INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 4, Month: August, Year: 2026
Article Number: 7527, https://doi.org/10.15837/ijccc.2026.4.7527

Multinode Real-Time Scheduling for Stable Networked Control of a Quanser 3DOF Hover Platform

Héctor Benítez-Pérez, Ricardo Federico Villarreal-Martínez, Rita Carolina Rodríguez-Martínez, Adrián Durán-Chavesti, Nora Isabel Pérez-Quezadas, José Alberto Aparicio-Santos

Héctor Benítez-Pérez
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) & Dirección General de Cómputo y Tecnologías de la Información y las Comunicaciones (DGTIC)
at UNAM, Mexico
hector.benitez@iimas.unam.mx

Ricardo Federico Villarreal-Martínez
Institute of Research in Applied Mathematics and Systems (IIMAS)
at UNAM, Mexico
ricardo.villarreal@iimas.unam.mx

Rita Carolina Rodríguez-Martínez
Institute of Research in Applied Mathematics and Systems (IIMAS)
at UNAM, Mexico
rita.rodriguez@iimas.unam.mx

Adrián Durán-Chavesti
Institute of Research in Applied Mathematics and Systems (IIMAS)
at UNAM, Mexico
adrian.chavesti@iimas.unam.mx

Nora Isabel Pérez-Quezadas
Institute of Research in Applied Mathematics and Systems (IIMAS)
at UNAM, Mexico
nora.perez@iimas.unam.mx

José Alberto Aparicio-Santos
Dirección General de Cómputo y Tecnologías de la Información y las Comunicaciones (DGTIC)
at UNAM, Mexico
alberto.aparicio@unam.mx

Abstract

This work presents an experimental evaluation of distributed real-time control architectures applied to a Quanser 3DOF Hover platform. Multinode configurations comprising 3 and 9 computing nodes communicating via TCP/IP are reviewed, with control models executed in MATLAB/Simulink within an interconnected data environment. For each configuration, sensing, control, and actuation modules were deployed on distinct nodes to assess the influence of communication-induced delay and jitter on closed-loop stability. The main contribution consists of the modelling of time-delay models based on a scheduling-oriented approximation, together with the synthesis of a control law that explicitly incorporates the boundary conditions imposed by the scheduling policy. The study introduces the integration of a multinode scheduler, implemented as an additional node, which temporally orchestrates the critical events of the control loop. The scheduler is designed according to its dynamic configuration, with the specific objective of coupling the control law to the delays introduced by the scheduling mechanism. The manuscript provides both a conceptual analysis and a detailed design of the scheduler and its associated control law. Experimental results demonstrate that the platform remains stable with up to nine nodes when the scheduling mechanism is employed, whereas, in architectures lacking explicit temporal coordination, oscillatory behaviour associated with jitter and desynchronisation is observed.

Keywords: Networked control systems, Distributed real-time control, Multinode scheduling, Time-delay systems, Jitter compensation

1 Introduction

In Networked Control Systems (NCS), sensors, controllers, and actuators communicate via a digital communication network, as reported by [23], [31], among others. This architectural paradigm, however, introduces several challenges, including network-induced phenomena such as communication delays, variations in sampling instants, and constraints arising from the scheduling of packet transmissions. These effects directly influence the closed-loop performance of the system and may, in extreme cases, jeopardise its stability.

In [3], a reconfigurable networked control architecture based on automata is proposed, in which the integration of three modelling techniques for the reconfiguration process is illustrated: the design of the control law, automata-based modelling, and the real-time scheduling strategy. This approach has been reviewed and extended in several subsequent works, such as [8, 9], as well as in more recent contributions [10]. In this framework, a real-time scheduling algorithm is used to approve or reject modifications to the behaviour of the data network, with the aim of bounding communication delays within a specific time window.

The study of failures and their consequences has become a crucial topic in high-criticality networked systems. In [5], the design of a Takagi–Sugeno fuzzy logic structure is proposed to address two challenges: local faults and the corresponding time delays in a real-time distributed system. This approach is presented as a reconfigurable strategy based on communication delays in a real-time distributed system. In [16], a scheduling strategy is presented that enables efficient management of network resources, especially when bandwidth is limited; the techniques used prevent delays during transmission.

Temporal delays can be modelled using real-time dynamic scheduling algorithms; however, the resulting delays are time-varying and stationary. Therefore, in [6] a fuzzy networked control system is presented that considers scheduling constraints. In that work, it is shown that using the dynamic scheduling approximation makes the system predictable and bounded, and thus, the temporal delays can be modelled.

Networked control systems tend to increase temporal complexity through delays, making it necessary to study their behaviour. In [7], dynamic priority scheduling is used to limit delays; it addresses multiple delays and failures that produce a variable structure in the model and in the dynamic response, based on a real-time scheduling approach. In [9], it is shown that the system, with a time delay shorter than the sampling period but exhibiting complex behaviour, remains stable based on classical foundations, as modelled in [31], [27], [28], and [29]. Time delays can be modelled by means of a bounded-frequency transmission control approach; however, the resulting delays are time-varying and stationary. Therefore, a local control law related to this characteristic must be designed, in which temporal integration is the key global aspect to be considered. In [8], it is shown that fuzzy control can be an attractive option to guarantee global stability, since any condition is bounded to be less than the worst-case sampling period without loss of generality.

Variable delay can degrade the stability of a control system, even when the controller is stable in a centralised implementation. In this context, the Quanser 3DOF Hover platform provides an ideal experimental testbed [25] for evaluating distributed architectures under real communication conditions [24]. This work evaluates increasingly complex multi-node architectures and proposes a centralised scheduling strategy to mitigate the adverse effects of fluctuation.

The main contribution here is that, unlike previous approximations, where delays are clearly defined values due to the current behaviour of Real-Time Scheduling under Priority Exchange plus Rate Monotonic, the scheduling algorithm produces a bounded probability that incorporates a bounded delay. These values are included in the formal definition of the control law. Additionally, the study introduces the integration of a multinode scheduler, implemented as an additional node, which temporally orchestrates the critical events of the control loop. The scheduler is designed according to its dynamic configuration, with the specific objective of coupling the control law to the delays intro-

duced by the scheduling mechanism. As a result, the proposed scheme unifies scheduling and control, reducing temporal uncertainty and improving the stability of NCS architectures.

2 Theoretical framework

2.1 Networked Control System

In a Networked Control System (NCS), sensor and actuator signals are transmitted over a communication network, which inherently introduces delays and time-varying characteristics. These phenomena can significantly affect the stability and overall performance of the resulting closed-loop system, as rigorously analysed in [18] and [33]. It is important to emphasise that the issue is not limited to the presence of delays in control-law design, but also to the specific manner in which information exchange is managed over Ethernet-based networks—for instance, through the deployment of traffic-blocking mechanisms that preserve the temporal order of transmissions. This strategy does not necessarily fall within the scope of Co-design [4], as it is predicated on the integration of established knowledge from both control and communication disciplines, rather than on their joint and systematic co-optimisation.

2.2 Delay and jitter

The total delay consists of the sampling, transmission, processing, and actuation intervals. While a constant delay can be partially compensated, jitter—defined as the temporal variability of this delay—typically induces oscillatory behaviour and degrades closed-loop performance [23, 31]. Consequently, it is essential to characterise this variability without compromising the determinism and reliability of the processing. Alternative strategies, such as those presented by Hermosillo et al. [17], may constrain jitter by treating it as an inherent property of communication frames, thereby making it a factor of interest. Nevertheless, synchronisation remains a critical component. In contrast, the scheduling approaches considered here do not perform synchronisation; instead, they focus solely on ordering communication sequences.

For the problem under consideration, it is crucial to formalise the coupling introduced by the communication network and the resulting modifications to the dynamics of the controlled plant, namely the 3DOF Hover. The delay variability is modelled as arising from stochastic fluctuations in the communication latencies. It affects the system model independently and is regarded as an inherent consequence of the scheduling and resource-allocation strategy adopted for the implementation with 9 networked nodes.

2.3 Need for Scheduling approximation

In real-time distributed systems, scheduling and local synchronisation of events are fundamental principles for ensuring temporal coherence and preventing desynchronisation among nodes [21]. Likewise, the aim is to determine an ordering for the processing and communication of tasks under key principles such as schedulability [13], as well as for the ordering of sampled communication and event-triggered responses. Strategies such as budget-based servers have been explored in previous publications, yielding satisfactory results [8], as well as schedulers based on ordering via continuous divisions of budget into submultiples of the minimum possible processing capacity [17, 26].

In this context, it is necessary to organise the processing and transmission of information, for which various strategies have been explored [14] for sampling information from sensing elements. With this in mind, the use of the *Priority Exchange* [9] [8] is proposed as the main scheduling algorithm for managing the sampling of sensor information, with the remaining nodes thus becoming dependent on the event that summons them, based on communication between sensors, controller, and actuators.

As its name indicates, the scheduler treats the budget task as a periodic task, swapping its priority with that of the current periodic task at the instant [10, 13] when an aperiodic request appears, following the diagram in Figure 1. In this Figure, two periods are presented (T_k, T_{k-1}), during the second period an aperiodic task appears A_i with **a constant time delay equal to** Δt, therefore,

Sensor task 3 S_3 is reordered. Time consumption from this task is presented as τ_{s3}. Local time delay Δt is the scheduled task given by the Priority Exchange algorithm.

