

Article

Multi-Thread Real-Time Control Based on Event-Triggered Mechanism

Yixuan Wang, Hui Gao *, Yang Yang and Pan Xu

School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China; 200612052@sust.edu.cn (Y.W.); 220612039@sust.edu.cn (Y.Y.); 200612039@sust.edu.cn (P.X.)

* Correspondence: gaohui@sust.edu.cn

Abstract: In this paper, we consider the control problem for systems in which the number of controllers does not match that of plants. As the model of the control plants becomes increasingly complex and diversified, a large number of control systems often share one or a small number of control centers, and each subsystem cannot receive control signals at the same time. A multi-thread control algorithm based on an event-triggered mechanism is proposed to solve the control problem yielding input-to-state stability and achieve the ability to save communication and computing resources. The concept of multi-threading in the computer field is first used to describe such complex systems in control systems, and the two situations of the continuous updating of the controller and the maintaining of the current state are considered in the modeling. The event-triggered rule of the multi-thread control system is designed using the constraint relationship between the errors and the state functions. The feasibility of the proposed algorithm is ensured by its ability to avoid the *Zeno phenomenon* that may occur in the controller switching process. Finally, the proposed algorithm is investigated using simulations, showing that it has excellent flexibility, robustness, and practicability.

Keywords: multi-thread control; event-triggered; *Zeno phenomenon*; input-to-state stability



Citation: Wang, Y.; Gao, H.; Yang, Y.; Xu, P. Multi-Thread Real-Time Control Based on Event-Triggered Mechanism. *Electronics* **2023**, *12*, 2129. <https://doi.org/10.3390/electronics12092129>

Academic Editor: Marcel A.M. Hendrix

Received: 26 March 2023

Revised: 28 April 2023

Accepted: 5 May 2023

Published: 6 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the rapid advances in mechanical equipment, electronic devices, and software designs in intelligent systems have brought both challenges and opportunities to engineering control algorithms. At the same time, the model of the control system is becoming more and more complex and diversified and contains multiple subsystems. It is generally believed that there is often a mismatch between the number of controllers and the controlled plants in the complex systems, which makes the control process more vulnerable to communication and computing resources. However, existing control algorithms are usually designed to solve the problem of the one-to-one characteristics of controllers and plants, which impairs their practicability in real-world engineering applications [1]. In order to solve the control problem of mismatch between the controllers and the controlled plants, it is important to establish a new model to describe these systems and to minimize the usage of resources in the control process under the proposed system form.

What inspired us to develop a new method to solve the mismatch problem between controllers and plants is the working mechanism of the human brain, which cannot handle more than two tasks at once. As studied in [2], the human brain has two frontal lobes, which can send two tasks to the left and right hemispheres of the brain, respectively, for co-processing. However, when there are more than two tasks, the brain is not able to complete them at the same time. When humans perform multiple actions, the brain only processes two tasks at a time, while other tasks are always in an inertial state after being triggered by the brain at the last triggered moment. The prefrontal areas are involved in resolving conflicts that occur when tasks in the same brain network are activated [3]. However, humans seem to be juggling multiple tasks at once, which is the result of the

rapid switching between tasks in the brain. Motivated by the brain working process, it has great significance to develop a systematic approach to practical engineering systems with few-to-many features. This systematic approach can be designed and applied to real-life systems where the numbers of controllers and controlled plants do not match, for example, the da Vinci surgical system with one control center and five controlled fingers. Currently, for such systems with a mismatch between controllers and plants, saving communication and computing resources has become an important issue.

Since there are multiple controlled plants and the control signals need to be switched, the overall structure of the system presents the characteristics of switching. At the same time, switching control causes the degradation of system control performance. Moreover, the event-triggered mechanism for the purpose of saving communication and computing resources provides a reasonable and feasible switching logic. To minimize the usage of resources in the control process, it is important to increase the functionality of control systems using real-time scheduling algorithms that execute control tasks based on event-triggered instead of time-triggered execution [4–9]. The essential concept behind event-triggered control is to cope with feedback problems so as to determine the tasks to execute in the next moment [10]. Event-triggered control, also known as interrupt-based feedback [11], Lebesgue sampling [12], asynchronous sampling [13], or state-triggered feedback [14], has been utilized in a number of applications as an effective alternative to more typical periodic methods. To reduce the number of recomputations and battery-powered energy consumption, a decentralized event-triggered implementation was presented for centralized nonlinear controllers in [15]. A new event-based control algorithm for SISO systems was proposed using input–output linearity, which proved to be as accurate as continuous control algorithms as long as a reasonable threshold was selected [16]. Donkers and Heemels developed a decentralized event-triggered mechanism, the stability of which was analyzed using the principle of pulse [5]. In the case of periodic event-triggered control, [17] proposed a systematic method for determining the sample time and revising the triggered condition.

Using the concept of multi-threading in the computer field, the system with the mismatch problem between controllers and plants can be called “Multi-thread control system”, in which the whole system can remain stable using ideal switching similar to the brain working process. Due to the switching of actuators in the process of multi-thread control, the whole system becomes a hybrid system, which can operate in real time and handle events as well as continuous computations [18]. Such systems with specific switching logic have been widely used to describe complex engineering applications, such as automobile braking [19], traffic control [20], and robot motion control [21]. For the scheduling decisions of the switching systems, in [22], the model presented uses feedback information from the controlled plants. Specifically, the events are triggered when an error signal exceeds a preset threshold, causing the sample period to take random values [23]. For linear switching systems, the event-triggered mechanism only transmits system states, which ensures exponential stability and aids in the avoidance of the *Zeno phenomenon* [24–26]. The *Zeno phenomenon* also happens when the switching event is not very reasonable in multi-thread control systems, so an event-triggered mechanism and a corresponding controller are worth studying for the multi-thread process.

Furthermore, in existing theories, there still exist some obvious limitations. Firstly, existing research on systems with multiple controlled plants usually matches the number of controllers and controlled plants. However, with the rapid development of intelligent systems, a large number of controlled plants share one or a small number of controlled centers. The system with this mismatch has not been described or mathematically modeled yet. Secondly, in traditional research on switching systems, there is a one-to-one correspondence between the controller and the controlled plant, but in real life, the control system has the characteristics of a mismatch between the number of controllers and the controlled plants. Update rules for controllers have not been systematically proposed either in systems with mismatched characteristics. Thirdly, since the system has fewer control centers, a

suitable switching logic is required to save computing and communication resources. At present, the research results on the event-triggered mechanism have not been applied to the system with mismatched characteristics. At the same time, in the study of event-triggered mechanisms for systems with multiple controlled plants, the selection of triggered conditions is crucial, as it directly determines the ability to save resources. Finally, in this class of systems exhibiting mismatched characteristics, how to avoid the Zeno phenomenon becomes a new challenge.

In this study, a multi-thread control algorithm with an event-triggered mechanism to address the resource waste problem in the proposed multi-thread system is provided. First, the article analyzes the characteristics of the mismatch between the numbers of controllers and actuators in the system and systematically proposes a multi-thread control structure with linear and nonlinear forms to cope with the switching problem. Then, the switching moment for the proposed system is explored using the event-triggered mechanism, and the response time of the control tasks is explicitly considered, as is the ability of the proposed event-triggered scheduling strategy to ensure input-to-state stability (ISS). Finally, under the proposed scheduling approach, the presence of a minimum interval is demonstrated, suggesting that the *Zeno phenomenon* can be avoided for the whole system.

The structure of the rest of the multi-thread control algorithm designed in this paper is as follows: Preliminaries are provided in Section 2, which includes certain definitions and basic properties of the event-triggered scheduling policy using ISS. In Section 3, it is demonstrated that there is a minimum interval between the execution of one control task and that of another, and the coordination among controlled tasks is discussed. Both simulation and experimental results are given and analyzed in Section 4. Finally, in Section 5, the conclusions are presented, showing that the proposed control algorithm based on the event-triggered mechanism can achieve the effect of saving resources in the multi-thread system.

