Asynchronous Lebesgue Approximation Model for Distributed Continuous-Time Nonlinear Systems

Ying Shen, Xiaofeng Wang and Zheng-Guang Wu

***Abstract*— An appropriate approximation model can significantly reduce the computational costs in model-based approaches. This paper aims to develop discrete-time models to approximate distributed continuous-time nonlinear dynamical systems, in which subsystems are physically coupled and can receive information from their neighbors. To approximate such a system, we present asynchronous Lebesgue approximation approach, where each subsystem is approximated by an individual Lebesgue approximation model (LAM). Each LAM updates its state, depending on its own state as well as the neighboring states. Different LAMs execute asynchronously. The proposed distributed LAM is cost-efficient because it can automatically adjust its iteration frequency based on state's variation. To show stability of the distributed LAM, we construct a distributed event-triggered feedback system and prove that it generates the same state trajectories as the LAM with linear interpolation. Through this specific distributed event-triggered system, we show that the distributed LAM is uniformly ultimately bounded. Finally, we carry out some simulations on a nonlinear system to show the efficiency of the proposed method.**

I. Introduction

System model always plays an important role in model-based approaches [1]–[3]. Therefore, system modeling becomes a fundamental issue in control area, which receives a lot of attentions from the community [4], [5]. It is known that most of the physical systems are essentially continuous-time, whereas discrete-time models are required in some scenarios for prediction and planning, e.g., model-based path planning [6] and model predictive control [7].

To accommodate the discrete-time scenarios, one traditional method is to discretize the continuous-time state with a fixed period [8]. The advantage and disadvantage of periodic discretization are both obvious. Certainly, the resulting model is easy to design and analyze. The computation cost, however, could be high because its iteration frequency keeps constant no matter the rate of state change is high or low. In other words, iterations have to take place even when the state remains unchanged and, on the other hand, the model state may not be able to track the continuous-time state when it varies rapidly. Thus, a more flexible method is required, which is capable of dynamically adjusting its iteration frequency.

One possible solution is the quantized state system (QSS) [9], which discretizes the system along the state axis instead of the time axis. The state value is confined in a set with pre-defined state values. The work in [10]–[13] improves the approach in [9] by solving the issue of illegitimate models, and indicates that the QSS promises a significant improvement of computational efficiency in real-time simulations of large and complex systems, such as power systems. However, all of this aforementioned work only focuses on scalar systems with uniform quantizers and did not provide systematic design methods to construct stabilizing quantizers.

The Lebesgue approximation model (LAM) is an efficient discretization method. To some extent, it is similar to the quantized state system whose states are quantized. The main difference is that the states of the LAM are not pre-defined and the quantization size is not necessarily uniform. It has been stated in [14] that the LAM of a single system is equivalent to an event-triggered system with a specific structure in terms of state trajectories, which greatly facilitates the analysis and design of the LAM. Despite of such a connection, it is important to note that the LAM is essentially different from event-triggered system. The states of the former evolve only based on the model itself and they are predicted values of the actual continuous-time states at certain time instants, whereas the states of the latter are sampled values of the actual states. Due to its cost efficiency, the LAM has been applied in many areas, e.g. fault diagnosis [15] and model predictive control [16]. Nevertheless, to the best of our knowledge, the development of distributed LAM has not been studied yet, which inspires our present work.

Given a continuous-time distributed nonlinear system composed of a number of connected subsystems (or called "agents") with information exchange through a communication network, the distributed LAM in this paper is developed as follows: for each subsystem we develop an individual LAM that involves its own state and the states of its neighboring LAM. All LAMs iterates at an asynchronous manner. The collection of the LAMs form the distributed LAM. To analyze the stability of the resulting distributed LAM, we construct a distributed event-triggered feedback system and prove the equivalence between this system and the LAM. It is shown that the constructed distributed event-triggered system is uniformly ultimately bounded (UUB), which indicates that the LAM is also UUB. The efficiency of the proposed algorithm is exhibited through numerical simulations.

This work was partially supported by the National Natural Science Foundation of China (61621002) and the National Science Foundation (IIS-1525900 and ECCS1739886).

Y. Shen and Z.-G. Wu are with the National Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Yuquan Campus, Hangzhou, Zhejiang, 310027, China yingshen@zju.edu.cn; nashwzhg@zju.edu.cn

X. Wang is with the Department of Electrical Engineering at the University of South Carolina, Columbia, SC, 29208, USC wangxi@cec.sc.edu

II. Problem Formulation

In this paper, we consider a distributed nonlinear dynamical system which consists of M agents. Let $\mathcal{M} = \{1, 2, \cdots, M\}$ be the set of the agents. These agents are physically coupled and this coupling is characterized by a coupling graph which is defined as follows:

Definition 1: A graph $\mathscr{G}_{cp} = (\mathcal{M}, \Xi_{cp})$ is called a coupling graph, where each node $i \in \mathcal{M}$ denotes an agent, the ordered pair (edge) (i, j) in Ξ_{cp} means that agent j is directly driven by agent i.