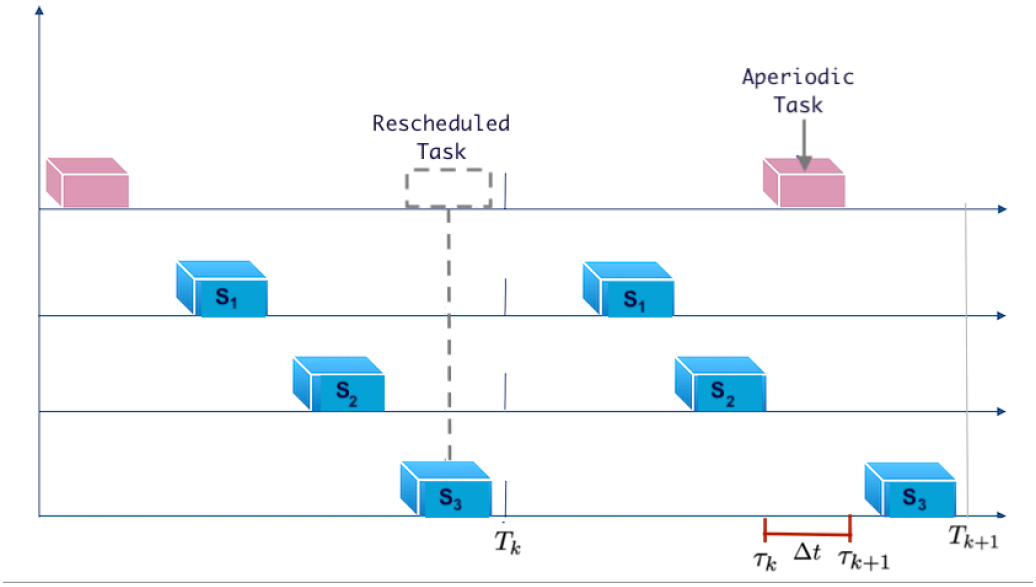


Figure 1: Timeline Management according to Priority Exchange

In this case, the delay Δt results from reprioritisation, which leads to a reorganisation of task performance. This Δt has a maximum equal to eqn 2

$$\Delta t = \{\tau_{k+1} - \tau_k\}$$
$$\text{Total Delay } \leq \{\tau_{k+1} - \tau_k\}\zeta_k \qquad \Delta t \ll \{T_k, (T_{k+1} - \zeta_k \Delta t)\} \tag{1}$$

In our case, only tasks from a segment equal to that of the sensor currently in turn, τ_{s3} (the consumption time of sensor 3), can be handled. Nevertheless, every time a delay exists, there is only one per period. Here, Δt is the maximum allowed delay.

Therefore, it will have a variation equal to ζ_k, $0 \leq \zeta \leq 1$ which is the maximum probability of current time delay that it is not exceeding a maximum time limit Δt within the sample, and as many times as indicated by the time delay (Total Delay), given the task to be performed, following the assumption proposed by [14], which is a non greater delay than the sample. Since the loss condition will have an effect only within the sampling interval, as stated in the assumption of [14] itself.

Taking into account the previous design of the *Priority Exchange* algorithm based upon *Rate Monotonic* has been presented by (Benítez et al. [7] and [9]) with the minimum modification that every delay is bounded to $\Delta t(\zeta_k)$ and given by a probability ζ_k. In that respect, the following table (Table 1) shows the consumption tasks, as well as the periodic Time and the aperiodic time reserved for budget utilisation. It is important to be notice that the only allowed task per period is TA_i with a maximum in the consumption time equal to *1ms* with the expression of $\Delta t(\zeta_k)$

Table 1: Time Table for Priority Exchange algorithm

Task Name	**Consumption Time**	**Periodic Time**
Sensor 1 t^{s_1}	1.0 ms	8 ms
Sensor 2 t^{s_2}	1.0 ms	9 ms
Sensor 3 t^{s_3}	1.0 ms	9.5 ms
Aperiodic Time $\Delta t(\zeta_k)$ given by TA_i	1.0 ms	8.5 ms

The following pseudocode outlines the task-specific decision-making for each sensor (see Table 2). Any aperiodic situation that arises at a given time for budget utilisation is addressed in $\Delta t(\zeta_k)$. If more time is needed, it is addressed at the next $\Delta t(\zeta_{k+1})$, which concerns the following period T_{k+1}.

Therefore, total consumption time, regardless of aperiodic tasks, appears to be equal to K

$$K = \left[\sum_{i=1}^{3}(t^{s_i})\right] + \Delta t(\zeta_k) \tag{2}$$

and

$$K \ll T_k \tag{3}$$

Table 2: Pseudocode from Priority Exchange plus Rate Monotonic in each sensor t^{s*}

Batch ordering of periodic Tasks (Sensors, Controller, Actuators)	
determination of Max $\Delta t(\zeta_k)$	
Online execution	
Main Loop	
if	aperiodic task TA_i appears given by the probability of ζ_k
	t^{s*} is on hold
	$\Delta t(\zeta_k)$ is executed with respect to TA_i until maximum bounded delay
else	t^{s*} is executed according to time left
end	
the rest of tasks are executed	this is happening within T_k
	thereafter T_{k+1} repeats to Main Loop

3 Methodology

3.1 Experimental platform

The Quanser 3DOF Hover model includes appropriate sensors and actuators for real-time control experimentation, and its integration with QUARC enables distributed models to run on multiple computers [24].

It starts with a 3DOF Hover reference-frame layout and the initial model construction as presented by several authors such as [1], [2], [11], [12]. The four rotors are arranged so that they can be identified as the Front Rotor, Right Rotor, Rear Rotor, and Left Rotor; for reference, they can be numbered 1, 2, 3, and 4, respectively. For stability reasons, the pair of propellers 1 and 3 rotate counterclockwise, while the remaining pair 2 and 4 rotate clockwise. Control of the aerial vehicle's motion in different directions and its rotation is achieved by adjusting the speed of each rotor. These speed changes can be applied individually, in pairs, or in any combination. Although the system has 6 degrees of freedom, it has only 4 controllable inputs (an underactuated system). The four motions that can be controlled with these inputs are: Thrust, Roll, Pitch, and Yaw, as we will see later. The following description assumes that the 3DOF Hover behaves as a formal Cuadracopter, which is not entirely certain, since the Inertial frame is not modified. Nevertheless, it is considered quite important to present the entire model to design a proper representation of this subactuated system.

In analysing the operation of the aerial vehicle, **two reference frames** are required to describe both its absolute position and its orientation and relative motions. These two frames are:

a) Inertial reference frame denoted here by I (world or Earth frame)

- It is used to describe the aircraft's **absolute position** in space.
- It is generally a Cartesian coordinate system (X, Y, Z), where Z is the height above the ground.
- It is a fixed system that does not rotate with the aerial vehicle.

b) Reference system of the body denoted here as C (air vehicle or local frame)

- It is used to describe the **orientation and movements** of the aerial vehicle.
- This system is fixed to the aerial vehicle and its origin is at the geometric centre of the mass or body and rotates with it.
- Its axes (x_c, y_c, z_c) are aligned with the structure of the aerial vehicle: x_c: Frontal direction, y_c: Lateral address, z_c: Vertical downward direction, aligned with gravity in steady state.

Both systems are needed to separate position and orientation

To mathematically describe the motion of the quadracopter, it is necessary to define two reference frames. The first is the inertial, or ground-fixed, one, and it is denoted by OI = $\{X_I, Y_I, Z_I, \phi, \theta, \psi\}$. The second is located on the aerial vehicle and is linked to the body's centre of mass; it is denoted as OC = $\{X_C, Y_C, Z_C, \text{p}, \text{q}, \text{r}\}$. These frameworks are illustrated in Figure 2.

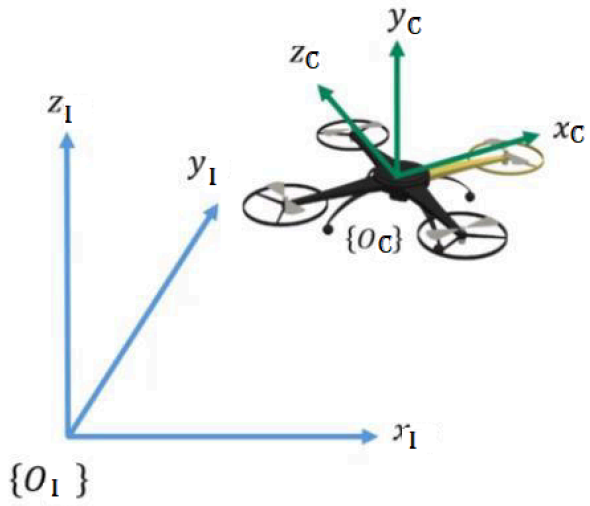


Figure 2: Frames of reference

The model used assumes the following hypotheses: The structure is rigid, the structure is symmetrical, the propellers are rigid, thrust and drag are directly proportional to the square of the propeller speeds.

In classical mechanics (Euler–Lagrange), dynamic models use generalised coordinates, where q is the state vector of the system:

$$q = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}^T$$

Under these considerations, the motion of the quadcopter can be described in terms of its position and orientation coordinates, which characterise the system:

$$q = \begin{bmatrix} \varepsilon \\ \eta \end{bmatrix}, \quad where \quad \varepsilon = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{4}$$

where (x, y, z) represents the linear positions (in this document it may appear as the vector ε) and (ϕ, θ, ψ) represents the orientation angles (Euler angles) respect to OC, which correspond to the standard 6-degree-of-freedom system model used in the literature for unmanned aerial vehicles [2, 12].

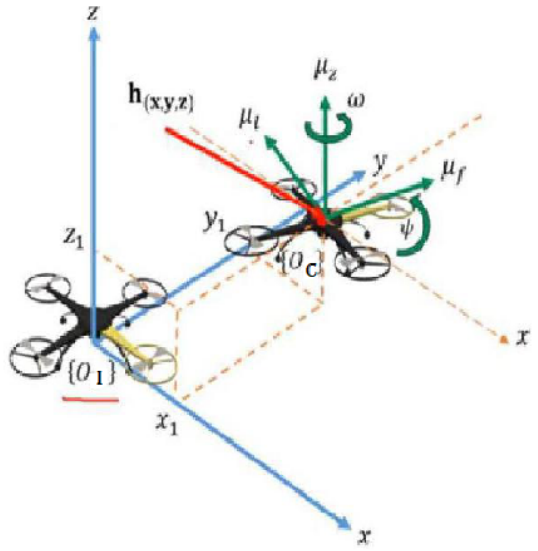
The 3DOF Hover is an underactuated system because it has six degrees of freedom but only four inputs to control its motion, such that the inputs U_1, U_2, U_3, U_4 are used in the system equations as local momentum per rotor. On the other hand, a full six-degree-of-freedom model uses the state vector and its related derivative $\dot{q}$

$$q = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}, \qquad \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5}$$

Taking the above into account, the quadcopter is analysed as a rigid body of mass m, and the aerodynamic forces due to the motion of the rotors are considered.

