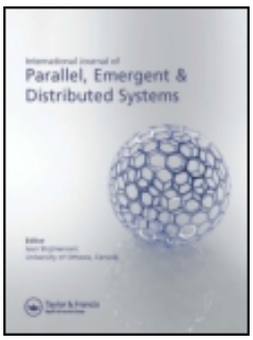

International Journal of Parallel, Emergent and Distributed Systems

ISSN: 1744-5760 (Print) 1744-5779 (Online) Journal homepage: http://www.tandfonline.com/loi/gpaa20

Bounded communication between nodes of a networked control system as a strategy of scheduling

O. Esquivel-Flores , H. Benítez-Pérez , P. Méndez-Monroy & J. Ortega-Arjona

To cite this article: O. Esquivel-Flores , H. Benítez-Pérez , P. Méndez-Monroy & J. Ortega-Arjona (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, DOI: 10.1080/17445760.2012.717941

To link to this article: http://dx.doi.org/10.1080/17445760.2012.717941

International Journal of Parallel, Emergent and Distributed Systems
Vol. 27, No. 6, December 2012, 481–502

Bounded communication between nodes of a networked control system as a strategy of scheduling

O. Esquivel-Flores[a], H. Benítez-Pérez[b]*, P. Méndez-Monroy[c] and J. Ortega-Arjona[d]

[a]Posgrado en Ciencias e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Mexico D.F., Mexico; [b]Departamento de Ingeniería de Sistemas Computacionales y Automatización, Instituto de Investigación en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Apdo. Postal 20-726. Del. A. Obregón, C.P. 01000, Mexico D.F., Mexico; [c]Programa de Doctorado en Ingeniería, Universidad Nacional Autónoma de México, Mexico D.F., Mexico; [d]Departamento de Matemáticas, F. Ciencias, Universidad Nacional Autónoma de México, Mexico, D.F., Mexico

(*Received 27 December 2010; final version received 31 July 2012*)

In a networked control system, several nodes exchange information through a network, to achieve specific control goals and thus increasing network traffic. This affects the overall system performance. Several approaches try to satisfy requirements of both control and communication performance. Particularly, some methodologies have been proposed to save bandwidth. One of such methodologies has been scheduling, which has been studied in depth through the last decade. Commonly, the objective of using scheduling to save bandwidth is to accurately use the computing resources. This paper shows two scheduling strategies, one performing static scheduling and the other carrying out dynamic scheduling, in order to expose the advantages of using dynamic scheduling in an ad hoc implementation. Both strategies execute on a real-time distributed system, and both are able to modify the frequency of transmission as well as the periods of tasks in individual components. Hence, both of them tend to impact on the quality of performance of the system, due to network use. The first scheduling strategy modifies the periods of task, and network access is assigned through a static scheduling algorithm. On the other hand, the second strategy, schedulability, is dynamically achieved by controlling the rate of frequency transmission into a frequency region, bounded by minimum and maximum transmission rates. Numerical simulations are used as implementations of both strategies.

Keywords: distributed systems; real time; frequency transmission; control

1. Introduction

A networked control system (NCS) is a current application of a real-time distributed system (RTDS), composed of a number of nodes capable of developing a complete control process. Important parts of such a control process are sensor and actuator activities, which rely on a real-time operating system and real-time communication network. NCSs have received increasing attention through the last few years, due to advantages such as easy maintenance, high reliability, less wiring and low cost. Besides, the use of control network architectures, when properly applied, may improve efficiency, flexibility and reliability, reducing installation, reconfiguration and maintenance in time and costs.

*Corresponding author. Email: hector@uxdea4.iimas.unam.mx

ISSN 1744-5760 print/ISSN 1744-5779 online

http://dx.doi.org/10.1080/17445760.2012.717941
http://www.tandfonline.com

Nevertheless, notice that the use of a single communication network for all control signals (for example, the feedback control loop) affects the performance of the NCS. Furthermore, conventional standard results in analysis and design of traditional digital control theory cannot be used, since many of its assumptions are not directly applicable to NCS [21], specifically considering time delays.

The use of common-bus network architecture introduces different forms of time-delay uncertainties between sensors, actuators and controllers. Several approaches have been developed to tackle this problem, mainly based on the work of Halevy and Ray [8]. In such approaches, a group of nodes make use of the same network, producing a traffic load. If there is no coordination among the nodes, data transmissions may simultaneously occur, and back off is needed to avoid collisions or bandwidth violations. This results in tasks which are unable to fulfil their deadlines. In order to achieve overall objectives of all tasks performed in a NCS, it is necessary for all nodes to properly exchange their own information through the network. Therefore the mechanism to exchange information plays an important role on the stability and performance of the control systems implemented over a communication network.

Here, in order to satisfy control requirements and communication performance, a co-design methodology [1,19,20] is used to generate proper control actions and to optimally utilise communication bandwidth. Regarding this, Lian et al. [10–13] provide a way to choose a sampling period to minimise the effect of delays. The effectiveness of the control system depends on such a sampling rate. A region is acceptable in digital control performance terms if it is contained within two sampling rate boundaries, which can be statistically determined. The lower boundary is a small sampling rate necessary to guarantee a level of control performance. The higher boundary implies a sampling rate that gets faster while the network traffic load increases, getting the possibility of contention time or data loss increment on a bandwidth-limited network. Thus, managing sampling rates is proposed as a key issue in network scheduling design. For instance, Branicky et al. [3] show experimental studies of control and feedback scheduling for a NCS, consisting of dynamic system simulations for the control nodes and the environment, and packet-level network simulations for the communication. They make use of the NS2 package [4] as a network simulator tool for studying packet dynamics. Another proposal for managing sampling rates, aiming to reduce the number of data transmitted over the network, is presented in [21]. In this work, it is assumed that a NCS is modelled using a single input single output (SISO) linear control system. This control system has state feedback, using incremental control input as an index of intensity of the system behaviour, in order to predict changes in sampling periods. Peng et al. [17] attempt to balance the sampling period and the amount of data transmitted over the network. This is achieved through an optimal scheduling algorithm with multiple control loops, considering control restrictions and limitations of network resource. Data transmission frequencies replace the sampling periods to facilitate the use of the generalised exponential distribution as an indicator of control performance. Finally, Menéndez-Leonel and Benítez-Pérez [15] proposed the design of a global scheduling strategy, based on the analysis of NCS. They show that the performance of the system depends not only on the sampling periods of its individual components but also on the time dispersions amongst these periods.

This paper discusses and extends the last approach, in order to show the effectiveness of managing sampling rates, to avoid long-time delays. This strategy is compared with a dynamical scheduling strategy, based upon modifications of frequency transmission of

individual components in the system [7]. Data exchange is carried out under specific frequency transmission, through a scheduler node, which applies a control frequency algorithm, based upon the information retrieved by smart sensors. This particular type of sensors is distinguished by its communication and consensus capabilities [2]. Hence, the main objective of this paper is to expose the importance of managing sampling rates, and a way to modify them to obtain an improvement in the scheduling strategy. This is demonstrated using a simulated case study based upon a degree of freedom (2-DOF) helicopter simulation benchmark [18]. This simulation provides an approximation to system response, in which, for demonstration purposes, the main results are obtained for a typical fault scenario [9]. Thus, for this simulation, two scheduling strategies are implemented using TrueTime[1] [5,6]: one performing dynamic scheduling and the other carrying out static scheduling. Both strategies execute on a real-time distributed system, modifying the frequency of transmission, as well as the periods of tasks in individual components. Thus, due to network use, both strategies impact on the quality of performance of the NCS. Notice, nevertheless, that the first strategy modifies the periods of tasks, and network access is assigned through a static scheduling algorithm; the second strategy achieves dynamic schedulability by controlling the rate of frequency transmission into a frequency region. This region is bounded by a minimum and a maximum transmission rate. The objective is to expose the advantages of using dynamic scheduling in an ad hoc implementation.

The rest of the paper is organised as follows: Section 2 presents a helicopter model, as a specific RTDS, used to implement scheduling approaches. Section 3 develops the scheduling strategy based on allocation bandwidth for the NCS. Section 4 describes the frequency transmission scheduling strategy, providing some results from the simulations which we use to compare the strategies exposed. Finally, conclusions are presented in Section 5.

2. A real-time distributed system – problem formulation

The RTDS used for implementation purposes of this paper is a 2-DOF helicopter prototype [18]. The following sections briefly introduce and describe this 2-DOF helicopter prototype and its controller design, as well as an experimental approach to express this prototype as a NCS.