For notational convenience, we further present the following notations:

$\mathcal{D}_{\to i} \triangleq \{j \in \mathcal{M} | (j, i) \in \Xi_{cp}\}$ are the set of agents that directly drive agent i;

$\mathcal{D}_{i\to} \triangleq \{j \in \mathcal{M} | (i, j) \in \Xi_{cp}\}$ are the set of agents which are directly driven by agent i;

$\bar{\Upsilon}_i \triangleq \Upsilon_i \cup \{i\}$ for any set $\Upsilon_i \in \{\mathcal{D}_{\to i}, \mathcal{D}_{i\to}\}$. $|\Upsilon|$ represents the number of elements in the set Υ for any $\Upsilon \subseteq \mathcal{M}$.

Based on the notations above, the dynamics of agent i can be described as follows:

$$\begin{aligned} \dot{x}_i(t) &= f_i(x_{\bar{\mathcal{D}}_{\to i}}(t), u_i(t)) \\ x_i(t_0) &= x_{i,0} \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^n$ is the agent i's state with initial value $x_{i,0}$, $u_i \in \mathbb{R}^m$ denotes control input, $x_{\bar{\mathcal{D}}_{\to i}} = \{x_j\}_{j \in \bar{\mathcal{D}}_{\to i}}$, $f_i : \mathbb{R}^{n|\bar{\mathcal{D}}_{\to i}|} \times \mathbb{R}^m \to \mathbb{R}^n$ is a locally Lipschitz function and satisfies $f_i(0, 0) = 0$. Without loss of generality, we make an assumption that the states and inputs of all agents in the distributed NCS have the same dimensions. In fact, it is easy to extend it to the case with different dimensions.

When we use the LAM to approximate the ith subsystem in (1), it should have the following structure based on [14]:

$$\begin{aligned} \hat{x}_i(t_{i,k+1}) &= \hat{x}_i(t_{i,k}) \\ &\quad + D_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k})) \frac{f_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}), u_i(t_{i,k}))}{||f_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}), u_i(t_{i,k}))||} \\ \hat{x}_i(t_0) &= x_{i,0} \\ t_{i,k+1} &= t_{i,k} + \frac{D_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}))}{||f_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}), u_i(t_{i,k}))||} \\ t_{i,0} &= t_0, i = 1, 2, \cdots, M \end{aligned} \tag{2}$$

The continuous-time dynamics (1) is approximated by (2) at time instants $t_{i,k}$ with the Lebesgue state $\hat{x}_i(t_{i,k})$ and input $u_i(t_{i,k})$. Note that $\hat{x}_i(t_{i,k})$ is not a sampled value but a predicted value of the continuous-time system's state $x_i(t)$. $D_i : \mathbb{R}^{n|\bar{\mathcal{D}}_{\to i}|} \to \mathbb{R}^+$ denotes agent i's quantization size which depends on appropriate state values $\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}) = \{\hat{x}_j(t_{j,k})\}_{j \in \bar{\mathcal{D}}_{\to i}}$. The function $D_i(\cdot)$ is an important element which helps determine when the next prediction takes place and what the next predicted value is.

The model in equation (2) is far from enough to be a satisfying approximation of the continuous-time model. Notice that the progress of different LAMs in (2) is different because of their aperiodic nature. It may take only a few iterations for some LAMs to predict their states over a long time horizon, while other LAMs are still predicting states over a very short time window. Therefore, it is very important to appropriately use the states of the neighboring LAMs; Otherwise, for example, it may happen that at the kth iteration the ith LAM uses the jth LAM's predicted state at time instant $t_{j,k}$ to predict its own state at $t_{i,k}$, while $t_{j,k} \gg t_{i,k}$. Using the wrong states, the approximation may deviate from the continuous-time system.

Given such a distributed approximation framework, this paper investigates asynchronous iterations in the LAM for nonlinear dynamical systems with appropriate scheduling methods.

III. The Asynchronous LAM

This section will develop the distributed approximation LAM with the right scheduling algorithm to approximate the system described by (1). In our scheme, an LAM will be established for each agent i as indicated in (2), and these established LAMs share the same coupling graph $\mathscr{G}_{cp}$ as the agents in $\mathcal{M}$ do. Unlike the model designed for only a single system, in the distributed LAM, the states and iteration time instants depend not only on individual LAM's sates but also the states of those LAMs in the set $\mathcal{D}_{\to i}$.

The proposed algorithm is composed of two stages: prediction and update. The ith LAM for agent i predicts its next Lebesgue state $\hat{x}_i(t_{i,k^i+1})$ and next time instant t_{i,k^i+1}. Then these predicted time instants t_{i,k^i+1} will be transmitted to its neighbors in $\mathcal{D}_{i\to}$. Let $t^* = \min_{i \in \mathcal{M}}\{t_{i,k^i+1}\}$. After the prediction stage, the ith LAM will update its Lebesgue state if $t_{i,k^i+1} = t^*$ or it receives some "important" states from other LAM $j \in \mathcal{D}_{\to i}$ at time instant t^*. By "important", it means that those states will be used in the next prediction. After that, those LAMs which have updated their states will go to the next prediction stage, while the other LAMs will hold.