Figure 3 uses a diagram to describe the elements of the kinematics and dynamics of the quadracopter, showing how its position $h(x, y, z)$ and its orientation $\eta(\phi, \theta, \psi)$ are related to the rotor forces (μ_l lateral, μ_f frontal, μ_z vertical) and to the transformations between the inertial reference frame and the body frame. The reference frames are indicated as:
OI, (x_I, y_I, z_I), known as the inertial or global reference frame, has its axes in blue and describes the absolute position of the 3DOF Hover in space.
OC, (x_c, y_c, z_c) is the body reference frame, fixed at the centre of mass of the 3DOF Hover. It moves and rotates together with it. Its axes are shown in brown.
The relationship between the two reference frames is expressed by rotation matrices. The motion variables and the position vector of the 3DOF Hover are represented in the inertial space, where the body is located with respect to the origin OI, and are drawn with the red vector. In Figure 3, ψ (yaw) appears (rotation about the z-axis), and Ω is the vector of angular velocities of the 3DOF Hover in the body frame. The lateral force μ_l is generated when the 3DOF Hover tilts its roll (ϕ), producing sideways motion. The frontal force (μ_f) is produced when the 3DOF Hover tilts its pitch (θ), resulting in forward or backward motion. The lift force (μ_z) is the vertical component of the thrust. Each rotor generates an upward force (along the body's z-axis). The sum of the four components yields the net force μ_z, which counteracts the weight and enables the vehicle to climb, descend, or hover. These forces are shown with green arrows (Figure 3).

The components of the vectors μ_f and μ_l, defined in the body frame, are projected onto the axes of the inertial frame according to the yaw angle ψ. Thus, the projections of these forces determine the components of the motion along the x and y axes, which are reflected in $\dot{h}_x$ and $\dot{h}_y$. The decomposition of μ_l can initially be expressed in terms of an angle β, defined with respect to its reference axis; however, due to the geometric relationship of the system, it holds that $\beta = \psi$. Consequently, the equations, derived from the figure, are expressed solely as functions of ψ.



$$h_x = x_1 \qquad \dot{h}_x = \mu_f \cos(\psi) - \mu_l \sin(\psi) \tag{6}$$

$$h_y = y_1 \qquad \dot{h}_y = \mu_f \sin(\psi) + \mu_l \cos(\psi) \tag{7}$$

$$h_z = z_1 + a \qquad \dot{h}_z = \mu_z \tag{8}$$

$$\dot{\psi} = \omega \tag{9}$$

Figure 3: Kinematic elements of the 3DOF Hover and projection of velocities from the body frame to the inertial frame.

The dynamic behaviour of the 3DOF Hover can be described using a state-space repre-

sentation. The state vector is defined as

$$X = \begin{bmatrix} h_x & h_y & h_z & \phi & \theta & \psi & \dot{h}_x & \dot{h}_y & \dot{h}_z & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T, \qquad U = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix}^T, \tag{10}$$

which makes it possible to model the system as a set of first-order differential equations [2, 11].
As seen at the beginning of the model, h_x, h_y, h_z represent the position in the inertial reference frame, ϕ, θ, ψ correspond to the Euler angles (roll, pitch, and yaw), and their derivatives represent the linear and angular velocities, respectively.
Defining the state variables as $x_1, \ldots, x_{12}$, the system can be expressed in nonlinear form as

$$\dot{X} = f(X, U). \tag{11}$$

The first equations describe direct kinematic relationships

$$\dot{x}_i = x_{i+6}, \qquad i = 1, \ldots, 6, \tag{12}$$

while the translational dynamics is given by

$$\dot{x}_7 = \frac{U_1}{m} \left(\cos x_6 \sin x_5 \cos x_4 + \sin x_6 \sin x_4\right), \tag{13}$$

$$\dot{x}_8 = \frac{U_1}{m} \left(\sin x_6 \sin x_5 \cos x_4 - \cos x_6 \sin x_4\right), \tag{14}$$

$$\dot{x}_9 = -g + \frac{U_1}{m} \left(\cos x_5 \cos x_4\right). \tag{15}$$

On the other hand, the rotational dynamics can be obtained from the Lagrangian formulation of the system,
Since the angular states were defined as $x_{10} = \dot{\phi}$, $x_{11} = \dot{\theta}$ and $x_{12} = \dot{\psi}$, their derivatives correspond to the angular accelerations: $\dot{x}_{10} = \ddot{\phi}$, $\dot{x}_{11} = \ddot{\theta}$ and $\dot{x}_{12} = \ddot{\psi}$. From the simplified rotational dynamics of the rigid body, and assuming decoupling between axes, it follows that the control torques satisfy $I_x\ddot{\phi} = U_2$, $I_y\ddot{\theta} = U_3$ and $I_z\ddot{\psi} = U_4$. Therefore, the last three equations of state are obtained as:

$$\dot{x}_{10} = \frac{U_2}{I_x}, \qquad \dot{x}_{11} = \frac{U_3}{I_y}, \qquad \dot{x}_{12} = \frac{U_4}{I_z}. \tag{16}$$

The inputs U_1, U_2, U_3, U_4 do not correspond directly to physical actuators; instead, they represent combinations of the forces and torques generated by the actuators.
For the purposes of Analysis and control design, "nonlinear systems are linearised around an equilibrium point" [20] corresponding to steady flight.

$$\phi_0 = 0, \qquad \theta_0 = 0, \qquad \psi_0 = 0, \tag{17}$$

$$\dot{h}_{x0} = 0, \qquad \dot{h}_{y0} = 0, \qquad \dot{h}_{z0} = 0, \tag{18}$$

$$\dot{\phi}_0 = 0, \qquad \dot{\theta}_0 = 0, \qquad \dot{\psi}_0 = 0, \tag{19}$$

$$U_{10} = mg. \tag{20}$$

Under the small-angle assumption, the approximation is obtained

$$\ddot{h}_x \approx g\theta, \qquad \ddot{h}_y \approx -g\phi, \qquad \ddot{h}_z \approx \frac{1}{m}\delta U_1, \tag{21}$$

and

$$\ddot{\phi} = \frac{U_2}{I_x}, \qquad \ddot{\theta} = \frac{U_3}{I_y}, \qquad \ddot{\psi} = \frac{U_4}{I_z}. \tag{22}$$

The resulting linear model can be written as

$$\dot{X} = AX + BU, \tag{23}$$

where A describes the internal dynamics of the system and B the influence of the control inputs. It is important to note that, in this linear representation, the matrix A does not explicitly depend on the inputs U_i, since these are incorporated through the matrix B. In contrast, the nonlinear model (11) allows the dynamics to be expressed directly as a function of U_i or, equivalently, as a function of the rotor speeds Ω_i.

Jacobian matrix of the non-linear model

Although the classical linear representation is expressed as

$$\dot{X} = AX + BU, \tag{24}$$

When it is desired to preserve the explicit dependence of the dynamics on the states and inputs, it is more appropriate to use the local linearisation of the nonlinear model around an operating point (X_0, U_0), given by

$$\delta\dot{X} = A(X_0, U_0)\,\delta X + B(X_0, U_0)\,\delta U, \tag{25}$$

where

$$A(X_0, U_0) = \left.\frac{\partial f}{\partial X}\right|_{(X_0,U_0)}, \qquad B(X_0, U_0) = \left.\frac{\partial f}{\partial U}\right|_{(X_0,U_0)}. \tag{26}$$

This expression is obtained by means of the first-order Taylor series expansion of the nonlinear system [20, 27].

For the state vector

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^T, \tag{27}$$

where $x_4 = \phi$, $x_5 = \theta$, $x_6 = \psi$, the Jacobian matrix with respect to the state is

$$A(X,U) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{U_1}{m}(-c_6 s_5 s_4 + s_6 c_4) & \frac{U_1}{m}(c_6 c_5 c_4) & \frac{U_1}{m}(-s_6 s_5 c_4 + c_6 s_4) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{U_1}{m}(-s_6 s_5 s_4 - c_6 c_4) & \frac{U_1}{m}(s_6 c_5 c_4) & \frac{U_1}{m}(c_6 s_5 c_4 + s_6 s_4) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{U_1}{m} c_5 s_4 & -\frac{U_1}{m} s_5 c_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{28}$$

$$c_i = \cos(x_i), \qquad s_i = \sin(x_i).$$

Alternatively, the Jacobian matrix with respect to the inputs is given by

$$B(X) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m}\left(\cos x_6 \sin x_5 \cos x_4 + \sin x_6 \sin x_4\right) & 0 & 0 & 0 \\ \frac{1}{m}\left(\sin x_6 \sin x_5 \cos x_4 - \cos x_6 \sin x_4\right) & 0 & 0 & 0 \\ \frac{1}{m}\cos x_5 \cos x_4 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix}, \tag{29}$$

Consequently, the locally linearised representation of the system is expressed as

$$\delta\dot{X} = A(X_0, U_0)\,\delta X + B(X_0)\,\delta U, \tag{30}$$

which explicitly preserves the dependence of the dynamics on the Euler angles, the lift input U_1, and the moments of inertia I_x, I_y, and I_z Regarding the model in its discrete form, we must consider the following representation, which allows us to express the system both in terms of the delay ζ_k and its corresponding representation by a fuzzy system.

As presented at the beginning of this section, the 3DOF Hover has been modelled as a formal Cuadracopter, which is not entirely certain, since the Inertial frame is not modified. Nevertheless, it is considered quite important to present the entire model to design a proper representation of this subactuated system. Therefore, the eqn 27 is reduced to six states as follows:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T, \tag{31}$$

Related to the angles and its derivatives.

$$X = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T, \tag{32}$$

If we adopt a piecewise modelling approach, analysing each scenario in terms of the time delay induced by the adaptation according to the scheduling model proposed in the previous section, we can proceed by considering a TKS-type fuzzy system that remains consistent with all admissible delay configurations resulting from the scheduler. This framework also incorporates the probability associated with the occurrence of external conditions that may arise in this context, as discussed in [8, 9, 14].