2.1 The 2-DOF helicopter dynamic model and its control design

The 2-DOF helicopter [18] system is a prototype with two propellers driven by DC motors, and integrating a CanBus network [12]. The front propeller controls the elevation of the helicopter nose on the pitch axis, and the back propeller controls the side-to-side movement on the yaw axis. These pitch and yaw angles describe the state of the helicopter and are measured using high-resolution encoders.

The dynamics of the model are based on kinetic and potential energy, and it is used for the design of a position controller: the helicopter centre of mass is described in xyz Cartesian coordinates regarding the pitch (θ) and yaw (ψ) angles (Figure 1).

The Euler–Lagrange equations are used to obtain nonlinear equations of motion for the 2-DOF helicopter, which are used to derive the linear state model, and subsequently, to design the position controller. Table 1 shows the specifications and parameters associated with the helicopter system.

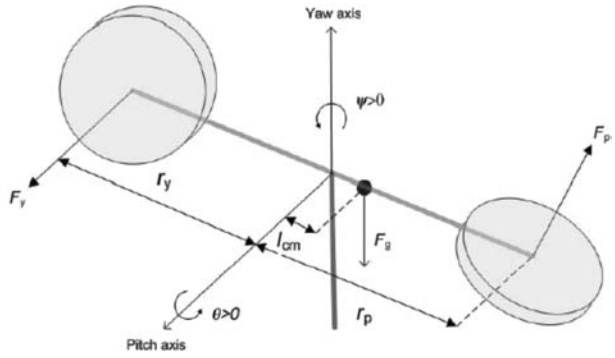

Figure 1. Helicopter dynamics.

As the helicopter represents a nonlinear system, it is required to perform a linearisation around a point. This linearisation point is

$$(\theta_0 = 0,\ \ \psi_0 = 0,\ \ \dot{\theta}_0 = 0,\ \ \dot{\psi}_0 = 0).$$

Table 1. Helicopter specification and parameters.

Symbol	Matlab notation	Description	Value	Unit
$B_{cq,p}$	B_eq_p	Equivalent viscous damping about pitch axis	0.800	N/V
$B_{cq,y}$	B_eq_y	Equivalent viscous damping about yaw axis	0.318	N/V
m_{heli}	m_heli	Mass of the helicopter	1.387	kg
$m_{m,p}$	m_motor_p	Mass of pitch motor	0.292	kg
$m_{m,y}$	m_motor_y	Mass of yaw motor	0.128	kg
m_{shield}	m_shield	Mass of propeller shield	0.167	kg
m_{props}	m_props	Mass of pitch and yaw propellers, propeller shields and motors	0.754	kg
$m_{body,p}$	m_body_p	Mass moving about the pitch axis	0.633	kg
$m_{body,y}$	m_body_y	Mass moving about the yaw axis	0.667	kg
m_{shaft}	m_shaft	Mass of the metal shaft rotating about the yaw axis	0.151	kg
L_{body}	L_body	Total length of helicopter body	0.483	m
l_{cm}	l_cm	Centre of mass length along helicopter body from pitch axis	0.186	cm
L_{shaft}	L_shaft	Length of metal shaft rotating about the yaw axis	0.280	m
$J_{body,p}$	J_m_p	Moment of inertia of helicopter body about the pitch axis	0.012	$kg\,m^2$
$J_{body,y}$	J_m_y	Moment of inertia of helicopter body about the yaw axis	0.013	$kg\,m^2$
J_{shaft}	J_shaft	Moment of inertia of metal shaft about yaw axis at end point	0.004	$kg\,m^2$
J_p	J_p	Moment of inertia of front motor/shield assembly about pitch pivot	0.018	$kg\,m^2$
J_y	J_y	Moment of inertia of back motor/shield assembly about yaw pivot	0.008	$kg\,m^2$
$J_{eq,p}$	J_eq_p	Total moment of inertia about pitch pivot	0.038	$kg\,m^2$
$J_{eq,y}$	J_eq_y	Total moment of inertia about yaw pivot	0.043	$kg\,m^2$

From this, the linearisation of the motion equation is obtained as follows:

$$\left(J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2\right)\ddot{\theta} = K_{\mathrm{pp}} V_{\mathrm{m,p}} + K_{\mathrm{py}} V_{\mathrm{m,y}} - B_{\mathrm{p}}\dot{\theta} - m_{\mathrm{heli}} g l_{\mathrm{cm}}, \tag{1}$$

$$\left(J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2\right)\ddot{\psi} = K_{\mathrm{yy}} V_{\mathrm{m,y}} + K_{\mathrm{yp}} V_{\mathrm{m,p}} - B_{\mathrm{y}}\dot{\psi} - 2m_{\mathrm{heli}} l_{\mathrm{cm}}^2 \theta\dot{\psi}\dot{\theta}. \tag{2}$$

Considering θ as the pitch angle, ψ as the yaw angle, $\dot{\theta}$ as the pitch derivative, $\dot{\psi}$ as the yaw derivative and the state vector $\mathbf{x}$ is defined as follows:

$$\mathbf{x} = \left[\theta,\ \ \psi,\ \ \dot{\theta},\ \ \dot{\psi}\right]'.$$

Substituting in Equations (1) and (2) and solving for $\dot{x}$, the following linear model of state spaces is obtained:

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{B_{\mathrm{p}}}{J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & 0 \\ 0 & 0 & 0 & -\frac{B_{\mathrm{y}}}{J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{\mathrm{pp}}}{J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & \frac{K_{\mathrm{py}}}{J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} \\ \frac{K_{\mathrm{yp}}}{J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & \frac{K_{\mathrm{yy}}}{J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} \end{bmatrix} u,$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x,$$

where K_{pp}, K_{yy}, K_{py} and K_{yp} are the torque constants used to obtain coupled torques acting on the pitch and yaw axes.

For the state space model, the input (**u**) and output (**y**) vectors are as follows:

$$\mathbf{u} = \left[V_{\mathrm{m,p}},\ \ V_{\mathrm{m,y}}\right]',$$

$$\mathbf{y} = \left[x_1,\ \ x_2,\ \ x_3,\ \ x_4\right]',$$

where $V_{\mathrm{m,p}}$ is the input pitch motor voltage and $V_{\mathrm{m,y}}$ is the input yaw motor voltage. Notice that, since the output matrix is the identity matrix, all states are measurable.

The model makes use of several Simulink and Matlab programs to develop the helicopter basic dynamics by running a simulation of the closed-loop response, using the position controller.

Regarding control issues, two controllers are designed: feed-forward (FF) − linear quadratic regulator (LQR) and FF + LQR + integrator (I). The FF + LQR regulates the pitch axis of the helicopter using FF and proportional-velocity (PV) compensators, while the yaw axis only makes use of a PV control. The FF + LQR + I controller uses an integrator in the feedback loop to reduce the steady-state error by a FF and proportional-integral-velocity (PIV) algorithm to regulate the pitch and only a PIV algorithm to control the yaw angle. This work focuses on the FF + LQR + I controller as follows.

The FF − LQR control converges $(\theta,\ \psi,\ \dot{\theta},\ \dot{\psi}) \rightarrow (\theta_{\mathrm{d}},\ \psi_{\mathrm{d}},\ 0,\ 0)$, where θ_{d} is the desired pitch angle and ψ_{d} is the desired yaw angle such that

$$\begin{bmatrix} u_{\mathrm{p}} \\ u_{\mathrm{y}} \end{bmatrix} = \begin{bmatrix} k_{\mathrm{ff}} \frac{m_{\mathrm{heli}} g l_{\mathrm{cm}} \cos\theta_{\mathrm{d}}}{K_{\mathrm{pp}}} \\ 0 \end{bmatrix} - \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix} \begin{bmatrix} \theta - \theta_{\mathrm{d}} \\ \psi - \psi_{\mathrm{d}} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}.$$

The addition of an integrator requires to introduce the states $\dot{x}_5 = \theta$ and $\dot{x}_6 = \psi$, so the linear state-space model is augmented as

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{B_{\mathrm{p}}}{J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{B_{\mathrm{y}}}{J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{\mathrm{pp}}}{J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & \frac{K_{\mathrm{py}}}{J_{\mathrm{eq,p}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} \\ \frac{K_{\mathrm{yp}}}{J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} & \frac{K_{\mathrm{yy}}}{J_{\mathrm{eq,y}} + m_{\mathrm{heli}} l_{\mathrm{cm}}^2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u,$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x.$$

Using the adequate **Q** and **R** weighting matrices, the control gain is as follows:

$$k = \begin{bmatrix} 18.9 & 1.98 & 7.48 & 1.53 & 7.03 & 0.77 \\ -2.22 & 19.4 & -0.45 & 11.9 & -0.77 & 7.03 \end{bmatrix}.$$

Thus the FF + LQR + I controller is

$$\begin{bmatrix} u_{\mathrm{p}} \\ u_{\mathrm{y}} \end{bmatrix} = \begin{bmatrix} k_{\mathrm{ff}} \frac{m_{\mathrm{heli}} g l_{\mathrm{cm}} \cos\theta_{\mathrm{d}}}{K_{\mathrm{pp}}} \\ 0 \end{bmatrix} - \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix} \begin{bmatrix} \theta - \theta_{\mathrm{d}} \\ \psi - \psi_{\mathrm{d}} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} \int k_{15}(\theta - \theta_{\mathrm{d}}) + \int k_{16}(\psi - \psi_{\mathrm{d}}) \\ \int k_{25}(\theta - \theta_{\mathrm{d}}) + \int k_{26}(\psi - \psi_{\mathrm{d}}) \end{bmatrix}.$$

Figure 2 shows the closed control loop of 2-DOF helicopter prototype.