More specifically, our distributed approximation algorithm executes as follows: For $j \in \mathcal{M}$, if $t_{j,k^j+1} = t^*$, then the jth LAM will update its state at time instant t^* as it was predicted; Also, the update in the jth LAM may trigger the updates of its neighbors in $\mathcal{D}_{j\to}$ at the same time instant t^*, based on the following condition:

$$||\hat{x}_j(t_{j,k}) - x^*_{j\to i}|| \geq \psi(x^*_{j\to i}) \tag{3}$$

where $x^*_{j\to i}$ denotes the jth LAM's Lebesgue state used by the ith LAM in its latest prediction, $\psi(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a pre-defined function of $x^*_{j\to i}$. Condition (3) implies that the distance from the last used Lebesgue state to the latest available state has exceeds a certain threshold. If (3) holds, the ith LAM will update its state at time instant t^* following the equations below:

$$\begin{aligned} t_{i,k^i+1} &= t^* \\ D^*_{i,k^i} &= (t_{i,k^i+1} - t_{i,k^i}) ||f_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k^i}), u_i(t_{i,k^i}))|| \\ \hat{x}_i(t_{i,k^i+1}) &= \hat{x}_i(t_{i,k^i}) + D^*_{i,k^i} \frac{f_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k^i}), u_i(t_{i,k^i}))}{||f_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k^i}), u_i(t_{i,k^i}))||} \end{aligned} \tag{4}$$

where D^*_{i,k^i} is the quantization size for updating when the update is triggered by its neighbors. D^*_{i,k^i} is inferred from the time difference between t^* and t_{i,k^i}. It should be noted that the ith LAM's update will possibly trigger the updates of its neighbors in $\mathcal{D}_{i\rightarrow}$ in the same manner described by (4).

IV. Stability Analysis

This section will study the stability of the distributed LAM. Directly analyzing its stability seems difficult. As an alternative, we first introduce a distributed event-triggered feedback system and show its state trajectories are identical to those of the distributed LAM. Consequently, we are able to know the stability of the distributed LAM by studying the equivalent distributed event-triggered system. In the LAM, the states between $\hat{x}_i(t_{i,k^i})$ and $\hat{x}_i(t_{i,k^i+1})$ are constructed by linear interpolation (to simplify the notations, we drop the index i in k^i if it is clear in context):

$$\hat{x}_i(t) = \hat{x}_i(t_{i,k}) + f(\hat{x}_i(t_{i,k}), u_i(t_{i,k}))(t - t_{i,k}) \tag{5}$$

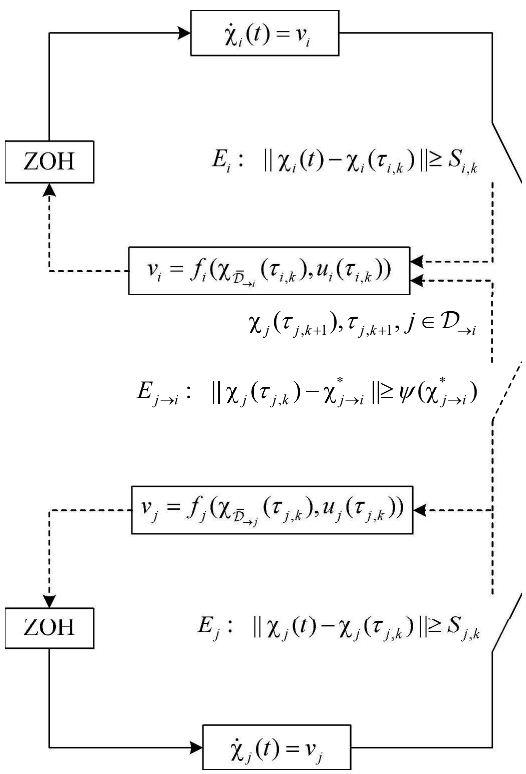

Fig. 1. The equivalent event-triggered feedback system

The constructed distributed event-triggered feedback system is shown in Fig. 1. It consists of M event-triggered systems and has the same coupling graph $\mathscr{G}_{cp}$ as the distributed LAM. The open-loop plant is described as follows:

$$\begin{aligned} \dot{\chi}_i(\tau) &= v_i(\tau) \\ \chi_i(\tau_0) &= x_{i,0} \\ \tau_0 &= t_0 \end{aligned} \tag{6}$$

where $\chi_i \in \mathbb{R}^n$ and $v_i \in \mathbb{R}^n$, $i = 1, 2, \cdots, M$ are the states and inputs of the plants, respectively. The feedback control is generated as follows:

$$v_i(\tau) = f_i(\chi_{\bar{\mathcal{D}}_{\rightarrow i}}(\tau_{i,k}), u_i(\tau_{i,k})), \ \forall \tau \in [\tau_{i,k}, \tau_{i,k+1}) \tag{7}$$

which is based on the latest sampled state values $\chi_{\bar{\mathcal{D}}_{\rightarrow i}}(\tau_{i,k}) = \{\chi_j(\tau_{j,k})\}_{j\in\bar{\mathcal{D}}_{\rightarrow i}}$, $v_i(\tau)$ keeps constant between adjacent sampling instants $\tau_{i,k}$ and $\tau_{i,k+1}$.

