Automatica 64 (2016) 234–239

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Lebesgue approximation model of continuous-time nonlinear dynamic systems ☆

Xiaofeng Wang, Bin Zhang

Department of Electrical Engineering, University of South Carolina, Columbia, SC, 29208, USA

ARTICLE INFO

Article history:
Received 30 March 2014
Received in revised form
29 June 2015
Accepted 31 October 2015
Available online 7 December 2015

Keywords:
System approximation
Lebesgue measurement
Sporadic model
Model-based
Fault prognosis

ABSTRACT

Traditional model-based approaches are based on periodic iterations, where the continuous-time model is discretized with a fixed period. Despite the easiness in analysis and design, such periodic approximation model may be undesirable from the computation-efficiency point of view. This paper presents the Lebesgue-approximation model (LAM) of continuous-time nonlinear systems, where the iteration is activated on an "as-needed" basis, but not periodically. We show that the proposed LAM behaves exactly the same as a specific event-triggered feedback system, through which the properties of the LAM can be studied. We provide a sufficient condition to ensure asymptotic stability of the LAM and derive theoretical bounds on the difference between the states of the LAM and the original continuous-time system. The LAM is then integrated in the particle-filtering approach for fault prognosis. Simulation results show that the LAM can dramatically reduce the number of iterations in prognosis without sacrificing accuracy and precision.

1. Introduction

Model-based approaches are widely used in control community. In such approaches, the physical dynamics can be described by a mathematical model, based on which one can address problems associated with controller synthesis and analysis, optimization, and prediction, etc. When implementing model-based algorithms in computers, the traditional method is to approximate the model with a fixed period over the time horizon and iterations take place periodically. Such a periodic approximation model is known as "Riemann Approximation model" (RAM) (Åström & Bernhardsson, 2002).

A nice feature of using RAM is the easiness in analysis and design. However, it may be undesirable in many situations, from the computation-efficiency point of view. On the one hand, since the iteration period is determined according to the worst-case operating scenario, the state of the model might be iterated even if there is little change in the actual states. In other words, it may take greater computational resources than it actually needs. This may result in significant over-provisioning of the real-time system hardware, in particular for embedded processors which have very limited computation ability (Tabuada, 2007; Wang & Lemmon, 2009). On the other hand, when the system becomes unstable, the state will diverge exponentially. In this case, it is expected to perform iterations more frequently so that the model can closely track the actual state, which cannot be met by RAM. It suggests cost-efficient approaches to construct the approximation models where iterations can be executed on an "as-needed" basis.

One possible solution is the discrete event simulation (DEVS) formalism, where the concept of the quantized state system (QSS) was introduced for real-time simulation (Zeigler, Praehofer, & Kim, 2000). The basic idea of QSS is to first quantize the state space based on a pre-defined quantizer $\mathcal{Q}(\cdot)$ and then define transitions between the pre-defined quantized states characterized by $\mathcal{Q}(\cdot)$. The frequency of state transition really depends on the state of the model, but not directly on the fixed amount of time elapsed. The existence of stabilizing quantizers has been established in Kofman (2002), Kofman and Junco (2001) and Zeigler et al. (2000). This approach was further explored in Cellier and Kofman (2006), Cellier, Kofman, Migoni, and Bortolotto (2008) and Kofman and Junco (2001), which extends the original approach to illegitimate models. These results indicate that the DEVS promises a significant improvement of computational efficiency in real-time simulations of large and complex systems such as power systems (Mamai, Smith, Kondratiev, & Dougal, 2011). Although QSS demonstrates great advantages in cost-efficiency, there is a lack of theoretical

☆ The authors gratefully acknowledge the partial financial support of the National Science Foundation (CNS-1329870). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Maurice Heemels under the direction of Editor Andrew R. Teel.

E-mail addresses: wangxi@cec.sc.edu (X. Wang), zhangbin@cec.sc.edu (B. Zhang).

http://dx.doi.org/10.1016/j.automatica.2015.11.016
0005-1098/© 2015 Elsevier Ltd. All rights reserved.

foundation and design methods to construct stabilizing quantizers. It is also worth mentioning that all of the prior work on QSS focuses on the uniform quantizer.

Inspired by the QSS approach, this paper considers sporadic discretized model of dynamic systems. Different from the work in Meng and Chen (2012) and Xu and Cao (2011) using stochastic approaches, the model developed in this paper is completely deterministic. Another approach similar in spirit is Lebesgue piecewise affine approximation in Azuma, Imura, and Sugie (2010), where the state space is partitioned into a number of sets and the nonlinear dynamics can be approximated by a linear system over each set. However, the state space is partitioned independent of the initial state in Azuma et al. (2010) and the nonlinearity has to take a special form $f(x) = \sum_{k=1}^{N} \prod_{i=1}^{n} f_{ik}(x_i)$ where x_i is the ith element of x. Moreover, the time information is not explicitly characterized in this approach.

This paper considers sporadic approximation models of multi-dimension continuous-time nonlinear systems, including aperiodic iterations in both state and time. We study stability of this model and its closeness to the continuous-time system. The contributions of this paper are as follows: (i) We present the Lebesgue approximation model (LAM), where the next state is computed only when it is certain distance away from the previous state. Different from the QSS, the states in the LAM do not have to be pre-defined. Another significant difference is that the quantizer in the LAM does not have to be uniform. Compared with Azuma et al. (2010), the initial state and iterations in time are taken into account in the state space partition; (ii) We develop sufficient conditions to ensure asymptotic stability of the LAM without exhibits of Zeno behavior. Theoretical bounds are derived to quantify the difference between the states of the LAM and the continuous-time system; and (iii) We apply the LAM in model-based fault prognosis. Experiments show that the LAM-based prognosis can dramatically reduce the computational costs without sacrificing accuracy.