2. Preliminaries

In a multi-thread control system, the numbers of controllers and plants usually do not match. More specifically, the number of controllers is generally smaller than that of plants. Figure 1 exemplifies a multi-thread control system with *one* controller and *n* actuators (*n* plants), and the same number of actuators and controlled plants are set. In this section, the form of the plants is described, and the problem is solved.

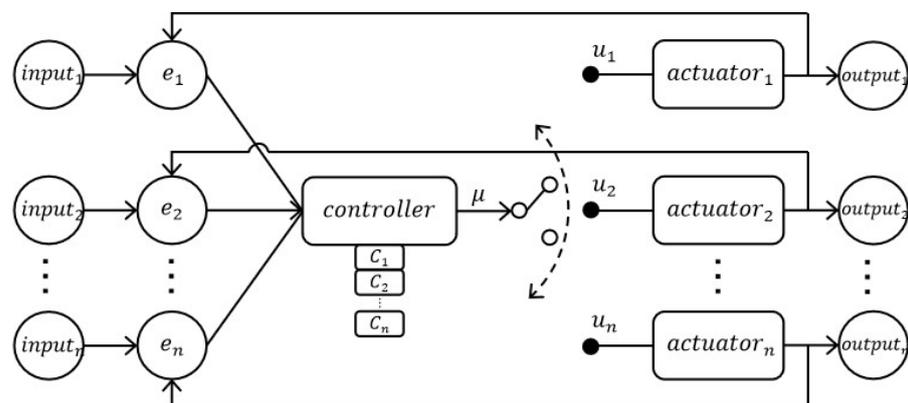


Figure 1. A multi-thread control system.

The following will analyze the multi-threaded control system proposed above.

Notation: This article uses standard notation, unless otherwise stated. \mathbb{R}^l denotes the *l*-dimensional Euclidean space. \mathbb{N}^* and \mathbb{Z}^+ denote a collection of positive integers. For the independent variables *x* and *u*, *f*(*x*, *u*) represents a function that varies with the independent variables. $\|x\|$ denotes the Euclidean 2-norm. $\lambda_m(A)$ denotes the largest

eigenvalue of matrix A . A^T denotes the transpose of matrix A . In this article, t_k denotes the triggered moment, not the sampling moment.

2.1. Plant Form

In this section, we consider a multi-thread control system with n nonlinear plants and m controllers ($m < n$) as follows:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, u_1) \\ \dot{x}_2 &= f_2(x_2, u_2) \\ &\dots \\ \dot{x}_n &= f_n(x_n, u_n) \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^l (i \in \{1, 2, \dots, n\}, n \in \mathbb{N}^*)$ is the state vector of each of the n plants and $u_i(x_i) (i \in \{1, 2, \dots, n\}, n \in \mathbb{N}^*)$ is the actual control law of the i th actuator. Let $\mu_j(x_i) (j \in \{1, 2, \dots, m\}, m \in \mathbb{N}^*)$ be the output signal of the j th controller and $C_i(x_i) \in \mathbb{R}^n (i \in \{1, 2, \dots, n\}, n \in \mathbb{N}^*)$ be the ideal control law of the i th plant, which can ensure the global stability of the i th plant during the continuous control process. When events are triggered, the output values $\mu_j(x_i)$ of the controllers are determined by the ideal control law $C_i(x_i)$. Because the number of controllers is smaller than that of plants, i.e., $m < n$, the controllers need to be switched to guarantee the stability of the plants. Now, we assume that t_k is the k th ($k \in \mathbb{N}^*$) switching time and $x_{i,t_k} (i \in \{1, 2, \dots, n\}, n \in \mathbb{N}^*)$ is the state vector of the i th plant at time t_k . Therefore, the actual controller update law u_i can be expressed as follows:

$$u_i(x_{i,t}) = \begin{cases} \mu_j(x_{i,t}) = C_i(x_{i,t}), & m \text{ updated } u_i \\ u_i(x_{i,t_k^-}), & n - m \text{ unchanged } u_i \end{cases} \tag{2}$$

where $i \in \{1, 2, \dots, n\}, (n \in \mathbb{N}^*), j \in \{1, 2, \dots, m\}, (m \in \mathbb{N}^*)$, and $t \in [t_k, t_{k+1})$. It can be seen, according to (2), that the plants with updated controllers have time-varying and continuous control signals in $t \in [t_k, t_{k+1})$, which ensures the stability of these plants until the next switching occurs. In fact, there are cases in which the controllers do not need to be continuously updated in $t \in [t_k, t_{k+1})$ after the switching occurs but simply at a certain time t_k , which helps to save both controller and computation resources [1]. Therefore, another actual controller update law is given as follows:

$$u_i(x_{i,t}) = \begin{cases} \mu_j(x_{i,t_k}) = C_i(x_{i,t_k}), & m \text{ updated } u_i \\ u_i(x_{i,t_{k-1}}), & n - m \text{ unchanged } u_i \end{cases} \tag{3}$$

where $i \in \{1, 2, \dots, n\}, (n \in \mathbb{N}^*), j \in \{1, 2, \dots, m\}, (m \in \mathbb{N}^*)$, and $t \in [t_k, t_{k+1})$. In (3), the controllers only change at time t_k and remain unchanged in $[t_k, t_{k+1})$.

Let the p th ($1 \leq p_1 < p_2 < \dots < p_m \leq n$) controller be updated at time t_k and the q th ($1 \leq q_1 < q_2 < \dots < q_{n-m} \leq n$) controller be unchanged at time t_k . Moreover, let $e_{i,t} \in \mathbb{R}^l, i \in \{1, 2, \dots, n\}, (n \in \mathbb{N}^*)$ be the measurement errors of n plants, respectively. Measurement errors $e_{i,t}$ for $t \in [t_k, t_{k+1})$ are defined as $e_{1,t} = x_{1,t_{1,*}} - x_{1,t}, e_{2,t} = x_{2,t_{2,*}} - x_{2,t}, \dots, e_{n,t} = x_{n,t_{3,*}} - x_{n,t}$, where $\{t_{i,*}\}$ is the last triggered time of plant i .

Remark 1. We assume a simple situation to illustrate the above definition of errors $e_{1,t}, e_{2,t}, \dots, e_{n,t}$. If the p_1, p_2, \dots, p_m th controllers are updated at t_k , and the q_1, q_2, \dots, q_{n-m} controllers are updated at t_{k-1} , then $e_{p_1,t} = x_{p_1,t_k} - x_{p_1,t}, e_{p_2,t} = x_{p_2,t_k} - x_{p_2,t}, \dots, e_{p_m,t} = x_{p_m,t_k} - x_{p_m,t}$ and $e_{q_1,t} = x_{q_1,t_{k-1}} - x_{q_1,t}, e_{q_2,t} = x_{q_2,t_{k-1}} - x_{q_2,t}, \dots, e_{q_{n-m},t} = x_{q_{n-m},t_{k-1}} - x_{q_{n-m},t}$.

