases: First, an initial stabilizing control policy with exploration noise is applied and the online information is recorded in matrices δ_{xx}, I_{xx}, and I_{xu} until the rank condition (12) is satisfied. Second, without requiring additional system information, the matrices δ_{xx}, I_{xx}, and I_{xu} are repeatedly used to implement the iterative process. A sequence of controllers, that converges to the optimal control policy, can be obtained.*

Remark 3.4 *In Algorithm 3.1, neither the information of A nor B is used. In addition, we do not directly identify A or B which together contain $n^2 + nm$ unknown constants. Instead, in each iteration we solve P_k and K_{k+1} comprised of only $\frac{1}{2}n(n+1) + nm$ unknown constants. Also, the state and input information is stored in matrices δ_{xx}, I_{xx}, and I_{xu}. Once the rank condition* (12) *is satisfied, the matrices are repeatedly used for all iterations and no new system information needs to be collected.*

Remark 3.5 *The choice of exploration noise is not a trivial task for general reinforcement learning problems and other related machine learning problems. In past literature, several types of exploration noise have been adopted, such as random noise [1], exponentially decreasing probing noise [31]. For the simulations in the next section, we will use the sum of sinusoidal signals with different frequencies, as in [13].*

4 APPLICATION TO THE GLUCOSE REGULATION PROBLEM

The model we study is the well-known *minimal model* developed by Bergman *et al.* [3]. It subdivides the the insulin

compartment into a plasma space and a compartment remote to the plasma which affects glucose uptake. This model can be described as follows:

$$\dot{G} = -p_1(G - G_B) - XG + D, \quad (14)$$
$$\dot{X} = -p_2 X + p_3(I - I_B), \quad (15)$$
$$\dot{I} = -nI + \frac{1}{V_I}u \quad (16)$$

where G is the blood glucose concentration; X is an auxiliary function representing insulin-excitable tissue glucose uptake activity, and is proportional to insulin concentration in the remote compartment; I is the blood insulin concentration; D is the rate of infusion of exogenous glucose from food intake; constants G_B and I_B are basal values of plasma glucose concentration and free plasma insulin concentration, respectively; p_1, p_2, p_3 are physiological constants implying the tissue glucose uptake rate and the insulin sensitivity; V_I is the insulin distribution volume and n is the fractional disappearance rate of insulin; u is the infusion rate of the insulin pump [8].

Without meal intake, we can assume $D = 0$. Otherwise, D can be modeled by decaying exponential function shown as follows:

$$D = e^{-b(t-t_0)}D(t_0), \ \ t_0 \geq 0. \quad (17)$$

where b is a positive constant, and $t = t_0$ is the time for meal input.

Notice that the steady-state insulin injection rate can be set to $u_c = nV_I I_B$ [8], which can be determined clinically. Further, define $G_\Delta = G - G_B$, $I_\Delta = I - I_B$, $X_\Delta = X$ and $u_\Delta = u - u_c$. Then, with $D = 0$, the (14)-(16) can be linearized in the following system:

$$\frac{d}{dt}\begin{bmatrix} G_\Delta \\ X_\Delta \\ I_\Delta \end{bmatrix} = \begin{bmatrix} -p_1 & -G_b & 0 \\ 0 & -p_2 & p_3 \\ 0 & 0 & -n \end{bmatrix}\begin{bmatrix} G_\Delta \\ X_\Delta \\ I_\Delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{V_I} \end{bmatrix} u_\Delta \quad (18)$$

The control objective is to design an adaptive optimal control scheme that finds online the optimal insulin pump rate that minimizes the following cost with $D = 0$

$$J = \int_0^\infty \left(\alpha G_\Delta^2 + \beta u_\Delta^2\right) d\tau \quad (19)$$

where $\alpha > 0$ and $\beta > 0$ are weighting constants.

For simulation purpose, we use three sets of physiological parameters describing three different patients. The parameters are based on [8], as shown in Table 1.

