

Article

# Self-Triggered Formation Control of Nonholonomic Robots

Carlos Santos , Felipe Espinosa , Miguel Martinez-Rey, David Gualda and Cristina Losada 

Electronics Department, University of Alcalá, Engineering School, Campus Universitario, 28871 Alcalá de Henares, Spain; felipe.espinosa@uah.es (F.E.); miguel.martinez@depeca.uah.es (M.M.-R.); david.gualda@uah.es (D.G.); cristina.losada@uah.es (C.L.)

\* Correspondence: carlos.santos@uah.es; Tel.: +34-918-856-545

Received: 23 April 2019; Accepted: 12 June 2019; Published: 14 June 2019



**Abstract:** In this paper, we report the design of an aperiodic remote formation controller applied to nonholonomic robots tracking nonlinear, trajectories using an external positioning sensor network. Our main objective is to reduce wireless communication with external sensors and robots while guaranteeing formation stability. Unlike most previous work in the field of aperiodic control, we design a self-triggered controller that only updates the control signal according to the variation of a Lyapunov function, without taking the measurement error into account. The controller is responsible for scheduling measurement requests to the sensor network and for computing and sending control signals to the robots. We design two triggering mechanisms: centralized, taking into account the formation state and decentralized, considering the individual state of each unit. We present a statistical analysis of simulation results, showing that our control solution significantly reduces the need for communication in comparison with periodic implementations, while preserving the desired tracking performance. To validate the proposal, we also perform experimental tests with robots remotely controlled by a mini PC through an IEEE 802.11g wireless network, in which robots pose is detected by a set of camera sensors connected to the same wireless network.

**Keywords:** self-triggered Lyapunov control; real-time scheduling; practical stability; remote guidance; formation control; nonlinear trajectory tracking

## 1. Introduction

Nowadays, multi-agent system is one of the most studied control topics; within this context, formation control has special relevance [1–3]. Advances in communication wireless technology and embedded devices, among other fields, have enabled the development of underwater robots [4], unmanned aerial vehicles [5] or terrestrial robots [2,3,6] working in formation. These groups of vehicles can properly deal with different challenges, such as navigation, mapping and monitoring works [2]. According to the way in which control actions are remotely applied to the units of the formation, centralized or decentralized control strategies can be defined. The centralized one is simpler in the design stage and is usually more robust against disturbances than decentralized alternatives because all units know the information of the rest of the units of the formation. However, its main disadvantage is that requires a large communication network to send all the information which makes its scalability limited [2,3]. On the contrary, decentralized control offers a solution to this problem [7,8] due to the control signal of each unit being computed according to its own state and the information of its neighbours, while at the same time guaranteeing the stability of the formation.

According to the mission carried out by the formation, three types of strategies can be differentiated [1]. When the position of each unit is measured with respect to a global coordinate system, this strategy is referred to as position-based control. If the unit measures the displacement of

its environment to establish the formation, it is called displacement-based control [9]. Finally, when the units compute the formation, control actions according to the distance with the other units are categorized as distance-based control [3].

One of the main drawbacks in remote formation control is the bandwidth limitation of the communication network [10]. Other common shortcomings are time-varying delays [11,12], packet dropouts [13,14], packet disorder [13,15], and network bandwidth constraints [16,17]. These problems are especially important in the case under study, where the remote controller has to send and receive information about multiple sensor and actuator wireless nodes. To deal with these drawbacks, we propose the use of aperiodic control strategies due to the fact that they offer flexibility to adapt the transmission of sensory and control information through the wireless channel. Aperiodic control techniques employ information about the system state to decide when the control action must be updated. The main aperiodic control techniques are event-triggered (ETC) [18–25] and self-triggered control (STC) [26–32]. In ETC, the triggering mechanism is based on the constant measurement of the plant state. In contrast, STC predicts when the system will fulfil the triggering condition. In order to carry out the forecast of the future states of the plant, STC employs the latest measurement of the plant as well as its model. An extensive description of these controllers is presented in [33] and a comparative study is described in [34].

Several studies have reported aperiodic control of nonholonomic robots tracking nonlinear trajectories. In [2], a consensus control solution to establish a certain formation with a set of LEGO robots is presented. They use the ETC approach to guarantee asymptotic stability of the formation and to reduce information interchange through a Zigbee network. However, the main disadvantage is that each robot position captured by an external camera is continuously evaluated to check the triggering condition. In [18], an event-triggered controller is implemented to track a nonlinear trajectory with a Khepera III robot. A Lyapunov function supports the triggering mechanism working on the Cartesian reference system. As in the previous case, the main disadvantage is the periodic sending of sensory information. In [8], an event-based formation control scheme is designed for wirelessly linked mobile robots. The implemented formation control follows a distributed consensus strategy using neural networks; however, only simulation results are reported. Ref. [22] presents an event-triggered controller applied to a mobile robot wirelessly connected with a remote centre that computes the triggering condition based on the difference between the measured position and the reference one. When this error reaches a predefined threshold, a new event is generated. In [23], a framework is devised to control a vehicle platoon using event-based communication and nonlinear controllers. A decentralized triggering strategy is designed considering a platoon of heterogeneous vehicles with nonlinear dynamics. However, only simulation results are reported. We have also designed and implemented aperiodic remote controllers [3,19]. In [19], we combine event-based estimation with an event-triggered controller for the remote control of only one P3DX robot. In [3], we present a consensus strategy for the remote platooning guidance of autonomous vehicles; however, to simplify the problem, we assume longitudinal and lateral decoupling and we only present simulation results.

The main contributions of the present paper are:

1. Design and implementation of a novel self-triggered Lyapunov-based control for nonlinear systems, using a dual stability approach in order to guarantee practical stability. When the Lyapunov function is greater than a given threshold, asymptotic stability is guaranteed. After that, the system is bounded on the Lyapunov threshold level.
2. Evaluation of centralized and decentralized triggering mechanisms for formation control of nonholonomic robots tracking nonlinear trajectories, comparing both with a periodic implementation. The experimental set-up includes three mobile robot remotely controlled in a scenario with four wireless camera sensors.
3. The design and implementation of a delay compensation strategy that leverages one of the main strengths of STC, namely that the next sampling instant is known in advance.

The paper is structured as follows: the problem statement is described in Section 2; the self-triggered Lyapunov control solution is presented in Section 3; Section 4 details the simulation results; in Section 5, the more relevant implementation aspects are described; the experimental outcomes are shown in Section 6; finally, Section 7 highlights the main paper contributions.

### Notation

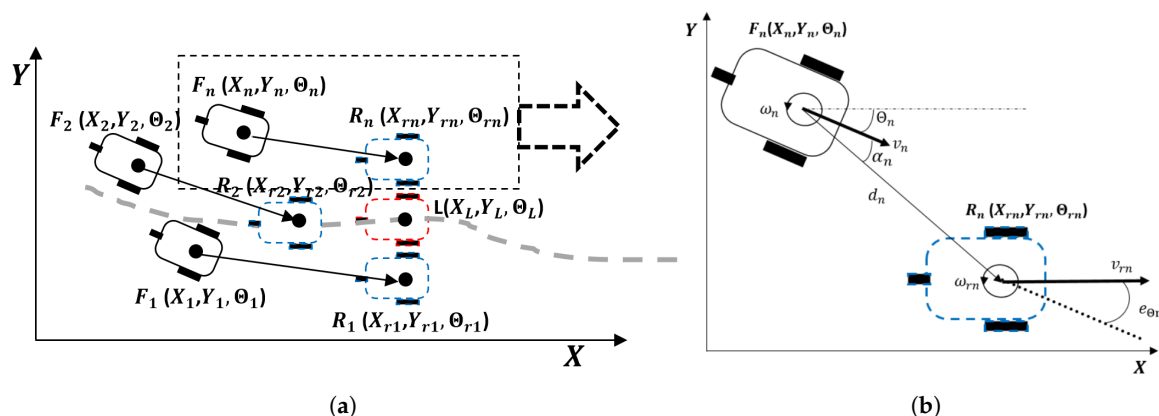
We use the following notation:  $\|v\|$  is the Euclidean norm of the vector  $v \in \mathbb{R}^n$  and  $\wedge$  is the truth-functional operator of logical conjunction. A function is of class  $C^0(\mathcal{D}_x)$  if it is continuous over  $\mathcal{D}_x$ , and it is  $C^l(\mathcal{D}_x)$ ,  $l > 0$  if its derivatives are of class  $C^{l-1}(\mathcal{D}_x)$ . A continuous function  $\rho : [0, a) \rightarrow +\infty, a > 0$  is of class  $\mathcal{K}$  if it is strictly increasing and  $\rho(0) = 0$ . We represent a Lyapunov level set by  $\Omega_{V_k} = \{\zeta(t) \in \mathbb{R}^{n_x} | V(\zeta(t)) \leq V_k\} \subset \mathcal{D}_x$ .

## 2. Problem Statement

A common problem in remote vehicle guidance is the design of control laws to follow a time-parameterized reference. This problem is especially challenging when considering nonlinear trajectories and nonholonomic vehicles. Such is the case of differential-drive robots, which only possess two actuation variables (linear velocity and angular velocity) for locomotion control, whereas the pose of the mobile unit is characterized by three degrees of freedom. In [19], the ETC theory for one robot unit is already presented; here, an extension to an STC formation control for a group of robot units is proposed. We report the design and implementation of an aperiodic formation controller to carry out this task through a wireless network.

### 2.1. Formation Control Problem

The left picture of Figure 1 displays the principal elements of the formation, where the reference pose for each robot  $R_n (X_{rn}, Y_{rn}, \Theta_{rn})$  is generated according to the position of a virtual leader robot  $L (X_L, Y_L, \Theta_L)$  and the current position of each robot is described by  $F_n (X_n, Y_n, \Theta_n)$ . The virtual leader tracks the nonlinear trajectory without error and is used to determine the position of each robot in the formation. To do this, we implement a road-following formation strategy to compute the position of each robot in each time and its velocity references.



**Figure 1.** (a) main variables describing the trajectory tracking in formation strategy: in blue, the reference poses of each robot  $R_n (X_{rn}, Y_{rn}, \Theta_{rn})$ , which are generated according the position of a virtual leader robot  $L (X_L, Y_L, \Theta_L)$  in red and in black the current position of each robot, described by  $F_n (X_n, Y_n, \Theta_n)$ ; (b) main variables describing trajectory tracking of one robot in the formation, where the robot reference is computed according to the position of the virtual leader:  $d_n$  is the distance error calculated from the robot point  $(X_n, Y_n)$  to the reference point  $(X_{rn}, Y_{rn})$ ,  $\alpha_n$  is the orientation error with respect to the target point,  $e_{\Theta_n}$  is the orientation error between the desired orientation to follow the trajectory  $(\Theta_{rn})$  and the orientation of the robot  $(\Theta_n)$ .