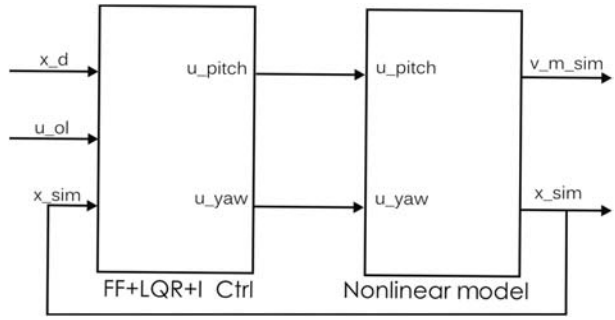

Figure 2. The closed control loop for the helicopter.

For a more complete description and detailed information of the 2-DOF helicopter system, see [18].

2.2 Experimental approach

In order to study the impact of network utilisation on closed control loop, the 2-DOF helicopter control model is built as a RTDS. Several nodes are connected through a common communication network. The experiments focus on network scheduling, and the main objective is to balance the amount of data sent through the network, in order to avoid latency and undersampling. Two network scheduling proposals are given to explore several aspects in control performance when the use of the network exceeds network bandwidth.

To have a performance criterion, and thus, quantify the system's quality performance, the integral of the absolute value of the error (IAE) is used:

$$\mathrm{IAE} = \int_{t_0}^{t_\mathrm{f}} |e(t)|\,\mathrm{d}t \approx \sum_{k=k_0}^{k_\mathrm{f}} |r(kh) - y(kh)|,$$

where $r(kh)$ is the reference signal, $y(kh)$ is the system output signal, $t_0(k_0)$ and $t_\mathrm{f}(k_\mathrm{f})$ are the minimum and maximum times of the evaluation period.

A fundamental requirement of a control system is stability. Control community has several versions of the definition of stability. For example, a system is stable if for any bounded input, the output is bounded; a system is unstable if the system output unboundedly diverges. Apart from stability, the transient behaviour is another focus of attention for control systems design.

In this paper, the control error $e(t)$, which is defined as the difference between the set point $r(t)$ and the system output $y(t)$, is computed using the IAE. This value gives an index to evaluate the performance of the system. For this criterion, the bigger value corresponds to the worst performance. In this paper, stable behaviour deals with small values of the IAE, and unstable behaviour deals with large values of the IAE.

The actual RTDS for the experiments consists of eight processors. These real-time kernel processors and the network are simulated using TrueTime [5,6,16]. The network used is a CSMA/AMP (CAN) with a transmission rate of 80,000 bits/s, minimum frame size of 40 bits, and not data loss. The TrueTime Network block simulates the physical and medium-access layer of various local-area networks. Other types of networks supported by TrueTime are CSMA/CD (Ethernet), CSMA/AMP (CAN), Round Robin (Token Bus),

FDMA, TDMA (TTP), Switched Ethernet, WLAN (802.11b) and ZigBee (802.15.4). Notice that the network blocks only simulate the medium access (the scheduling), possible collisions or interference, as well as point-to-point/broadcast transmissions. Higher layer protocols such as TCP/IP are not supported but may be implemented as processes in the block nodes.

Four sensor nodes execute periodic tasks to sense control signals as well as other additional periodic tasks. Each task has a period p_i and time consumption c_i (Figure 3). The sensed control signals are θ, ψ, $\dot{\theta}$ and $\dot{\psi}$. This model has a controller node, depicted on the left side (Figure 3). This controller takes the control law from the FF + LQR module by means of a task, which activates by event. The time consumption of the controller task is the maximum average time it takes to compute the control law. The controller node uses the sensor readings and sends control outputs u_{p} and u_{y} that correspond to the pitch and yaw voltages. Two actuator nodes, located on the bottom right corner (Figure 3), receive signals from the controller node. Finally, the scheduler node, located on the top right corner (Figure 3), organises the activity of the other seven nodes, and it is responsible for periodic allocation bandwidth used by these nodes.

Each node initialises, specifying the number of inputs and outputs of the respective TrueTime kernel block, defining a scheduling policy and creating periodic tasks for the simulation. These tasks involve parameters about the periodic times and the consumption times. The task periodic times define the time interval between tasks, whereas the consumption times refer to the execution time of the task. Figure 4 shows the 2-DOF helicopter model, with a RTDS, where feedback control loop is closed through a communication network.

Changes on the real-time task parameters of the RTDS commonly impact on network utilisation, and therefore, on the control performance [11,13]. The problem to tackle, thus, is to find a proper way to schedule the common communication network of the RTDS,

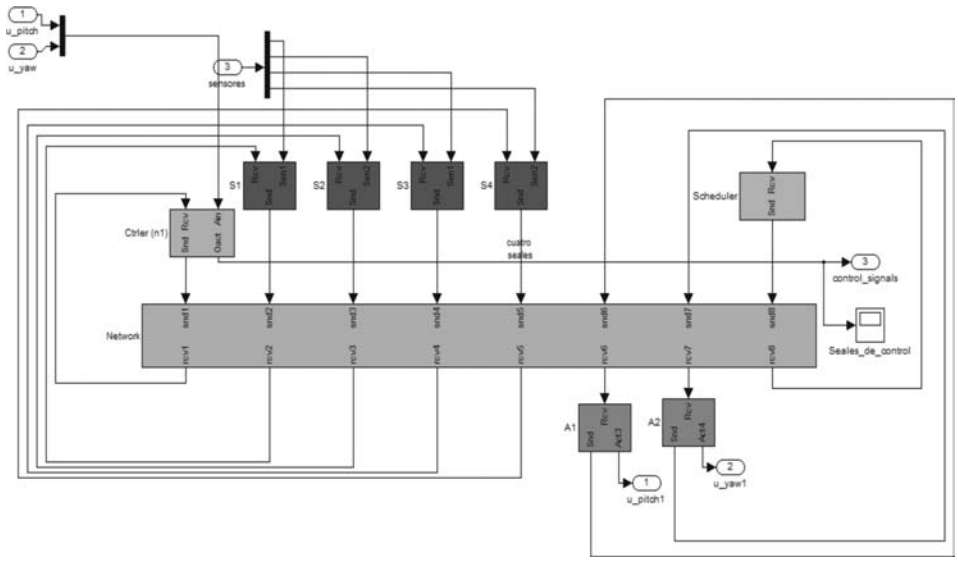

Figure 3. RTDS into helicopter model.

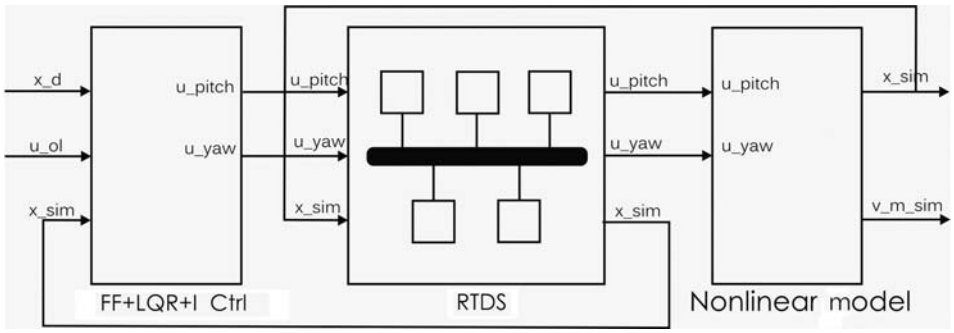

Figure 4. NCS for the helicopter model.

based on managing an accurate sampling period, capable of keeping both the network load and the required integrated performance.