The particular linear equation that is considered for the first time delay

$$\bar{A} = e^{AT} \qquad \bar{A}_1 = \int_0^{\Delta t(\zeta_k)} e^{A(T-s)}ds \qquad \bar{A}_2 = \int_{\Delta t(\zeta_k)}^{T} e^{A(T-s)}ds \tag{33}$$

where $\Delta t(\zeta_k)$ is given by ζ_k defined in Section 2.

Now, at the construction of the fuzzy equation using several time delays $\Delta t_i(k)$, due to several i_{esm} possible discrete plant representations based upon the related probability ζ_k^i, therefore $\Delta t(\zeta_k^i)$ are given as:

$$\bar{A}_1^i = \int_0^{\Delta t(\zeta_k^i)} e^{A(T-s)}ds \qquad \bar{A}_2^i = \int_{\Delta t(\zeta_k^i)}^{T} e^{A(T-s)}ds \tag{34}$$

$$\bar{X}(k+1) = \sum_{j=1}^{N} R_j((\bar{A}_1^j + \bar{A}_2^j)\bar{X}(k) + B_j u(k)) \tag{35}$$

with

$$0 < R_j < 1 \qquad ; \qquad \sum_{j=1}^{N} R_j(x) = 1 \tag{36}$$

$$R_j = \frac{h_j}{\sum_{i=1}^{N} h_i} \qquad ; \qquad h_i(V(k)) = \pi_{j=1}^{L} V_j^i(k) \qquad ; \qquad V_j^i = e^{-}(\frac{x_j(k) - \alpha_{ij}}{\beta_{ij}})^2 \tag{37}$$

where ζ_k^i is the result of a probability for each combination of the given delay and the given outcome produced by the scheduler (Section 2) for each event, taking into account that they are not cumulative but only reactive to the event. Furthermore, N and L represent the number of total Fuzzy Rules and the total number of Membership Functions, respectively.

In this regard, each equation represents the plant and accounts for a time delay corresponding to a particular combination, determined by a selection process via the *Priority Exchange plus Rate Monotonic* where the implicit variation is given by ζ_k^i.

Now, if we consider the set of delays

$$K = \sum_{i=1}^{N} \sum_{j=1 i \neq j}^{N} \zeta_k^i \Delta t_i(k) \tag{38}$$

$$0 \leq K \leq \Delta t_{max} * N \ll T_k$$

However, the delay $\Delta t_i(k)$ cannot be cumulative given its random nature (ζ_k^i), even if it is bounded by a local delay shorter than the period, with a maximum delay. (Figure 1)

$$\Delta t_{max} = [\tau_{k+1} - \tau_k] - \sum_{i=1}^{3} \tau_{s_i} \tag{39}$$

Based on the proposed scheduling model, one can formulate a probabilistic framework that characterises delays as bounded and non-accumulative. Now, the current delay $\Delta t(\zeta_k^i)$ from a task TA_i at T_k sampling period given by ζ_k^i probability the related Fuzzy Rule R_i that is selected by the modification of state $\bar{X}$ according to the Discretisation presented in 35

3.2 Multinode configurations

Two configurations are evaluated, summarised in Table 3. It is worth noting that each configuration poses clear challenges for sampling, control-law design, and scheduler handling of transmission. The first configuration is formulated by modifying the control law to account for communication-induced time delays; in contrast, the last configuration adopts a new scheduling-based approach to handling delays.

Table 3: Evaluated multi-node configurations

Architecture	Sensors	Control	Actuators	Scheduler
3 nodes	1	1	1	No
9 nodes	3	1	4	Yes

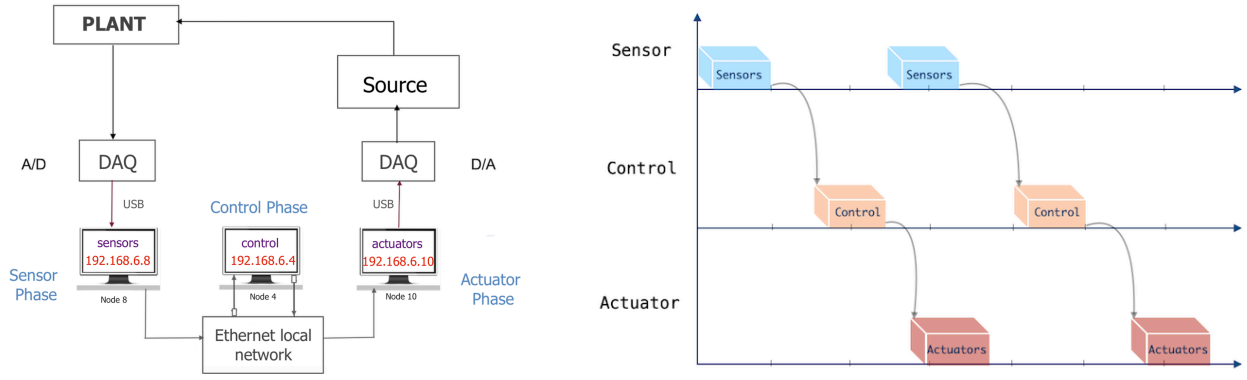
3.2.1 Architecture: 3 nodes

The objective of the three-node architecture is to segregate, at the node level, the functionalities associated with the Sensors, the Controller, and the Actuators. These nodes, which are distributed across the network, implement a synchronization mechanism based on message passing. This mechanism constitutes the means by which the components of the system presented in this work are coordinated. Each node is assigned an IP address and the TCP protocol to study the quadcopter's behaviour, observe its response to this protocol, and assess its stability under noise or turbulence,

seeking stability across multiple events. The model is partitioned as follows: node 8 serves as the sensor, node 4 as the controller, and node 10 as the actuator. (Figure 4).

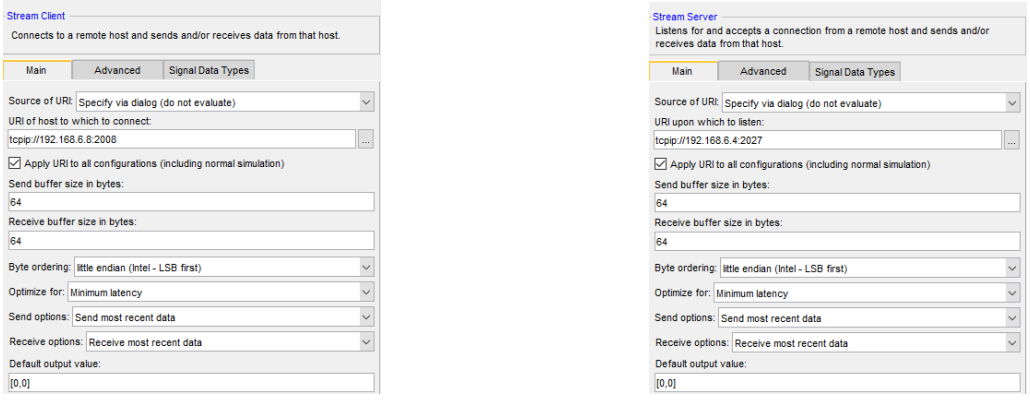
The nodes (PCs) have the following hardware configuration: an Intel Core i5-3470 processor at 3.2 GHz, 16 GB of RAM, a Gigabit Ethernet network interface, and a 1 TB solid-state drive (SSD). Regarding software, the nodes run the Windows 10 Pro operating system, and are equipped with the C++ compiler provided by Microsoft Visual Studio Enterprise 2019, as well as the MATLAB R2019b programming environment and the QUARC 2019 SP1 real-time control package.

The 3DHover, together with the nodes used in this architecture, is connected to an Ethernet LAN (192.168.6.x) that transmits data at 1 Gb/s. Each node is assigned an IP address that matches the node number to which the sensors, controller, and actuators are connected. In Figure 4, it can be seen that the Plant (Quanser 3 DOF Hover or Quadracopter) and the four-channel amplifier are connected directly to the data acquisition board (DAQ), which, in turn, is connected to node 192.168.6.8 (sensors) via a USB cable and sends analogue-digital signals to that node. In node 8, the sensor coupling is modelled, as is the Stream Client block shown in Figure 5, in which parameters such as the controller's IP address, port number, TCP protocol, and packet sizes for sending and receiving are configured. In node 4, corresponding to the controller, its model includes communication with both the sensor and the actuator. The controller receives the sensed data via the Stream Server block as shown in Figure 5, which is configured to use this node's IP address. For the controller design, we follow the scheme in [14]. Since the sensed data are discretised, they are manipulated to obtain the appropriate values, which are then processed as a time series under the control law. The control law adjusts the voltage parameters sent to the Actuator node via the Stream Client block, which is configured with node 10's IP address.



(a) Three nodes (b) Three nodes

Figure 4: Architecture for 3 nodes

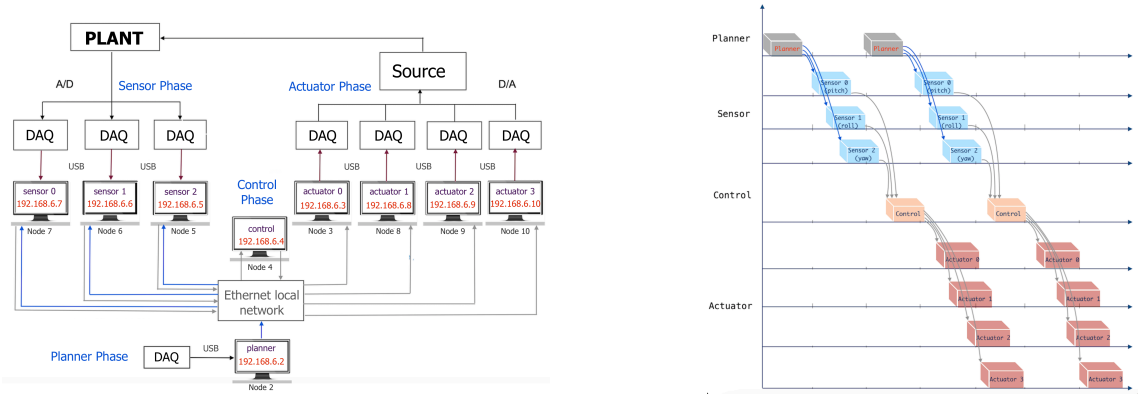


(a) Stream Client (b) Stream Server

Figure 5: Configuring network parameters for the Stream Client and Stream Server blocks.

