

Document downloaded from:

<http://hdl.handle.net/10251/38306>

This paper must be cited as:

Salt Llobregat, JJ.; Casanova Calvo, V.; Cuenca Lacruz, ÁM.; Pizá Fernández, R. (2014). Multirate control with incomplete information over Profibus-DP network. *International Journal of Systems Science*. 45(7):1589-1605. doi:10.1080/00207721.2013.844286.



The final publication is available at

[http://www.tandfonline.com/doi/abs/10.1080/00207721.2013.844286#.U6ldLZR\\_tPN](http://www.tandfonline.com/doi/abs/10.1080/00207721.2013.844286#.U6ldLZR_tPN)

Copyright Taylor & Francis

## RESEARCH ARTICLE

### *Multirate Control with incomplete information over Profibus-DP Network*

J. Salt\*, V. Casanova, A. Cuenca, R. Pizá

*Departamento de Ingeniería de Sistemas y Automática, Instituto de Automática e Informática Industrial. Universitat Politècnica de Valencia, Spain*

(May 2013)

When a Profibus-DP network is used in an industrial environment, a deterministic behaviour is usually claimed. But actually, due to some concerns such as bandwidth limitations, lack of synchronization among different clocks, and existence of time-varying delays, a more complex problem must be faced. This problem implies the transmission of irregular and, even, random sequences of incomplete information. The main consequence of this issue is the appearance of different sampling periods at the different network devices. In this paper, this aspect is checked by means of a detailed Profibus-DP time-scale study. In addition, in order to deal with the different periods, a delay-dependent dual-rate PID control is introduced. Stability for the proposed control system is analysed in terms of Linear Matrix Inequalities.

**Keywords:** Multirate Control Systems; Network Delay; Networked Control System; PID Controller; Stability Analysis; Profibus-DP

## 1. Introduction

Networked Control Systems (NCS) have raised a growing interest in the control community some years ago. The term NCS refers to a particular kind of control systems in which the main feedback loop is closed over a shared communication medium. Figure 1 shows a representation of a generic NCS. As it can be seen, the communication between controller and controlled plant in the control loop is performed through a link, which is also used by other devices belonging to different control loops or applications. NCS are gaining a significant presence in industrial environments. The usually large distances between main controller and processes to be controlled in some industrial plants, and the weight and space limitations in some systems like airplanes and other vehicles, suggest using an NCS because of wiring reduction, easier and cheaper maintenance, and cost optimization. The reader can see references such as Liou and Ray (1991), Schickhuber and McCarthy (1997), Bauer et al. (1999), Lian et al. (2001, 2002), Chow and Tipsuwan (2001) for interesting descriptions of NCS from the seminal works Halevi and Ray (1988), Ray and Halevi (1988), Ray (1989). Some previous authors' works referred to this kind of systems can be found in Casanova and Salt (2003), Salt et al. (2006), Cuenca et al. (2011, 2012), Piza et al. (2013). This non-exclusive availability of the link to perform the communication between control and plant causes a number of drawbacks including:

- (1) Bandwidth limitation: The shared medium is not able to transmit the amount of information that would be desirable from the point of view of the control specifications.