For every system i, there are two classes of event sources: inner event E_{in} and outer event E_{out}. E_{in} is defined by

$$E_{in} : ||\chi_i(\tau) - \chi_i(\tau_{i,k})|| \geq D_i(\chi_{\bar{\mathcal{D}}_{\rightarrow i}}(\tau_{i,k})) \tag{8}$$

and we further define the inner triggering time instant:

$$\tau^{in}_{i,k+1} = \inf_{\tau \geq \tau_{i,k}} \{\tau \mid E_{in}\} \tag{9}$$

Outer event E_{out} comes from the systems in $\mathcal{D}_{\rightarrow i}$, it happens when both E_j and $E_{j\rightarrow i}$, $\forall j \in \mathcal{D}_{\rightarrow i}$ take place:

$$E_j : ||\chi_j(\tau) - \chi_j(\tau_{j,k})|| \geq S_{j,k} \tag{10}$$

$$E_{j\rightarrow i} : ||\chi_j(\tau_{j,k}) - \chi^*_{j\rightarrow i}|| \geq \psi(\chi^*_{j\rightarrow i}) \tag{11}$$

where $\chi^*_{j\rightarrow i}$ denotes system j's state used by system i in the current time interval. $S_{j,k}$ will be specified in the sequel. Similarly, we define the outer triggering time instant:

$$\tau^{out}_{i,k+1} = \min_{j\in\mathcal{D}_{\rightarrow i}} \bar{\tau}_{j,k} \tag{12}$$

and

$$\bar{\tau}_{j,k} = \inf_{\tau \geq \tau_{j,k}} \{\tau \mid E_j \ and \ E_{j\rightarrow i}\}$$

If E_{out} from system $j \in \mathcal{D}_{\rightarrow i}$ does not happen, $\bar{\tau}_{j,k} = +\infty$. Note that $\tau^{in}_{i,k+1}$ and $\tau^{out}_{i,k+1}$ are not the actual sampling instant, but they together determine the next sampling instant. The system will take the next sample at time instant $\tau^{out}_{i,k+1}$ if E_{out} happens earlier than E_{in}; Otherwise, the next sampling instant will be $\tau^{in}_{i,k+1}$. It can be inferred that $\tau^{out}_{i,k+1} \in \{\tau^*, +\infty\}$ with $\tau^* \triangleq \min_{i\in\mathcal{M}}\{\tau^{in}_{i,k+1}\}$. We merge the inner event and outer event into one, i.e.,

$$E_i : ||\chi_i(\tau) - \chi_i(\tau_{i,k})|| \geq S_{i,k} \tag{13}$$

where $S_{i,k}$ switches between $D_i(\chi_{\bar{\mathcal{D}}_{\rightarrow i}}(\tau_{i,k}))$ and $D^*_{i,k}$, $D^*_{i,k}$ is defined as follows:

$$D^*_{i,k} = (\tau^*_{k+1} - \tau_{i,k})||f_i(\chi_{\bar{\mathcal{D}}_{\rightarrow i}}(\tau_{i,k}), u_i(\tau_{i,k}))|| \tag{14}$$

$S_{i,k}$ is defined as follows:

$$S_{i,k} = \begin{cases} D^*_{i,k}, & \tau^{out}_{i,k+1} < \tau^{in}_{i,k+1} \\ D_i(\chi_{\bar{\mathcal{D}}_{\rightarrow i}}(\tau_{i,k})), & else \end{cases} \tag{15}$$

where $\tau^{out}_{i,k+1} < \tau^{in}_{i,k+1}$ means the sampling will be triggered by outer event, otherwise, it is triggered by inner event. Then we are able to define the next sampling instant $\tau_{i,k+1}$ as

$$\tau_{i,k+1} = \inf_{\tau \geq \tau_{i,k}} \{\tau \mid ||\chi_i(\tau) - \chi_i(\tau_{i,k})|| \geq S_{i,k}\} \tag{16}$$

Remark 1: It should be noted that, in our framework, if the outer event E_{out} happens earlier than inner event E_{in}, i.e. $\tau^{out}_{i,k+1} < \tau^{in}_{i,k+1}$, then it will lead to the occurrence of system sampling at τ^*, because in this case, $S_{i,k} = D^*_{i,k}$.

Meanwhile, $\tau^{out}_{i,k+1} < \tau^{in}_{i,k+1}$ implies $D^*_{i,k} < D_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,k}))$ holds.

Next, we show the equivalence between the distributed LAM and the distributed event-triggered feedback system.

Theorem 1: Consider the distributed LAM described by (2) and (4) with linear interpolation (5)and the distributed event-triggered feedback system described by (6), (7), and (16). The state trajectories of these two systems are identical, i.e. $\chi_i(\tau) \equiv \hat{x}_i(t)$ and $t \equiv \tau$ for any $t, \tau \geq t_0$. Also, $\tau_{i,k} = t_{i,k}, \forall i \in \mathcal{M},\ k \geq 0$.