3.2.2 Architecture: 9 nodes

Once all components of the Plant (3DOF Hover) are distributed across the nodes, the results are favourable, provided the system is modelled properly to account for **Priority Exchange**, the related Time Delay $\Delta t_i(k)$, and proper Control Law design. It is decided to use the eight-node configuration as the basis and add a final node representing the scheduler. As shown in Figure 6, the configuration is essentially the same. In this case, the model is divided as follows: node 3 as actuator 0, node 8 as actuator 1, node 9 as actuator 2, node 10 as actuator 3, node 7 as sensor 0, node 6 as sensor 1, node 5 as sensor 2, node 4 as controller. The scheduling node modifies the communication behaviour of the sensing nodes. In each model, a Stream Server block was added, and the sensor node's IP address and a port number were assigned, enabling communication over the Gigabit Ethernet network with node 2, which corresponds to the scheduler. In the scheduling model, three Stream Client blocks are added, each configured with the IP address assigned to that sensor and the same port number as the Stream Server for that sensor. In this architecture, the configuration sequence starts at node 2, which communicates with each sensor via TCP. The sensors receive the signal through the Stream Server block, forward it to the Stream Client in their models, and it is then received by node 4 (controller). The remaining steps are the same as those described in the eight-node architecture.



(a) Nine nodes (b) Nine nodes

Figure 6: Architecture for 9 nodes

Now, the controller design for this strategy uses a full-state representation, allowing us to observe the effects of scheduling-system time delays discussed in Section 2.

Given the fuzzy system providing the possible combinations of delays as a function of $\Delta t_i(\zeta_k^i)$ with respect to the scheduler eqns 35 and 37, so that

$$(if \quad V(\bar{X})) \quad then \quad \bar{X}(k+1) = A_j\bar{X}(k) + B_j u(k)) \tag{40}$$

Given the complete distribution of the plant–control system and considering that the variations in V correspond only to the process at k, implying $\Delta t_i(\zeta_k^i)$, it follows that

$$\frac{\sum_{i=1}^{N} h_i[(V(k))(A_i\bar{X}(k) + B_i u)]}{\sum_{i=1}^{N} h_i(V(k))} = \bar{X}(k+1) \tag{41}$$

having

$$h_i(V(k)) = \pi_{j=1}^{i}(V_j(k)) \tag{42}$$

Where the controller is:

$$u(k) = \frac{-\sum_{j=1}^{N}[h_j(V(k))(F_j\bar{X}(k))]}{\sum_{j=1}^{N}(V_j(k))]} \tag{43}$$

Therefore

$$\bar{X}(k+1) = \sum_{i=1}^{N}[h_i(V(k))(A_i\bar{X}(k) + B_i(-\sum_{j=1}^{N}[h_j(V(k))(F_j\bar{X}(k))]))] \tag{44}$$

following the delay system proposed in Section 3.1 with respect to eqn 35:

$$\bar{X}(k+1) = \sum_{i=1}^{N}[h_i(V(k))(\bar{A}_1^i + \bar{A}_2^i\bar{X}(k)) - B_i\sum_{j=1}^{N}[h_j(V(k))F_j\bar{X}(k)])] \tag{45}$$

$$\bar{X}(k+1) = \sum_{i=1}^{N} h_i(\mathrm{V}(k))\, h_i(\mathrm{V}(k))((\bar{A}_1^i + \bar{A}_2^i) - B_iF_i)\bar{X}(k) + 2\sum_{i=1}^{N}\sum_{j=i+1}^{N} h_i(\mathrm{V}(k))\, h_j(\mathrm{V}(k))\, G_{ij}\, \bar{X}(k) \tag{46}$$

having the configuration (Figure 7)

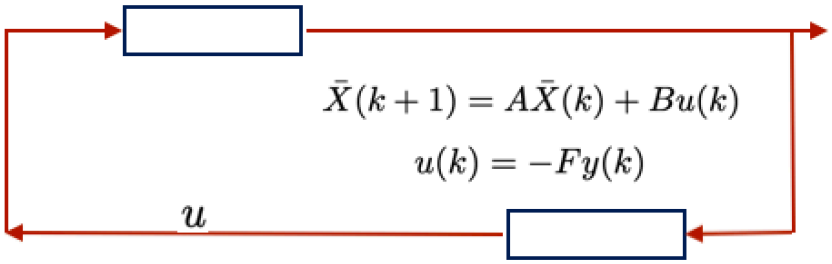


Figure 7: Basic NCS configuration

Given that

$$G_{ij} = \frac{((\bar{A}_1^i + \bar{A}_2^i) - B_iF_j) + ((\bar{A}_1^j + \bar{A}_2^j) - B_jF_i)}{2},\ \mathrm{i} > \mathrm{j} \tag{47}$$

Theorem 1: A stability process with a globally asymptotic equilibrium exists if there is a common P, a positive definite matrix, such that the following conditions are satisfied

$$\bar{V} = \bar{X}^T P\bar{X} \tag{48}$$

and obtaining the respective discrete derivative from $\bar{V}$

$$\Delta\bar{V} = \bar{V}(k+1) - \bar{V}(k) \tag{49}$$

By expanding this final equation and expressing it in a form decomposed from eqs. 47 through 48, in accordance with eq. 46, we obtain a representation that is primarily derived via Linear Matrix Inequality (LMI) decomposition, given that the condition on G_{ij} is completely satisfied.

$$\{(\bar{A}_1^i + \bar{A}_2^i) - B_iF_i\}^T P\{(\bar{A}_1^i + \bar{A}_2^i) - B_iF_i\} - \mathrm{P} < 0\ \mathrm{i} = 1, \ldots, \mathrm{N} \tag{50}$$

$$G_{ij}^T PG_{ij} - \mathrm{P} < 0\ \mathrm{i} < \mathrm{j} =< \mathrm{N} \tag{51}$$

Initially, G is a Hurwitz matrix such that

$$(\bar{A}_1^i + \bar{A}_2^i) - B_iF_i = G_{ii} \tag{52}$$

which leads us to view this as a commonly linear system. If it is considered the implementation using the LMI approaches (Tanaka [30]), we have, based on an initial definition:

$$F(x) = F_0 + \sum_{i=0}^{m} X_iF_i > 0 \tag{53}$$

Considering the system with controlled time variations, we have that

$$((\bar{A}_1^i + \bar{A}_2^i) - B_i F_i)^T P((\bar{A}_1^i + \bar{A}_2^i) - B_i F_i) - \mathrm{P} =< 0 \tag{54}$$

having then

$$G_{ij}^T P G_{ij} - \mathrm{P} < 0 \tag{55}$$

$$(\frac{G_{ij} + G_{ji}}{2})^T P(\frac{G_{ij} + G_{ji}}{2}) - \mathrm{P} =< 0 \tag{56}$$

thus,this is giving a proper solution to Theorem 1.

Nevertheless, it is possible to formulate a second theorem that accounts for two internal time delays without altering the overall system representation. This theorem enables accurate modelling of both (i) the case in which there are no time delays arising from aperiodic tasks and (ii) the case in which a time delay is introduced in accordance with specific conditions associated with an aperiodic task TA_i.

Consider the variable ζ_k^i, defined as the probability associated with the delay induced by the scheduler (see (34)) in an integrated system comprising multiple sensors and actuators. The delay arising from the occurrence of an aperiodic task (Table 1) can be modelled by means of a specific discrete-time transformation, such that:

$$\bar{A}_1^{iD} = \int_0^{\Delta t(\zeta_k^{i\,D})} e^{A(T-s)} ds \qquad \bar{A}_2^{iD} = \int_{\Delta t(\zeta_k^{i\,D})}^{T} e^{A(T-s)} ds \tag{57}$$

Let $\Delta t(\zeta_k^{i\,D})$ denote the delay associated with a probabilistic event occurring after the initial delay, acting on the same plant before completion of the first processing cycle represented by $\Delta t(\zeta_k^i)$. In this framework, two local and internal time delays, $\Delta t(\zeta_k^i)$ and $\Delta t(\zeta_k^{i\,D})$, are considered, both reflecting two possible temporal variations within the sampling interval T_k, neither of which violates the constraint specified in eqn. 1. Therefore:

$$\bar{X}(k+1) = \sum_{i=1}^{N} h_i(V(k))\{(\bar{A}_1^i + \bar{A}_2^i)\bar{X}(k) + B_i u(k) + (\bar{A}_1^i + \bar{A}_2^i)^D \bar{X}(k - \Delta t((\zeta_k^i)^D)) + B^D u(k - \Delta t((\zeta_k^i)^D))\} \tag{58}$$

such that

$$u(k) = -\sum_{j=1}^{N} h_j(V(k))(F_j \bar{X}(k) + F_j^D \bar{X}(k - \Delta t(\zeta_k^{i\,D}))) \tag{59}$$

obtaining then

$$\bar{X}(k+1) = \sum_{i=1}^{N} \Big(h_i(\mathrm{V}(k))\Big)^2 \Big[((\bar{A}_1^i + \bar{A}_2^i) - B_i F_i)\bar{X}(k) + ((\bar{A}_1^i + \bar{A}_2^i)^D - B_i^D F_i)\bar{X}(k - \Delta t((\zeta_k^i)^D))\Big] \tag{60}$$

$$+ 2\sum_{i=1}^{N} \sum_{j=i+1}^{N} h_i(\mathrm{V}(k))\, h_j(\mathrm{V}(k)) \Big[G_{ij}\bar{X}(k) + G_{ij}^D \bar{X}(k - \Delta t((\zeta_k^i)^D))\Big]. \tag{61}$$

Therefore, a second Theorem arise:

Theorem 2:

As in theorem 1, the following Lyapunov eqn is presented considering this proposal of both delays.