---

\*Corresponding author. Email: julian@isa.upv.es

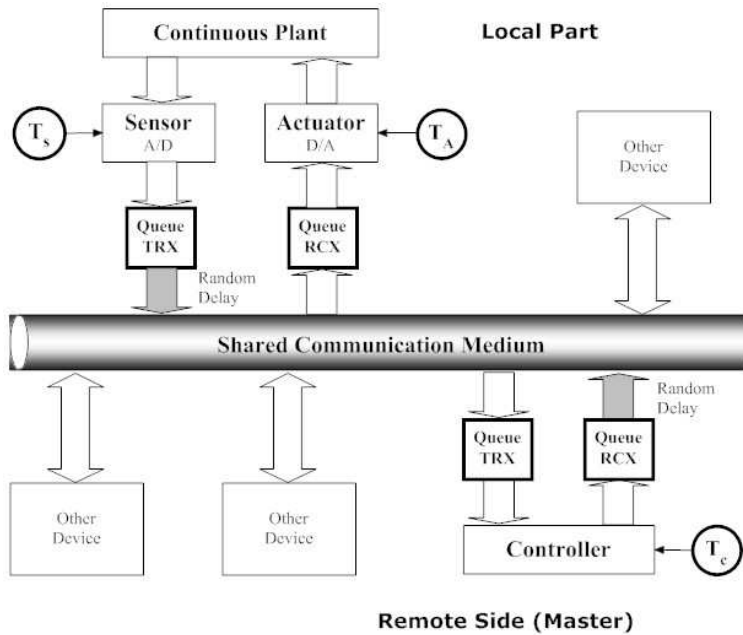


Figure 1. Networked Control System.

- (2) Time-varying delays: The lack of regularity involved in the communication process causes the presence of non-constant (and even random) delays when transmitting information between control and plant. As any other delay, these ones are a potential source of instability, more difficult to avoid due to the irregularity.
- (3) Lack of synchronization: In a conventional control loop every control action is applied during the same time period. Also, the time between two consecutive samples is constant. This is far from true when the loop is closed through a shared link. In fact there are different time-scales with different clocks and consequently with different phases among these time-sequences at different points of the network. The problem of synchronization errors and their effects on NCS is addressed for instance in Lorand and Bauer (2006) or in Lin et al. (2013).

If a Direct Control topology Tipsuwan and Chow (2003) is considered (see Figure 1), different sampling frequencies in different parts of the NCS can appear. So, in general, a multirate system with skewed sequences (due to the offsets of the time-scales) can be assumed.

The goal of this work is to consider a specific Profibus-DP based NCS. The special features of this field bus justify the different time sequences and lead to a Random Incomplete Information (RII)<sup>1</sup> challenge problem. The control law proposal is developed from a gain scheduling multirate control strategy Sala et al. (2009) properly adapted to this advanced specific case. Other works related to this field can be found in Xie et al. (2013), Khan et al. (2012), Peñarrocha et al. (2012), Cuenca et al. (2007), Lall and Dullerud (2001), Salt and Albertos (2005), Salt et al. (2011), Cuenca and Salt (2012).

After this introduction, the paper will provide a brief description of the Profibus-DP operation mode in section 2. Section 3 is devoted to expose the problem statement and to describe a previous analysis of data with the proposed set-up. A simulation model and some results that enable to understand the true dimensions of the problem are presented as well. Real and simulated data will be compared as well in this section. Section 4 introduces the multirate control design established for this situation assuming the time-varying transmission delays. In addition,

<sup>1</sup>Actually, an irregular sequence with random delays is found.

a stability analysis of this kind of Linear Time Variant (LTV) systems is developed by means of Linear Matrix Inequalities (LMIs). Some simulation results from experimental data obtained in the laboratory with a real equipment will be showed in section 5. Conclusions and description of future work will conclude the paper.

## 2. Profibus DP Performance

### 2.1. The fieldbus Profibus-DP

Profibus is an open fieldbus that accomplishes the EN50170 European standard Specification (1998). There are three different versions of the communication protocol: FMS, DP and PA. Profibus-DP is designed for communication between programmable controllers and input/output devices, distributed in a network at a very high baud rate. The devices in a Profibus-DP NCS can be masters (which control the communication in the fieldbus) or slaves (which do not have right to access to the fieldbus and only can send messages to its master device by demand). The most used transmission medium is based on the EIA RS-485 standard. The baud rate is configurable, and the maximum baud rate depends on the length of the segment. Profibus uses a hybrid system to control the access to the bus: the token ring procedure and the master-slave procedure. When one master gets the token then it runs a slaves polling list. The master devices connected to the Profibus-DP network have a direction and form a logical ring. The token represents the right to use the fieldbus, and it is sent between the masters using a special frame. The parameter  $T_{TR}$  sets the maximum token rotation time along the ring. When a master device receives the token, it sends messages to its slaves and then they answer to their master. A deeper explanation of Profibus can be found in Weigmann and Kilian (2000), Vitturi (2004), Yang (2006) and a practical introduction in Lee et al. (2003), Chang Lee et al. (2004)