Proof. Mathematical induction method will be used. Note that both the distributed LAM and the distributed event-triggered system start at the same time instant $t_{i,0} = \tau_{i,0} = t_0$ and the same initial state $\hat{x}_{i,0} = \chi_{i,0} = x_{i,0}$. During the time interval $[\tau_{i,0}, \tau_{i,1})$, the feedback $v_i(\tau) = f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))$ keeps constant, then the state will grow in a linear fashion, i.e.

$$\chi_i(\tau) = \chi_i(\tau_{i,0}) + f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))(\tau - \tau_{i,0}) \quad (17)$$

until

$$||\chi_i(\tau_{i,1}) - \chi_i(\tau_{i,0})|| = S_{i,0} \quad (18)$$

Since $\chi_i(t)$ is continuous, it follows from (17) that

$$\chi_i(\tau_{i,1}) = \chi_i(\tau_{i,0}) + f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))(\tau_{i,1} - \tau_{i,0}) \quad (19)$$

Then (18) together with (19) yield that

$$\begin{aligned} \tau_{i,1} &= \tau_{i,0} + \frac{S_{i,0}}{||f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))||} \\ \chi_i(\tau_{i,1}) &= \chi_i(\tau_{i,0}) + S_{i,0}\frac{f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))}{||f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))||} \end{aligned} \quad (20)$$

By comparing (20) with (2) and (4), we can find that (20) is equivalent to (2) when $S_{i,0} = D_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}))$, and is equivalent to (4) when $S_{i,0} = D^*_{i,0}$ defined in (14). Note that $S_{i,0}$ is determined by (15), where $\tau^{in}_{i,1}$ is defined by (9), i.e.,

$$\tau^{in}_{i,1} = \tau_{i,0} + \frac{D_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}))}{||f_i(\chi_{\bar{\mathcal{D}}_{\to i}}(\tau_{i,0}), u_i(\tau_{i,0}))||} \quad (21)$$

Considering the prediction stage of the distributed LAM, we are able to know that predicted time instant of the ith LAM is equal to $\tau^{in}_{i,1}$, and then $t^*_1 = \tau^*_1$ holds. Consequently, the switching mechanism of $S_{i,0}$ is the same as the update scheme of the distributed LAM depending on (2) or (4). Comparing (17) with (5) and considering the continuity of $\hat{x}_i(t)$ and $\chi_i(\tau)$, we conclude that $\hat{x}_i(t) = \chi_i(\tau)$ and $t = \tau$ for any $t, \tau \in [\tau_{i,0}, \tau_{i,1}]$.

Assume that $\hat{x}_i(t) = \chi_i(\tau)$ and $t = \tau$ holds for $t, \tau \in [\tau_{i,0}, \tau_{i,k}]$ and $t_{i,k} = \tau_{i,k}$. Following a similar analysis, we can readily prove that $t_{i,k+1} = \tau_{i,k+1}$, $t = \tau$, and $\hat{x}_i(t) = \chi_i(\tau)$ hold for $t, \tau \in [\tau_{i,k}, \tau_{i,k+1}]$. This completes the proof. ■

Assume that the state feedback control law takes the form of $u_i = h_i(x_{\bar{\mathcal{D}}_{\to i}})$. Then the corresponding closed-loop continuous-time system is

$$\begin{aligned} \dot{x}_i(t) &= f_i(x_{\bar{\mathcal{D}}_{\to i}}(t), h_i(x_{\bar{\mathcal{D}}_{\to i}})) \\ x_i(t_0) &= x_{i,0} \end{aligned} \quad (22)$$

We denote $f^{cl}_i(x_{\bar{\mathcal{D}}_{\to i}}) \triangleq f_i(x_{\bar{\mathcal{D}}_{\to i}}, h_i(x_{\bar{\mathcal{D}}_{\to i}}))$, $f^{cl}_i(0) = 0$. Consequently, $f_i(\cdot,\cdot)$ in the LAM and event-triggered system will be replaced by $f^{cl}_i(\hat{x}_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}))$ and $f^{cl}_i(\chi_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}))$ respectively. In the sequel, we will not distinguish $\tau_{i,k}$ from $t_{i,k}$, and use the notation $\hat{\chi}_i(t) \triangleq \chi(t_{i,k})$ for $t \in [t_{i,k}, t_{i,k+1})$. Let $e_i(t) \triangleq \chi_i(t) - \hat{\chi}_i(t)$ and $e_j(t) \triangleq \chi_j(t) - \chi^*_{j\to i}$ for $j \in \mathcal{D}_{\to i}$, then the following inequalities hold:

$$\begin{aligned} ||e_i(t)|| &\leq S_{i,k} \\ ||e_j(t)|| &\leq S_{i,k} + \psi(\chi^*_{j\to i}) \end{aligned} \quad (23)$$

for any $t \in [t_{i,k}, t_{i,k+1})$.