In this paper, the NS2 network simulator [3,4] is not applied, since the use of NS2 has been reviewed in terms of network bandwidth and some scheduling strategies such as earliest deadline first (EDF) or Token Ring. From this, several strategies may be usable for scheduling. However, none of them guaranteed a valid response in terms of time constraints, since real time is an important restriction for the purposes here. Even though NS2 allows simulation of complex situations, in terms of data exchange, it neglects the local jitter that is inherent of the simulated transactions, since it makes use of a scale factor that is quite significant compared to the delay of data transfer. Hence, the accuracy of NS2 simulator is out of the scope of this paper, since the work here necessarily deals with modelling the features of the data exchange.

3. Network scheduling based on allocation bandwidth

The first scheduling strategy to be experimented consists of determining a base period ρ, which allows obtaining the operational periods of all nodes that participate in the RTDS. Even though all the nodes can use this base period as their operational period, the strategy includes the option of having different periods for each node, by means of the base period and a dispersion factor, namely λ. The base period is the operational period for the controller periodic task in the corresponding controller node. The base period and dispersion factor are used to obtain the actual sampling periods of the four sensor nodes according to the following equations:

$$\text{sensor1} = \rho(1 + \lambda),$$

$$\text{sensor2} = \rho(1 - \lambda),$$

$$\text{sensor3} = \rho(1 + 1.1\lambda),$$

$$\text{sensor4} = \rho(1 - 1.1\lambda).$$

The value of the dispersion ranges from 0 to 0.20, and it means that, in the tightest case, the dispersion factor is 0, and the four sensors have the same period of the controller. On the other extreme, when λ is 0.20, the sensors have periods of 20% or 22% above or below the base period. A λ-value $>25\%$ causes an unacceptable performance. In the scheduling strategy proposed, RTDS scheduler node allocates a bandwidth share to every node, by means of assigning a time-window when to transmit. This is independent from

the network protocol used, which must be considered solely as the network access controller.

Menéndez-Leonel and Benítez-Pérez [15] provide some tests to quantify the performance of the NCS, based on the IAE as a metric. The base period and dispersion parameters are fixed in specific values. These values are used to evaluate such a performance when all nodes are competing to get the network and bandwidth. According to the network access control algorithm, the system tends to be unstable, since the IAE increases.

Figure 5 shows a typical, stable behaviour of the RTDS, using a base period within the interval [0.015, 0.020] seconds and dispersion λ fixed to 0.15 for all nodes. The simulation is carried out during 50 s.

Notice that, however, when the base period is out of the proposed interval and/or the scheduler does not assign a proper bandwidth, the system easily becomes unstable. Figure 6 shows such unstable behaviour.

The access to network is defined using a scheduling algorithm, executing on the scheduler node as a periodic task. The scheduler generates an id, which each sensor uses to access the network, in order to send data to the controller. Figure 7 shows the communication between sensor and controller nodes, supervised by the scheduler node.

The scheduler node periodically assigns network bandwidth to the sensor nodes. This implies that there is an access network distribution for each sensor. Hence, the scheduler plans the NCS activities. It is also responsible, as a part of the scheduling strategy, to periodically assign network bandwidth to the nodes of the RTDS. Such a periodical assignment, particularly for the sensors, is based on an equal access distribution. This procedure ensures that unique data are transmitted through the network during each period of the scheduler. So, the steps to get access to the network are as follows:

1. Scheduler randomly selects a node identifier (id).
2. Scheduler writes the selected id in the distributed memory.

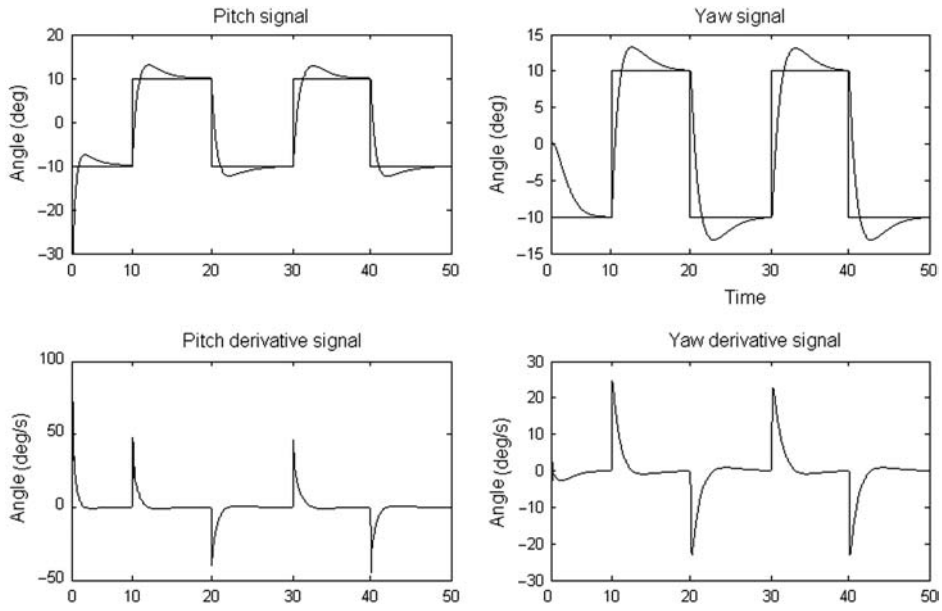

Figure 5. Helicopter stable behaviour.

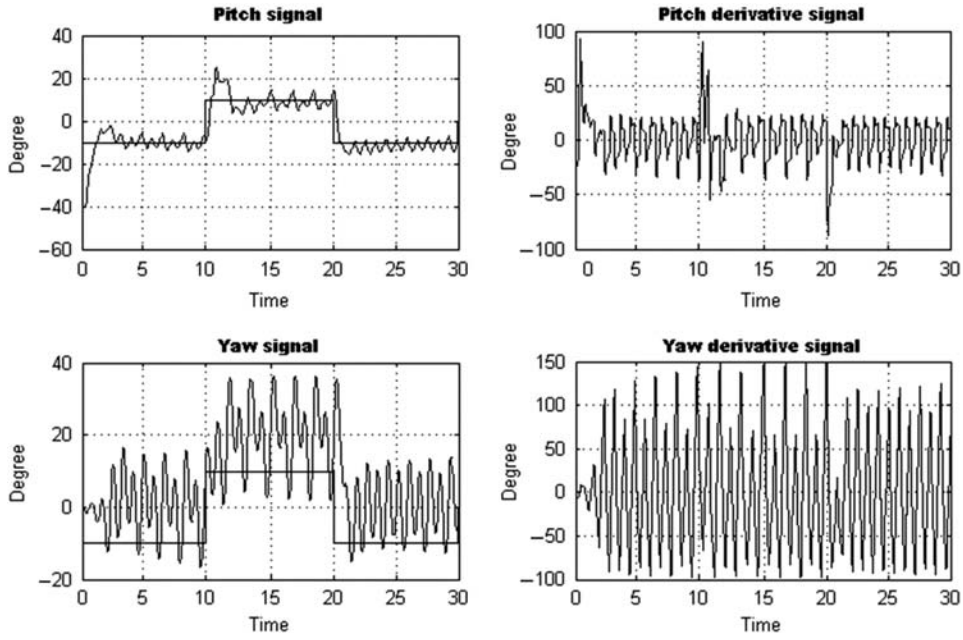

Figure 6. Helicopter unstable behaviour.

3. Sensors and controller nodes read distributed memory during each period of the task.
4. If node id of the sensors and controller is equal to id on distributed memory, then some of these nodes have access to network, else wait for next period to check the distributed memory which has a new id.

These points can be explained as follows. All nodes in the RTDS have an identifier (id). Each node communicates this identifier to other nodes by sending and receiving messages. When the RTDS operation starts, the scheduler node executes a periodic task with period p_s; this node will assign a time window to each node which needs to transmit. Each node i performs different tasks, the main task of each node has period p_i and execution time c_i; this task is divided into segments that perform some actions such as sensing, computation of the control law, reception and data transmission.