Road-following formation allows two kinds of coordination: the spatial coordination and temporal coordination [35]. In the case of spatial coordination, the trajectories of each unit within the formation are coordinated, thus the shape of formation adapts to the conditions of the trajectory; therefore, the distances between the different agents of the formation can vary at certain times. On the contrary, in a temporally-coordinated formation, the distance between the different agents is permanently kept constant. Considering that our challenge is the nonlinear, trajectory tracking of a robot formation, we apply the spatial coordination despite the fact that this does not guarantee a constant relative position between the mobile units every time. For this reason, the trajectory generator is responsible for limiting the curvature of the trajectories and establishing an equal linear velocity for all the units in the formation in order to avoid possible collision situations of the different units. To implement this trajectory generator, whose mission is to compute the position and velocity references of each real robot in the formation, we decided to use a virtual leader as a reference.

Figure 1a details the tracking problem for each robot in the formation. The current pose of each robot is characterized by  $(X_n, Y_n, \Theta_n)$ . The robot kinematics allows for establishing the relationship between pose time derivative and the related linear  $v$  and angular  $\omega$  velocities. For the nonholonomic robot model, we have:

$$\begin{aligned}\dot{X}_n(t) &= v_n(t) \cos(\Theta_n(t)), \\ \dot{Y}_n(t) &= v_n(t) \sin(\Theta_n(t)), \\ \dot{\Theta}_n(t) &= \omega_n(t).\end{aligned}\quad (1)$$

We also define these error variables for each agent (Figure 1b): angle error  $\alpha_n$  is defined as the different between the current orientation and the orientation needed to reach the target point, orientation error  $e_{\Theta_n}$  is the different between the desired  $(\Theta_{rn})$  and actual  $(\Theta_n)$  orientations; finally, distance error  $d_n$  is calculated as the distance between the current and reference positions:

$$d_n(t) = \sqrt{(X_{rn}(t) - X_n(t))^2 + (Y_{rn}(t) - Y_n(t))^2}, \quad (2)$$

$$\alpha_n(t) = \text{atan2}(Y_{rn}(t) - Y_n(t), X_{rn}(t) - X_n(t)) - \Theta_n(t). \quad (3)$$

Contrary to previous studies of aperiodic tracking controllers such as [18,27] that use Cartesian coordinates, we work with polar coordinates due to the fact that they allow for reach the target point asymptotically using a time-invariant and smooth feedback controller [36,37]. On the contrary, if the robot is localized with Cartesian coordinates, the control presents some limitations indicated by Brockett's results [38].

Considering the kinematic model of each robot in the formation, Equation (1), the time derivative of its orientation and distance errors in polar coordinates [37] are:

$$\dot{d}_n(t) = -v_n(t) \cos(\alpha_n(t)) + v_{rn}(t) \cos(\alpha_n(t) - e_{\Theta_n}(t)), \quad (4)$$

$$\dot{\alpha}_n(t) = -\omega_n(t) + v_n(t) \frac{\sin(\alpha_n(t))}{d_n(t)} - v_{rn}(t) \frac{\sin(\alpha_n(t) - e_{\Theta_n}(t))}{d_n(t)},$$

$$\dot{e}_{\Theta_n}(t) = \omega_{rn}(t) - \omega_n(t).$$

The state vector of each unit  $\zeta_n(t)$  includes the variables presented in the last equation  $\zeta_n(t) = [d_n(t), \alpha_n(t), e_{\Theta_n}(t)]$ .

**Assumption 1.** The virtual leader reference velocities,  $v_{rL}(t)$  and  $\omega_{rL}(t)$ , and therefore the reference velocities of each robots,  $v_{rn}(t)$  and  $\omega_{rn}(t)$ , are defined piecewise constant and are known beforehand in order to introduce their variations in the self-triggered scheduler.

## 2.2. Lyapunov Formation Controller

In this section, we present the nonlinear trajectory controller used by each robot in the formation. Inspired by [37], we obtain the following Lyapunov based linear and angular control laws applied to Equation (4):

$$v_n(t) = K_v d_n(t) \cos(\alpha_n(t)) + v_{rn}(t) \cos(e_{\Theta_n}(t)), \quad (5)$$

$$\omega_n(t) = \dot{\Theta}_{md_n}(t) + v_{md_n}(t) (K_\omega (K_v d_n(t) \sin(\alpha_n(t)) + v_{rn}(t) \sin(e_{\Theta_n}(t))) + d_n(t) \sin(\alpha_n(t))), \quad (6)$$

where  $\Theta_{md_n}$  is the modified desired heading angle of each robot:

$$\Theta_{md_n}(t) := \text{atan2}(K_v d_n(t) \sin(\alpha_n(t) - e_{\Theta_n}(t)), v_{rn}(t) + K_v d_n(t) \cos(\alpha_n(t) - e_{\Theta_n}(t))) + \Theta_{rn}(t), \quad (7)$$

and  $v_{md}$  is equal to:

$$v_{md_n}(t) := \sqrt{v_{rn}(t)^2 + (K_v d_n(t))^2 + 2v_{rn}(t)K_v d_n(t) \cos(\alpha_n(t) - e_{\Theta_n}(t))}, \quad (8)$$

and  $K_v > 0$  and  $K_\omega > 0$  are control gains related to the linear and angular velocities, respectively. The closed-loop Lyapunov function for each robot is:

$$V_n(\xi_n(t)) = \frac{1}{2} d_n(t)^2 + 1 - \cos(\Theta_{md_n}(t) - \Theta_n(t)). \quad (9)$$

As it is demonstrated in [37], Equation (9) fulfils Lyapunov function properties.

Finally, the Lyapunov function that describes the stability of our road-following formation control is:

$$V(\xi(t)) = \sum_{i=1}^n V_n(\xi_n(t)). \quad (10)$$

This function also fulfils the Lyapunov conditions because, as demonstrated in [39], a sum of Lyapunov functions is itself a Lyapunov function.

**Remark 1.** The choice of Equation (10) is based on the aggregation of the different Lyapunov functions of each unit. This way, the overall stability of the formation is analysed. This is possible because, in the road following strategy, the objectives of each unit are independent of the state of the rest of the units.

## 3. Lyapunov Based Self-Triggering Control Proposal

First, we introduce briefly the problem formulation of nonlinear aperiodic control. Consider an autonomous nonlinear system:

$$\dot{\xi}(t) = f(\xi(t), u(t)), \quad (11)$$

where  $\xi(t) \in \mathcal{D}_\xi \subset \mathbb{R}^{n_x}$  and  $u(t) \in \mathcal{D}_u \subset \mathbb{R}^{n_u}$ , both domains containing the origin.

**Assumption 2.** There exists a differentiable state feedback law  $K : \mathcal{D}_\xi \rightarrow \mathcal{D}_u$  such that the origin of the closed-loop continuous system

$$\dot{\xi}(t) = f(\xi(t), K(\xi(t))) \quad (12)$$

is the unique locally asymptotically stable equilibrium point in  $\mathcal{D}_\xi$ .

If Assumption 2 is fulfilled, we can obtain a Lyapunov function  $V(\xi(t))$  and apply it to the system described in Equation (12) with the following properties [40,41]:

$$\begin{aligned} \underline{\gamma}(\|\xi(t)\|) &\leq V(\xi(t)) \leq \bar{\gamma}(\|\xi(t)\|), \\ \dot{V}(\xi(t)) &= \frac{\partial V(\xi(t))}{\partial \xi} f(\xi(t), K(\xi(t))) \leq -\gamma_1(\|\xi(t)\|), \\ \left\| \frac{\partial V(\xi(t))}{\partial \xi} \right\| &\leq \gamma_2(\|\xi(t)\|), \end{aligned} \quad (13)$$

where  $\underline{\gamma}, \bar{\gamma}, \gamma_1, \gamma_2$  are  $\mathcal{K}$ -class functions.

**Assumption 3.** Assume that:

1. The function  $f \in C^l(\mathcal{D}_\xi \times \mathcal{D}_u)$ , with  $l \geq 3$ .
2. The functions  $\underline{\gamma}, \gamma_1 \in \mathcal{K}$  in Equation (13) are such that  $\underline{\gamma}^{-1}, \gamma_1$  are Lipschitz continuous on the working compact set  $(\mathcal{D}_\xi)$ . The Lipschitz constants on  $\mathcal{D}_\xi$  of functions  $\underline{\gamma}^{-1}$  and  $\gamma_1$  are represented by  $L_{\underline{\gamma}^{-1}}$  and  $L_{\gamma_1}$ , respectively.

Between updates, the control signal is held constant with a zero-order hold (ZOH) implementation. When the update times  $t_k$  are reached, the signal is recalculated with the new measurement:

$$u(t) = K(\xi(t_k)), \quad t \in [t_k, t_{k+1}[, \quad k \in \mathbb{N}. \quad (14)$$

The sampled-data system dynamics with this implementation are:

$$\dot{\xi}(t) = f(\xi(t), K(\xi(t_k))), \quad t \in [t_k, t_{k+1}[, \quad k \in \mathbb{N}. \quad (15)$$

After the introduction of the fundamentals of nonlinear aperiodic control, we present our Lyapunov based self-triggering condition proposal. We design a self-triggering condition assuming that the full state information is available at the measurement instants. Unlike most previous aperiodic control published works [18,21,27,29], we propose a self-triggering condition that triggers the controller taking into account the variation of the Lyapunov function and not the measurement error.

**Definition 1.** Semiglobal practical stability [42]: a system  $\dot{\xi}(t) = f(\xi(t), K(\xi(t)))$  is said to be semiglobally practically stable if for any (arbitrarily large) compact set  $\mathcal{D}_\xi$  and any arbitrarily small compact set  $\mathcal{D}_{V_0}$  including the origin, every trajectory of the system with  $\xi \in \mathcal{D}_\xi$  is defined for all  $t \in [0, \infty[$  and there exists  $T \in [0, \infty[$  such that  $\xi \in \mathcal{D}_{V_0}$  for all  $t \in [T, \infty[$ , where  $\mathcal{D}_{V_0} \subset \mathcal{D}_\xi$ .