Assumption 1: Consider a distributed continuous-time system described by (22). Assume that for any $i \in \mathcal{M}$, there exists a differentiable, positive-definite function $V_i(x_i) : \mathbb{R}^n \to \mathbb{R}^+$ satisfying:

$$\underline{\alpha}_i(||x_i||) \leq V_i(x_i) \leq \overline{\alpha}_i(||x_i||) \quad (24)$$

$$\begin{aligned} &\frac{\partial V_i(\chi_i)}{\partial \chi_i} f^{cl}_i(\chi_{\bar{\mathcal{D}}_{\to i}} - e_{\bar{\mathcal{D}}_{\to i}}) \\ &\quad \leq -\sigma_i||\chi_i|| + \sum_{j\in\mathcal{D}_{\to i}} \pi_j||\chi_j|| + \sum_{j\in\bar{\mathcal{D}}_{\to i}} \mu_j||e_j|| \end{aligned} \quad (25)$$

and the following inequality holds:

$$\sigma_i - \pi_i|\mathcal{D}_{i\to}| > 0 \quad (26)$$

where $\underline{\alpha}_i(\cdot)$ and $\overline{\alpha}_i(\cdot)$ are class $\mathcal{K}$ functions, σ_i, π_i and μ_i are some positive constants.

Theorem 2: If Assumption 1 holds, the proposed distributed LAM is uniformly ultimately bounded.

Proof. Now that it has been proved in Theorem 1 that the distributed LAM and distributed event-triggered system have identical state trajectories, we will demonstrate the boundedness of $\hat{x}(t)$ by investigating $\chi(t)$.

Let $V(\chi) = \sum_{i\in\mathcal{M}} V_i(\chi_i)$, then we can readily obtain the time derivative of $V(\chi)$:

$$\begin{aligned} \dot{V} =& \sum_{i\in\mathcal{M}} \frac{\partial V_i(\chi_i)}{\partial \chi_i} f^{cl}_i(\chi_{\bar{\mathcal{D}}_{\to i}} - e_{\bar{\mathcal{D}}_{\to i}}) \\ \leq& \sum_{i\in\mathcal{M}} \Big\{ -\sigma_i||\chi_i|| + \sum_{j\in\mathcal{D}_{\to i}} \pi_j||\chi_j|| + \sum_{j\in\bar{\mathcal{D}}_{\to i}} \mu_j||e_j|| \Big\} \\ <& \sum_{i\in\mathcal{M}} \Big\{ -\sigma_i||\chi_i|| + \sum_{j\in\mathcal{D}_{\to i}} \pi_j||\chi_j|| \\ &+ \mu_i S_{i,k} + \sum_{j\in\mathcal{D}_{\to i}} \mu_j(S_{i,k} + \psi(\chi^*_{j\to i})) \Big\} \\ \leq& \sum_{i\in\mathcal{M}} \Big\{ -\sigma_i||\chi_i|| + \sum_{j\in\mathcal{D}_{\to i}} \pi_j||\chi_j|| \\ &+ \sum_{j\in\bar{\mathcal{D}}_{\to i}} \mu_j D^{max}_j + \sum_{j\in\mathcal{D}_{\to i}} \mu_j\psi^{max} \Big\} \\ =& \sum_{i\in\mathcal{M}} \{ -(\sigma_i - \pi_i|\mathcal{D}_{i\to}|)||\chi_i|| \\ &+ \mu_i(|\bar{\mathcal{D}}_{i\to}|D^{max}_i + |\mathcal{D}_{i\to}|\psi^{max}) \} \end{aligned} \quad (27)$$

where D_i^{max} and ψ^{max} are the maximum values of $D_i(\chi_{\bar{\mathcal{D}}_{\to i}}(t_{i,k}))$ and $\psi(\chi^*_{j\to i})$, respectively. When

$$|\bar{\mathcal{D}}_{i\to}|D_i^{max} + |\mathcal{D}_{i\to}|\psi^{max} \leq \frac{\rho_i(\sigma_i - \pi_i|\mathcal{D}_{i\to}|)}{\mu_i}||\chi_i|| \quad (28)$$

with $0 < \rho_i < 1$, then

$$\dot{V} \leq -(1-\rho_i)(\sigma_i - \pi_i|\mathcal{D}_{i\to}|)||\chi_i||. \quad (29)$$

Equation (28) together with (29) ensure that $\chi(t)$ is uniformly ultimately bounded [17, Theorem 4.18]. As a result, the boundedness of $\hat{x}(t)$ is proved. This completes the proof. ■

V. Simulations

In this section we will carry out some simulations on a nonlinear system to verify the developed algorithm. This system consists of four agents described as follows:

$$\begin{cases} \dot{x}_1(t) = a_1 x_1(t) + b_1 \tan(x_4(t)) \\ \dot{x}_2(t) = a_2 x_1^2(t) + b_2 x_2(t) \\ \dot{x}_3(t) = a_3\sqrt{x_2(t)} + b_3 x_3(t) \\ \dot{x}_4(t) = a_4 x_3(t) + b_4 x_4(t) \end{cases}$$

with initial condition $t_0 = 0$, $x(0) = [0.21\ 0.19\ 0.17\ 0.22]^{\mathrm{T}}$, a_i, b_i, $i = 1, 2, 3, 4$ are pre-known parameters.