Table 1: Physiological Parameters for the Patients

Parameters	**Patient 1**	**Patient 2**	**Patient 3**
p_1	1.0×10^{-7}	1.0×10^{-5}	1.0×10^{-6}
p_2	0.025	0.03	0.02
p_3	1.3×10^{-5}	2×10^{-5}	1×10^{-5}
G_B (mmol/L)	4.5	4	5
I_b (mU/L)	15	13	17
n	0.0926	0.1000	0.0741
b	0.05	0.1	0.07
V_I (L)	12	10	15

In the simulation, it is assumed that variables G, X, I of the patients are in steady-state at $t = 0$, and there is no meal input in the first 200 minutes. The weighting parameters in (19) are set to be $\alpha = 100$ and $\beta = 1$. The initial control input is defined as $u_\Delta = G_\Delta + e$, where e is the sum of one hundred sinusoids with different frequencies randomly taking from $[-500,\ 500]$. Algorithm 3.1 is applied on the time interval of $[0,\ 180]$, and approximated optimal controllers are obtained after three hours. The patients have meals at $t = 200\text{min}$, which is simulated by setting $D(200) = 0.5$ [8]. Then, blood glucose levels of all the patients are regulated under ADP-based controllers. Convergence performance of the cost matrices is illustrated in Figure 1, and trajectories of the glucose level, insulin level, control input and meal input are shown in Figures 2-5.

5 CONCLUSIONS AND FUTURE WORK

A novel computational policy iteration approach for finding online adaptive optimal controllers for continuous-time linear systems with completely unknown system dynamics has been presented in this paper. Based on the ADP methodology, the algebraic Riccati equation was iteratively solved using system state and input information collected online. By implementing this computational adaptive optimal control scheme, knowledge of the system matrices can be completely unknown. In addition, all iterations can be conducted by using repeatedly the same information collected on some finite intervals. A practical model-free adaptive optimal control algorithm was proposed and has been applied to the optimal glucose regulation for patients with different physiological parameters.

The methodology developed in this paper may serve as a computational tool to extend the current ADP-based framework of continuous-time adaptive optimal control in three directions. First, the proposed scheme can be extended for solving nonlinear problems [33] by employing neural networks (see [9] for example) to approximate the solution of the Hamilton-Jacob-Bellman equation via state and input information. Second, the proposed scheme can be extended for online adaptive learning for nonzero-sum differential games. Third, in our recent work on robust-ADP [13, 14], ADP-based online learning method was combined with tools from nonlinear control theory (for example, Lyapunov designs [19], input-to-state stability theory [27, 28], and nonlinear small-gain techniques [15, 16]). However, partial system knowledge was still required in [13, 14]. Employing the proposed control scheme, we aim to further generalize our results to a class of nonlinear systems with unknown system dynamics and system order.

Appendix

Proof of Lemma 3.1: It amounts to show that the following linear equation

$$\Theta_k X = 0 \quad (20)$$

has only the trivial solution $X = 0$.

To this end, we prove by contradiction. Assume $X = \begin{bmatrix} Y_v^T & Z_v^T \end{bmatrix}^T \in \mathbb{R}^{\frac{1}{2}n(n+1)+mn}$ is a nonzero solution of (20), where $Y_v \in \mathbb{R}^{\frac{1}{2}n(n+1)}$ and $Z_v \in \mathbb{R}^{mn}$. Then, a symmetric matrix $Y \in \mathbb{R}^{n\times n}$ and a matrix $Z \in \mathbb{R}^{m\times n}$ can be uniquely determined, such that $\hat{Y} = Y_v$ and $\text{vec}(Z) = Z_v$.