**Theorem 1.** Consider that Assumptions 2 and 3 hold for  $\mathcal{D}_\xi$  and  $\xi(t_0) \in \mathcal{D}_\xi$ . If the control signal is updated according to the following triggering condition,

$$t_{k+1} = \min\{t > t_k | (\dot{V}(\xi(t)) \geq 0 \wedge V(\xi(t)) \geq V_0)\}, \quad (16)$$

the system (15) converges asymptotically to the bounded set  $\mathcal{D}_{V_0}$ , where  $\mathcal{D}_{V_0} = \{\xi(t) | V(\xi(t)) < V_0\}$ .

**Proof.** By Theorem 3.2 of [43], the triggering condition (16) presented in Theorem 1 enforces semiglobal practical stability of the system described in Equation (15), as long no Zeno executions are presented. To show that the proposed triggering condition does not introduce Zeno executions, we employ Theorem 4.1 of [29] to guarantee a strictly positive minimum dwell-time between updates.

Theorem 4.1 of [29] is based on the error function  $g(t)$ :

$$\begin{aligned} g(t) &:= f(\xi(t), K(\xi(t_k))) - f(\xi(t), K(\xi(t))), \\ &t \in [t_k, t_{k+1}[, \quad k \in \mathbb{N}. \end{aligned} \quad (17)$$

Using this error function, the dynamics of the sample-data system (15) are rewritten as:

$$\dot{\xi}(t) = f(\xi(t), K(\xi(t))) + g(t), \quad t \in [t_k, t_{k+1}[ , \quad k \in \mathbb{N}, \quad (18)$$

and the following sampling rule is proposed by [29]:

$$t_{k+1} = \min\{t > t_k \mid \|g(t)\| > e_{V_0}\}. \quad (19)$$

This way, Equation (19) ensures semiglobal practical stability of the closed-loop system for any  $e_{V_0} > 0$  and guarantees a positive minimum dwell-time if Assumptions 2 and 3 hold for any (arbitrarily large) compact set  $\mathcal{D}_{\xi}$  and  $\xi(t_0) \in \mathcal{D}_{\xi}$ .

If we can prove that condition described in Equation (16) generates equal or greater inter-execution times than (19) for some selection of  $e_{V_0} > 0$ , then, by virtue of Theorem 4.1 of [29], we obtain the desired result: the existence of a strictly positive minimum dwell-time, and thus no Zeno execution is possible. Observe that:

$$\begin{aligned} \dot{V}(\xi(t)) &= \frac{\partial V(\xi(t))}{\partial \xi} f(\xi(t), K(\xi(t_k))) = \\ &= \frac{\partial V(\xi(t))}{\partial \xi} f(\xi(t), K(\xi(t))) + \\ &= \frac{\partial V(\xi(t))}{\partial \xi} (f(\xi(t), K(\xi(t_k))) - f(\xi(t), K(\xi(t)))) \leq \\ &= -\gamma_1(\|\xi(t)\|) + \gamma_2(\|\xi(t)\|)\|g(t)\|, \end{aligned} \quad (20)$$

which, together with Equation (19), implies:

$$\dot{V}(\xi(t)) \leq -\gamma_1(\|\xi(t)\|) + \gamma_2(\|\xi(t)\|_{\mathcal{L}_{\infty,k}})e_{V_0}. \quad (21)$$

Thus, Equation (19) enforces the following implication:

$$\dot{V}(\xi(t)) < 0 \quad \forall t, \quad \text{if} \quad \|\xi(t)\| > \gamma_1^{-1}(\gamma_2(\|\xi(t)\|_{\mathcal{L}_{\infty,k}})e_{V_0}). \quad (22)$$

Inspecting the triggering condition (16), we observed that it *only* demands new updates if:

$$\dot{V}(\xi(t)) \geq 0 \quad \wedge \quad V(\xi(t)) > V_0. \quad (23)$$

Thus, from Equation (13), we have that Equation (16) forces updates when:

$$\dot{V}(\xi(t)) \geq 0 \quad \wedge \quad \|\xi(t)\| \geq \bar{\gamma}^{-1}(V_0). \quad (24)$$

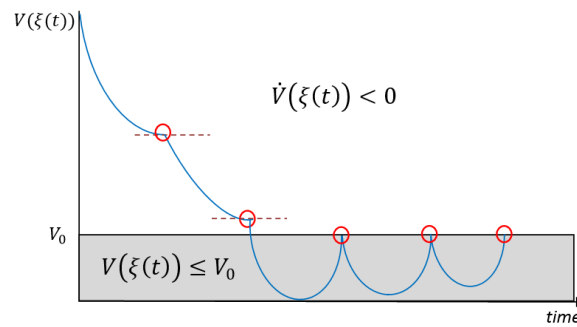
Select now an  $e_{V_0}$  such that:

$$\gamma_1^{-1}(\gamma_2(\|\xi(t)\|_{\mathcal{L}_{\infty,k}})e_{V_0}) \leq \bar{\gamma}^{-1}(V_0). \quad (25)$$

Such an  $e_{V_0} > 0$  always exists as long as  $\|\xi(t)\|_{\mathcal{L}_{\infty,k}}$  is upper bounded, from the properties of  $\mathcal{K}_{\infty}$  functions and  $V_0 > 0$ . Note that  $\|\xi(t)\|_{\mathcal{L}_{\infty,k}}$  is upper bounded as  $V(\xi(0)) < \infty$  and thus due to Equation (16)  $V(\xi(t)) < \infty$  for all positive times, which by Equation (13) implies the required boundedness.

Finally, by virtue of Equation (25), we have that, when triggering function (24) takes place, the triggering condition from Equation (19) is certainly violated.  $\square$

Figure 2 describes graphically the Theorem 1 triggering condition.



**Figure 2.** Description of the Theorem 1 triggering condition (16). If the Lyapunov function is greater than  $V_0$ , the system is updated every time the derivative of the Lyapunov function is non-negative. If the system converges to the invariant set defined by  $V_0$ , the system is triggered only when the Lyapunov function reaches the threshold  $V_0$ .

**Remark 2.** Theorem 1 guarantees  $\dot{V}(\xi(t)) < 0$  when the system is outside the invariant set  $\mathcal{D}_{V_0}$ . The choice of  $V_0$  establishes a trade-off between the invariant set  $\mathcal{D}_{V_0}$  and the inter-execution times  $[t_k, t_{k+1}]$ . Decreasing the value of  $V_0$  reduces the size of the set  $\mathcal{D}_{V_0}$  as well as the inter-execution times.

**Remark 3.** To find a dwell-time, we use Theorem 4.1 from [29] because our triggering condition (16) presents no inter-execution times lower than [29] outside of the bounded set defined by  $\mathcal{D}_{V_0}$ .

Finding a lower bound of the inter-sampling time  $t_{min}$  is still an open issue [29]. Nevertheless, using Remark 3, we guarantee that  $t_{min}$  is positive and, in order to avoid Zeno-executions, we force a  $t_{min}$ , taking into account the hardware constraints.

We design two different triggering strategies, one synchronous and centralized and the other asynchronous and decentralized. For the centralized strategy, we evaluate the triggering condition (16) with the global Lyapunov function of the formation (10). In this case, the triggering instants are:

$$t_{k+1} = \min\{t > t_k | (\dot{V}(\xi(t)) \geq 0 \wedge V(\xi(t)) \geq V_0)\}. \quad (26)$$

We also implement a decentralized solution for the formation, in which each robot unit triggers its own events asynchronously. Formation stability is also guaranteed because this strategy is more triggering demanding than the global one:

$$t_{nk+1} = \min\{t > t_{nk} | (\dot{V}_n(\xi_n(t)) \geq 0 \wedge V_n(\xi_n(t)) \geq V_0/n)\}. \quad (27)$$

**Remark 4.** For the decentralized triggering strategy (27), the threshold  $V_0$  is divided by the number of robot units, as can be seen in the equation. Thus, we guarantee that the formation converges asymptotically to the same bounded set as when we implement the centralized condition (26).

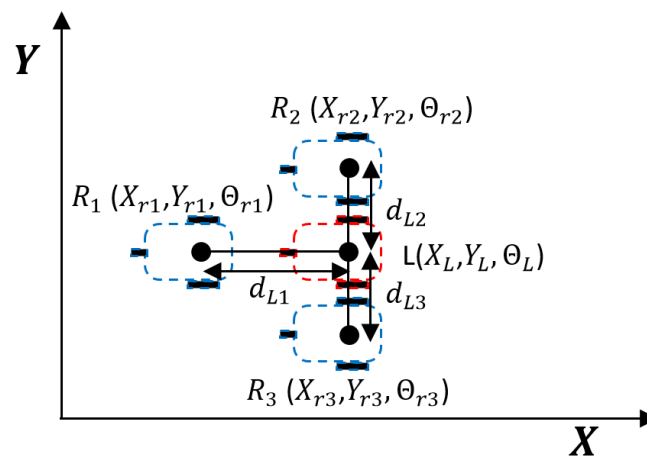
Another important parameter of STC is  $t_{max}$ , which means the maximum time the system is allowed to run in open loop and it is a design parameter [26]. This implies that, if the inter-execution time  $(t_{k+1} - t_k)$  obtained with the triggering condition is greater than the  $t_{max}$ , the next update instant  $t_{k+1}$  is modified to  $t_k + t_{max}$ . This parameter is used to reduce the time that the system works in open loop, which is the main disadvantage of STC with respect to ETC. This way the designer can fix this parameter to limit the possible uncertainties or disturbances that can affect the system.

#### 4. Simulation Results

In this section, we describe a simulation of formation control of three robots based on the previously mentioned robot kinematic model (1). A virtual leader tracks the desired nonlinear trajectory



without error and determines the position of each robot in the formation and its velocity references using a road-following formation strategy, as described in Figure 3.