$$\bar{V} = \bar{X}^T P \bar{X} + \bar{X}^T R \bar{X} \tag{62}$$

where the respective derivative is given as

$$\Delta\bar{V} = \bar{V}(k+1) - \bar{V}(k) \tag{63}$$

Therefore the definition of a matrix P as well as the resulting R, both being global and positive definite such that

$$G_{ij} = \frac{((\bar{A}_1^i + \bar{A}_2^i) - B_i F_j) + ((\bar{A}_1^j + \bar{A}_2^j) - B_j F_i)}{2} \quad \text{i} \quad < \quad \text{j} \tag{64}$$

$$T_{ij} = \frac{(\bar{A}_1^i + \bar{A}_2^i)^D - B_i^D F_j^D) + ((\bar{A}_1^j + \bar{A}_2^j)^D - B_j^D F_i^D)}{2} \quad \text{i} \quad < \quad \text{j} \tag{65}$$

Taking the previous equations, now expressed as

$$G_{ij}^T P G_{ij} - P + T_{ij}^T R T_{ij} - \ \text{R} < 0 \tag{66}$$

By applying an analogous procedure to the linear matrix inequality (LMI) feasibility problem, it becomes possible to determine an appropriate solution for the following ultimate condition:

$$\frac{(G_{ij} + G_{ji})}{2} P \frac{(G_{ij} + G_{ji})^T}{2} - P + \frac{(T_{ij} + T_{ji})^T}{2} R \frac{(T_{ij} + T_{ji})}{2} - \ \text{R} < 0 \ / \ \text{i} < \text{j} \tag{67}$$

Where R and P are functions that are identical across all combinations and are determined by the variation of $\Delta t(\zeta_k^i)$. Consequently, if the unitary solution is feasible, **the associated stability is guaranteed.**

4 Experimental Results

Architecture without a Scheduler In the 3-node configuration, stable behaviour is observed with limited degradation. As slide desynchronisation among actuators increases, the effective jitter rises, leading to persistent oscillations, consistent with reports in the literature [23, 33].

Architecture with a scheduler By incorporating the multinode scheduler (9 nodes), temporal coherence among the sensors, the controller, and the actuators is restored. Reducing jitter yields a smoother, more stable system response.

Discussion The results indicate that the scalability of an NCS depends not only on the number of nodes but also on the temporal coordination among them. Centralised scheduling mitigates jitter without modifying the base controller, consistent with the design principles of real-time distributed systems [21].

4.1 Configuration for 3 nodes

For a 3-node configuration, a Linear Quadratic Integral (LQI) controller is designed that incorporates integral action to eliminate steady-state errors and improve the rejection of constant disturbances. Its design follows the same procedure used for the classical LQR controller, but applied to an

$$A_a = \begin{bmatrix} A_d & 0 \\ -C\,T_s & I \end{bmatrix}, \qquad B_a = \begin{bmatrix} B_d \\ 0 \end{bmatrix}. \tag{68}$$

Where A_d is the discretised state matrix, B_d is the discretised input matrix, C is the output matrix, and I is the 3×3 identity matrix.

Solving the discrete Riccati equation yields a gain matrix $K_m = [K_x \ K_I]$, where K_x is the gain associated with the physical states and K_I is the gain associated with the integral states, which are formed by integrating the orientation angles ϕ, θ, and ψ. The gains obtained were:

$$K_x = \begin{bmatrix} -307.9798 & 277.0141 & 0 & -74.4121 & 43.8171 & 0 \\ -307.9798 & -277.0141 & 0 & -74.4121 & -43.8171 & 0 \\ 307.9798 & 0 & 219.4617 & 74.4121 & 0 & 42.5294 \\ 307.9798 & 0 & -219.4617 & 74.4121 & 0 & -42.5294 \end{bmatrix}, \qquad K_I = \begin{bmatrix} 475.5451 & -565.6656 & 0 \\ 475.5451 & 565.6656 & 0 \\ -475.5451 & 0 & -254.6696 \\ -475.5451 & 0 & 254.6696 \end{bmatrix}. \tag{69}$$

The resulting control law is given by

$$u(k) = -K_x x(k) - K_I x_I(k), \tag{70}$$

where $x(k)$ represents the vector of physical states and $x_I(k)$ the vector of integral states. The stability of the augmented closed-loop system is verified using the eigenvalues of

$$A_{cl} = A_a - B_a K_m. \tag{71}$$

The eigenvalues obtained were

$$\lambda(A_{cl}) = \left\{ \begin{array}{c} 0.6878 \\ 0.6878 \\ 0.9671 + 0.0277i \\ 0.9671 - 0.0277i \\ 0.9663 \\ 0.9830 \\ 0.9592 \\ 0.9645 + 0.0165i \\ 0.9645 - 0.0165i \end{array} \right\}. \tag{72}$$

The spectral radius of the closed-loop system is given by

$$\rho(A_{cl}) = \max_i |\lambda_i| = 0.9830. \tag{73}$$

Dado que

$$\rho(A_{cl}) < 1, \tag{74}$$

All eigenvalues lie within the unit circle in the complex plane. Consequently, the matrix A_{cl} is a Schur matrix, and the augmented discrete-time system controlled via LQI is asymptotically stable.

In this regard, the loss due to the delay is so small that there is no degradation in the response of the sensors, Figure 8, and therefore the four actuators respond instantaneously to a motion with an oscillatory effect and a saturation characteristic of an oscillatory motion, Figure 9, without any loss of information. Nevertheless, slide Saturation is presented as a common response. It is important to highlight that perturbations from aperiodic tasks appear at the second 100 without losing track of the system's response.

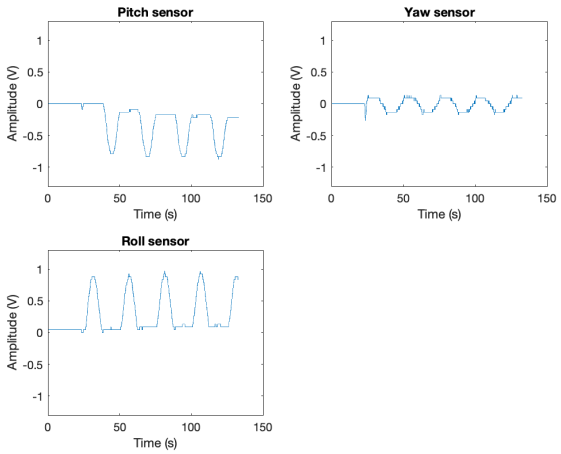


Figure 8: Signal from the sensors for 3 nodes.

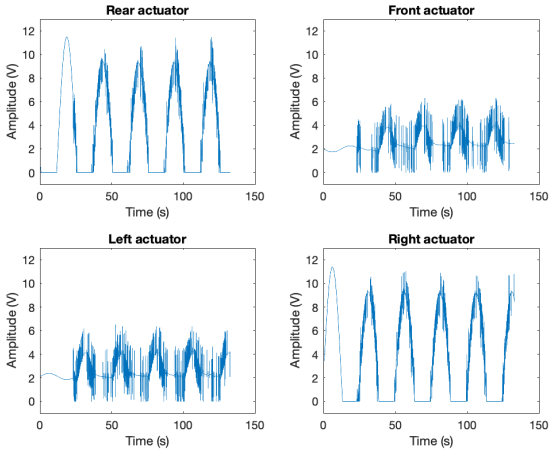


Figure 9: Response of the actuators for 3 nodes.

4.2 Configuration for 9 nodes

The Takagi–Sugeno model of the quadcopter is composed of four fuzzy rules associated with the vertices of the operating space defined by the roll ($-30° \leq \phi \leq 30°$) and pitch ($-30° \leq \theta \leq 30°$) angles. For each local subsystem the gain matrices F_i, $i = 1, \ldots, 4$ following LQR design, are presented below:

$$F_1 = \begin{bmatrix} 16.2649 & 38.2864 & -126.7914 & 1.3002 & 29.4059 & -45.8277 \\ -6.5882 & -131.6217 & -22.4033 & -0.3276 & -29.0807 & -42.6931 \\ 129.6799 & 39.3431 & 86.8428 & 18.3268 & 0.7912 & 38.6072 \\ -139.3566 & 53.9922 & 62.3519 & -19.2994 & -1.1165 & 49.9137 \end{bmatrix}, \tag{75}$$

$$F_2 = \begin{bmatrix} -16.2649 & 38.2864 & -126.7914 & -1.3002 & 29.4059 & -45.8277 \\ 6.5882 & -131.6217 & -22.4033 & 0.3276 & -29.0807 & -42.6931 \\ 139.3566 & 53.9922 & 62.3519 & 19.2994 & -1.1165 & 49.9137 \\ -129.6799 & 39.3431 & 86.8428 & -18.3268 & 0.7912 & 38.6072 \end{bmatrix}, \tag{76}$$

$$F_3 = \begin{bmatrix} -6.5882 & 131.6217 & -22.4033 & -0.3276 & 29.0807 & -42.6931 \\ 16.2649 & -38.2864 & -126.7914 & 1.3002 & -29.4059 & -45.8277 \\ 129.6799 & -39.3431 & 86.8428 & 18.3268 & -0.7912 & 38.6072 \\ -139.3566 & -53.9922 & 62.3519 & -19.2994 & 1.1165 & 49.9137 \end{bmatrix}, \tag{77}$$

$$F_4 = \begin{bmatrix} 6.5882 & 131.6217 & -22.4033 & 0.3276 & 29.0807 & -42.6931 \\ -16.2649 & -38.2864 & -126.7914 & -1.3002 & -29.4059 & -45.8277 \\ 139.3566 & -53.9922 & 62.3519 & 19.2994 & 1.1165 & 49.9137 \\ -129.6799 & -39.3431 & 86.8428 & -18.3268 & -0.7912 & 38.6072 \end{bmatrix}. \tag{78}$$

In here, scheduling approximation based on **Priority Exchange** plays a key role in Time Delays management due to aperiodic task allocation. Two tests are carried out to verify the stability of the system: an analysis based on the spectral radius and, alternatively, an evaluation based on linear matrix inequality (LMI) conditions. In both cases, the stability matrices were created from equations (64) and (65); let the spectral radius of the system be ρ, and the spectral radius of the time-delay system be ρ^D, we have.

$$\rho = \begin{bmatrix} 0.9418 & 0.9416 & 0.9447 & 0.9444 \\ 0.9416 & 0.9418 & 0.9444 & 0.9447 \\ 0.9447 & 0.9444 & 0.9418 & 0.9416 \\ 0.9444 & 0.9447 & 0.9416 & 0.9418 \end{bmatrix}, \qquad \rho^D = \begin{bmatrix} 0.4401 & 0.4380 & 0.4335 & 0.4358 \\ 0.4380 & 0.4401 & 0.4358 & 0.4335 \\ 0.4335 & 0.4358 & 0.4401 & 0.4380 \\ 0.4358 & 0.4335 & 0.4380 & 0.4401 \end{bmatrix}. \tag{79}$$

giving

$$\max(\rho_{ij}) = 0.9447 < 1, \qquad \max(\rho_{ij}^D) = 0.4401 < 1 \tag{80}$$

which indicates that all the matrices considered are Schur, which implies asymptotic stability.