Scheduler node selects a particular id using a random probability function and writes it in a shared table which contains identifiers of nodes. After every lapse of time p_s, the id in

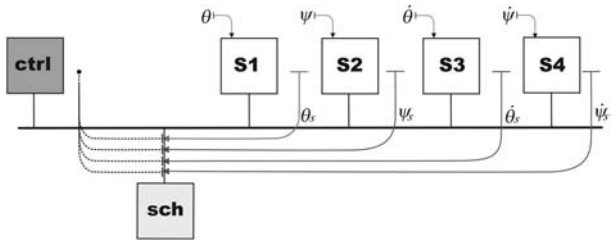

Figure 7. Sensor–controller communication supervised by the scheduler.

the shared table changes and bandwidth is assigned to a particular node to transmit. This process begins when the scheduler places the selected id in the shared memory of the system, and the node that is ready to transmit consults the shared table. If the node identifier is written in the table, the node has access to the network to transmit; otherwise the node waits a period of time p_i and repeats the previous process.

In summary, a node has access to the network only if the scheduler selects the node identifier equal to the id of the node ready to transmit.

Sensor and controller activities are of particular interest, thus the scheduler node assigns greater access to the network at these nodes. The controller gets 1/3 of the bandwidth, while the rest 2/3 is shared equally amongst the sensors. This distribution could be changed using a different random probability function to elect node identifiers.

Figure 8 shows an assignment of network bandwidth to the controller node and the sensor nodes. Notice that the controller node has the highest transmission rate.

In order to show the network activity, Figure 9 exposes the messages sent by the sensors and controller through the network during 1 s.

This procedure is feasible since we assume that the RTDS is a dynamical process in which delays are bounded and time-invariant. Offline some scenarios were studied in order to determine specific parameters of scheduling which do not lead to large delays.

As an important issue, notice that a successful network management is a key point to achieve system schedulability, and thus obtain an acceptable performance. Even though a distributed scheduling does not represent a quite novel approach, it seems necessary to exploit the distributed nature of the architecture, and so, to achieve a sort of consensus between the nodes of the RTDS, thereby efficiently managing the network resources when these are limited. A disadvantage of this proposal is that it is static. This means that real time parameters are computed offline, and there is no mechanism to modify these parameters online. An online modification is desirable, since it seems useful to change parameters during fault scenarios or loss of control performance.

4. Network scheduling based on frequency transmission

This approach modifies the frequency transmission using three parameters: minimum frequency rate (f_m), real frequency rate (f_r) and maximum frequency rate (f_x). The RTDS dynamics, then, is modelled as a linear time-invariant system, whose state variables are

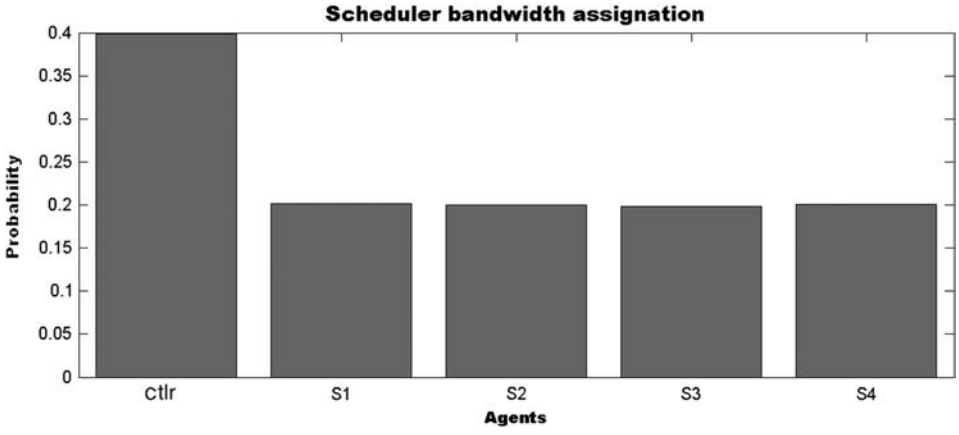

Figure 8. Allocation bandwidth distribution.

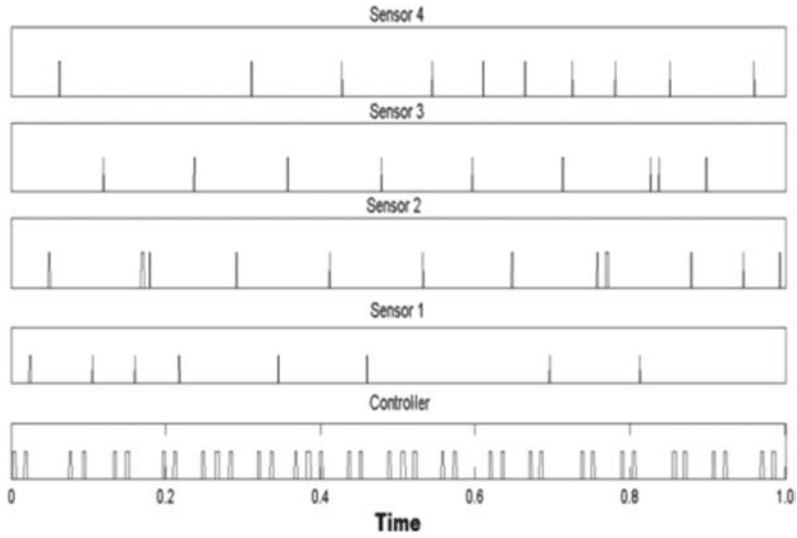

Figure 9. Sensors and controller nodes sending messages through the network.

transmission frequencies ($f_i = 1/p_i$) of the n nodes that compose the RTDS [7]. There is a relationship between the frequency rates of a node and some external input frequency rates, which serve as coefficients of the linear system. So, it is possible to control the NCS using the input vector **u**, such that the output vector **y** contains the frequency rates of a node within a nonlinear region L, bounded by the maximum and minimum transmission frequency rates (Figure 10).

The objective of controlling the frequency rate is to achieve clustering of frequency rates through converging values. For this, each sensor node f_{m} is aware of its minimum and maximum frequency rates (f_{m}, f_{x}), and based upon messages sent to the controller, it could estimate its own real transmission frequency rate (f_{r}). Thus, let a RTDS with k nodes be such that each node performs n tasks, and each task τ_i has a period p_i and a consumption c_i.

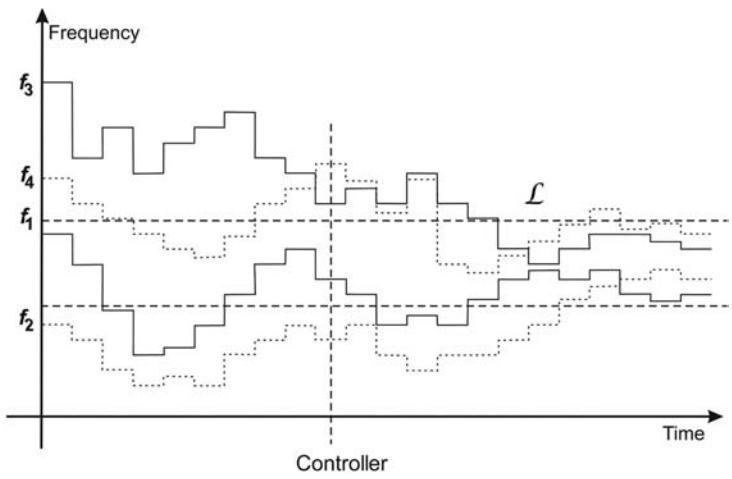

Figure 10. The schedulability region L by modifying the frequency transmission rates.

4.1 Scheduling analysis

Since an optimum fixed-priority scheduler has an upper bound of processor utilisation [14], it is necessary to consider each processor that uses the network in the RTDS:

$$U = \sum_{i=1}^{n} \frac{c_i}{p_i} < 1, \tag{3}$$

where c_i is the task consumption time and p_i is the task period.

For our actual purposes, let us assume that there are k processors, each one having n tasks, and there is a set of possible task configurations that satisfy Equation (3). For example, consider the set H_j^* which contains all task configurations corresponding to those parameters in which all n tasks of each processor j are schedulable:

$$H_j^* = \left\{ H_j^1, H_j^2, \ldots, H_j^s \right\},$$

where each element of the set, $H_j^1, H_j^2, \ldots, H_j^s$, is local schedulable subsets. Thus, a global schedulable set is

$$\bar{H} = \left\{ H_j^* \middle| j = 1, 2, \ldots, k \right\} = \left\{ H_1^*, H_2^*, \ldots, H_k^* \right\},$$

which comprises all k processors of the RTDS, and hence, the overall RTDS is schedulable.