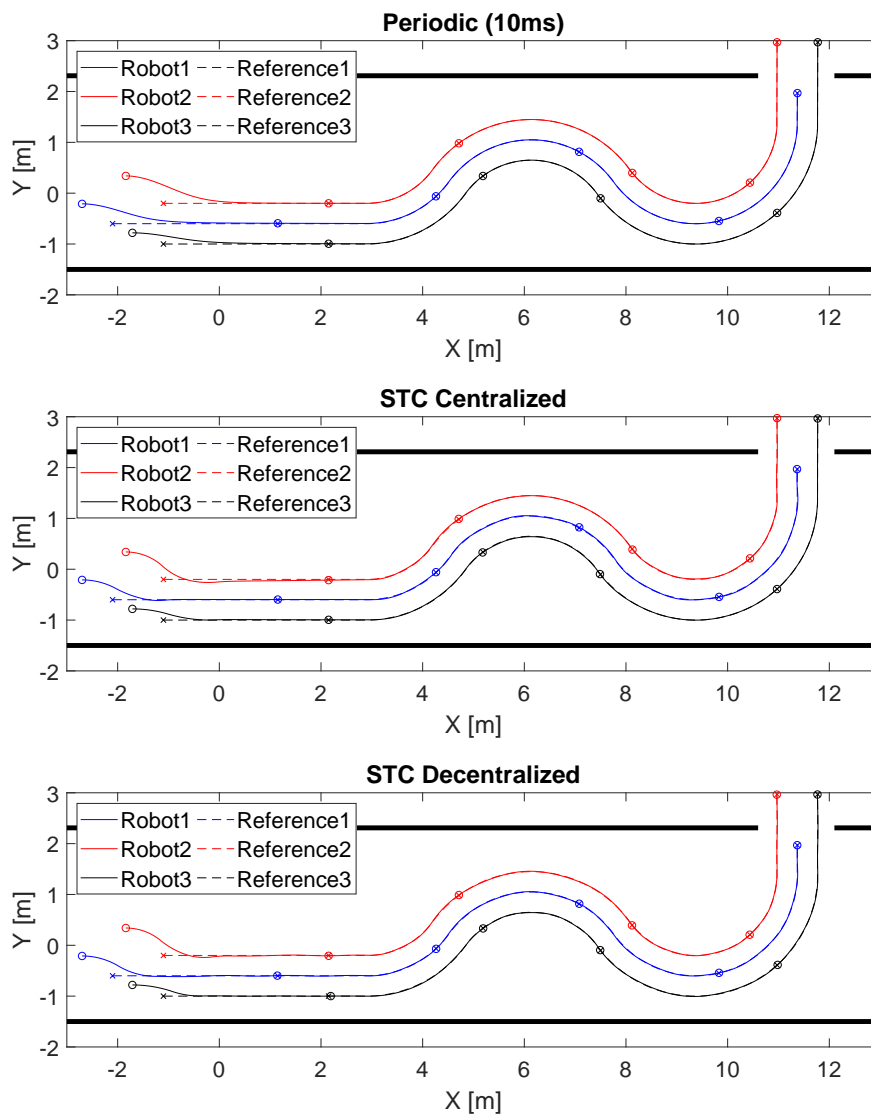


**Figure 3.** Main variables describing the formation strategy: in blue color the reference poses of each robot  $R_n (X_{rn}, Y_{rn}, \Theta_{rn})$ , which are generated according the position of the Virtual Leader  $L (X_L, Y_L, \Theta_L)$  in red.

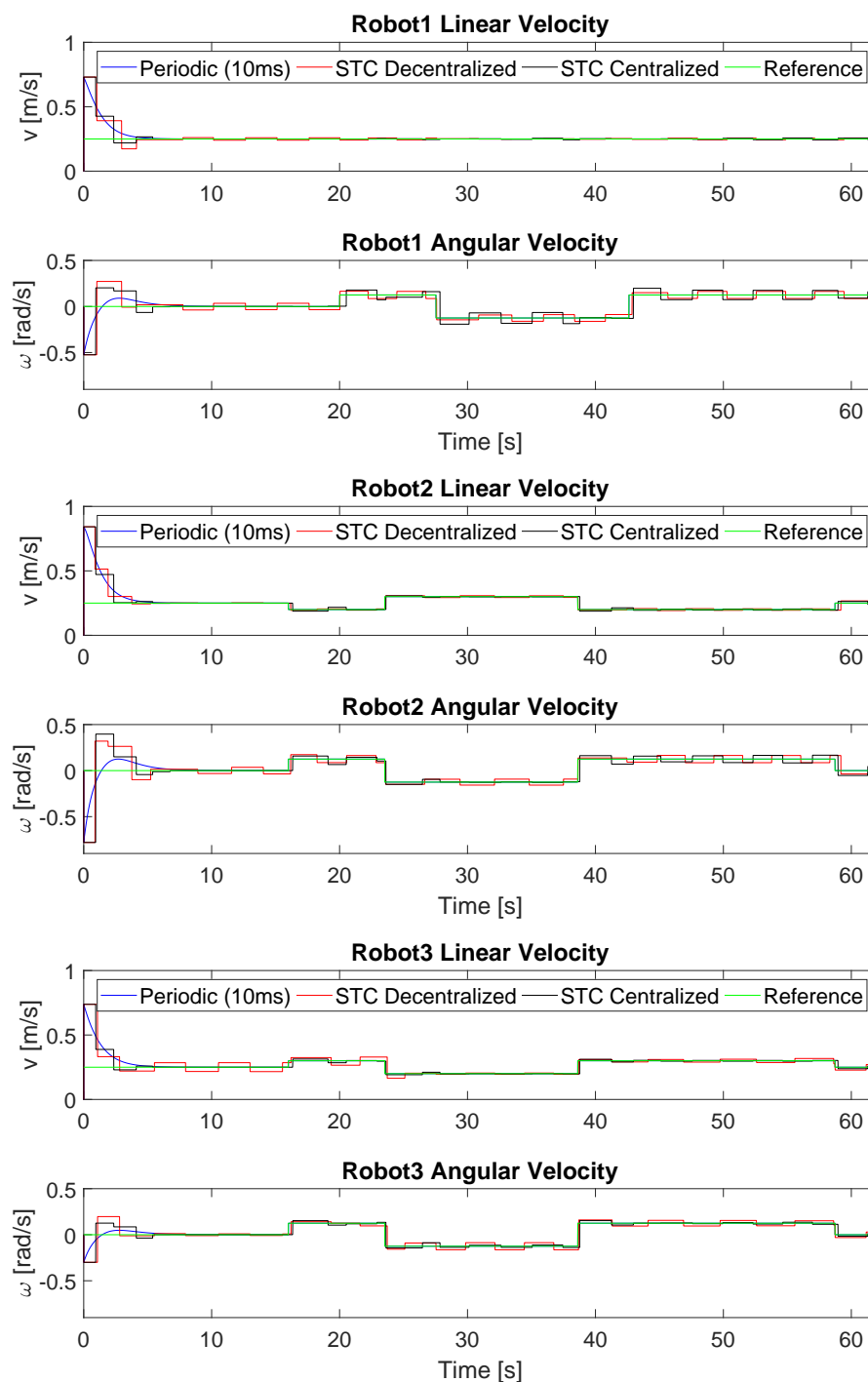
The distance between the reference robots and the virtual leader is determined by  $d_{Ln}$ . In our example, we set  $d_{L1} = 1$  m and  $d_{L2} = d_{L3} = 0.4$  m. To implement the spatially coordinated road-following formation [35], delayed velocity references of the Virtual Leader are sent to the robots depending on the longitudinal distance from the robot to the Virtual Leader, and adjusted linear velocity references for the Virtual Leader are sent to the robots depending on the curvature of the route in each track and the lateral distance from the robot to the Virtual Leader. In the formation shown in Figure 3, Robot 1 receives the delayed velocity reference and Robot 2 and Robot 3 the references adjusted for the curvature of each track. Note that, with our formation strategy, the relative positions of the vehicles at a given point in time are not coordinated at every moment, but the trajectories of the vehicles are always spatially aligned.

The control parameters of the robots are:  $K_v = 0.8$ ,  $K_w = 0.05$ ,  $t_{min} = 10$  ms,  $t_{max} = 5$  s and  $V_0 = 10^{-4}$ . The initial robot locations [m,m,rad] are  $F_1 = [-2.70; -0.21; 0]$ ,  $F_2 = [-1.84; 0.34; 0]$  and  $F_3 = [-1.71; -0.78; 0]$ . The initial formation reference points are  $R_1 = [-2.1; -0.6; 0]$ ,  $R_2 = [-1.1; -0.2; 0]$  and  $R_3 = [-1.1; -1; 0]$ . The nonlinear trajectory tracking by the formation of robots is shown in Figure 4. We obtained good tracking performance with all the control solutions tested: periodic ( $T_s = 10$  ms), STC with centralized triggering (26) and STC with decentralized triggering (27). The sampling period is chosen according to the hardware restrictions imposed by the P3DX robots and the minimum time between events of the STC control ( $T_s = 10$  ms).

Figure 5 shows the linear and angular velocity references and commands of the formation applying the three different strategies. In the case of periodic control, there is a practically continuous evolution in which the error is corrected gradually until being accurately linked to the reference at the cost of a large number of updates compared to the STC strategies. In the case of the STCs, changes in the velocity commands only occur at specific moments in time. For the centralized solution, each robot update is activated at the same time, with the corresponding additional reduction in the number of updates, while, for the decentralized strategy, this is performed at different moments according to triggering conditions (26) and (27), respectively.