By (10), we have

$$\Theta_k X = I_{xx}\mathrm{vec}(M) + 2I_{xu}\mathrm{vec}(N) \tag{21}$$

where

$$M = A_k^T Y + Y A_k + K_k^T(B^T Y - RZ) + (YB - Z^T R)K_k, \tag{22}$$
$$N = B^T Y - RZ. \tag{23}$$

Notice that since M is symmetric, we have

$$I_{xx}\mathrm{vec}(M) = I_{\bar{x}}\hat{M} \tag{24}$$

where

$$I_{\bar{x}} = \left[\int_{t_0}^{t_1} \bar{x}d\tau \quad \int_{t_1}^{t_2} \bar{x}d\tau \quad \cdots \quad \int_{t_{l-1}}^{t_l} \bar{x}d\tau \right]^T \in \mathbb{R}^{l\times\frac{n(n+1)}{2}}.$$

Then, (20) implies the following matrix form of linear equations

$$\left[\begin{array}{cc} I_{\bar{x}}, & 2I_{xu} \end{array} \right] \left[\begin{array}{c} \hat{M} \\ \mathrm{vec}(N) \end{array} \right] = 0. \tag{25}$$

Under the rank condition (12), we know $\left[\begin{array}{cc} I_{\bar{x}}, & 2I_{xu} \end{array} \right]$ has full column rank. Therefore, the only solution to (25) is $\hat{M} = 0$ and $\mathrm{vec}(N) = 0$. As a result, we have $M = 0$ and $N = 0$.

Now, by (23) we know $Z = R^{-1}B^T Y$, and (22) is reduced to the following Lyapunov equation

$$A_k^T Y + Y A_k = 0. \tag{26}$$

Since A_k is Hurwitz for all $k \in \mathbb{Z}_+$, the only solution to (26) is $Y = 0$. Finally, by (23) we have $Z = 0$.

In summary, we have $X = 0$. But it contradicts with the assumption that $X \neq 0$. Therefore, Θ_k must have full column rank for all $k \in \mathbb{Z}_+$.

The proof is complete. □

References

[1] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 43(3): 473-481, 2007.

[2] Baird, L. C. III, Reinforcement learning in continuous time: Advantage updating, in *Proc. of IEEE International Conference on Neural Networks*, 1994: 2448-2453.

[3] R. N. Bergman, Y. Z. Ider, C. R. Bowden, and C. Cobelli, Quantitative estimation of insulin sensitivity. *American Journal of Physiology-Endocrinology And Metabolism*, 236(6): 667-677, 1979.

[4] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, Adaptive linear quadratic control using policy iteration, in *Proc. of American Control Conference*, 1994: 3475-3479.

[5] F. Chee, T. Fernando, *Closed Loop Control of Blood Glucose.* Berlin: Springer-Verlag, 2007.

[6] F. Chee, A. V. Savkin, T. L. Fernando, and S. Nahavandi, Optimal H_∞ insulin injection control for vlood glucose regulation in diabetic patients, *IEEE Transactions on Biomedical Engineering*, 52(10): 1625-1631, 2005.

[7] S. Ferrari, J. E. Steck, and R. Chandramohan, Adaptive feedback control by constrained approximate dynamic programming, *IEEE Transcations on System, Man, and Cybernetics, Part B: Cybernetics*, 38(4): 982-987, 2008.

[8] M. Fisher, A semiclosed-loop algorithm for the control of blood glucose levels in diabetics, *IEEE Transactions on Biomedical Engineering,* 38(1):57-61, 1991.

[9] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive neural network control of robotic manipulators*, World Scientific Pub. Co. Inc., 1998.

[10] J. A. Ho, S. C. Zeng, M. R. Huang, and H. Y. Kuo, Development of liposomal immunosensor for the measurement of insulin with femtomole detection, *Analytica Chimica Acta*, 556(1): 127-132, 2006.

[11] R. Hovorka, Continuous glucose monitoring and closed-loop systems, *Diabetic medicine*, 23(1): 1-12, 2006.

[12] P. A. Ioannou and J. Sun, *Robust Adaptive Control*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[13] Y. Jiang and Z. P. Jiang, Robust approximate dynamic programming and global stabilization with nonlinear dynamic uncertainties, in *Proc. of the Joint IEEE Conference on Decision and Control and European Control Conference*, pp. 115-120, 2011.

[14] Y. Jiang and Z. P. Jiang, Robust adaptive dynamic programming, in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, F. L. Lewis and D. Liu, Eds. John Wiley and Sons, 2012.