The problem to tackle is to modify the task periods of each sensor node, based on frequency transition model, such that it ensures a local scheduling. Each processor j could change the task configuration through time:

$$H_j' \underset{\varphi_1}{\rightarrow} H_j^* \underset{\varphi_2}{\rightarrow} H_j' \underset{\varphi_3}{\rightarrow} H_j^* \underset{\varphi_4}{\rightarrow} \ldots \underset{\varphi_r}{\rightarrow} H_j^*, \tag{4}$$

where $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_r$ are the r local frequency transitions, and H_j' is the task configuration subset. Even though it may be the case that H_j' is not schedulable, nevertheless, if after all frequency transitions there is a sequence similar to Equation (4), then it converges to a task configuration $\bar{H}$, where the RTDS is globally schedulable. The frequency transition model presented here deals with sensor frequency modifications using an LQR controller, in which a gain K changes any subset of task configurations H_j' into $\bar{H}$.

Figure 11 shows a time diagram, in which at time t_0 several tasks are executed on four processors, and thus having H_j^1, H_j^2, H_j^3 and H_j^4 as task configuration subsets. Three of these subsets are schedulable (the tasks on processors P_1, P_2 and P_4), and one is not schedulable (the tasks on processor P_3). After the first transition (φ_1) at time t_1, one task configuration subset is still not schedulable (the tasks on processors P_1); after the second transition (φ_2), all task configuration subsets are schedulable. If any task configuration subsets remain in $\bar{H}$ after several transitions, the RTDS reaches global scheduling. Here, it is assumed that this fact implies system stability.

The RTDS can also be modelled as a linear time-invariant system, in which the state variables $x_1, x_2, \ldots, x_n$ take the values of the real transmission frequencies. Let us assume that there are several ratios between real transmission frequencies $f_1, f_2, \ldots, f_n$ and the external input frequencies $u_1, u_2, \ldots, u_n$ [7]. These ratios serve as coefficients of the

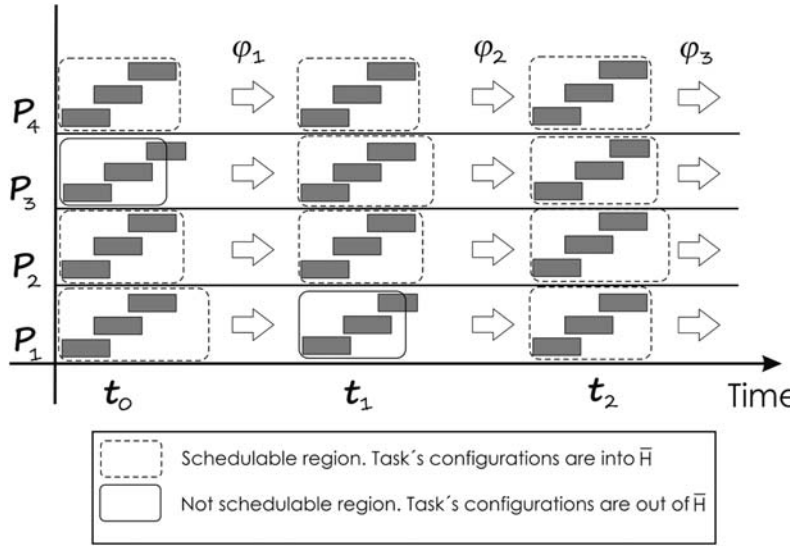

Figure 11. Tasks of schedulable and non-schedulable regions.

matrices **A** and **B** of the discrete time linear system:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k), \end{aligned} \tag{5}$$

where $\mathbf{A} \in \Re^{n\times n}$ is the matrix of ratios between frequencies of all nodes, $\mathbf{B} \in \Re^{n\times n}$ is the scale frequencies matrix, $\mathbf{C} \in \Re^{n\times n}$ is the matrix with ordered frequencies, $\mathbf{x} \in \Re^n$ is the real frequencies vector and $\mathbf{y} \in \Re^n$ is the output frequencies vector. The input $\mathbf{u} \in \Re^n$ is a vector of reference frequencies of the nodes in the RTDS. Let $a_{ij} \in A$ given by a function η of minimal frequencies f_{m} of node i and $b_{ij} \in B$ given by a γ function of maximal frequencies f_{x}, so that

$$a_{ij} = \eta\left(f_{\mathrm{m}}^1, f_{\mathrm{m}}^2, \ldots, f_{\mathrm{m}}^n\right),$$

$$b_{ij} = \gamma\left(f_{\mathrm{x}}^1, f_{\mathrm{x}}^2, \ldots, f_{\mathrm{x}}^n\right).$$

The control input is given by a function of the minimal frequencies and the real frequencies of node i:

$$\mathbf{u} = h(r - x) = K^s(\mathbf{f_m} - \mathbf{f_r}), \tag{6}$$

where $\mathbf{f_m}$ and $\mathbf{f_r}$ are the vectors:

$$\mathbf{f_m} = \left[f_{\mathrm{m}}^1, f_{\mathrm{m}}^2, \ldots, f_{\mathrm{m}}^n\right]',$$

$$\mathbf{f_r} = \left[f_{\mathrm{r}}^1, f_{\mathrm{r}}^2, \ldots, f_{\mathrm{r}}^n\right]'.$$

Substituting Equation (6) in Equation (5):

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{x}(k+1) &= \mathbf{A}\mathbf{f}_{\mathbf{r}}(k) + \mathbf{B}(K^s(\mathbf{f}_{\mathbf{m}}(k) - \mathbf{f}_{\mathrm{m}}(k))), \end{aligned} \tag{7}$$

where K^s is the control gain defined in the LQR algorithm. Matrices **A**, **B** and **C** are dimensionally correct with the schedulability restriction expressed in Equation (3).

Figure 12 shows the data transmission from a sensor node to a scheduler node, sending frequency data through the network. Each time, the scheduler node uses frequency information to obtain a new frequency transmission, aiming for an efficient utilisation of the network.

Previous procedure is based on exploration of minimal and maximal frequencies where the RTDS uses the network avoiding undersampling or overload. The LQR scheme used to control frequencies of transmission requires a proper parameter calibration which involves several tests offline to adjust these frequencies into a schedulability region.

5. Comparative analysis

The scheduling parameters used in allocation bandwidth strategy are base period and dispersion. The analysis of these parameters explores the value of IAE using different values of base period and dispersion; therefore it is possible to find a schedulability region in which the use of the network is balanced. This region bounds the values of scheduling parameters and ensures a level of system performance. Nevertheless, under fault scenarios the system does not adjust these parameters.

Table 2 contains the values of IAE changing base period and dispersion. Figure 13 shows a schedulability region where the amount of IAE is minimal and regions where IAE increased.

Using a base period bounded in the interval $[0.015, 0.020]$ seconds and dispersion fixed in 15% result in minimal IAE, thus the RTDS behaviour is stable with respect to IAE; however, when base period is out of this interval the value of IAE increases, thus the system becomes unstable (Figure 14).

Network scheduling simulations based on frequency transmission were performed for the values of maximum, minimum and real frequencies and execution time as shown in Table 3.

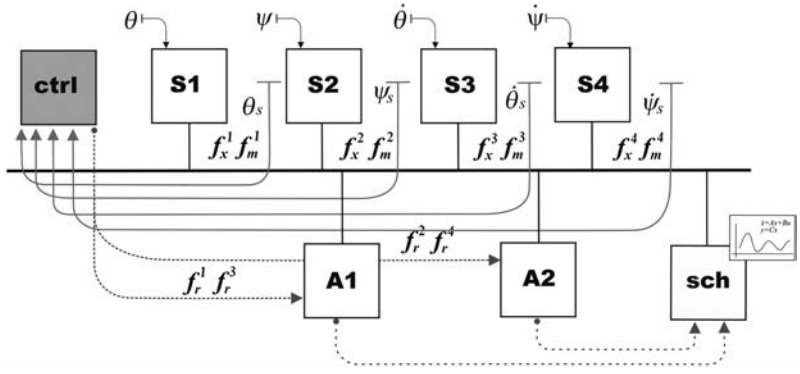

Figure 12. Communication of frequency data from sensor node to scheduler node.

Table 2. IAE values using several base periods and dispersion values.