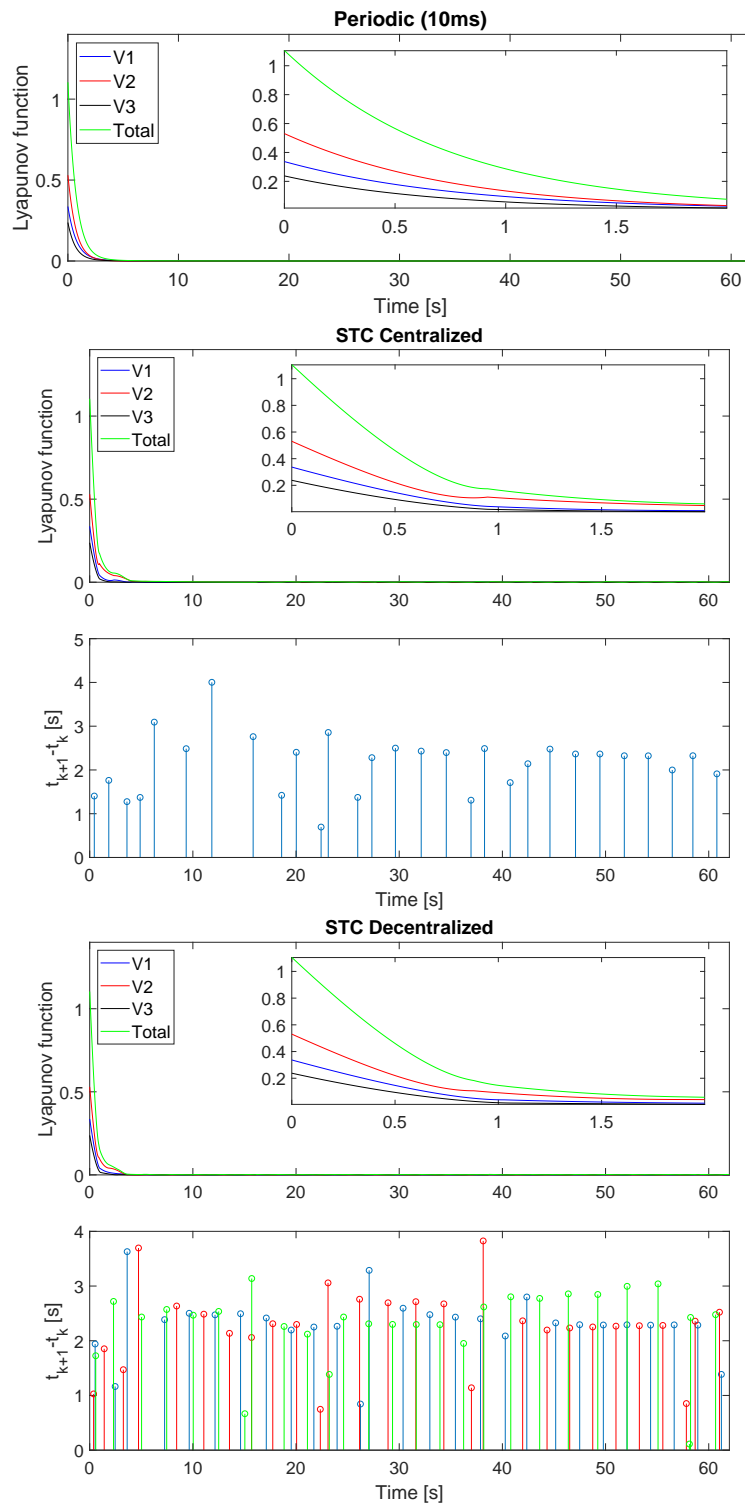
**Figure 4.** Nonlinear trajectory tracked by the formation of three robotic units. Robot 1 is represented in blue, Robot 2 in red and Robot 3 in black. The route followed by each robot is presented with a continuous line, the reference with a discontinuous line, and every 13 s the punctual position of each of them is shown with a circle and an x, respectively. The top figure represents the periodic implementation, the middle one the STC with centralized triggering (26) and the bottom the STC with decentralized triggering (27).



**Figure 5.** Linear and angular velocity references and commands of the formation. Robot 1 is represented in top figure, Robot 2 in middle one and Robot 3 in the bottom. The velocity done by each robot is presented in blue for the periodic implementation, in red for the STC with decentralized triggering (27) and in black for the centralized triggering (26). The velocity reference of each unit is presented in green.

Figure 6 describes the evolution of the Lyapunov function of the formation (10), the Lyapunov function of each robot (9) and the inter-execution-times of the STC implementations. As can be seen, the system reaches the equilibrium point properly with all three strategies. In the case of decentralized triggering (27), the gradient of the Lyapunov function of all the robots is negative at all times in the transient regime. In contrast, in the case of centralized triggering (26), there are specific moments in

which a robot Lyapunov function slope may be positive; however, the formation slope is negative at all times.



**Figure 6.** Lyapunov function of the formation including a zoom of the first 2 s and inter-execution times for STC strategies. Robot 1 is drawn in blue, Robot 2 in red, Robot 3 in black and the Formation in green. Top figure represents the periodic implementation, the middle one the STC with centralized triggering (26) and the bottom the STC with decentralized triggering (27).

Table 1 quantifies the number of updates and Root Mean Squared (RMS) value for distance error (2) of the formation with the control strategies tested. With the designed control, STC significantly reduces the number of updates in the control signals applied to the robot, as well as achieving a performance of the same magnitude as periodic implementation. With centralized triggering, there is a greater reduction in the number of updates because the commands are sent to the three robots at the same time. On the other hand, with decentralized triggering, the event is evaluated at the precise moment for each robot, but at the cost of a larger number of updates.

**Table 1.** Comparison of the update number and RMS value of distance error concerning the different control strategies for nonlinear, trajectory tracking (see Figure 4): periodic implementation, STC with centralized triggering (26) and STC with decentralized triggering (27).

	Periodic [10 ms]				STC Centralized (26)				STC Decentralized (27)			
	R1	R2	R3	Formation	R1	R2	R3	Formation	R1	R2	R3	Formation
Updates	6500	6500	6500	<b>6500</b>	29	29	29	<b>29</b>	29	30	28	<b>87</b>
$d_{RMS}$ [m]	1.31	1.39	1.04	<b>3.74</b>	1.12	1.58	0.92	<b>3.62</b>	1.19	1.39	1.37	<b>3.95</b>

We also present a statistical study to validate the controllers. In this study, we carry out 100 simulations with the different strategies. We use a random combination for the initial pose conditions, all of them within a radius of 2 m from the previous starting positions,  $F_1$ ,  $F_2$  and  $F_3$ . Table 2 shows the mean and standard deviation for performance and updates of each control technique. For the case under study, our STC solutions provides the best results with a relevant reduction in the controller updates comparing with the periodic implementation, without diminishing performance.

**Table 2.** Comparison of the average (AVG) and the standard deviation (STD) values of update number and RMS value of distance after 100 simulation results concerning the different control strategies for nonlinear, trajectory tracking: periodic implementation, STC with centralized triggering (26) and STC with decentralized triggering (27).

	Periodic [10 ms]				STC Centralized (26)				STC Decentralized (27)			
	R1	R2	R3	Formation	R1	R2	R3	Formation	R1	R2	R3	Formation
AVG Updates	6200	6200	6200	<b>6200</b>	32.70	32.70	32.70	<b>32.70</b>	30.37	30.89	30.22	<b>91.48</b>
STD Updates	0	0	0	<b>0</b>	4.78	4.78	4.78	<b>14.34</b>	3.26	3.69	6.99	<b>10.51</b>
AVG $d_{RMS}$ [m]	3.54	3.56	3.82	<b>10.91</b>	3.35	3.23	3.16	<b>9.74</b>	3.17	3.26	3.28	<b>9.71</b>
STD $d_{RMS}$ [m]	1.68	1.71	1.84	<b>4.96</b>	1.25	1.23	1.27	<b>3.64</b>	1.25	1.23	1.28	<b>3.69</b>

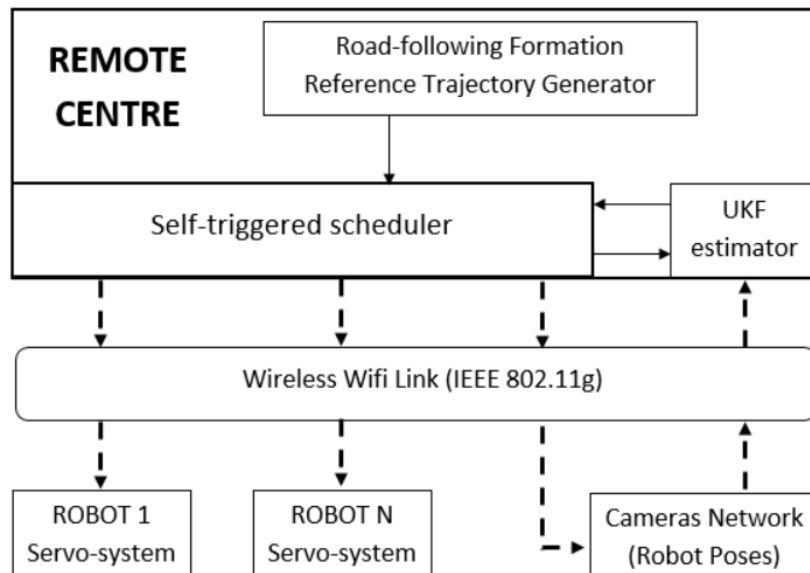
## 5. Remote Centre Task Scheduler

In this section, we describe one of the most critical aspects of our remote centre: the design of our task scheduler and its interaction with the robots and the wireless sensor nodes. To enable a better understanding of the experimental validation of the aperiodic strategies applied to trajectory tracking of a robot formation, we detail the role of the remote center. Figure 7 shows the main elements of our implementation scenario.

The implementation scenario has four main elements:

1. A non-holonomic mobile robot formation. Each robot locally implements a periodic servosystem for linear and angular velocity tracking.
2. A set of sensor nodes covers the entire experimental area and provides each robot with pose information using computer vision.
3. An IEEE 802.11g standard wireless network that links the remote centre to the robots and the set of sensor nodes.
4. A remote centre that performs the principal tasks: trajectory generation for the virtual leader considered the reference for the road-following formation, trajectory generation for each real

robot with respect to the virtual leader, new measurement request to the camera network, pose estimation of each robot unit based on the UKF, and application of the self-triggered control strategy.



**Figure 7.** Description of the main elements in our implementation scenario: the remote centre, the robots formation, the camera network and the wireless communication channel. The remote centre carry out the principal tasks: reference trajectory generation of each robot according to the road-following strategy, unscented Kalman filter(UKF) and self-triggered control based on Lyapunov functions for asynchronous request of measurements and actuations on the robots.

### 5.1. Delay Compensation

In the network control system practical implementation, the time delays must be taken into account. Leveraging one of the main strengths of STC, namely that the next sampling instant is known in advance, we designed a channel delay compensation strategy inspired by [30].

We consider the following delays:

1.  $\tau_t$  is the maximum network delay, it is the maximum time to transmit a message via the wireless communication network.
2.  $\tau_s$  is the maximum sensor delay, i.e., the time between the start of a measurement acquisition in the sensor node and the instant it is ready to be sent to the remote centre. This time includes image acquisition and processing.
3.  $\tau_c$  is the control computing time of the remote centre.
4.  $\tau_r$  is the dominant constant time that characterizes the robot dynamics.