Then, the following system form is given:

m updated plants :

$$\begin{cases} \dot{x}_{p_1,t} = f_{p_1}(x_{p_1,t}, u_{p_1}(x_{p_1,t} + e_{p_1,t})) \\ \dot{x}_{p_2,t} = f_{p_2}(x_{p_2,t}, u_{p_2}(x_{p_2,t} + e_{p_2,t})) \\ \dots \\ \dot{x}_{p_m,t} = f_{p_m}(x_{p_m,t}, u_{p_m}(x_{p_m,t} + e_{p_m,t})) \end{cases} \quad (4)$$

n - m upchanged plants :

$$\begin{cases} \dot{x}_{q_1,t} = f_{q_1}(x_{q_1,t}, u_{q_1}(x_{q_1,t} + e_{q_1,t})) \\ \dot{x}_{q_2,t} = f_{q_2}(x_{q_2,t}, u_{q_2}(x_{q_2,t} + e_{q_2,t})) \\ \dots \\ \dot{x}_{q_{n-m},t} = f_{q_{n-m}}(x_{q_{n-m},t}, u_{q_{n-m}}(x_{q_{n-m},t} + e_{q_{n-m},t})) \end{cases}$$

To better illustrate the above system structure and the upcoming multi-thread control algorithm, we consider a linear time-invariant control puzzler with 4 plants and 2 controllers as follows:

$$\begin{cases} \dot{x}_1 = A_1x_1 + b_1u_1 \\ y_1 = c_1x_1 + d_1u_1 \end{cases}, \begin{cases} \dot{x}_2 = A_2x_2 + b_2u_2 \\ y_2 = c_2x_2 + d_2u_2 \end{cases}, \begin{cases} \dot{x}_3 = A_3x_3 + b_3u_3 \\ y_3 = c_3x_3 + d_3u_3 \end{cases}, \begin{cases} \dot{x}_4 = A_4x_4 + b_4u_4 \\ y_4 = c_4x_4 + d_4u_4 \end{cases}. \quad (5)$$

The ideal controllers, C_i , of the system are described as $C_1 = k_1x_1$, $C_2 = k_2x_2$, $C_3 = k_3x_3$, and $C_4 = k_4x_4$, where k_1, k_2, k_3 , and k_4 are the feedback gains of the controller. Controllers u_1 and u_3 are assumed to be updated at time t_k . By (2), we have

$$\begin{aligned} u_1(x_{1,t}) &= \mu_1(x_{1,t}) = k_1x_{1,t} \\ u_2(x_{2,t}) &= u_2(x_{2,t_k^-}) \\ u_3(x_{3,t}) &= \mu_2(x_{3,t}) = k_3x_{3,t} \\ u_4(x_{4,t}) &= u_4(x_{4,t_k^-}) \end{aligned} \quad (6)$$

where $t \in [t_k, t_{k+1})$. Since two controllers are not continuously updated in $t \in [t_k, t_{k+1})$, another actual controller update law is given as follows:

$$\begin{aligned} u_1(x_{1,t}) &= \mu_1(x_{1,t}) = k_1x_{1,t} \\ u_2(x_{2,t}) &= u_2(x_{2,t_{k-1}}) \\ u_3(x_{3,t}) &= \mu_2(x_{3,t}) = k_3x_{3,t} \\ u_4(x_{4,t}) &= u_4(x_{4,t_{k-1}}) \end{aligned} \quad (7)$$

Considering the two conditions of the controllers, the state space expression of the tasks at time t ($t \in [t_k, t_{k+1})$) can be replaced by

$$\begin{cases} \dot{x}_{1,t} = A_1x_{1,t} + b_1k_1x_{1,t} \\ y_{1,t} = c_1x_{1,t} + d_1k_1x_{1,t} \end{cases}, \begin{cases} \dot{x}_{2,t} = A_2x_{2,t} + b_2u_2(x_{2,t_k^-}) \\ y_{2,t} = c_2x_{2,t} + d_2u_2(x_{2,t_k^-}) \end{cases}, \begin{cases} \dot{x}_{3,t} = A_3x_{3,t} + b_3k_3x_{3,t} \\ y_{3,t} = c_3x_{3,t} + d_3k_3x_{3,t} \end{cases}, \begin{cases} \dot{x}_{4,t} = A_4x_{4,t} + b_4u_4(x_{4,t_k^-}) \\ y_{4,t} = c_4x_{4,t} + d_4u_4(x_{4,t_k^-}) \end{cases} \quad (8)$$

or

$$\begin{cases} \dot{x}_{1,t} = A_1 x_{1,t} + b_1 k_1 x_{1,t} \\ y_{1,t} = c_1 x_{1,t} + d_1 k_1 x_{1,t} \end{cases}, \begin{cases} \dot{x}_{2,t} = A_2 x_{2,t} + b_2 u_2(x_{2,t_{k-1}}) \\ y_{2,t} = c_2 x_{2,t} + d_2 u_2(x_{2,t_{k-1}}) \end{cases} \\ \begin{cases} \dot{x}_{3,t} = A_3 x_{3,t} + b_3 k_3 x_{3,t} \\ y_{3,t} = c_3 x_{3,t} + d_3 k_3 x_{3,t} \end{cases}, \begin{cases} \dot{x}_{4,t} = A_4 x_{4,t} + b_4 u_4(x_{4,t_{k-1}}) \\ y_{4,t} = c_4 x_{4,t} + d_4 u_4(x_{4,t_{k-1}}) \end{cases} \end{cases} \quad (9)$$

Remark 2. The purpose of designing two update laws of controllers is to better approach the control methods of real-life devices. Although the control accuracy of the second control law scheme is not as good as the first one in the multi-thread control system, it can save resources better. This scheme is equivalent to adding a zero-order keeper behind each controller.

2.2. Problem Statement

As previously given, the structure of multi-thread control is designed for the multi-objective system with few controllers. Considering the influence of measurement errors on system stability, we will list the basic ISS condition definitions that can ensure the stability of the multi-tread control structure as follows:

Definition 1 ([27]). The smooth function $V_N = [V_1, V_2, \dots, V_n]: \mathbb{R}^n \rightarrow \mathbb{R}_0^+$ is defined as an ISS Lyapunov function for closed-loop system (1) or (4) if there exist class \mathcal{K}_∞ functions $\alpha_1, \alpha_2, \beta,$ and γ satisfying

$$\begin{aligned} \alpha_1(\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|) &\leq \\ V_N &\leq \alpha_2(\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|) \end{aligned} \quad (10)$$

$$\begin{aligned} \frac{\partial V_N}{\partial x} &= \sum_{i=1}^{n-m} \frac{\partial V_{q_i}}{\partial x_{q_i,t}} f_{q_i}(x_{q_i,t}, u_{q_i}(x_{q_i,t} + e_{q_i,t})) + \sum_{j=1}^m \frac{\partial V_{p_j}}{\partial x_{p_j,t}} f_{p_j}(x_{p_j,t}, u_{p_j}(x_{p_j,t} + e_{p_j,t})) \\ &\leq -\beta(\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|) \\ &\quad + \gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|). \end{aligned} \quad (11)$$

where $1 \leq p_1 < p_2 < \dots < p_m \leq n$ and $1 \leq q_1 < q_2 < \dots < q_{n-m} \leq n$.

The status of a multi-thread system is sampled at time instants $t_1, t_2, t_3, \dots, t_n$ to determine the controller. These time instants are neither pre-determined nor pre-programmed but are instead implicitly specified by an event-triggered execution rule based on the condition of the plant. We assume that $\Delta \geq 0$ is the time it takes to calculate the control law and update the actuator from the state read by the sensor, so that $t_1 + \Delta, t_2 + \Delta, t_3 + \Delta, \dots, t_n + \Delta$ are the time intervals for each controller to update completely. Moreover, at time $t \in [t_k + \Delta, t_{k+1} + \Delta)$, measurement errors $e_{1,t}, e_{2,t}, \dots, e_{n,t}$ are the same as defined above. By (11), assuming that the limit of mistake to fulfill is

$$\begin{aligned} &\gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|) \\ &\leq \xi \beta(\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|) (\xi > 0) \end{aligned} \quad (12)$$

the dynamics of V_1, V_2, \dots, V_n are bounded by

$$\begin{aligned} &\sum_{i=1}^{n-m} \frac{\partial V_{q_i}}{\partial x_{q_i,t}} f_{q_i}(x_{q_i,t}, u_{q_i}(x_{q_i,t} + e_{q_i,t})) + \sum_{j=1}^m \frac{\partial V_{p_j}}{\partial x_{p_j,t}} f_{p_j}(x_{p_j,t}, u_{p_j}(x_{p_j,t} + e_{p_j,t})) \\ &\leq (\xi - 1)\beta(\|x_{p_1,t}\| + \|x_{p_2,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \|x_{q_2,t}\| + \dots + \|x_{q_{n-m},t}\|). \end{aligned} \quad (13)$$