[15] Z. P. Jiang and L. Praly, Design of robust adaptive controllers for nonlinear systems with dynamic uncertainties, *Automatica*, 34(7): 825-840, 1998.

[16] Z. P. Jiang, A. R. Teel, and L. Praly, Small-gain theorem for ISS systems and applications, *Mathematics of Control, Signals, and Systems*, 7(2):95-120, 1994.

[17] P. Kaveh and Y. Shtessel, Blood glucose regulation using higher order sliding mode control, *International Journal of Robust and Nonlinear Control*, 18: 557-569, 2008.

[18] D. Kleinman, On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1): 114-115, 1968.

[19] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*, John Wiley, 1995.

[20] F. L. Lewis and V. L. Syrmos, *Optimal Control*, Wiley, 1995.

[21] F. L. Lewis and D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Transactions on Circuits and Systems Magazine*, 9(3): 32-50, 2009.

[22] 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*, 41(1): 14-23, 2011.

[23] J. J. Murray , C. J. Cox, G. G. Lendaris, and R. Saeks, Adaptive dynamic programming, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 32(2): 140-153, 2002.

[24] C. Owens, H. Zisser, L. Jovanovic, B. Srinivasan, and D. Bonvin, and J. Doyle III, Run-to-run control of blood glucose concentrations for people with type 1 diabetes mellitus, *IEEE Transactions on Biomedical Engineering*, 53: 996-1005, 2006.

[25] R. Parker, F. Doyle, and N. Peppas, A model-based algorithm for blood glucose control in type 1 diabetic patients, *IEEE Transactions on Biomedical Engineering*, 46: 148-157, 1999.

[26] S. D. Patek, M. D. Breton, Y. Chen, C. Solomon, and B. Kovatchev, Linear quadratic Gaussian-based closed-loop control of type 1 diabetes, *Journal of Diabetes Science and Technology*, 1(6):834-841, 2007.

[27] E. D. Sontag, Further facts about input to state stabilization, *IEEE Transactions on Automatic Control*, 35(4): 473-476, 1990.

[28] E. D. Sontag, Y. Wang, On characterizations of the input-to-state stability property, *Systems & Control Letters*, 24: 351-359, 1995.

[29] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[30] G. Tao, *Adaptive Control Design and Analysis*, Wiley, 2003.

[31] K. G. Vamvoudakis and F. L. Lewis, Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton-Jacobi equations, *Automatica*, 47(8): 1556-1569, 2011.

[32] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, *Automatica*, 45(2): 477-484, 2009.

[33] D. Vrabie and F. L. Lewis, Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems, *Neural Networks*, 22(3): 237-246, 2009.

[34] F.-Y. Wang, H. Zhang, and D. Liu, Adaptive dynamic programming: An introduction, *IEEE Computational Intelligence Magazine*, 4(2): 39-47, 2009.

[35] J. Wang, A. Ibáñez, M. P. Chatrathi, On-chip integration of enzyme and immunoassays: simultaneous measurements of insulin and glucose, *Journal of the American Chemical Society*, 125(28): 8444-8445, 2003.

[36] C. Watkins, *Learning from delayed rewards*, Ph.D. thesis, King's College of Cambridge, 1989.

[37] Q. Wei, H. Zhang, J. Dai, Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions, *Neurocomputing* , 72: 1839-1848, 2009.

[38] P. J. Werbos, *Beyond regression: New tools for prediction and analysis in the behavioural sciences*, Ph.D. thesis, Harvard University, 1974.

[39] P. J. Werbos, Neural networks for control and system identification, in *Proceedings of IEEE Conference on Decision and Control*, 1989: 260-265.

[40] P. J. Werbos, Intelligence in the brain: A theory of how it works and how to build it, *Neural Networks*, 22(3): 200-212, 2009.

[41] H. Zhang, Q. Wei, and D. Liu, An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games, *Automatica*, 47(1): 207-214, 2011.