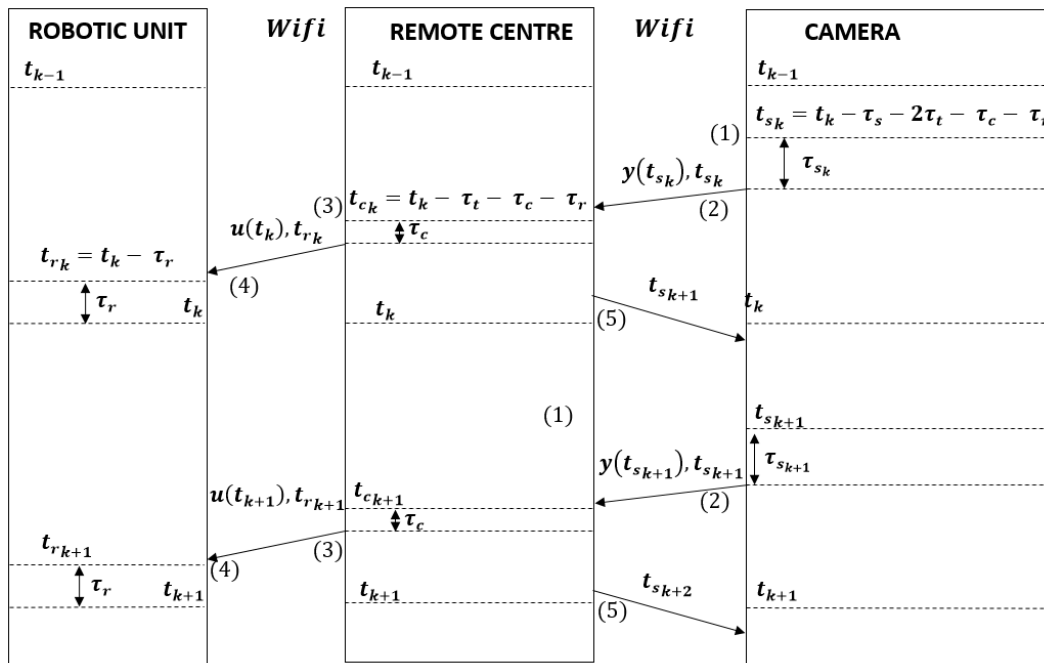
Quantifying these delays requires clock synchronization between the sensors, robots and remote centre. We decided to use Network Time Protocol (NTP) due to it allowing a correct synchronization in the order of magnitude of our time step ( $\Delta = 10$  ms). In Figure 8, we describe our strategy to compensate delays and robot dynamics:

The main idea of the global delay compensation strategy is to pre-calculate robot control actions using the predictive capability of the self-triggered control. This strategy involves the following five steps:

1. At time instant  $t_{s_k}$ , the cameras start the measurement process with image acquisition. This time is previously indicated to the cameras by the STC of the remote centre. Computation of this time is explained in Step 5.

2. When the pose measurement ( $t_{s_k}$ ) is ready, the cameras send it and the acquisition time ( $t_{s_k}$ ) to the remote centre.
3. At time instant  $t_{c_k}$ , the UKF of the remote centre corrects the prediction of the states for time instant  $t_{s_k}$  with the measurement sent by the camera ( $y(t_{s_k})$ ). Next, the UKF predicts the states at time instant  $t_k$  and sends this information to the STC controller. With this information, the STC generates linear and angular speed commands for each robot ( $u(t_k)$ ) and computes the next update instant ( $t_{k+1}$ ). The control signal ( $u(t_k)$ ) and the application time are sent to the robot ( $t_{r_k}$ ).
4. At time instant  $t_{r_k}$ , the control signal is applied to the robot; thus, the desired control signal is reached at time instant  $t_k$  compensating the robot dynamics.
5. After sending the control signal to the robots, the remote centre sends the next measurement acquisition time to the cameras ( $t_{s_{k+1}}$ ). This time is computed taking into account the next sampling instant  $t_{k+1}$  and all the delays ( $t_{s_{k+1}} = t_{k+1} - \tau_s - 2\tau_t - \tau_c - \tau_r$ ).

**Remark 5.** In certain situations, the time between update times ( $t_{k+1} - t_k$ ) will be less than the sum of all the delays ( $\tau_s + 2\tau_t + \tau_c + \tau_r$ ). In this case, the system will work with the state prediction provided by the UKF until a new measurement is available to include in the UKF correction.



**Figure 8.** Communication protocol between the remote centre, sensor and robot, with the delay compensation strategy.

## 5.2. Control Design Dependent on State Estimation

We implement an UKF which takes care of bounding the estimation error between measurement. Our self-triggered controller computes the update times based on the output of this filter. The principal drawbacks of this implementation is that bounded peaks may be obtained in the Lyapunov function when the estimation error grows; consequently, we have designed an aperiodic controller that minimizes this effect by guaranteeing practical stability of the system instead of asymptotic stability. Thus, our strategy achieves longer times between control updates than periodic or aperiodic alternatives aimed at guaranteeing asymptotic or exponential stability.

## 6. Experimental Tests

The tests are carry out with three Pioneer P3DX robots (Product specs can be found at <http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx>) with additional hardware elements [44]. The remote centre control is implemented in a NUC5i3RYH mini PC (Model number NUC5i3RYH. Product specs can be found at <https://www.intel.com/content/www/us/en/nuc/nuc-kit-nuc5i3ryh-brief.html>). The test area is sensed by four Kinect RGB cameras (Microsoft, Redmond, WA, USA) connected to an identical mini PC. All mini PCs run Ubuntu 12.04, as an operating system. An AprilTag marker is situated on the top of each P3DX robot to obtain its pose using the AprilTags fiducial system presented in [45]. Figure 9 shows this experimental scenario.



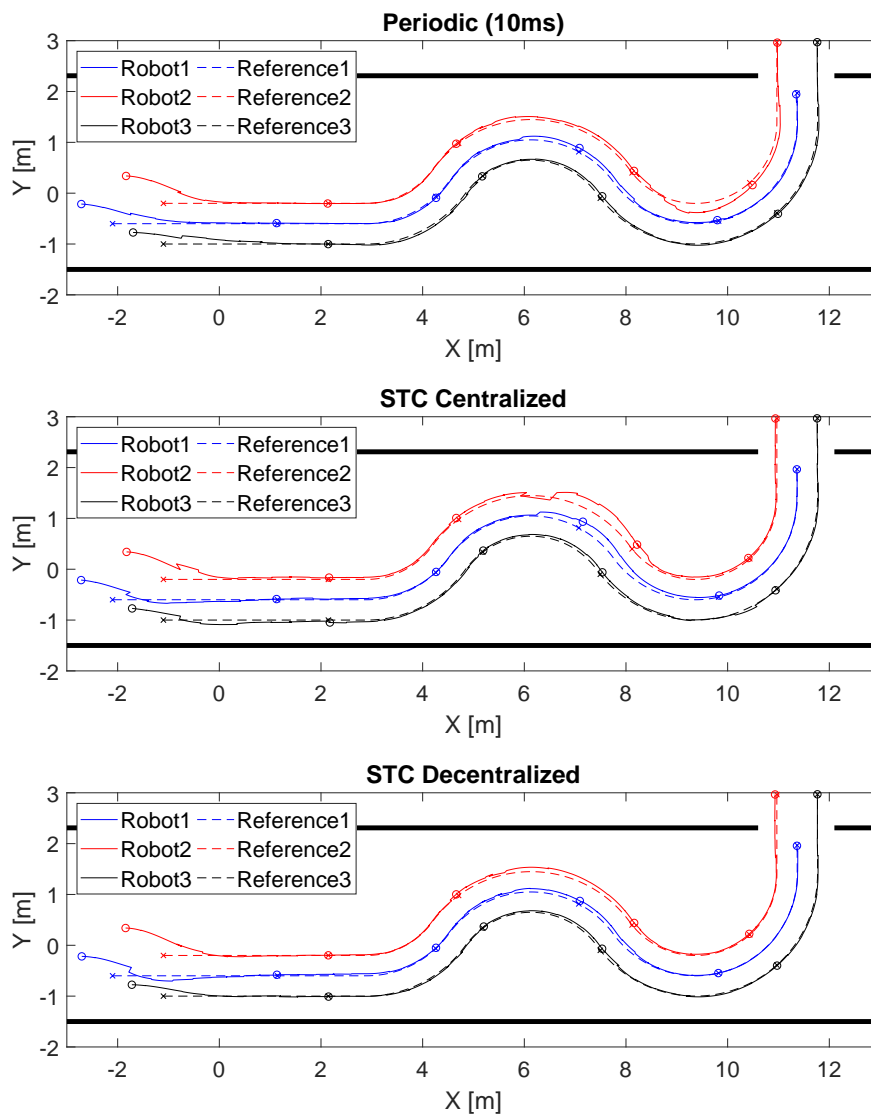
**Figure 9.** Picture of the working area with four Kinect 2 camera sensor and the formation of three P3DX robot wirelessly controlled by a miniPC.

### Results

Below, we present three experimental tests with the road-following formation strategy, the control parameters and the initial conditions described in Section 4. The control parameters of each robots are:  $K_v = 0.8$ ,  $K_w = 0.05$ ,  $t_{min} = 10$  ms,  $t_{max} = 5$  s and  $V_0 = 10^{-4}$ . The initial robot locations are  $F_1 = [-2.70; -0.21; 0]$ ,  $F_2 = [-1.84; 0.34; 0]$  and  $F_3 = [-1.71; -0.78; 0]$ . The initial formation points are  $R_1 = [-2.1; -0.6; 0]$ ,  $R_2 = [-1.1; -0.2; 0]$  and  $R_3 = [-1.1; -1; 0]$ .