Let $\xi < 1$; then, (13) can be replaced with

$$\sum_{i=1}^{n-m} \frac{\partial V_{q_i}}{\partial x_{q_i,t}} f_{q_i}(x_{q_i,t}, u_{q_i}(x_{q_i,t} + e_{q_i,t})) + \sum_{j=1}^m \frac{\partial V_{p_j}}{\partial x_{p_j,t}} f_{p_j}(x_{p_j,t}, u_{p_j}(x_{p_j,t} + e_{p_j,t})) < 0 \quad (14)$$

thus ensuring that $V_1 + V_2 + \dots + V_n$ decreases. By carrying out the control tasks, inequality (12) can be enforced when

$$\begin{aligned} & \gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|) \\ & = \xi\beta(\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|)(\xi > 0) \end{aligned} \quad (15)$$

Then, the previous execution time, t_{k+1} , is given as

$$t_{k+1} = \inf\{t > t_k | \gamma(\|e_{1,t}\| + \|e_{2,t}\| + \dots + \|e_{n,t}\|) > \xi\beta(\|x_{1,t}\| + \|x_{2,t}\| + \dots + \|x_{n,t}\|)\} \quad (16)$$

and we also have the multi-thread event-triggered condition as follows:

$$\gamma(\|e_{1,t}\| + \|e_{2,t}\| + \dots + \|e_{n,t}\|) > \xi\beta(\|x_{1,t}\| + \|x_{2,t}\| + \dots + \|x_{n,t}\|) \quad (17)$$

Remark 3. Equation (15) is the ISS-based event-triggered condition to guarantee that the Lyapunov function does not increase. Since the number of controllers, m , is less than the number of controlled plants, n , it is impossible for us to force errors $e_{1,t}, e_{2,t}, \dots, e_{n,t}$ of all plants to 0 with the updated controller when the event-triggered condition is satisfied. The p th ($1 \leq p_1 < p_2 < \dots < p_m \leq n$) controller whose e is updated to 0 at time t_k is selected using certain rules.

Although stability is guaranteed by constructing simple execution rules, it is critical to ensure that the execution time is not arbitrarily close to the accumulation point to evaluate the feasibility of the scheduling policy. In other words, the proposed multi-thread event-triggered condition should promise that the *Zeno phenomenon*, which can lead to degraded system performance, controller failure, and even other problems, can be avoided.

3. Analysis of Stability

Based on multi-thread control structure (4) and event-triggered conditions (17), the main result is given as shown below.

Theorem 1. Consider the multi-thread system described in (1) and (4); input $u_{p_1}, u_{p_2}, \dots, u_{p_m}$ ($1 \leq p_1 < p_2 < \dots < p_m \leq n$) is updated with controller output μ_i ($i = 1, 2, \dots, m$) and ideal global control law C_i ($i = 1, 2, \dots, n$) described in (3). The closed-loop system is input-to-state stable with respect to the measurement errors, and all the signals are uniformly bounded with event-triggered conditions (17). Moreover, Zeno behavior can be avoided.

Proof of Theorem 1. Using (16), we define a compact set E for all $e_{1,t}, e_{2,t}, \dots, e_{n,t} \in \mathbb{R}^l$ and all $x_{1,t}, x_{2,t}, \dots, x_{n,t} \in \mathbb{R}^l$ that satisfies

$$\begin{aligned} & \|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\| \\ & \leq \gamma^{-1}(\xi\beta(\|x_{p_1,t}\| + \|x_{p_2,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \|x_{q_2,t}\| + \dots + \|x_{q_{n-m},t}\|)) \end{aligned} \quad (18)$$

Since β^{-1} and γ are Lipschitz continuous, $\beta^{-1}(\gamma(\|r_{p_1,t}\| + \dots + \|r_{p_m,t}\| + \|r_{q_1,t}\| + \dots + \|r_{q_{n-m},t}\|)/\xi)$ is continuous. For compact set E , let L be the Lipschitz constant [28], so that

$$\begin{aligned}
 & |\beta^{-1}(\gamma(\|r_{p_1,t}\| + \|r_{p_2,t}\| + \dots + \|r_{p_m,t}\| + \|r_{q_1,t}\| + \|r_{q_2,t}\| + \dots + \|r_{q_{n-m},t}\|)/\xi) \\
 & - \beta^{-1}(\gamma(\|s_{p_1,t}\| + \|s_{p_2,t}\| + \dots + \|s_{p_m,t}\| + \|s_{q_1,t}\| + \|s_{q_2,t}\| + \dots + \|s_{q_{n-m},t}\|)/\xi)| \quad (19) \\
 & \leq L(\|r_{p_1,t}\| - \|s_{p_1,t}\|) + \dots + (\|r_{p_m,t}\| - \|s_{p_m,t}\|) \\
 & + (\|r_{q_1,t}\| - \|s_{q_1,t}\|) + \dots + (\|r_{q_{n-m},t}\| - \|s_{q_{n-m},t}\|)
 \end{aligned}$$

Let $r_{p_1,t} = e_{p_1,t}, r_{p_2,t} = e_{p_2,t}, \dots, r_{p_m,t} = e_{p_m,t}, r_{q_1,t} = e_{q_1,t}, r_{q_2,t} = e_{q_2,t}, \dots, r_{q_{n-m},t} = e_{q_{n-m},t}, s_{p_1,t} = \dots = s_{p_m,t} = s_{q_1,t} = \dots = s_{q_{n-m},t} = 0$, so that (19) can be replaced with

$$\begin{aligned}
 & \beta^{-1}(\gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|)/\xi) \quad (20) \\
 & \leq L(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|)
 \end{aligned}$$

Note that by enforcing the more conservative inequality $L(\|e_{p_1,t}\| + \dots + \|e_{q_{n-m},t}\|) \leq \|x_{p_1,t}\| + \dots + \|x_{q_{n-m},t}\|$, we guarantee $\beta^{-1}(\gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \|e_{q_{n-m},t}\|)/\xi) \leq \|x_{p_1,t}\| + \dots + \|x_{q_{n-m},t}\|$. The above proves that the time interval between successive triggered condition actions is bounded.

Considering the closed-loop form in (4), $f_i(x_{i,t}, u_i(x_{i,t} + e_{i,t}))$ ($i \in \mathbb{Z}^+, i_{max} = n$) are Lipschitz continuous on compacts, that is,

$$|F(\Pi, u(\Pi, \Psi)) - F(\Pi', u(\Pi', \Psi'))| \leq L|(\Pi, \Psi) - (\Pi', \Psi')| \quad (21)$$

where

$$\begin{aligned}
 F(\Pi, u(\Pi, \Psi)) = & (f_{p_1}(r_{p_1,t}, u_{p_1}(r_{p_1,t} + s_{p_1,t})) + f_{p_2}(r_{p_2,t}, u_{p_2}(r_{p_2,t} + s_{p_2,t})) \\
 & + \dots + f_{p_m}(r_{p_m,t}, u_{p_m}(r_{p_m,t} + s_{p_m,t})) + f_{q_1}(r_{q_1,t}, u_{q_1}(r_{q_1,t} + s_{q_1,t})) \\
 & + f_{q_2}(r_{q_2,t}, u_{q_2}(r_{q_2,t} + s_{q_2,t})) + \dots + f_{q_{n-m}}(r_{q_{n-m},t}, u_{q_{n-m}}(r_{q_{n-m},t} + s_{q_{n-m},t})))
 \end{aligned}$$

and

$$\begin{aligned}
 (\Pi, \Psi) = & ((r_{p_1,t}, s_{p_1,t}) + (r_{p_2,t}, s_{p_2,t}) + \dots + (r_{p_m,t}, s_{p_m,t}) \\
 & + (r_{q_1,t}, s_{q_1,t}) + (r_{q_2,t}, s_{q_2,t}) + \dots + (r_{q_{n-m},t}, s_{q_{n-m},t}))
 \end{aligned}$$