### 2.2. Performance description

In a typical NCS the information is sent from the controller to the plant (control actions) and from the plant to the controller (samples of the controlled variable) using a shared resource. The control algorithm is implemented in a device which can compute the control actions, using the information measured by a slave device (A/D conversion) and the slave applies the control actions to the continuous plant (D/A conversion). In figure 2 a structure of a control loop formed by a master and a slave is shown<sup>1</sup>. Master and slave devices are integrated in a Profibus-DP, becoming a real NCS. The control task is executed in the master, with a sample period  $T_c$ . The controller needs the measures of the controlled variable and they are available in its input buffer. The computed control actions are written in its output buffer. The slave applies the control actions to the plant and measures the controlled variable for the control algorithm. The actions are available in the slave input buffer and the slave needs a D/A converter to apply them to the plant with a period  $T_A$ . The controlled variable is measured by an A/D converter, with sample period  $T_s$ , and the samples are written in the slave output buffer. After being transmitted the samples are available in the master input buffer to be used in the control algorithm. A/D and D/A conversions can be periodic or event-based tasks and use slave input/output buffers. Communication task is periodic and transfers samples and actions between master and slaves. Profibus-DP transmits the information through the fieldbus between the master buffers and the slave buffers, and it does not depend on the rest of the tasks. However, the fieldbus is used by another devices distributed in the network. The following subjects must be considered:

- The capacity of the input/output buffers is one.

---

<sup>1</sup>Although an NCS with sensor and actuator usually share one side, in general they will be separated.

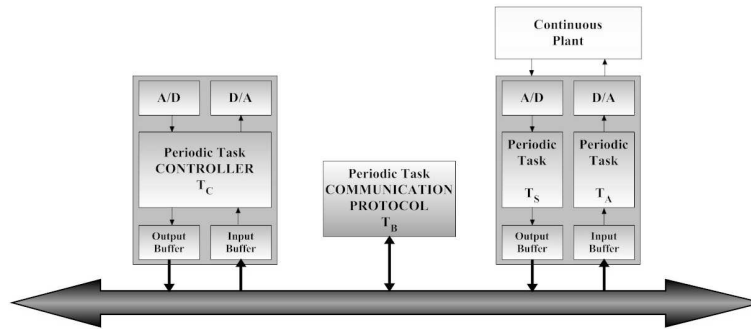


Figure 2. Structure of a control loop in an NCS.