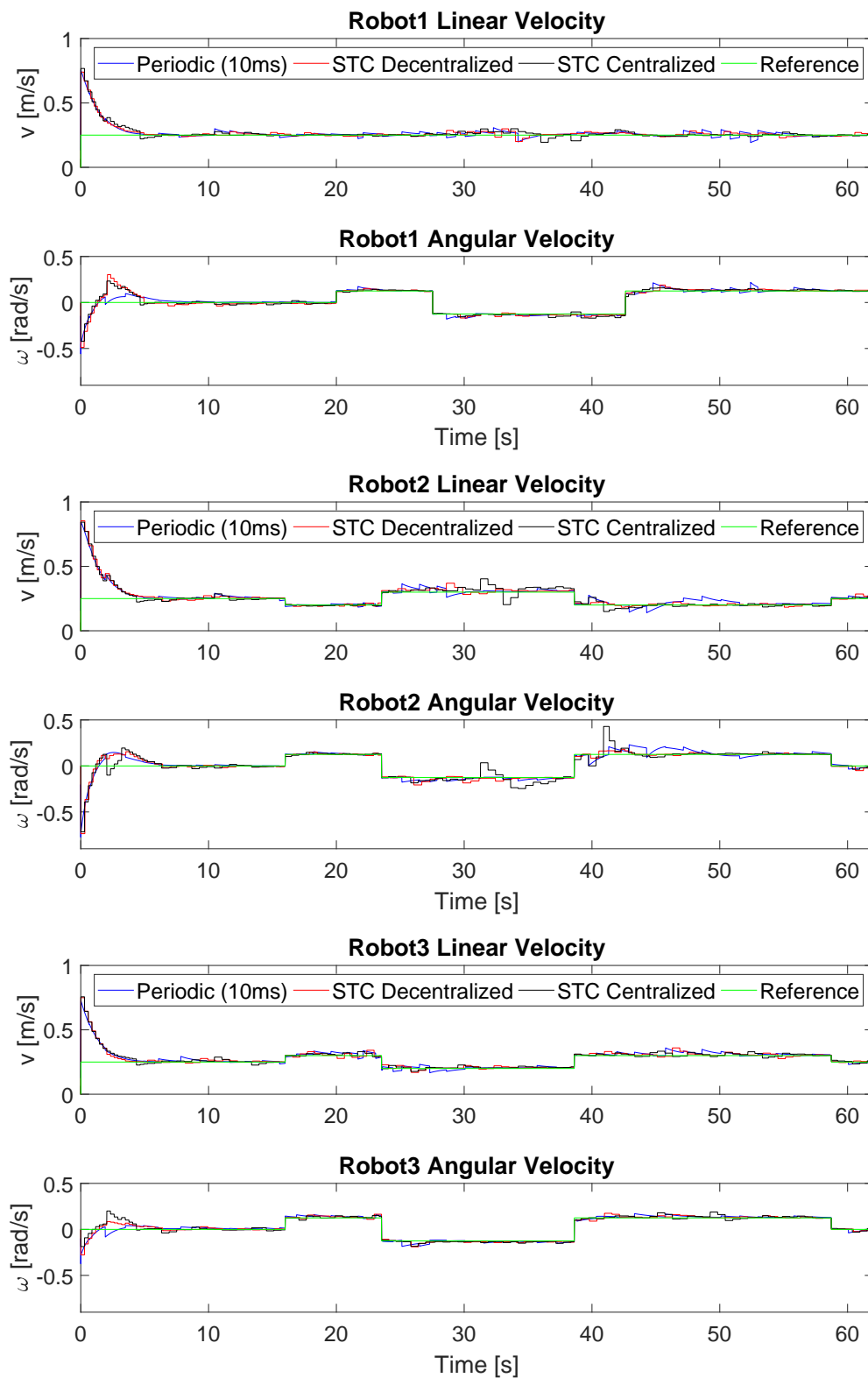
Figure 10 depicts the formation of P3DX robot trajectory tracking for the three solutions tested (periodic ( $T_s = 10$  ms), STC with centralized triggering (26) and STC with decentralized triggering (27). We include a video, see Supplementary Materials, showing the experimental test using the STC with decentralized triggering). If we compare it with Figure 4, it can be seen that performance deteriorates a little because the simulation assumes a perfect communication channel, robot models based only on kinematics and position sensing without error.





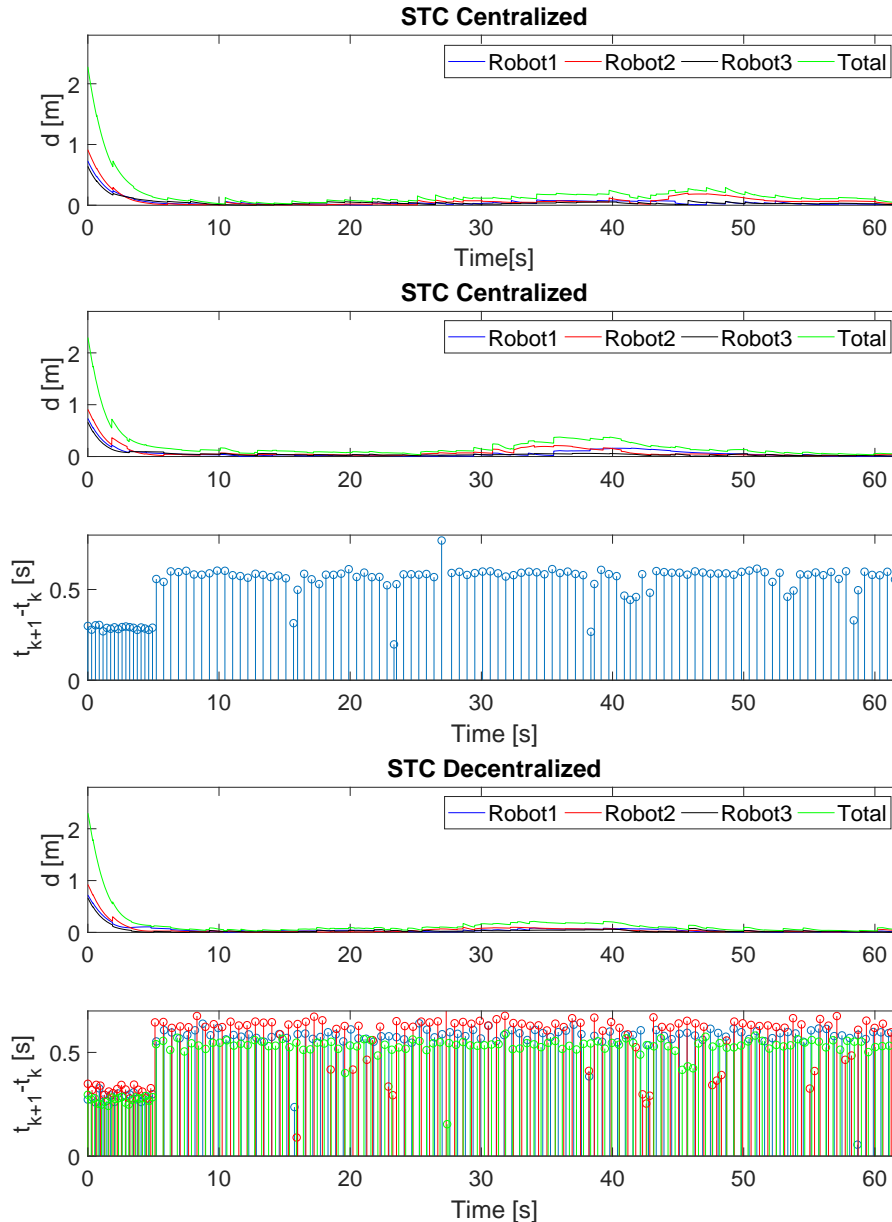
**Figure 10.** The nonlinear, trajectory tracking by the formation of three robotic units. Robot 1 is represented in blue, Robot 2 in red and Robot 3 in black, the route done by each robot is presented with a continuous line, the reference with a discontinuous line, and every 13 s the punctual position of each of them is shown with a circle and an x, respectively. The top figure represents the periodic implementation, the middle one the STC with centralized triggering (26) and the bottom the STC with decentralized triggering (27).

Figure 11 shows the linear and angular velocity reference and commands of the formation for each robot applying the three different strategies. As in simulation, the STC implementations applied few updates in the velocity commands; these updates are activated at the same time in the centralized solution and at different moments in the decentralized solution.



**Figure 11.** Linear and angular velocity references and commands of the formation. Robot 1 is represented in the top figure, Robot 2 in the middle one and Robot 3 in the bottom. The velocity done by each robot is presented in blue for the periodic implementation, in red for the STC with decentralized triggering (27) and in black for the centralized triggering (26). The velocity reference of each unit is presented in green.

Figure 12 describes the evolution of the formation distance error (2), the error of each robot (9) and the inter-execution times for the STC strategies. As can be seen, the system is bounded around the null distance error with the three strategies. When a new measurement is obtained, the UKF corrects its prediction and a small jump appears in the distance error.



**Figure 12.** Lyapunov function of the formation and inter-execution times for STC strategies. Robot 1 is represented in blue, Robot 2 in red, Robot 3 in black and the Formation in green. The top figure represents the periodic implementation, the middle one the STC with centralized triggering (26) and the bottom the STC with decentralized triggering (27).

Table 3 quantifies the number of updates and the Root Mean Squared (RMS) value for distance error (2) of the formation with the different control strategies. The experimental test results follow the same trend as those for simulation. Thus, STC significantly reduces the number of changes in the control signals applied to the robot, as well as achieving a performance of the same magnitude as periodic implementation.

**Table 3.** Comparison of the update number and RMS value of distance concerning the different control strategies for trajectory tracking, including the periodic implementation, the STC with centralized triggering (26) and the STC with decentralized triggering (27).

	Periodic [10 ms]				STC Centralized (26)				STC Decentralized (27)			
	R1	R2	R3	Formation	R1	R2	R3	Formation	R1	R2	R3	Formation
Updates	6500	6500	6500	<b>6500</b>	125	125	125	<b>125</b>	123	119	136	<b>378</b>
$d_{RMS}$ [m]	2.89	4.61	3.12	<b>10.62</b>	2.94	4.90	3.72	<b>11.56</b>	2.24	3.34	3.05	<b>8.63</b>

## 7. Conclusions

This paper describes the design and implementation of a self-triggered remote road-following formation controller applied to nonholonomic robots tracking nonlinear trajectories using an external positioning sensor network.

We design a novel self-triggered Lyapunov-based controller, using a dual stability approach in order to guarantee practical stability. Unlike most previous work in the field of aperiodic control, the measurement error is not taken into account when the controller is triggered, thus achieving longer inter-execution times as the solution is less conservative. This triggering condition is implemented adopting both a centralized and a decentralized approach, highlighting the advantages of each strategy.

STC solutions have been implemented taking into account practical drawbacks such as communication network delays, actuation time delays due to robot dynamics, and acquisition and processing times associated with the camera sensor to obtain the pose of each unit. To minimize the effect of these problems on the stability and performance of the closed loop system, we designed and implemented a delay compensation strategy. This strategy leverages the predictive capability of self-triggered control to pre-calculate control actions and compensate delays.

To validate the theoretical proposal, several experimental tests are conducted using the designed self-triggered controllers and a periodic implementation. We report simulation results, showing that our control solution significantly reduces the need for communication in comparison with periodic implementations while preserving the desired tracking performance. To validate the proposal, we also perform experimental tests with a formation of three P3DX robots remotely controlled through an IEEE 802.11g wireless network with a mini PC, in which robot pose is detected by a set of camera sensors connected to the same wireless network. The results obtained for our self-triggered control solution indicate that using our aperiodic proposals rather than periodic solution yields a significant reduction in the number of updates without degrading performance, rendering this kind of control solution especially interesting for networked control systems and wireless sensor network integration.

For future work, we are going to consider the design of an ETC to implement strategies of obstacle avoiding combined with collision avoiding between the different units of the formation.

**Supplementary Materials:** A video showing the results of one of the performed experimental tests is available online at <http://www.geintra-uah.org/idi/demostraciones/demostraciones#self19>.

**Author Contributions:** C.S. and F.E. are responsible of the theoretical contribution. M.M.-R. worked on the simulation results of the robotic formation. D.G. and C.L. were involved in the design and implementation of the demonstrator.