Similarly, $F(\Pi', u(\Pi', \Psi'))$, (Π', Ψ') is obtained. If the set

$$\begin{aligned}
 r_{p_1,t} = x_{p_1,t}, \dots, r_{p_m,t} = x_{p_m,t} \\
 r_{q_1,t} = x_{q_1,t}, \dots, r_{q_{n-m},t} = x_{q_{n-m},t} \\
 s_{p_1,t} = e_{p_1,t}, \dots, s_{p_m,t} = e_{p_m,t} \\
 s_{q_1,t} = e_{q_1,t}, \dots, s_{q_{n-m},t} = e_{q_{n-m},t}
 \end{aligned}$$

and otherwise are 0. Then, the formula can be obtained:

$$\begin{aligned}
 & |f_{p_1}(r_{p_1,t}, u_{p_1}(r_{p_1,t} + s_{p_1,t})) + \dots + f_{p_m}(r_{p_m,t}, u_{p_m}(r_{p_m,t} + s_{p_m,t})) \\
 & + f_{q_1}(r_{q_1,t}, u_{q_1}(r_{q_1,t} + s_{q_1,t})) + \dots + f_{q_{n-m}}(r_{q_{n-m},t}, u_{q_{n-m}}(r_{q_{n-m},t} + s_{q_{n-m},t}))| \quad (22) \\
 & \leq L(\|x_{p_1,t}\| + \|x_{p_2,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \|x_{q_2,t}\| + \dots + \|x_{q_{n-m},t}\| \\
 & + \|e_{p_1,t}\| + \|e_{p_2,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \|e_{q_2,t}\| + \dots + \|e_{q_{n-m},t}\|)
 \end{aligned}$$

Now, we define $X_N = \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\| + \|x_{p_1,t}\| + \dots + \|x_{p_m,t}\|$ and $E_N = \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\| + \|e_{p_1,t}\| + \dots + \|e_{p_m,t}\|$; the inter-event times can be decided according to the dynamics of $\frac{E_N}{X_N}$.

$$\begin{aligned}
 \frac{E_N}{X_N} &= \frac{d[(e_{q_1}^T e_{q_1})^{\frac{1}{2}} + (e_{q_2}^T e_{q_2})^{\frac{1}{2}} + \dots + (e_{p_m}^T e_{p_m})^{\frac{1}{2}}]}{dt[(x_{q_1}^T x_{q_1})^{\frac{1}{2}} + (x_{q_2}^T x_{q_2})^{\frac{1}{2}} + \dots + (x_{p_m}^T x_{p_m})^{\frac{1}{2}}]} \\
 &= \frac{-\frac{1}{\|e_{q_1}\|} e_{q_1}^T \dot{x}_{q_1} - \frac{1}{\|e_{q_2}\|} e_{q_2}^T \dot{x}_{q_2} - \dots - \frac{1}{\|e_{p_m}\|} e_{p_m}^T \dot{x}_{p_m}}{X_N} \\
 &\quad - \frac{E_N(\frac{1}{\|x_{q_1}\|} x_{q_1}^T \dot{x}_{q_1} + \dots + \frac{1}{\|x_{p_m}\|} x_{p_m}^T \dot{x}_{p_m})}{X_N^2} \tag{23} \\
 &\leq \frac{\frac{1}{\|e_{q_1}\|} \|e_{q_1}\| \|\dot{x}_{q_1}\| + \dots + \frac{1}{\|e_{q_{n-m}}\|} \|e_{q_{n-m}}\| \|\dot{x}_{q_{n-m}}\|}{X_N} \\
 &\quad + \frac{\frac{1}{\|e_{p_1}\|} \|e_{p_1}\| \|\dot{x}_{p_1}\| + \dots + \frac{1}{\|e_{p_m}\|} \|e_{p_m}\| \|\dot{x}_{p_m}\|}{X_N} \\
 &\quad + \frac{E_N(\frac{1}{\|x_{q_1}\|} \|x_{q_1}\| \|\dot{x}_{q_1}\| + \dots + \frac{1}{\|x_{q_{n-m}}\|} \|x_{q_{n-m}}\| \|\dot{x}_{q_{n-m}}\|)}{X_N^2} \\
 &\quad + \frac{E_N(\frac{1}{\|x_{p_1}\|} \|x_{p_1}\| \|\dot{x}_{p_1}\| + \dots + \frac{1}{\|x_{p_m}\|} \|x_{p_m}\| \|\dot{x}_{p_m}\|)}{X_N^2} \\
 &\leq (1 + \frac{E_N}{X_N}) \frac{L(X_N + E_N)}{X_N} \\
 &= L(1 + \frac{E_N}{X_N})(1 + \frac{E_N}{X_N})
 \end{aligned}$$

We define $g = \frac{E_N}{X_N}$, so that $\dot{g} \leq L(1 + g)^2$ and $g(t) \leq \Phi(t, \Phi_0)$, where $\Phi(t, \Phi_0)$ is the solution of $\dot{\Phi} = L(1 + \Phi)^2$ satisfying $\Phi(0, \Phi_0) = \Phi_0$. The time of delay for the multi-thread control system is denoted by $\Delta \geq 0$.

We assume that $\Delta = 0$, where the execution time is limited by Φ equaling 0 to $\frac{1}{M}$ and the interval time is limited by solution $\tau \in \mathbb{R}^+$ of $\Phi(0, \Phi_0) = \frac{1}{M}$. It is obtained, using integration, that $\Phi(0, \Phi_0) = \frac{\tau L}{1 - \tau L}$, so $\tau = \frac{1}{L + LM}$.

Now, we discuss the circumstances when $\Delta > 0$. First, we choose the ζ' satisfying $\zeta < \zeta' < 1$. M' is the Lipschitz constant of $\beta^{-1}(\gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|)) / \zeta'$. We assume that $\zeta_1 \in \mathbb{R}^+$ satisfies $\Phi(\zeta_1, \frac{1}{M}) = \frac{1}{M'}$, where ζ_1 always exists when $\Phi > 0$ is continuous and $\frac{1}{M} < \frac{1}{M'}$. We guarantee that for $t \in [t_k, t_k + \zeta_1)$, we have $\frac{\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|}{\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|} \leq \frac{M'}{M}$ and thus also $\gamma(\|e_{p_1,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \dots + \|e_{q_{n-m},t}\|) \leq \zeta' \beta(\|x_{p_1,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \dots + \|x_{q_{n-m},t}\|)$. Since $\zeta < 1$, asymptotic stability still ensures that the execution time is bounded by $\tau + \Delta$, where τ is defined as the time that Φ evolves from

$$\frac{\|e_{p_1,t_k+\Delta}\| + \dots + \|e_{q_1,t_k+\Delta}\| + \dots + \|e_{q_{n-m},t_k+\Delta}\|}{\|x_{p_1,t_k+\Delta}\| + \dots + \|x_{q_1,t_k+\Delta}\| + \dots + \|x_{q_{n-m},t_k+\Delta}\|} = \frac{\|x_{p_1,t_k} - x_{p_1,t_k+\Delta}\| + \dots + \|x_{q_{n-m},t_k} - x_{q_{n-m},t_k+\Delta}\|}{\|x_{p_1,t_k+\Delta}\| + \dots + \|x_{q_1,t_k+\Delta}\| + \dots + \|x_{q_{n-m},t_k+\Delta}\|}$$

to $\frac{1}{M}$. Since $\Phi > 0$, a small enough Δ value to satisfy the mechanism is picked. It now follows, from the continuity of $\frac{\|x_{p_1,t_k} - x_{p_1,t_k+\Delta}\| + \dots + \|x_{q_{n-m},t_k} - x_{q_{n-m},t_k+\Delta}\|}{\|x_{p_1,t_k+\Delta}\| + \dots + \|x_{q_1,t_k+\Delta}\| + \dots + \|x_{q_{n-m},t_k+\Delta}\|}$ with respect to Δ , the existence of $\zeta_2 > 0$, such that for any $0 \leq \Delta \leq \zeta_2$, we have

$$\frac{\|e_{p_1,t_k+\Delta}\| + \dots + \|e_{q_1,t_k+\Delta}\| + \dots + \|e_{q_{n-m},t_k+\Delta}\|}{\|x_{p_1,t_k+\Delta}\| + \dots + \|x_{q_1,t_k+\Delta}\| + \dots + \|x_{q_{n-m},t_k+\Delta}\|} < \frac{1}{M}.$$

Finally, we choose $\zeta = \min[\zeta_1, \zeta_2]$.