The solution of the linear matrix inequality was successfully obtained using the SeDuMi solver, yielding the Lyapunov matrices

$$P = \begin{bmatrix} 2.0139 & 0 & 0 & 0.6212 & 0 & 0 \\ 0 & 2.2104 & 0 & 0 & 0.6791 & 0 \\ 0 & 0 & 1.2704 & 0 & 0 & -0.4206 \\ 0.6212 & 0 & 0 & 0.4185 & 0 & 0 \\ 0 & 0.6791 & 0 & 0 & 0.4252 & 0 \\ 0 & 0 & -0.4206 & 0 & 0 & 1.0494 \end{bmatrix} \qquad R = \begin{bmatrix} 2.9066 & 0 & 0 & -0.4243 & 0 & 0 \\ 0 & 0.9157 & 0 & 0 & -0.2809 & 0 \\ 0 & 0 & 1.6952 & 0 & 0 & -0.0751 \\ -0.4243 & 0 & 0 & 0.6555 & 0 & 0 \\ 0 & -0.2809 & 0 & 0 & 0.5198 & 0 \\ 0 & 0 & -0.0751 & 0 & 0 & 0.8040 \end{bmatrix} \tag{81}$$

Finally, the condition is verified

$$E_{ij} = G_{ij}^T P G_{ij} - P + T_{ij}^T R T_{ij} - R \tag{82}$$

calculating the maximum eigenvalue of each matrixE_{ij}:

$$\lambda_{\max} = \begin{bmatrix} -0.5203 & -0.6308 & -0.7083 & -0.7052 \\ -0.6308 & -0.5203 & -0.7053 & -0.7083 \\ -0.7083 & -0.7053 & -0.5203 & -0.6308 \\ -0.7052 & -0.7083 & -0.6308 & -0.5203 \end{bmatrix}. \tag{83}$$

since

$$\lambda_{\max}(i,j) < 0, \qquad \forall i,j \in \{1,\ldots,4\}, \tag{84}$$

The proposed LMI condition is feasible and, consequently, the quadratic stability of the considered TS system under the designed control law is guaranteed.

Giving a proper response from sensors and actuators signals as shown in Figs 10 and 11. Although there is a perturbation from the second 200, this is compensated through our proposed strategy

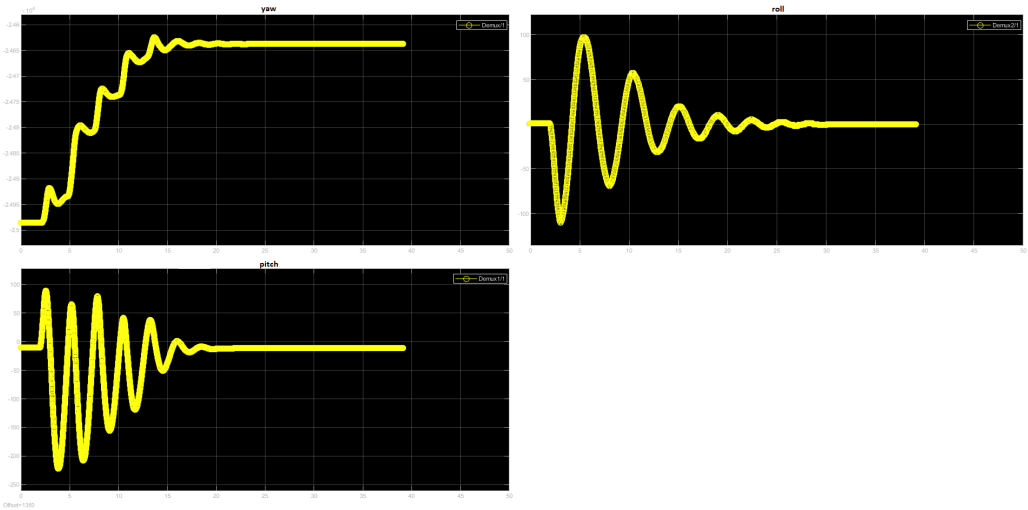


Figure 10: Sensor signal for 9 nodes.

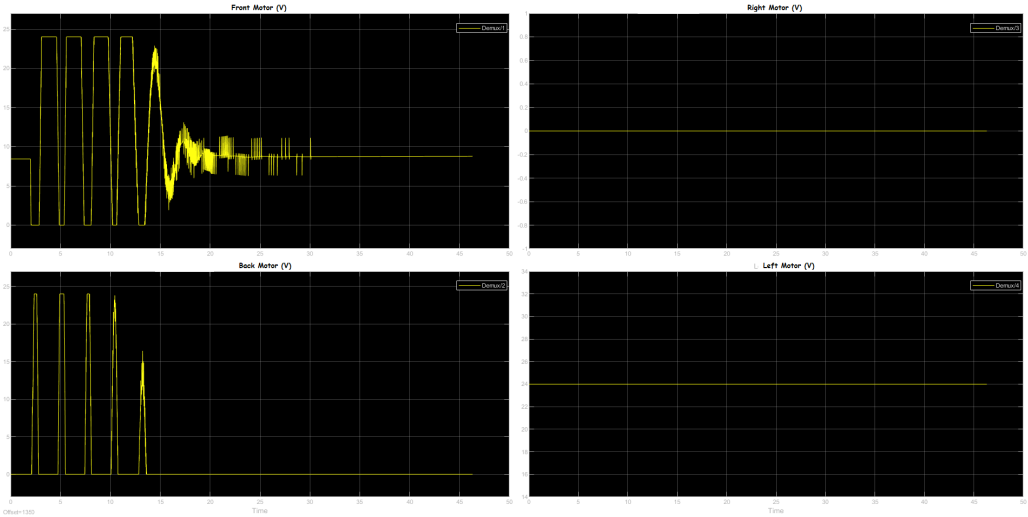


Figure 11: Response of the actuators for 9 nodes.

Regarding the communication network, the Round Trip Time (RTT) for Sensor 0 is shown in Fig. 12. This parameter exhibits a relatively large variation during the initial communication packets. However, the overall dispersion remains limited, as the mean jitter is *0.0512 ms*, which is comparable to the mean jitter observed for the remaining sensors, *0.05 ms*.

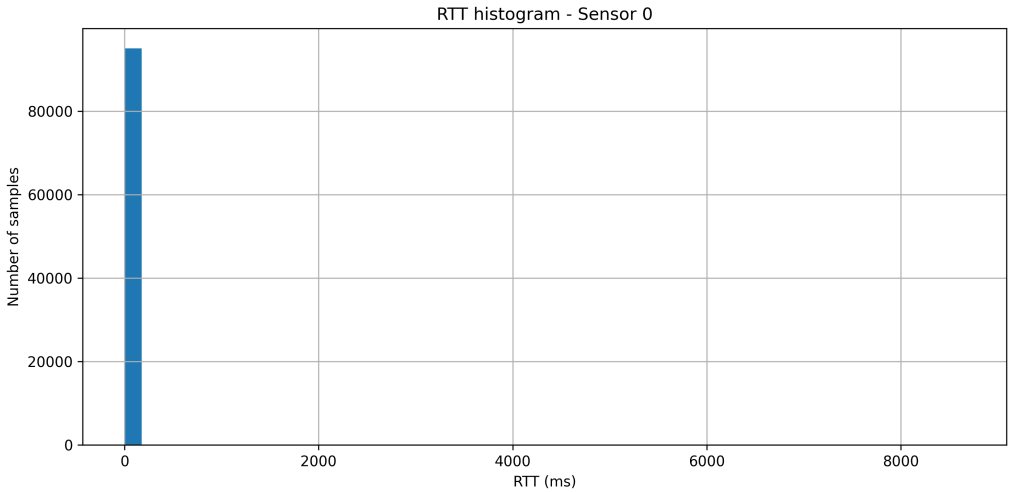


Figure 12: Response from RTT related to sensor 0.

In terms of the Total Round Trip Time, the dispersion is larger, as expected, due to the overhead introduced by the initial communication links. It is important to emphasise that there is no actual synchronisation among the nodes; instead, there is only communication between them in accordance with the adopted scheduling strategy.

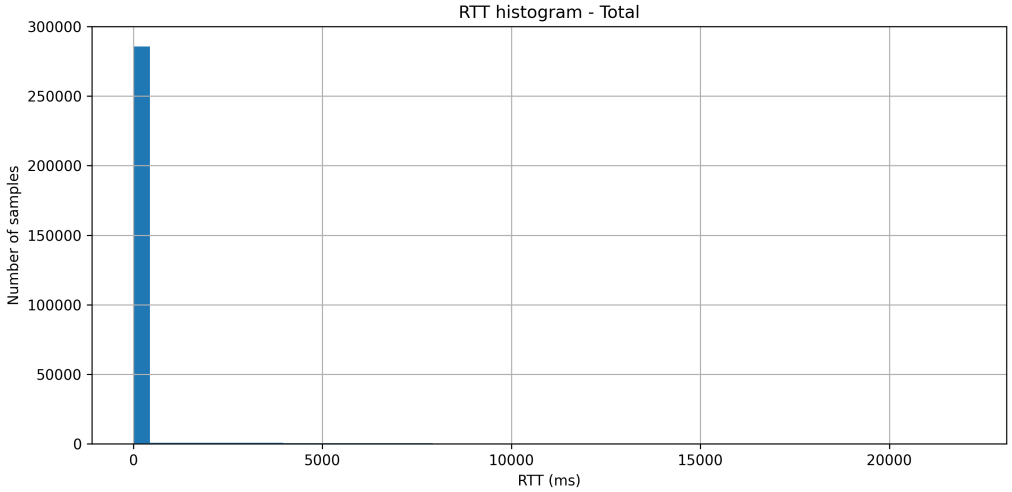


Figure 13: Response from RTT total for 9 nodes.