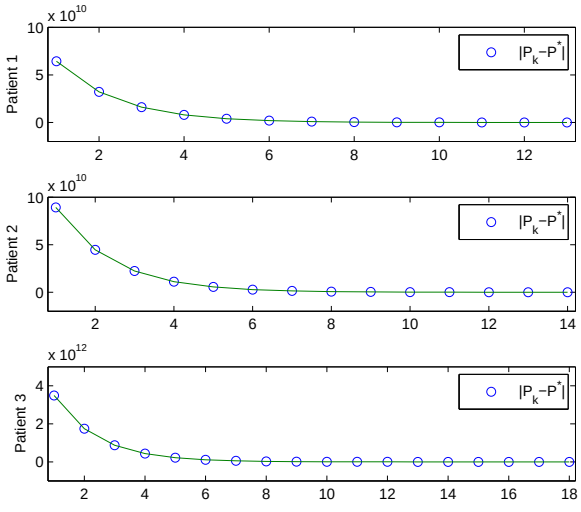

Fig. 1: Convergence performance of the cost matrices.

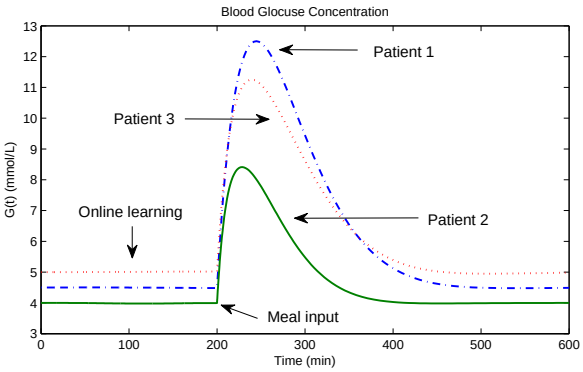

Fig. 2: Trajectories of glucose level.

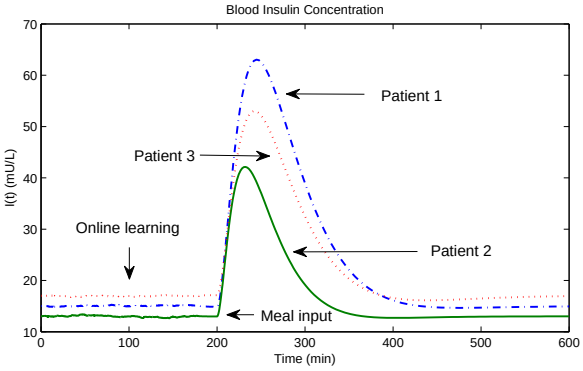

Fig. 3: Trajectories of insulin level.

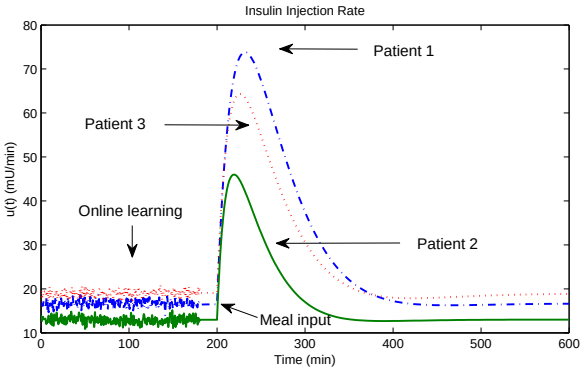

Fig. 4: Trajectories of insulin injection rate.

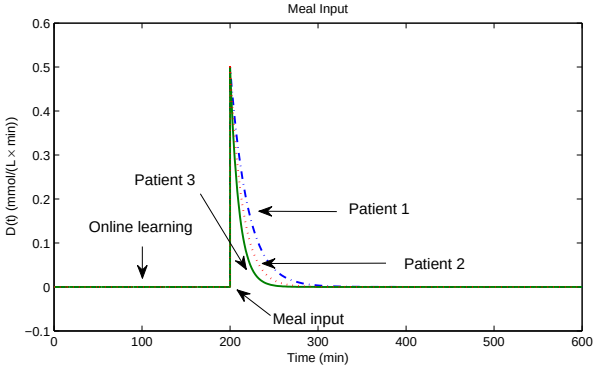

Fig. 5: Trajectories of meal input.