For the above system, a distributed LAM will be established to predict system's states sporadically. In the algorithm, the quantization size $D_i(\cdot)$ and the threshold $\psi(\cdot)$ are set as follows:

$$D_1 = 0.1\cdot\sqrt{x_1^2 + x_4^2}, \quad D_2 = 0.1\cdot\sqrt{x_1^2 + x_2^2}$$
$$D_3 = 0.1\cdot\sqrt{x_2^2 + x_3^2}, \quad D_4 = 0.1\cdot\sqrt{x_3^2 + x_4^2}$$
$$\psi(x^*_{4\to1}) = 0.2\cdot|x^*_{4\to1}|, \quad \psi(x^*_{1\to2}) = 0.1\cdot|x^*_{1\to2}|$$
$$\psi(x^*_{2\to3}) = 0.1\cdot|x^*_{2\to3}|, \quad \psi(x^*_{3\to4}) = 0.1\cdot|x^*_{3\to4}|$$

Then the following two cases are both simulated:
Case 1:

$$(a_1, a_2, a_3, a_4) = (-0.95, 0.23, -0.86, 0.43)$$
$$(b_1, b_2, b_3, b_4) = (0.90, -0.31, -0.83, -0.6)$$

Case 2:

$$(a_1, a_2, a_3, a_4) = (-0.095, 0.023, -0.086, 2.15)$$
$$(b_1, b_2, b_3, b_4) = (0.090, -0.031, -0.083, -3.0)$$

Compared with Case 1, in Case 2 the dynamics of x_1, x_2, x_3 are slowed down and that of x_4 is speeded up. For comparison purpose, we also present the simulation result using centralized method provided by [14], where we use the parameters in Case 2 and set $D = \max\{D_1, D_2, D_3, D_4\}$.

The simulation results are presented in Table II and Fig. 2–5. Figs. 2 and 4 display the state trajectories generated by the distributed LAM and the continuous-time system in Case 1 and 2, respectively. We found that the errors between actual continuous-time states and the Lebesgue states are very small, which means that the proposed algorithm has excellent approximation performance.

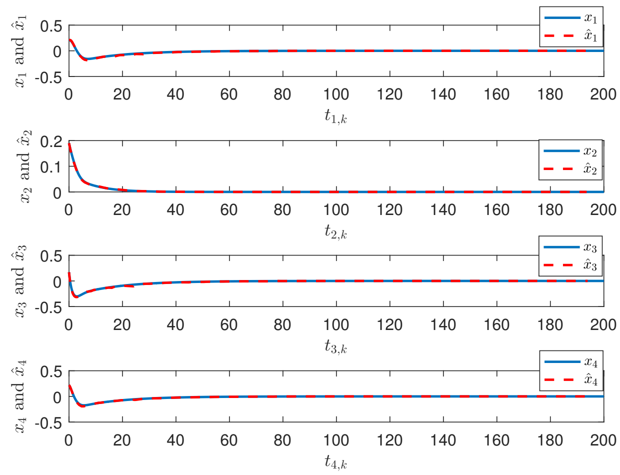

Fig. 2. Case 1: State trajectories generated by distributed LAM and continuous-time system

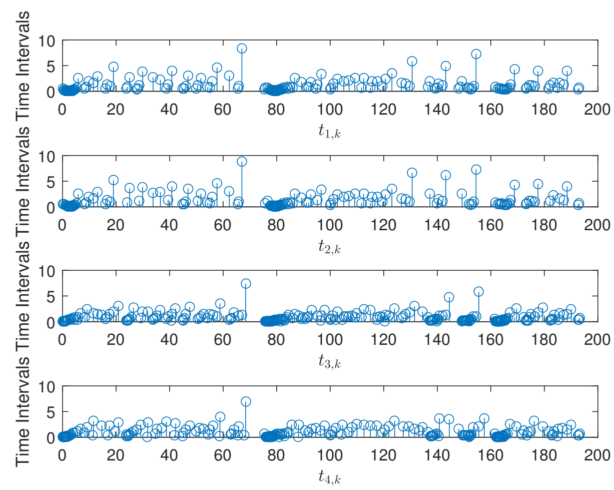

Fig. 3. Case 1: Inter-sampling time intervals

Fig. 3 and 5 plot the inter-iteration time intervals generated by individual LAMs in Case 1 and Case 2, respectively. We can see that compared with Case 1, the inter-iteration time intervals for x_1, x_2, x_3 in Case 2 become much larger. It means that the iteration frequency of x_1, x_2, x_3 becomes low because the dynamics of x_1, x_2, x_3 in Case 2 are slower. On the contrary, x_4 in Fig. 5 is updated more frequent because the dynamics of x_4 in Case 2 is faster. By this fact, it shows that the distributed LAM can update individual states at different frequencies according to the individual dynamics.