	Base period (ms)															
Dispersion	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0.0	16.85	15.38	17.58	18.09	10.38	9.96	9.51	9.01	8.98	55.86	13.55	60.55	55.68	10.29	21.86	10.52
0.05	15.45	9.41	9.99	14.67	10.40	9.23	9.41	8.74	8.79	45.41	8.94	65.33	17.24	10.61	24.37	9.89
0.10	13.59	8.59	9.20	11.84	9.73	9.68	8.67	8.85	8.92	28.99	9.18	67.43	14.19	10.73	24.03	11.33
0.15	13.40	9.18	9.90	13.65	9.86	9.60	9.08	8.82	8.69	29.06	9.59	41.15	20.55	10.81	22.95	10.42
0.20	12.43	7.39	9.19	17.97	10.09	9.29	7.89	7.97	7.99	45.05	8.58	26.95	15.19	10.06	23.15	7.80

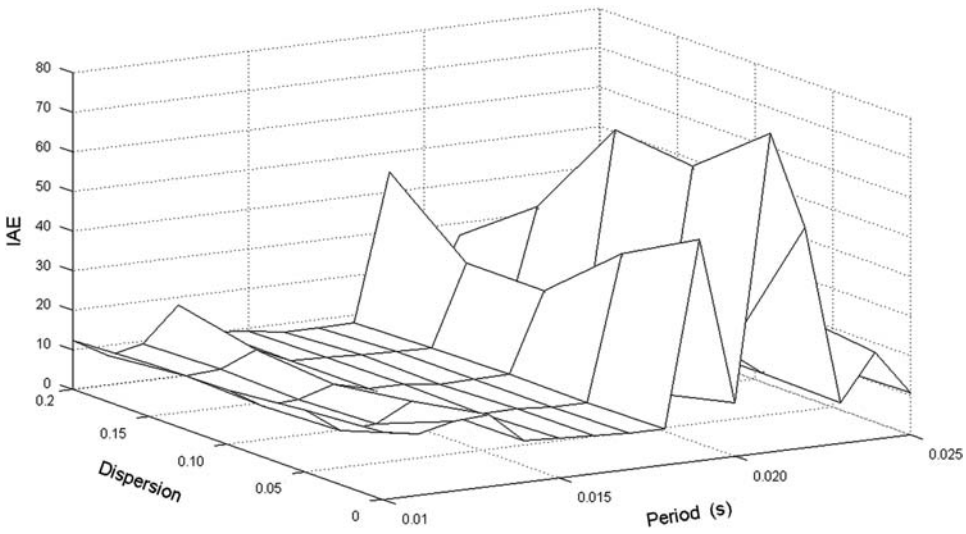

Figure 13. Schedulability regions where the IAE is minimal and maximal using base period and dispersion parameters.

Figure 15 shows frequency dynamics using an LQR controller. Initial frequencies are out of a schedulability region, the controller moves these frequencies online into a region where the system behaviour is stable.

Figure 16 shows a modification on transmission rates using the frequency transmission model. Initially, the task periods of the sensor nodes begin outside the schedulable region.

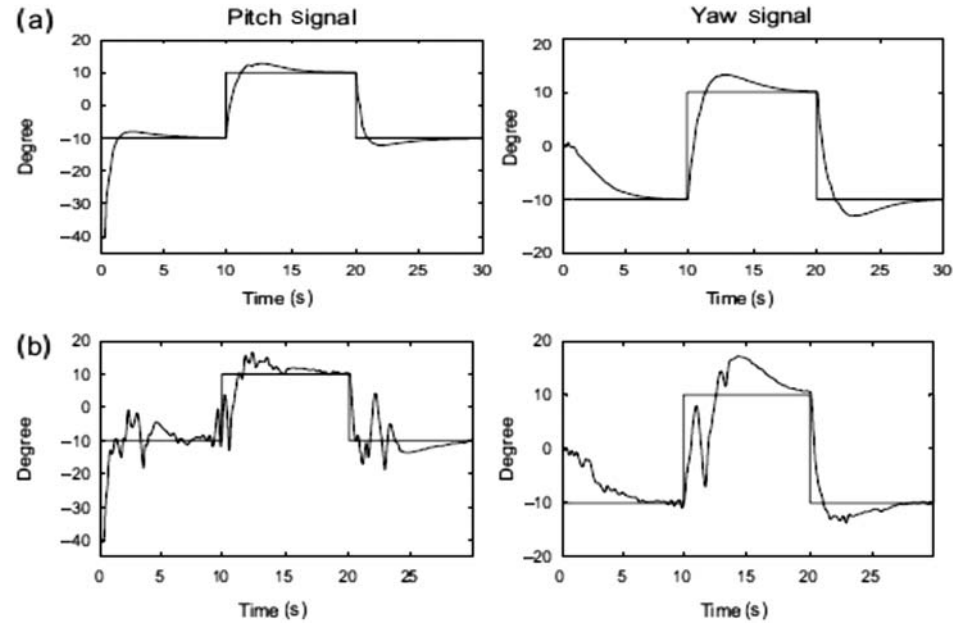

Figure 14. System response with different base periods: (a) base period = 0.017 s; (b) base period = 0.022 s.

Table 3. Initial frequencies and execution time of the sensors in the RTDS.

Node	Minimum frequency	Maximum frequency	Real frequency	Consumption
1	60	310	40	0.001
2	50	270	250	0.001
3	50	270	100	0.001
4	45	300	50	0.001

Frequency transmission rates are modified for two intervals: $[12, 30]$ seconds and $[50, 60]$ seconds. Frequencies inside these intervals cause controlled data transmission rates, thus systems behaviour is stable with respect to IAE.

Figure 17 shows the network activity using the frequency transmission model. Sensors and controllers equally use the network when the frequency transmission rates are controlled into schedulability region.

We consider important to emphasise that the controllability in both strategies mainly depends on time delays. Network scheduling based on allocation bandwidth and network scheduling based on frequency transmissions are the two controllable processes because both comply with controllability conditions: all states are observable and measurable. Thus if time delays are bounded, the RTDS could move into a schedulability region where all system's tasks fulfil its deadlines. Therefore, in a RTDS where, time delays are bounded, the system is still being controllable.

6. Conclusions and future work

This paper shows two scheduling strategies, one performing dynamic scheduling and the other carrying out static scheduling, in order to expose the advantages of using dynamic scheduling in an ad hoc implementation for the network of a NCS. A helicopter model is used as a case study, adding a RTDS that performs flight control functions. Both scheduling strategies are implemented by numerical simulations that modify the sampling periods, and therefore the frequencies of data transmission.

In the first strategy, the RTDS results are schedulable but have a large IAE, which means that the RTDS has an unstable behaviour. Therefore, this strategy is effective only for a particular interval of the sampling periods of the nodes of the RTDS. In particular

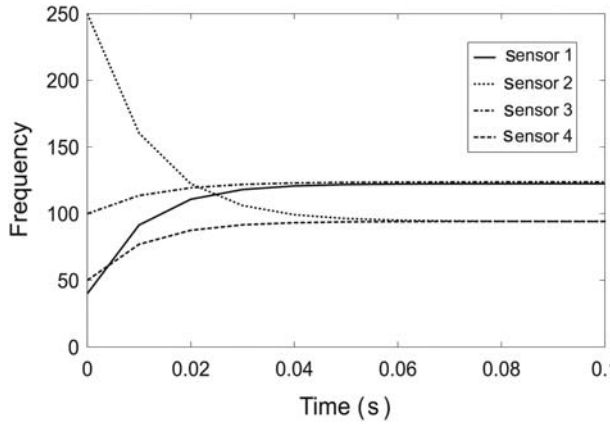

Figure 15. Frequency response controlled by an LQR controller.