It can be obtained according to the above proof that there exists $\zeta > 0$ in such a way that for all response times $\Delta \in [0, \zeta]$, there exists a time $\tau \in \mathbb{R}^+$ such that inter-execution times $\{t_{k+1} - t_k\}_{k \in \mathbb{N}} \geq \tau$ implicitly defined by execution rule (15) with $\xi < 1$. So, Zeno behavior can be avoided for the system. \square

Remark 4. In a multi-thread control structure, the controller is quickly updated under the action of the triggered condition, and even in a limited time, it requires an infinite number of updates. This situation is called Zeno phenomenon. This leads to problems such as system performance degradation and controller failure. In the event-triggered structure with the “one-to-one” feature, infinite switching can be regarded as a continuous update process of the controllers. However, in the control structure with the characteristics of “one-to-many” or “minority-to-majority”, it is impossible for resources to serve multiple controlled plants at the same time. So, the Zeno phenomenon in multi-thread control leads to greater risk of instability. The above proof process shows that the proposed algorithm can avoid the Zeno phenomenon.

We assume that there are n tasks in the system, $T_1, T_2, T_3, \dots, T_n$, controlled by m controllers. These controlled tasks are ranked in a priority order that determines when they are implemented. Since $\gamma(\|e_{p_1,t}\| + \|e_{p_2,t}\| + \dots + \|e_{p_m,t}\| + \|e_{q_1,t}\| + \|e_{q_2,t}\| + \dots + \|e_{q_{n-m},t}\|) = \xi\beta(\|x_{p_1,t}\| + \|x_{p_2,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \|x_{q_2,t}\| + \dots + \|x_{q_{n-m},t}\|)$ is the execution rule, we judge the norm of $\|e_{i,t}\|, i \in \mathbb{Z}^+, (i_{max} = n)$. If $\|e_{1,t}\| > \|e_{2,t}\| > \|e_{3,t}\| > \dots > \|e_{n,t}\|$, then $T_1 > T_2 > T_3 > \dots > T_n$. The m controllers are assigned to tasks T_1, T_2, \dots, T_m .

With the event-triggered mechanism, the proposed multi-thread control algorithm can thoroughly address the mismatch problem between m controllers and n actuators online, which makes the algorithm well suited to the needs of real-world engineering applications.

4. Simulation and Experiments

Now, the three plants of the system are considered in Section 2. For the convenience of simulation with the proposed algorithm, we selected the linear control system in (5) (see [29]),

$$\begin{aligned} \text{plant1} : \begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1 \\ \text{plant2} : \begin{bmatrix} \dot{x}_{21} \\ \dot{x}_{22} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2 \\ \text{plant3} : \begin{bmatrix} \dot{x}_{31} \\ \dot{x}_{32} \end{bmatrix} &= \begin{bmatrix} 0 & 2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x_{31} \\ x_{32} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_3 \end{aligned} \tag{24}$$

stabilized by the linear feedback $u_1 = x_{11} - 4x_{12}$, $u_2 = x_{21} - 4x_{22}$, and $u_3 = x_{31} - 4x_{32}$. Therefore, we obtain

$$K_1 = K_2 = K_3 = \begin{bmatrix} 1 & -4 \end{bmatrix}. \tag{25}$$

4.1. Traditional Event-Triggered Mechanism

Firstly, the traditional event-triggered rules were applied to the above-mentioned system with a mismatch between the number of controllers and the number of controlled plants. The traditional event-triggered mechanism is a fixed threshold strategy. The

threshold under this triggered mechanism is fixed and does not change with the control signal, and the threshold is closely related to the errors of the system.

The traditional event-triggered mechanism is given as follows:

$$t_{k+1} = \inf\{t > t_k | \gamma(\|e_{1,t}\| + \|e_{2,t}\| + \dots + \|e_{n,t}\|) > L\} \tag{26}$$

where L is a fixed threshold related to the errors. At the triggered moment, only the non-triggered controlled plants have error values, and the triggered-controlled plants have errors of 0. When the next trigger time approaches, L approaches the error values of the system; then, the calculation formula of L is as follows:

$$\lim_{t \rightarrow t_{k+1}} \|e_{1,t}\| + \|e_{2,t}\| + \dots + \|e_{n,t}\| = L \tag{27}$$

Figure 2 shows how and at what time the plants were triggered based on the traditional control algorithm. It can be seen from the simulation diagram that the traditional event-triggered mechanism could make three events be alternately controlled. However, if the triggered condition is designed for the purpose of saving resources in a multi-thread control system, the traditional form is not suitable. According to the simulation results, it can be seen that this traditional event-triggered scheme with a fixed threshold did not achieve the desired effect.

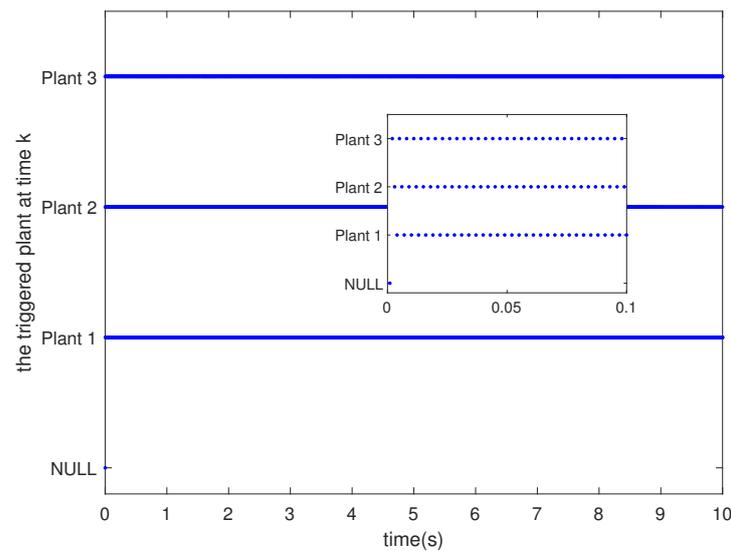


Figure 2. The plants triggered in the system at time k based on the traditional event-triggered mechanism.

4.2. Proposed Event-Triggered Mechanism

These three plants are in one system, using a decision-making process prompted by an event to decide which task to perform at time k . Using $V = x^T Y x$ as a Lyapunov function, we obtain $(A + BK)^T Y + Y(A + BK) = -\Omega$, where Y_1, Y_2 , and Y_3 , and Ω_1, Ω_2 , and $e\Omega_3$ are defined as

$$Y_1 = Y_2 = Y_3 = \begin{bmatrix} 1 & \frac{1}{4} \\ \frac{1}{4} & 1 \end{bmatrix} \tag{28}$$

$$\Omega_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{2} \end{bmatrix}, \Omega_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{5}{2} \end{bmatrix}, \Omega_3 = \begin{bmatrix} \frac{1}{2} & -\frac{3}{4} \\ -\frac{3}{4} & \frac{3}{2} \end{bmatrix}.$$

Using (25), the smallest eigenvalues of Ω_1, Ω_2 , and Ω_3 are $\lambda_m(\Omega_1) > 0.44, \lambda_m(\Omega_2) > 0.17$, and $\lambda_m(\Omega_3) > 0.57$, and we obtain $|K_1^T B_1^T Y_1 + Y_1 B_1 K_1|, |K_2^T B_2^T Y_2 + Y_2 B_2 K_2|$, and $|K_3^T B_3^T Y_3 + Y_3 B_3 K_3|$. We choose $\xi = 0.04$ since $4 * \xi$ must be less than 0.17, and obtain the execution rule $\|e_{1,t}\| + \|e_{2,t}\| + \|e_{3,t}\| \leq \xi(\|x_{1,t}\| + \|x_{2,t}\| + \|x_{3,t}\|)$, in the same way as in [1].

