Control Engineering Practice 26 (2014) 151–159

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A learning automata-based blood glucose regulation mechanism in type 2 diabetes

Javad Akbari Torkestani [a,*], Elham Ghanaat Pisheh [b]

[a] Department of Computer Engineering, Arak Branch, Islamic Azad University, Arak, Iran
[b] Research Center of Health in Emergency and Disaster, University of Social Welfare and Rehabilitation Sciences, Tehran, Iran

ARTICLE INFO

Article history:
Received 13 March 2012
Accepted 6 January 2014
Available online 14 February 2014

Keywords:
Blood glucose regulation
Type 2 diabetes
Insulin therapy
Run-to-run controller
Learning automata

ABSTRACT

This paper proposes a learning automata-based mechanism for blood glucose regulation in type 2 diabetics. The proposed mechanism takes into account the past history of the blood glucose level to determine the correct dosage of the insulin. This method uses the learning automata theory to predict the required dosage of insulin and records the patient history in parameters of a Gaussian probability distribution function. The parameters of the distribution function are updated based on the difference between the actual glucose level regulated by the learning automata and the normal range in such a way that the gap between the actual glucose level and the normal one is minimized. As the proposed algorithm proceeds, it can be seen that it converges to the optimal insulin dosage that keeps the glucose level in normal range for a long time. Convergence of the proposed algorithm to the optimal insulin dosage is theoretically proven. A clinical study is conducted to show the performance of the proposed insulin therapy system for regulation of the blood glucose level of type 2 diabetics.

1. Introduction

The cells of the Pancreas (islets of Langerhans) are responsible for monitoring and regulating the blood glucose level. The normal blood glucose level in the human body varies in a narrow range for example between 70 and 110 mg/dl (milligrams per deciliter) or 4.0 and 6.30 mmol/l (millimoles per litre). If the blood glucose level falls down to a dangerous level (due to a heavy exercise or lack of food), the Alpha cells of the pancreas release glucagon that effects on the liver cells to increase the blood glucose level. On the other side, when the blood glucose level rises, a different hormone so called insulin is released from the Beta cells of the Pancreas. Insulin brings about more glucose is converted into glycogen by the Liver. If for some reasons the Pancreas does not perform normally (i.e., it is unable to control the normal glucose–insulin interaction), diabetes is diagnosed. Besides physical exercise and managing diet, insulin therapy is a necessary and often irreplaceable partner to tackle the hyperglycemia. Improper administration of insulin is quite dangerous and may severely harm the body in long term. Intensive insulin therapy requires close monitoring of the glucose level, a great deal of patient education, and even a good understanding of the insulin pharmacokinetics that cannot be properly done by the diabetics specially children and older people. An automatic closed-loop blood glucose control system is a promising approach to mitigate these problems (Bequette, 2005; Chee, Fernando, Savkin, & van Heeden, 2003; Lin et al., 2004).

A closed-loop insulin delivery system generally has three major components. First, a monitoring device (blood glucose sensor) that automatically measures the blood glucose level at appropriate intervals and electronically sends the results to the decision system. Second, an automated decision system that assesses the patient condition and determines the amount of insulin needs to be administered to keep the blood glucose in a good control. This is the main part of the closed-loop system that the performance of the system strongly depends on its decisions. Third, an insulin pump that is instructed to inject the right dose of the insulin to the person's body. Future closed-loop systems are expected to continuously monitor the blood glucose level and automatically deliver the correct insulin dosage to the patient to keep the blood glucose within an acceptable range (Lin et al., 2004; Ibbini, 2006; Hernjak & Doyle, 2005).

During the last decades, lot of researchers investigated the glucose–insulin interaction issue and several automated control systems were proposed. PID (proportional–integral derivative) classical blood sugar controller (Chee et al., 2003), optimal glucose level control mechanism (Ibbini, Masadeh, & BaniAmer, 2004), adaptive glucose controller (Lin et al., 2004), neuro-fuzzy control algorithms (Ibbini, 2006; Dazzia et al., 2001), model predictive control methods (Hernjak & Doyle, 2005; Hovorka et al., 2004; Lynch & Bequette, 2002), H_∞ control technique (Parker, Doyle,

* Corresponding author.
E-mail addresses: j-akbari@iau-arak.ac.ir (J. Akbari Torkestani), el.ghanaatpisheh@uswr.ac.ir (E. Ghanaat Pisheh).

0967-0661/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.conengprac.2014.01.010

Ward, & Peppas, 2000; Ruiz-Velazquez, Femat, & Campos-Delgado, 2004), μ-synthesis controller (Kovacs, Kulcśar, & Benýo, 2006), Linear Parameter Varying (LPV) control method (Kovacs, Kulcśar, Bokor, & Benýo, 2006), and H_2/H_∞ controller (Kovacs, Paíancz, Almassy, & Benýo, 2004) are well-known glucose control approaches reported in the literature (Kovacs, Benýo, Benýo, & Kovacs, A, 2009).

Almost all known automated closed-loop glucose control mechanisms aim at regulating the glucose level within the normal range based solely on the results of the current sampling tests. Such methods do not take into consideration the long-term history of the blood glucose level, while an effective decision on the drug dosage by which the glucose level for a long time remains in a normal range strongly depends on the patient's past history. In this paper, a Learning automata-based Blood Glucose Regulation mechanism so called LBGR is designed in which the past history of the blood glucose level is recorded to make a long-term decision on the insulin dosage. By using a continuous action-set learning automaton, LBGR prescribes an insulin dosage for the patient based on the history recorded in a probability distribution function. The selected dose is injected and the blood glucose level is measured. Depending on the difference between the actual glucose level and the normal range, LBGR updates the parameters of the probability distribution function in such a way that the gap between the actual glucose level and the normal one decreases in the next tests. After a number of sampling tests, it can be seen that LBGR selects the optimal insulin dosage by which the glucose level is kept in a normal range as long as possible. The convergence of the learning automata to the optimal insulin dosage is theoretically proved.