5 Conclusion

An experimental evaluation of distributed control architectures applied to a Quanser 3DOF Hover platform is presented. It has been demonstrated and implemented that incorporating a multinode scheduler maintains system stability even with 9 nodes, mitigating the adverse effects of jitter and the concurrency behaviour of the **Priority Exchange** algorithm. Although the scheduler does not allow a specific time delay to be determined, given the random nature of aperiodic tasks, delay bounds can be determined by ordering the periodic tasks. This allows the bounded random effect on the timing order of aperiodic tasks to be determined without disturbing more than a portion of the sample when

a service request for an aperiodic task arrives. In this sense, the time-ordered scheduling scheme can be used to determine a stable system for any delay condition. The side effects of time delays are bounded to a portion of time that can be managed through a gain scheduler such as a TKS Fuzzy Controller based upon main probability condition ζ_k^i.

The approach proposed in this manuscript demonstrates that it is particularly useful for bounding delay; it reduces the synchronisation time window, allowing us to determine a viable stability region within the environment of bounded time delays. In Fact, by considering the probabilities bounded by the intervention time of aperiodic tasks, the inherent variations due to jitter are also taken into account, as they are much smaller than those of the non-periodic task itself. The importance of managing known and bounded delays in time enables us to define both the minimum and maximum effects on the system's dynamics. Indeed, unlike what is presented in [6] [4] [9], the fundamental consideration is the probability of the event occurring ζ_k^i; if it does, the necessary time to process a non-periodic task is known $\Delta t(\zeta_k^i)$. This leads us to a fundamental definition of the control law given these delay limits; furthermore, Fuzzy Logic Control design remains a feasible strategy, even under bounded conditions. In fact, as presented in [9], it remains a valid algorithm for Priority Exchange plus Rate Monotonic; nevertheless, the current implementation is more flexible for aperiodic tasks by considering bounded probabilities.

As future work, the integration of precise synchronisation using PTP (IEEE 1588) [19] is under consideration, both for a fully synchronous communication block and for migration to ROS2- and DDS-based architectures [15, 22] to enable more flexible handling of sensors, actuators, and other electromechanical systems to be controlled over the same computing network. In this regard, packet loss warrants the development of a strategy, such as the robustness study in the presence of actual packet loss proposed in [18]. In fact, the use of deterministic networks via TSN [32] is a topic of study in various electromechanical systems that share a common data network.

Acknowledgmnets

The authors would like to thank the project PAPIIT UNAM IN106426 and IT101323 for its financial support

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Beard, R. W. (2008). Quadrotor Dynamics and Control, *IEEE Control Systems Magazine*, 28(5), 38–45, 2008.

[2] Beard R. W. and McLain T. W. (2012). Small Unmanned Aircraft: Theory and Practice, *Princeton University Press*, 2012.

[3] Benítez-Pérez, H.; Cárdenas-Flores, F.; García-Nocetti, F. (2007). An Implementation of Reconfigurable Network Control based upon Automata Proposal for Three Conveyor Belt Case Study, *International Journal of Computers, Communications & Control*, 2(4), 314-327, 2007.

[4] Benítez-Pérez H., Cardenas-Flores F., and García-Nocetti F. (2011) Reconfigurable Takagi Sugeno Fuzzy Logic Control for a Class of Nonlinear System Considering Communication Time Delays on Peripheral Elements, *International Journal of Computers Communications & Control* 6(3),384-399, 2011

[5] Benítez-Pérez, H.; Cárdenas-Flores, F.; García-Nocetti, F. (2011). Reconfigurable Takagi-Sugeno Fuzzy Logic Control for a Class of Nonlinear System considering Communication Time Delays on Peripheral Elements, *Int. J. of Computers, Communications and Control*, 6(3), 387-402, 2011.

[6] Benítez-Pérez, H.; Benítez-Pérez, A.; Ortega-Arjona, J.; Esquivel-Flores, O. (2012). Fuzzy Networked Control Systems Design Considering Scheduling Restrictions, *Hindawi Publishing Corporation, Advances in Fuzzy Systems*, 2012.

[7] Benítez-Pérez, H.; Benítez-Pérez, A.; Ortega-Arjona, J. (2013). Networked Control Systems Design Considering Scheduling Restrictions and Local Faults Using Local State Estimation, *International Journal of Innovative Computing, Information & Control*, 9(8), 2013.

[8] H. Benítez-Pérez, J. Ortega-Arjona, O. Esquivel-Flores, J.A. Rojas-Vargas, A. Alvarez-Cid, (2016) A Fuzzy Networked Control System Following Frequency Transmission Strategy, *International Journal of Computers Communications & Control* 11(1), 10-25, 2016.

[9] Benítez-Pérez H.; Ortega-Arjona J.; Rojas-Vargas J.A; Durán-Chavesti A. (2016) Design of a Fuzzy Networked Control Systems Priority Exchange Scheduling Algorithm, *International Journal of Computers Communications & Control*, 11(2), 179-193, 2016.

[10] Benítez-Pérez H., Hermosillo-Gomez, J. (2021) Job-shop scheduling over a heterogeneous platform, *International Journal of Computers Communications & Control* 16(3), 1-11, 2021.

[11] Bouabdallah S., Murrieri P., and Siegwart R. (2004). Design and Control of an Indoor Micro Quadrotor, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 4393–4398, 2004.

[12] Bouabdallah, S., Murrieri, P., and Siegwart, R. (2007). Design and Control of an Indoor Micro Quadrotor, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 4393–4398, 2007.

[13] Buttazo, G. (2004) Hard Real-Time Computing Systems, Springer 2004.

[14] Castillo O.; Benítez-Pérez H. (2020) Improving NCS Stabilisation Using a Predictive Pulsed Control Law, *International Journal of Computers Communications & Control*, 15(6), 2020.

[15] Data Distribution Service (DDS) Specification, *Object Management Group*, version 1.4, 2015

[16] Esquivel-Flores, O.; Benítez-Pérez, H.; Méndez-Monroy, P.; Ortega-Arjona,J. (2012). Bounded communication between nodes of a networked control system as a strategy of scheduling, *International Journal of Parallel, Emergent and Distributed Systems*, 27(6), 481-502,2012.

[17] J. Ángel Hermosillo-Gómez and Héctor Benítez-Pérez. (2019), RUN enhancement through Bayesian networks *International Journal of Parallel, Emergent and Distributed Systems* 34(5), 523–537, 2019.

[18] Hespanha, João P. and Naghshtabrizi, Payam and Xu, Yonggang. A Survey of Recent Results in Networked Control Systems, *Proceedings of the IEEE*, 95(1), 138-162, 2007.

[19] IEEE Standard for a Precision Clock Synchronisation Protocol for Networked Measurement and Control Systems, *IEEE*, IEEE Std 1588-2008, 2008.

[20] Khalil H. K. (2002). Nonlinear Systems, *Prentice Hall*, 3rd Edition, 2002.

[21] Kopetz, Hermann. Real-Time Systems: Design Principles for Distributed Embedded Applications, *Springer*, 2011.

[22] Maruyama, Yu and Kato, Shinpei and Azumi, Takuya. Exploring the Performance of ROS2, *Proceedings of the International Conference on Embedded Software (EMSOFT)*, 2016.

[23] Nilsson, Johan. Real-Time Control Systems with Delays, *Lund Institute of Technology*, 1998.

[24] Quanser 3DOF Hover Experiment User Manual, *Quanser Inc.*, 2020

[25] Ramos-Fernández A. (2021). Sistema de Control Sobre Redes Utilizando Redes Neuronales y Modelos Ocultos de Markov, Master Thesis, *Universidad Nacional Autónoma de México.*, Programa de Maestría y Doctorado en Ingeniería, 2021.

[26] F. M. S. Nascim nto and G. Lima. (2023). Dealing with Soft Aperiodic Tasks in RUN *2023 XIII Brazilian Symposium on Computing Systems Engineering (SBESC)* doi: 10.1109/SBESC6096.2023.10324231. 1-6. 2023.

[27] Slotine J.-J. E. and Li W. (1991). Applied Nonlinear Control, *Prentice Hall*, 1991.

[28] Spong M. W., Hutchinson S., and Vidyasagar M. (2006). Robot Modeling and Control, *John Wiley & Sons*, 2006.

[29] Stevens B. L., Lewis F. L., and Johnson E. N. (2015). Aircraft Control and Simulation, *John Wiley & Sons*, 2015.

[30] Tanaka K., and Wang H.(2008). Fuzzy Control Systems Design and Analysis, *John Wiley & Sons*, 2008.

[31] Tipsuwan, Yodyium and Chow, Mo-Yuen. Control Methodologies in Networked Control Systems, *Control Engineering Practice*, 11(10), 1099-1111, 2003.

[32] Time-Sensitive Networking (TSN) Overview, *IEEE 802.1 Time-Sensitive Networking Task Group*, 2018.

[33] Zhang, Wei and Branicky, Michael S. and Phillips, Stephen M. Stability of Networked Control Systems, *IEEE Control Systems Magazine*, 21(1), 84-99, 2001.

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Benítez-Pérez, H.; Villarreal-Martínez, F.D.; Rodríguez-Martínez, R.C.; Durán-Chavesti, A.; Pérez-Quezadas, N.I.; Aparicio-Santos, A.; (2026). Experimental Evaluation of a Real-Time Distributed Control System for a Quanser Quadcopter: Multinode Architectures with Centralised Scheduling, *International Journal of Computers Communications & Control*, 21(4), 7527, 2026.
https://doi.org/10.15837/ijccc.2026.4.7527