In Figure 3, we can see a lot of intersections in the states of the plants, but the general trends are consistent. The relationship between inputs $u_1, u_2,$ and u_3 and states $x_1, x_2,$ and x_3 of the three plants is given in (25). In fact, since $B, Y,$ and K are the same in the three plants, the assumed relationship is the same, only with respect to the state of each plant. In Figure 4, inputs $u_1, u_2,$ and u_3 of the system are displayed.

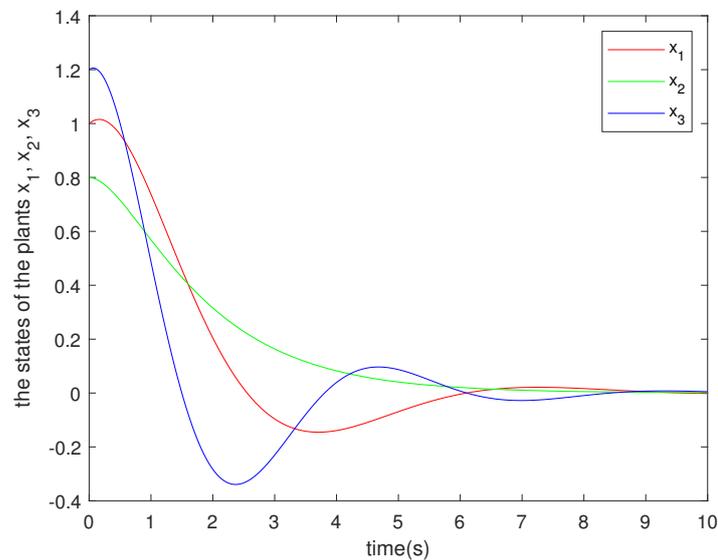


Figure 3. States $x_1, x_2,$ and x_3 of plants in the system.

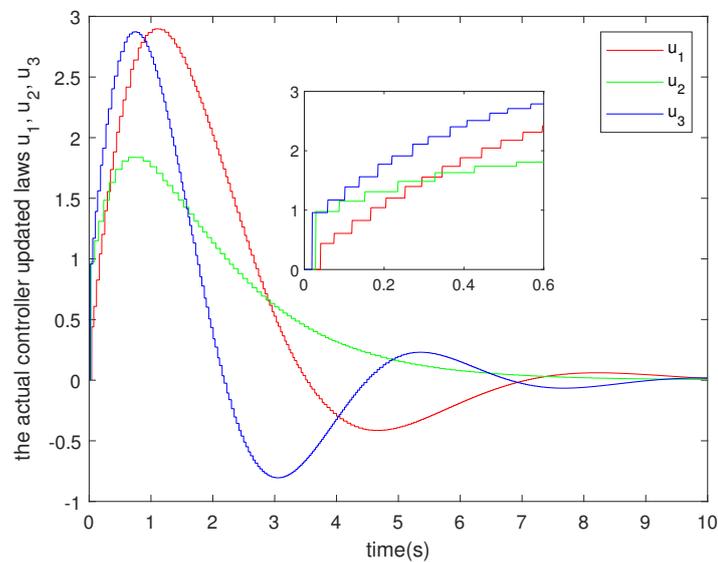


Figure 4. Inputs $u_1, u_2,$ and u_3 of plants in the system.

Figure 5 shows how and at what time the plants were triggered based on the proposed multi-thread real-time control algorithm. During the period from 0 to 10, system 1 was triggered 260 times; system 2, 115 times; and system 3, 305 times. No controller update was triggered for the remaining 9319 times. The suggested algorithm implements the underlying notion of devoting extra Central Processing Unit (CPU) time to control activities in order to increase control performance. In Figure 6, we see that the error norm of the system never reached $\zeta(\|x_{p_1,t}\| + \|x_{p_2,t}\| + \dots + \|x_{p_m,t}\| + \|x_{q_1,t}\| + \|x_{q_2,t}\| + \dots + \|x_{q_{n-m},t}\|)$, and the goal was achieved.

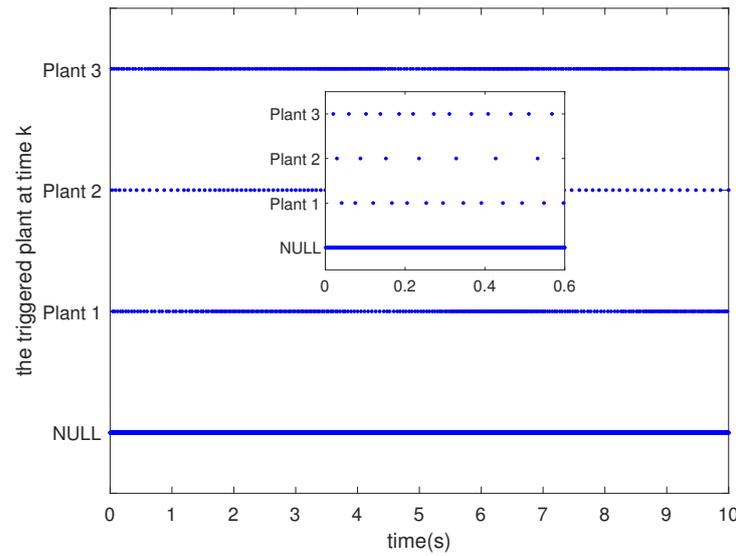


Figure 5. The plants triggered in the system at time k .

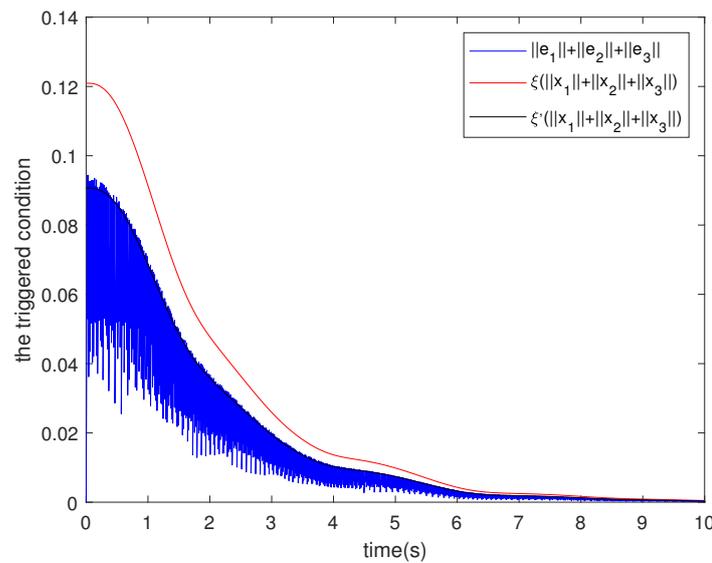


Figure 6. Evolution of $\|e_1\| + \|e_2\| + \|e_3\|$, $\xi(\|x_1\| + \|x_2\| + \|x_3\|)$, and $\xi'(\|x_1\| + \|x_2\| + \|x_3\|)$ for $\Delta = 0.001$.

Comparing Figure 2 and Figure 5, it can be seen that firstly, the modeling scheme of the multi-thread control system is feasible and the description of the update of the controller is reasonable. Secondly, the proposed event-triggered mechanism can be applied to multi-thread control systems and achieve the effect of saving resources.