**Funding:** This work was supported by the University of Alcala through the project with reference UAH-GP2018-3.

**Acknowledgments:** Authors would like to thank the support of the GEINTRA research group at the University of Alcala.

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

## References

- Oh, K.K.; Park, M.C.; Ahn, H.S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [[CrossRef](#)]

2. Guinaldo, M.; Fábregas, E.; Farias, G.; Dormido-Canto, S.; Chaos, D.; Sánchez, J.; Dormido, S. A Mobile Robots Experimental Environment with Event-Based Wireless Communication. *Sensors* **2013**, *13*, 9396–9413. [[CrossRef](#)] [[PubMed](#)]
3. Santos, C.; Espinosa, F.; Santiso, E.; Martínez, M.; Mazo, M. Aperiodic Consensus Control for Tracking Nonlinear Trajectories of a Platoon of Vehicles. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 1983–1988.
4. Yan, Z.; Xu, D.; Chen, T.; Zhang, W.; Liu, Y. Leader–Follower Formation Control of UUVs with Model Uncertainties, Current Disturbances, and Unstable Communication. *Sensors* **2018**, *18*, 662. [[CrossRef](#)] [[PubMed](#)]
5. Dong, X.; Yu, B.; Shi, Z.; Zhong, Y. Time-Varying Formation Control for Unmanned Aerial Vehicles: Theories and Applications. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 340–348. [[CrossRef](#)]
6. Lee, W.; Kim, D. Autonomous Shepherding Behaviors of Multiple Target Steering Robots. *Sensors* **2017**, *17*, 2729. [[CrossRef](#)] [[PubMed](#)]
7. Huang, C.C.; Chiu, Y.H.; Wen, C.Y. Using Hybrid Angle/Distance Information for Distributed Topology Control in Vehicular Sensor Networks. *Sensors* **2014**, *14*, 20188–20216. [[CrossRef](#)] [[PubMed](#)]
8. Guzey, H.M.; Vignesh, N.; Jagannathan, S.; Dierks, T.; Acar, L. Distributed consensus-based event-triggered approximate control of nonholonomic mobile robot formations. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 3194–3199.
9. Li, X.; Chen, H. An Interactive Control Algorithm Used for Equilateral Triangle Formation with Robotic Sensors. *Sensors* **2014**, *14*, 7229–7247. [[CrossRef](#)] [[PubMed](#)]
10. Pawlowski, A.; Guzmán, J.L.; Rodríguez, F.; Berenguel, M.; Sánchez, J.; Dormido, S. The influence of event-based sampling techniques on data transmission and control performance. In Proceedings of the 2009 IEEE Conference on Emerging Technologies Factory Automation, Mallorca, Spain, 22–25 September 2009; pp. 1–8.
11. Cloosterman, M.B.G.; van de Wouw, N.; Heemels, W.P.M.H.; Nijmeijer, H. Stability of Networked Control Systems With Uncertain Time-Varying Delays. *IEEE Trans. Autom. Control* **2009**, *54*, 1575–1580. [[CrossRef](#)]
12. Luan, X.; Shi, P.; Liu, F. Stabilization of Networked Control Systems With Random Delays. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4323–4330. [[CrossRef](#)]
13. Cuenca, A.; García, P.; Albertos, P.; Salt, J. A Non-Uniform Predictor-Observer for a Networked Control System. *Int. J. Control Autom. Syst.* **2011**, *9*, 1194–1202. [[CrossRef](#)]
14. Li, H.; Shi, Y. Network-Based Predictive Control for Constrained Nonlinear Systems With Two-Channel Packet Dropouts. *IEEE Trans. Ind. Electron.* **2014**, *61*, 1574–1582. [[CrossRef](#)]
15. Zhao, Y.B.; Liu, G.P.; Rees, D. Actively Compensating for Data Packet Disorder in Networked Control Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2010**, *57*, 913–917. [[CrossRef](#)]
16. Casanova, V.; Salt, J.; Cuenca, A.; Piza, R. Networked Control Systems: Control structures with bandwidth limitations. *Int. J. Syst. Control Commun.* **2009**, *1*, 267–296. [[CrossRef](#)]
17. Ojha, U.; Chow, M.Y. Realization and validation of Delay Tolerant Behavior Control based Adaptive Bandwidth Allocation for networked control system. In Proceedings of the 2010 IEEE International Symposium on Industrial Electronics, Bari, Italy, 4–7 July 2010; pp. 2853–2858.
18. Postoyan, R.; Bragagnolo, M.; Galbrun, E.; Daafouz, J.; Nesic, D.; Castellan, E. Nonlinear event-triggered tracking control of a mobile robot: Design, analysis and experimental results. *IFAC Proc. Vol.* **2013**, *46*, 318–323. [[CrossRef](#)]
19. Santos, C.; Martínez-Rey, M.; Espinosa, F.; Gardel, A.; Santiso, E. Event-Based Sensing and Control for Remote Robot Guidance: An Experimental Case. *Sensors* **2017**, *17*, 2034. [[CrossRef](#)] [[PubMed](#)]
20. Borgers, D.P.; Heemels, W.P.M.H. Event-Separation Properties of Event-Triggered Control Systems. *IEEE Trans. Autom. Control* **2014**, *59*, 2644–2656. [[CrossRef](#)]
21. Chen, X.; Hao, F.; Ma, B. Periodic event-triggered cooperative control of multiple non-holonomic wheeled mobile robots. *IET Control Theory Appl.* **2017**, *11*, 890–899. [[CrossRef](#)]
22. Socas, R.; Dormido, S.; Dormido, R.; Fabregas, E. Event-Based Control Strategy for Mobile Robots in Wireless Environments. *Sensors* **2015**, *15*, 30076–30092. [[CrossRef](#)]
23. Linsenmayer, S.; Dimarogonas, D.V.; Allgöwer, F. Event-based vehicle coordination using nonlinear unidirectional controllers. *IEEE Trans. Control Netw. Syst.* **2018**, *5*, 1575–1584. [[CrossRef](#)]
24. Miskowicz, M. *Event-Based Control and Signal Processing*; CRC Press: Boca Raton, FL, USA, 2016.

25. Hu, Y.; Lu, Q.; Hu, Y. Event-Based Communication and Finite-Time Consensus Control of Mobile Sensor Networks for Environmental Monitoring. *Sensors* **2018**, *18*, 2547. [[CrossRef](#)]
26. Mazo, M.; Anta, A.; Tabuada, P. On self-triggered control for linear systems: Guarantees and complexity. In Proceedings of the 2009 European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 3767–3772.
27. Eqtami, A.; Heshmati-alamdari, S.; Dimarogonas, D.V.; Kyriakopoulos, K.J. Self-triggered Model Predictive Control for nonholonomic systems. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 638–643.
28. Santos, C.; Mazo, M., Jr.; Espinosa, F. Adaptive self-triggered control of a remotely operated P3DX robot: Simulation and experimentation. *Robot. Auton. Syst.* **2014**, *62*, 847–854. [[CrossRef](#)]
29. Tiberi, U.; Johansson, K. A simple self-triggered sampler for perturbed nonlinear systems. *Nonlinear Anal. Hybrid Syst.* **2013**, *10*, 126–140. [[CrossRef](#)]
30. Santos, C.; Espinosa, F.; Santiso, E.; Mazo, M. Aperiodic Linear Networked Control Considering Variable Channel Delays: Application to Robots Coordination. *Sensors* **2015**, *15*, 12454–12473. [[CrossRef](#)] [[PubMed](#)]
31. Velasco, M.; Marti, P.; Fuertes, J.M. The self-triggered task model for real-time control systems. In Proceedings of the 24th IEEE Real-Time Systems Symposium RTSS 2003, Cancun, Mexico, 3–5 December 2003; pp. 1–4.
32. Zhu, Z.; Sun, J.; Qi, G.; Chai, Y.; Chen, Y. Frequency Regulation of Power Systems with Self-Triggered Control under the Consideration of Communication Costs. *Appl. Sci.* **2017**, *7*, 688. [[CrossRef](#)]
33. Heemels, W.P.M.H.; 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.
34. Santos, C.; Espinosa, F.; Santiso, E.; Martinez-Rey, M. A simplified event-triggering condition non-dependent on measurement error. In Proceedings of the 2017 3rd International Conference on Event-Based Control, Communication and Signal Processing (EBCCSP), Funchal, Portugal, 24–26 May 2017; pp. 1–6.
35. Ono, M.; Droge, G.; Grip, H.; Toupet, O.; Scrapper, C.; Rahmani, A. Road-following formation control of autonomous ground vehicles. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 4714–4721.
36. Wang, Z.; Liu, Y. Visual regulation of a nonholonomic wheeled mobile robot with two points using Lyapunov functions. In Proceedings of the 2010 International Conference on Mechatronics and Automation (ICMA), Xi'an, China, 4–7 August 2010; pp. 1603–1608.
37. Amoozgar, M.; Zhang, Y. Trajectory tracking of Wheeled Mobile Robots: A kinematical approach. In Proceedings of the 2012 IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA), Suzhou, China, 8–10 July 2012; pp. 275–280.
38. Brockett, R.W. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*; Birkhauser: Boston, MA, USA, 1983; pp. 181–191.
39. Grines, V.; Medvedev, T.; Pochinka, O. *Dynamical Systems on 2- and 3-Manifolds*; Springer: Cham, Switzerland, 2016.
40. Khalil, H. *Nonlinear Systems*; Prentice Hall: Upper Saddle River, NJ, USA, 2002.
41. Kurzweil, J. On the inversion of Lyapunov's second theorem on stability of motion. *Czechoslov. Math. J.* **1956**, *6*, 217–259.
42. Lamnabhi-Lagarrigu, F.; Loria, A.; Panteley, E.; Laghrouche, S. *Taming Heterogeneity and Complexity of Embedded Control*; Wiley-ISTE: London, UK, 2007.
43. Chaillet, A.; Loria, A. Necessary and sufficient conditions for uniform semiglobal practical asymptotic stability: Application to cascaded systems. *Automatica* **2006**, *42*, 1899–1906. [[CrossRef](#)]
44. Espinosa, F.; Salazar, M.; Pizarro, D.; Valdes, F. Electronics Proposal for Telerobotics Operation of P3DX Units. In *Remote and Telerobotics*; Mollet, N., Ed.; InTech: London, UK, 2010; Chapter 1.
45. Wang, J.; Olson, E. AprilTag 2: Efficient and robust fiducial detection. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016.