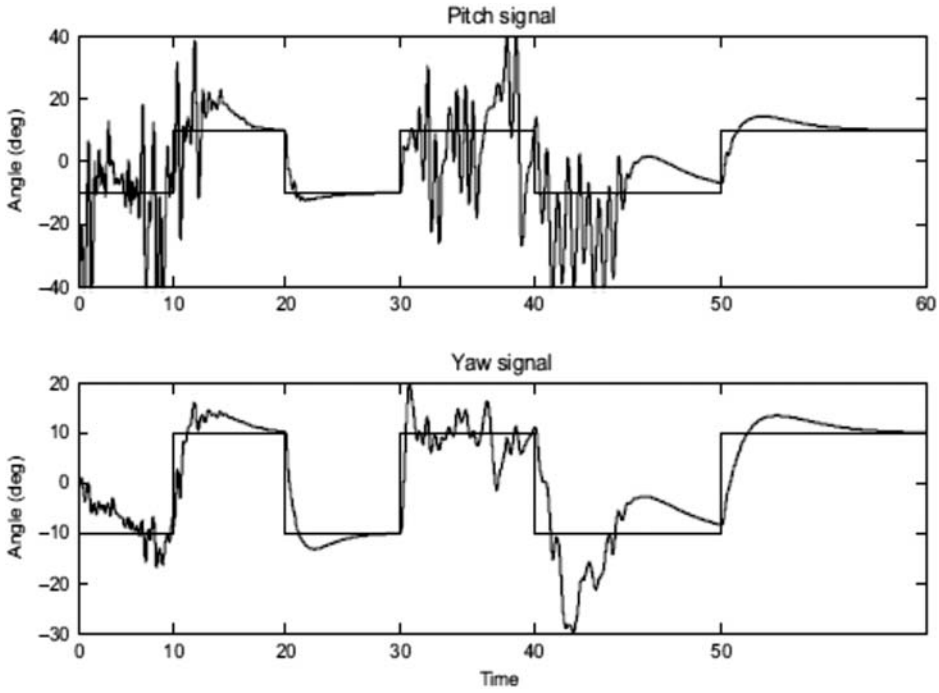

Figure 16. System behaviour using frequency transition model in different time intervals.

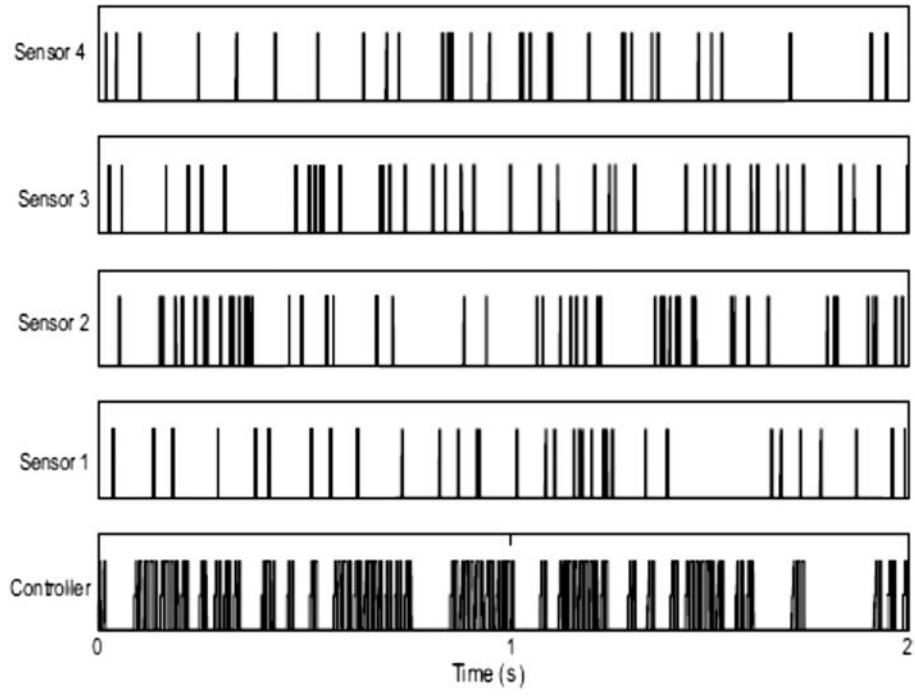

Figure 17. Network activity using frequency transition model from 20 to 30 s.

circumstances, such as fault scenarios that modify the sampling periods, the RTDS is affected, diminishing its performance.

The second strategy dynamically adjusts the frequencies, considering the participation of several nodes of the RTDS. However, this approach is centralised, since a frequency transmission dynamical system is obtained only by the scheduler node. Nevertheless, the dynamical changes of this strategy improve the RTDS response under fault scenarios.

These scheduling approaches show a way to manage the network resources, especially with a limited network bandwidth. These techniques avoid network delays during transmission.

As future work, it is considered that each node may compute the control transmission frequency and thus reach consensus from all involved nodes to globally manage the access to network and hence obtain a distributed scheduling. Another future work refers to exploring the effect of increasing the network load to calculate an accurate frequency transmission.

Acknowledgement

The financial support given by Grants PAPIIT 103310, PICCO-ICYTDF 10-53 and CONACYT (Ph.D. Grant) is specially appreciated.

Note

1. A real-time simulation tool based on Matlab and Simulink.

References

[1] K.E. Årzén, B. Bernhardsson, J. Eker, A. Cervin, K. Nilsson, P. Persson, and L. Sha, *Integrated control and scheduling*, Tech. Rep. ISRN LUTFD2/TFRT7586SE. Lund Institute of Technology, Sweden 1999.
[2] H. Benítez-Pérez and F. García-Nocetti, *Reconfigurable Distributed Control*, Springer, Berlin, Germany, 2005.
[3] M. Branicky, V. Liberatore, and S. Phillips, *Networked control system co-simulation for co-design*, Proc. Am. Control Conf. 4 (2003), pp. 3341–3346.
[4] D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Ya Xu, and Haobo Yu. *Advances in network simulation*, Computer 33(5) (2000), pp. 59–68.
[5] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K. Arzen, *How does control timing affect performance?* Control Syst. Mag. 23(3) (2003), pp. 16–30.
[6] A. Cervin, M. Ohlin, and D. Henriksson, *Simulation of networked control systems using truetime*, in Proceedings of the 3rd International Workshop on Networked Control Systems: Tolerant to Faults. Nancy, France, 2007.
[7] O. Esquivel-Flores, H. Benítez-Pérez, E. Méndez, and A. Menéndez, *Frequency Transition for scheduling management using dynamic system approximation for a kind of NCS*, ICIC Express Lett. B Appl. 1(1) (2010), pp. 93–98.
[8] Y. Halevy and A. Ray, *Integrated communication and control Systems: Part I – analysis*, J. Dyn. Syst. Meas. Control 110 (1988), pp. 367–373.
[9] G. Heredia, A. Ollero, M. Bejar, and R. Mahtani, *Sensor and actuator fault detection in small autonomous helicopters*, Mechatronics 8 (2008), pp. 90–99.
[10] F. Lian, J. Moyne, and D. Tilbury, *Time delay modeling and sample time selection for networked control systems*, in *Proceedings of ASME-DSC*, Vol. XX, DCS, New York, 2001, pp. 5, 6.
[11] F. Lian, J. Moyne, and D. Tilbury, *Network design considerations for distributed networked for distributed control systems*, IEEE Trans. Control Syst. Technol. 10(2) (2002), pp. 297–307.
[12] F. Lian, J. Moyne, and D. Tilbury, *Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems*, IEEE Trans. Ind. Inf. 2(1) (2006), pp. 18–27.

[13] F. Lian, J. Yook, P. Otanez, D. Tilbury, and J. Moyne, *Design of sampling and transmission rates for achieving control and communication performance in networked multi-agent system*, Proc. Am. Control Conf. vol. 4, (2003), pp. 3329–3334.
[14] C. Liu and J. Layland, *Scheduling algorithms for multiprogramming in a hard real-time environment*, J. Assoc. Comput. Mach. 20(1) (1973), pp. 46–61.
[15] A. Menéndez-Leonel and H. Benítez-Pérez, *Scheduling strategy for real-time distributed systems*, J. Appl. Res. Technol. 8(2) (2010), pp. 177–185.
[16] M. Ohlin, D. Henriksson, and A. Cervin, *TrueTime 1.5 Reference Manual*, Department of Automatic Control, Lund University, 2007.
[17] C. Peng, D. Yue, Z. Gu, and F. Xia, *Sampling period scheduling of networked control systems with multiple-control loops*, Math. Comput. Simul. 79 (2009), pp. 1502–1511.
[18] Quanser, *2-DOF Helicopter, User and Control Manual*, Quanser, Speciality Experiments: 2-DOF Helicopter, 2007.
[19] F. Xia and Y. Sun, *Control-scheduling codesign: A perspective on integrating control and computing*, Dyn. Contin. Discrete Impulsive Syst. Ser. B 13(S1) (2006), pp. 1352–1358.
[20] H. Yan, J. Wan, D.L. Yuqing, and P. Zhang, *Codesign of networked control systems: A review from different perspectives*, IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, 2011.
[21] M. Yu, L. Wang, T. Chu, and G. Xie, *Stabilization of networked control systems with data packet dropout and network delays via Schitching system approach*, 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, 2004, pp. 14–17.