The simulations show that the controller switching logic of the proposed multi-thread control system is feasible. Existing triggered mechanisms include event-triggered mechanisms, self-triggered mechanisms, time-triggered mechanisms, etc., mainly for the “one-to-one” control system structure. When the triggered condition is met, the system controller is immediately updated. However, a multi-thread control structure, when multiple triggered conditions are met at the same time, faces the problem of how to allocate controller resources first, and it is impossible to simply perform the “trigger and update” operation. The proposed algorithm solves how to design a flexible event-triggered mechanism to allocate system resources according to the different requirements of each subsystem to dynamic perfor-

mance and static performance, so as to ensure the feasibility and stability of the multi-thread control algorithm.

5. Conclusions

This paper presents a novel multi-thread real-time control algorithm based on an event-triggered mechanism to address the mismatch problem in hybrid systems. Both linear and nonlinear system forms are considered to describe the mismatch problem, i.e., the number of controllers is different from that of plants, for multi-thread controllers. To solve the switching problem, an event-triggered mechanism is proposed, and ISS-based triggered conditions in the multi-thread control algorithm are studied. Finally, a minimal time interval is proved to exist between the consecutive execution of one control task and that of another, which avoids the *Zeno phenomenon* for the whole system. The simulation results exhibit that the proposed algorithm is effective and can achieve the resource saving goal for control systems. For the application of the above control methods in multi-thread control systems, our future research trends and development directions mainly include the following aspects: (a) The case where different controllers have different event-triggered mechanisms can be investigated. This event-triggered control system can improve the fault tolerance and reliability of the system and reduce communication overhead and calculation load. (b) Design and optimization of event-triggered interval. The interval between event triggers directly affects the performance and stability of the system. How to design and optimize the event-triggered interval to achieve the optimal performance and maximum stability of the system is still one of the main research objectives of this field.

Author Contributions: Conceptualization, Y.W. and H.G.; methodology, Y.W.; software, Y.Y.; formal analysis, P.X.; writing, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors disclose receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by Natural Science Foundation of Shaanxi under grant 2022JQ-651.

Data Availability Statement: We did not create new data sets in this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tabuada, P. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans. Autom. Control* **2007**, *52*, 1680–1685. [[CrossRef](#)]
2. Charron, S.; Koechlin, E. Divided representation of concurrent goals in the human frontal lobes. *Science* **2010**, *328*, 360–363. [[CrossRef](#)]
3. Fan, J.; Flombaum, J.I.; McCandliss, B.D.; Thomas, K.M.; Posner, M.I. Cognitive and brain consequences of conflict. *Neuroimage* **2003**, *18*, 42–57. [[CrossRef](#)] [[PubMed](#)]
4. Donkers, M.C.F.; Heemels, W.P.M.H. Output-based event-triggered control with guaranteed L_∞ -gain and improved event-triggering. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 3246–3251.
5. Heemels, W.H.; Donkers, M.C.F.; Teel, A.R. Periodic event-triggered control for linear systems. *IEEE Trans. Autom. Control* **2012**, *58*, 847–861. [[CrossRef](#)]
6. Heemels, W.P.; Johansson, K.H.; Tabuada, P. An introduction to event-triggered and self-triggered control. In Proceedings of the 2012 IEEE 51st IEEE conference on decision and control (CDC), Maui, HI, USA, 10–13 December 2012; pp. 3270–3285.
7. Marti, P.; Fuertes, J.M.; Fohler, G.; Ramamritham, K. Improving quality-of-control using flexible timing constraints: Metric and scheduling. In Proceedings of the 23rd IEEE Real-Time Systems Symposium, Austin, TX, USA, 3–5 December 2002; pp. 91–100.
8. Mazo, M., Jr.; Anta, A.; Tabuada, P. An ISS self-triggered implementation of linear controllers. *Automatica* **2010**, *46*, 1310–1314. [[CrossRef](#)]
9. Postoyan, R.; Tabuada, P.; Nešić, D.; Anta, A. A framework for the event-triggered stabilization of nonlinear systems. *IEEE Trans. Autom. Control* **2014**, *60*, 982–996. [[CrossRef](#)]
10. Åaržén, K.E. A simple event-based PID controller. *IFAC Proc. Vol.* **1999**, *32*, 8687–8692. [[CrossRef](#)]
11. Hristu-Varsakelis, D.; Kumar, P.R. Interrupt-based feedback control over a shared communication medium. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 3, pp. 3223–3228.

12. Astrom, K.J.; Bernhardsson, B.M. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 2, pp. 2011–2016.
13. Voulgaris, P. Control of asynchronous sampled data systems. *IEEE Trans. Autom. Control* **1994**, *39*, 1451–1455. [[CrossRef](#)]
14. Tabuada, P.; Wang, X. Preliminary results on state-triggered scheduling of stabilizing control tasks. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 282–287.
15. Mazo, M.; Tabuada, P. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Trans. Autom. Control* **2011**, *56*, 2456–2461. [[CrossRef](#)]
16. Stöcker, C.; Lunze, J. Event-based control of nonlinear systems: An input-output linearization approach. In Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011; pp. 2541–2546.
17. Postoyan, R.; Anta, A.; Heemels, W.P.M.H.; Tabuada, P.; Nešić, D. Periodic event-triggered control for nonlinear systems. In Proceedings of the 52nd IEEE Conference on Decision and Control, Firenze, Italy, 10–13 December 2013; pp. 7397–7402.
18. Alur, R.; Courcoubetis, C.; Halbwachs, N.; Henzinger, T.A.; Ho, P.H.; Nicollin, X.; Yovine, S. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.* **1995**, *138*, 3–34. [[CrossRef](#)]
19. Chen, Z.; Xia, B.; You, C.; Mi, C.C. A novel energy management method for series plug-in hybrid electric vehicles. *Appl. Energy* **2015**, *145*, 172–179. [[CrossRef](#)]
20. Zhang, L. VirtualClock: A new traffic control algorithm for packet-switched networks. *ACM Trans. Comput. Syst. TOCS* **1991**, *9*, 101–124. [[CrossRef](#)]
21. Zheng, Y.F.; Fan, Y. Robot force sensor interacting with environments. *IEEE Trans. Robot. Autom.* **1991**, *7*, 156–164. [[CrossRef](#)]
22. Cervin, A.; Eker, J.; Bernhardsson, B.; Årzén, K.E. Feedback–feedforward scheduling of control tasks. *Real-Time Syst.* **2002**, *23*, 25–53. [[CrossRef](#)]
23. Lemmon, M.; Chantem, T.; Hu, X.S.; Zyskowski, M. On self-triggered full-information h-infinity controllers. In *International Workshop on Hybrid Systems: Computation and Control: 10th International Workshop, Pisa, Italy, April 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 371–384.
24. Li, T.F.; Fu, J. Event-triggered control of switched linear systems. *J. Frankl. Inst.* **2017**, *354*, 6451–6462. [[CrossRef](#)]
25. Li, T.F.; Fu, J.; Deng, F.; Chai, T. Stabilization of switched linear neutral systems: An event-triggered sampling control scheme. *IEEE Trans. Autom. Control* **2018**, *63*, 3537–3544. [[CrossRef](#)]
26. Mahmoud, M.S.; Memon, A.M. Aperiodic triggering mechanisms for networked control systems. *Inf. Sci.* **2015**, *296*, 282–306. [[CrossRef](#)]
27. Liu, A.; Zhang, W.A.; Chen, M.Z.; Yu, L. Moving horizon estimation for mobile robots with multirate sampling. *IEEE Trans. Ind. Electron.* **2016**, *64*, 1457–1467. [[CrossRef](#)]
28. Heinonen, J. *Lectures on Analysis on Metric Spaces*; Springer Science & Business Media: San Francisco, CA, USA, 2001; pp. 20–40.
29. Gao, H.; Wang, Y. A multi-thread control algorithm with update delays based on event-triggered mechanism. *Meas. Control* **2022**, *2022*, 00202940221139568. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.