The rest of the paper is organized as follows. Section 2 briefly reviews the learning automata theory. In Section 3, the proposed blood glucose regulation mechanism is presented. In Section 4, a clinical study is conducted to show the performance of the proposed insulin therapy system. Section 5 concludes the paper. In Appendix A, the correctness of the proposed system is theoretically proved. This section shows the convergence of the proposed glucose regulation system to the optimal insulin dose with which the blood glucose level remains in the normal range.

2. Learning automata theory

A learning automaton (Narendra & Thathachar, 1989; Thathachar & Harita, 1987) is an adaptive decision-making unit that improves its performance by learning how to choose the optimal action from a finite set of allowed actions through repeated interactions with a random environment. The action is chosen at random based on a probability distribution kept over the action-set and at each instant the given action is served as the input to the random environment. The environment responds the taken action in turn with a reinforcement signal. The action probability vector is updated based on the reinforcement feedback from the environment. The objective of a learning automaton is to find the optimal action from the action-set so that the average penalty received from the environment is minimized. Learning automata have been found to be useful in systems where incomplete information about the environment, in which those systems operate, exists. Learning automata are also proved to perform well in complex, dynamic and random environments with a large amount of uncertainties (Poznyak & Najim, 1997). To name just a few, learning automata have a wide variety of applications in combinatorial optimization problems (Akbari Torkestani, 2013e, 2013g), computer networks (Akbari Torkestani, 2013a, 2013b, 2013d, 2013h), Grid computing (Akbari Torkestani, 2012b, 2012e, 2013c, 2013f), and Web engineering (Akbari Torkestani, 2012a, 2012c, 2012d).

The environment can be described by a triple $\{\alpha,\beta,c\}$, where $\alpha\equiv\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$ represents the finite set of the inputs, $\beta\equiv\{\beta_1,\beta_2,\ldots,\beta_m\}$ denotes the set of the values that can be taken by the reinforcement signal, and $c\equiv\{c_1,c_2,\ldots,c_r\}$ denotes the set of the penalty probabilities, where the element c_i is associated with the given action a_i. If the penalty probabilities are constant, the random environment is said to be a stationary random environment, and if they vary with time, the environment is called a non stationary environment. The environments depending on the nature of the reinforcement signal β can be classified into *P*-model, *Q*-model and *S*-model. The environments in which the reinforcement signal can only take two binary values 0 and 1 are referred to as *P*-model environments. Another class of the environment allows a finite number of the values in the interval [0, 1] can be taken by the reinforcement signal. Such an environment is referred to as *Q*-model environment. In *S*-model environments, the reinforcement signal lies in the interval $[a,b]$.

Learning automaton can be generally classified into two main families: Finite action-set learning automata (FALA) and continuous action-set learning automata (CALA) (Narendra & Thathachar, 1989). The action-set of FALA is finite, for example for an *r*-action $(2\le r<\infty)$ FALA, the action probability distribution is represented by an *r*-dimensional probability vector and is updated by a learning algorithm. When the FALA is used for solving the optimization problems, we need to discretize the parameter space so that the actions of the learning automaton can be possible values of the corresponding parameter. A large action-set leads to slow convergence of the learning algorithm. To provide a higher convergence rate, the continuous action-set learning automaton is proposed. The action-set of such a learning automaton is the real line. The following provides a brief review of FALA and CALA.

2.1. Finite action-set learning automata

FALA can be classified into two main families: fixed structure learning automata and variable structure learning automata (the readers are referred toNarendra & Thathachar (1989) for more information). Variable structure learning automata are represented by a triple $<\beta,\alpha,T>$, where β is the set of inputs, α is the set of actions, and T is learning algorithm. The learning algorithm is a recurrence relation which is used to modify the action probability vector. Let $\alpha_i(k)\in\alpha$ and $p(k)$ denote the action selected by learning automaton and the probability vector defined over the action set at instant k, respectively. Let a and b denote the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities, respectively. Let r be the number of actions that can be taken by learning automaton. At each instant k, the action probability vector $p(k)$ is updated by the linear learning algorithm given in Eq. (1), if the selected action $\alpha_i(k)$ is rewarded by the random environment, and it is updated as given in Eq. (2) if the taken action is penalized.

$$p_j(k+1)=\begin{cases}p_j(k)+a[1-p_j(k)] & j=i\\ (1-a)p_j(k) & \forall j\neq i\end{cases}. \tag{1}$$

$$p_j(k+1)=\begin{cases}(1-b)p_j(k) & j=i\\ \left(\frac{b}{r-1}\right)+(1-b)p_j(k) & \forall j\neq i\end{cases}. \tag{2}$$

If $a=b$, the recurrence Eqs. (1) and (2) are called linear reward-penalty (L_{R-P}) algorithm, if $a\gg b$ the given equations are called linear reward-ϵ penalty ($L_{R-\epsilon P}$), and finally if $b=0$ they are called linear reward-Inaction (L_{R-I}). In the latter case, the action probability vectors remain unchanged when the taken action is penalized by the environment.