Table II provides detailed data on the average inter-iteration time intervals (unit: seconds) in three cases, including the time intervals generated by the centralized LAM [14]. We can find that the average time intervals of LAMs 1-3 in Case 2 are significantly longer than that in Case 1, which is consistent with our observations from Fig. 3 and 5. We found that although the dynamics of x_4 becomes faster in

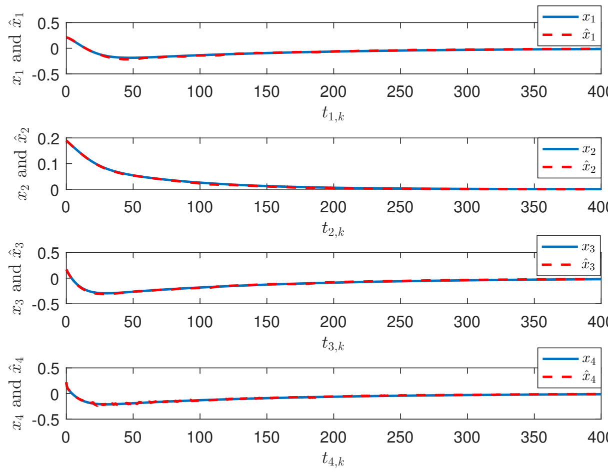

Fig. 4. Case 2: State trajectories generated by distributed LAM and continuous-time system

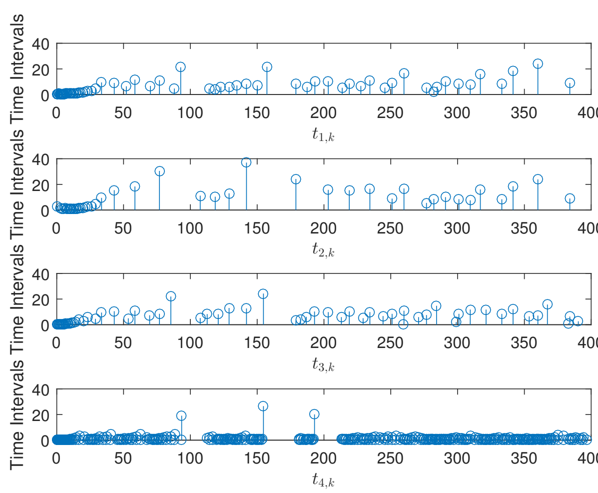

Fig. 5. Case 2: Inter-sampling time intervals

Case 2, the related average time intervals of LAM 4 is still longer than that in Case 1. This is because the updates of $\hat{x}_4$ triggered by $\hat{x}_3$ becomes fewer since the dynamics of x_3 becomes slow. Also note that the centralized algorithm updates much more frequently than the distributed algorithm. All these results demonstrate that the proposed distributed LAM can improve cost efficiency without degrading the approximation performance.

VI. Conclusions

In this paper a discrete-time approximation method based on the LAM has been proposed for a nonlinear dynamical system, which is developed under a distributed framework. The distributed LAM has been proved to be equivalent to a specific distributed event-triggered feedback system in a sense that they have identical state trajectories. Based on the equivalent event-triggered feedback system, the stability of the distributed LAM analyzed. Simulations show that the proposed algorithm is computationally efficient.

TABLE I
Average Inter-Sampling Time Interval

	LAM_1	LAM_2	LAM_3	LAM_4
Case 1	1.1173	1.3072	0.7547	0.8929
Case 2	4.7059	8.5106	5.9701	1.0
Centralized Case [14]	0.7143	0.7143	0.7143	0.7143

References

[1] W. P. M. H. Heemels and M. C. F. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.

[2] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 422–434, 2013.

[3] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE Transactions on Systems, Man, and cybernetics-Part A: Systems and Humans*, vol. 37, no. 3, pp. 348–361, 2007.

[4] X. Deng, L. Schenato, W. C. Wu and S. S. Sastry, "Flapping flight for biomimetic robotic insects: part i-system modeling," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 776–778, 2006.

[5] C. Altafini and F. Ticozzi, "Modeling and control of quantum systems: An introduction," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 1898–1917, 2012.

[6] N. Leonard, and J. Graver, "Model-based feedback control of autonomous underwater gliders," *IEEE Journal of oceanic engineering*, vol. 26, no. 4, pp. 633–645, 2001.

[7] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[8] K. J. Astrom and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2 IEEE, pp. 2011–2016, 2002.

[9] B. P. Zeigler, H. Praehofer, and T. G. Kim, "Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems." *Academic press*, 2000.

[10] F. E. Cellier and E. Kofman, "Continuous system simulation." *Springer*, 2006.

[11] F. E. Cellier, E. Kofman, G. Migoni, and M. Bortolotto, "Quantized state system simulation," *Proceedings of Grand Challenges in Modeling and Simulation*, pp. 504–510, 2008.

[12] E. Kofman and S. Junco, "Quantized-state systems: A DEVS approach for continuous system simulation," *Transactions of the Society for Modeling and Simulation International*, vol. 18, no. 3, pp. 123–132, 2001.

[13] A. Jakobsson, A. Serban and S. Gong, "Implementation of quantized-state system models for a PLL loop filter using verilog-AMS," *IEEE Transactions on Circuits & Systems I: Regular Papers*, vol. 62, no. 3, pp. 680–688, 2015.

[14] X. Wang and B. Zhang, "Lebesgue approximation model of continuous-time nonlinear dynamic systems," *Automatica*, vol. 64, pp. 234–239, 2016.

[15] W. Yan, B. Zhang, X. Wang, W. Dou and J. Wang, "Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1804–1812, 2016.

[16] X. Wangand L. Yang, "Sporadic model predictive control using Lebesgue approximation," *American Control Conference IEEE*, pp. 5768–5773, 2017.

[17] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, New Jersey, 2007, pp. 172.