2. Problem formulation

Notations. We denote by $\mathbb{R}^n$ the n-dimensional real vector space, by $\mathbb{R}^+$ the set of the real positive numbers. We use $\|\cdot\|$ to denote the Euclidean norm of a vector and the induced 2-norm of a matrix. The symbol e is used for exponential function to distinguish it from the error e. Given two functions $\phi, \psi : \mathbb{R} \to \mathbb{R}$, we define $\phi \circ \psi(t) = \phi(\psi(t))$. Given a differentiable function $f(x)$, its gradient is denoted by $\nabla f(x)$. For a function of time $x(t)$, sometimes we drop t for brevity if it is clear in context.

Consider an autonomous system:

$$\dot{x}(t) = f(x(t)), \qquad x(t_0) = x_0 \tag{1}$$

where $x : \mathbb{R} \to \mathbb{R}^n$ is the system state, $f : \mathbb{R}^n \to \mathbb{R}^n$ is a known function, and $x_0 \in \mathbb{R}^n$ is the initial state. In general, the discretized model of this continuous-time system can be described by the iterative equations:

$$\hat{x}(t_{k+1}) = \hat{f}\left(\hat{x}(t_k)\right) \tag{2a}$$

$$\hat{x}(t_0) = x_0 \tag{2b}$$

$$t_{k+1} = t_k + \hat{g}\left(\hat{x}(t_k)\right) \tag{2c}$$

where the functions $\hat{f}$ and $\hat{g}$ describe the iteration in the state and the time, respectively, which are to be determined. The discrete-time signal $\hat{x}(t_k)$ is the state of this discrete-time model at t_k. Starting from $\hat{x}(t_0)$, this model will generate two sequences: the time sequence $\{t_k\}_{k=0}^{\infty}$ and the state sequence $\{\hat{x}(t_k)\}_{k=0}^{\infty}$.

Once the discrete-time model is established, we can use different interpolation methods to construct the states over (t_k, t_{k+1}) such as $\hat{x}(t) = h\left(\hat{x}(t_k), \hat{x}(t_{k+1}), t\right)$ for any $t \in (t_k, t_{k+1})$. Thus, the discrete-time model (2) and the interpolation generate a continuous-time trajectory $\hat{x}(t)$, which is expected to approximate the actual state $x(t)$ of the continuous-time system in (1).

The objective is to construct sporadic real-time models in the form of (2) and develop guidelines on the quantization design that ensure stability of the resulting model as well as its closeness to the continuous-time system in (1), with the cost-efficient purpose.

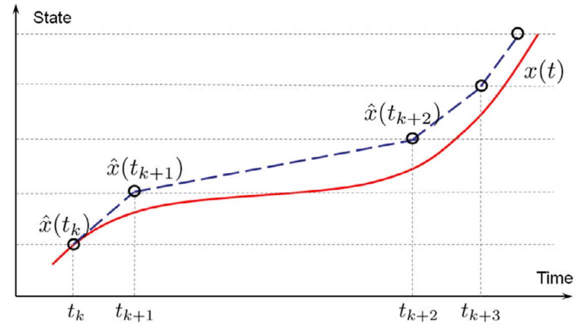

Fig. 1. The state trajectories generated by the LAM and the continuous-time system.

Remark 2.1. Notice that the LAM is completely different from the self-triggered feedback scheme (Anta & Tabuada, 2010; Wang & Lemmon, 2009). Self-triggered control only needs the iteration equation on time (Eq. (2c)), while the states are always sampled from the plant. The signal $\hat{x}(t_k)$ in the LAM is not the sampled state, but the predicted state that is completely based on the model itself.

3. Lebesgue approximation model

The basic idea of the LAM is to have iterations executed only when the state is "significantly" different from the previous one. The "significance" is measured by the threshold $D(x)$, which is also referred to the quantization size (Kofman & Junco, 2001). The LAM is described as follows:

$$\hat{x}(t_{k+1}) = \hat{x}(t_k) + D(\hat{x}(t_k)) \cdot \frac{f\left(\hat{x}(t_k)\right)}{\left\|f\left(\hat{x}(t_k)\right)\right\|} \tag{3a}$$

$$\hat{x}(t_0) = x_0 \tag{3b}$$

$$t_{k+1} = t_k + \frac{D(\hat{x}(t_k))}{\left\|f\left(\hat{x}(t_k)\right)\right\|} \tag{3c}$$

where $\hat{x}(t_k)$ is called the Lebesgue state and $D : \mathbb{R}^n \to \mathbb{R}^+$ represents the quantization size.

We consider linear interpolation to construct the states between $\hat{x}(t_k)$ and $\hat{x}(t_{k+1})$: for any $t \in (t_k, t_{k+1})$,

$$\hat{x}(t) = \hat{x}(t_k) + f\left(\hat{x}(t_k)\right) \cdot (t - t_k). \tag{4}$$

Fig. 1 shows the state trajectories generated by a scalar continuous-time system ("solid") and the related LAM ("dashed") with uniform quantization size D. Assume that at t_k, $\hat{x}(t_k) = x(t_k)$. The next state $\hat{x}(t_{k+1})$ is simply $\hat{x}(t_k) + D$. The inter-transition time $t_{k+1} - t_k$ is based on the slope at $\hat{x}(t_k)$. This iteration will generate a gap between $\hat{x}(t_{k+1})$ and $x(t_{k+1})$ and this gap will be propagated into the next iterations, which may possibly be amplified. Thus, it is important to study stability of the LAM and how close it is compared with the continuous-time system.

4. Equivalence to event-triggered systems

To show the properties of the LAM, it might be difficult to directly analyze the model by itself. Alternatively, we build up