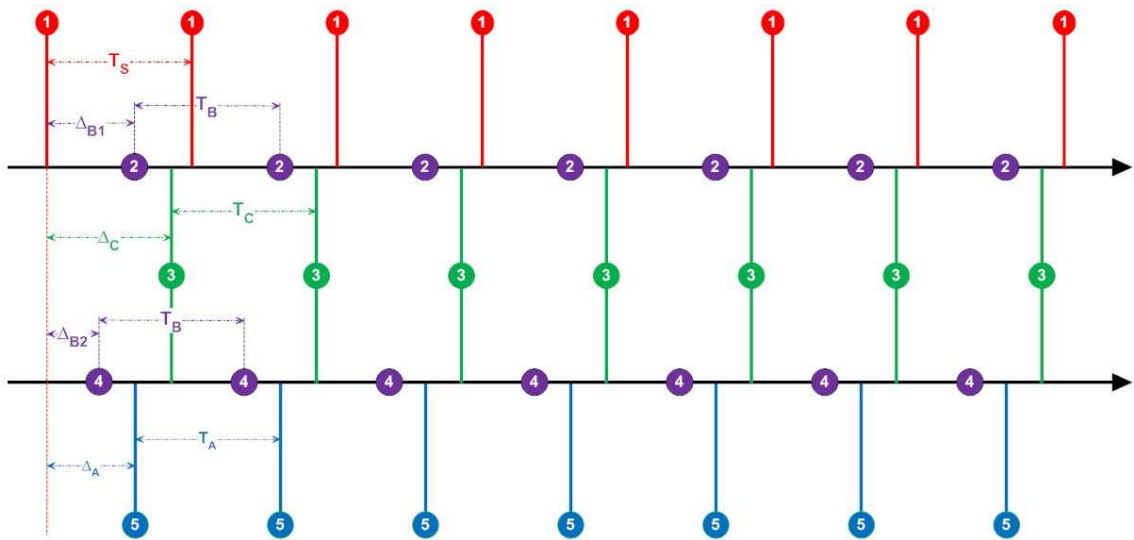


Figure 3. Profibus-DP performance.

- The communication protocol is a periodic task and the period is  $T_{DP}$ . This period is related with the token rotation time along the logical ring,  $T_{TR}$ .

After analyzing some real implementations, it can be checked that  $T_{DP}$  is smaller than  $T_{TR}$ . The value  $T_{TR}$  is a parameter given by the system but the measurement of  $T_{DP}$  has to be done experimentally. Both parameters depend on:

- The number of devices distributed in the network.
- The amount of information to be sent by the devices.
- The baud rate which depends on the length of the bus segment.

Therefore, if the number of devices distributed in the network is increased or the baud rate is decreased (due to the length of the segment),  $T_{DP}$  and  $T_{TR}$  will be increased as the devices need more time to perform a communication cycle.

### 2.3. Parameters and Time Scales

A number of parameters were defined to determine the performance in a Profibus-DP network. If a single-master installation is considered, it can be assumed just one time-parameter in the sense of introduced  $T_{DP}$  and  $T_{TR}$ . In this case the master gets the token and the period is used for the transmission of information and some maintenance operations. This period is going to

denote like  $T_B$  and it is fixed by the application when the slave list and pooling is known. There are two modes of operation: constant and non-constant bus cycle. The first of them is more conservative (and longer) but allows to assure an equidistant cycle. Taking into account these circumstances and the other parameters above introduced, a graphical representation of the different time-scale performances is depicted in figure 3. Specifically, the explanation of these parameters is as follows:

- $T_c$ : The control clock determines the control instants in which a new action is calculated. It is the time between two control instants assuming a time-driven application at remote side.
- $T_s$ : The sensor clock determines the sampling instants in which the controlled variable is captured.
- $T_A$ : The actuator clock determines the application instants in which the action is applied to the plant.
- $T_B$ : The communication clock determines the beginning of a master-slave transmission cycle. (In a multimaster system it will be  $T_{DP}$ , that is, the time one master needs to explore its slaves).
- $\Delta_s$ : The time offset of the sensor sampling sequence with respect to some time reference. In this study, the sensor sampling is considered as the time reference, that is,  $\Delta_s = 0$ . So every sampling sequence will be skewed with respect to this one.
- $\Delta_B$ : The time offset of the bus sampling sequence. That is, the cycle of the bus will present a different phase with respect to different clocks in the NCS.
- $\Delta_c$ : The time offset of the controller sampling sequence. The sensor clock and the control clock are unsynchronized because they work independently.
- $\Delta_A$ : The time offset of the actuator sampling sequence.
- $\Delta_{B_1}$ : The time offset of the sequence representing the instants when the bus picks up the information from the slave output buffer (sensor).
- $\Delta_{B_2}$ : The time offset of the sequence representing the instants when the bus picks up the information from the master output buffer (control action).
- $d_c$ : The controller computation time at the master device.
- $d_s$ : The A/D conversion time.
- $d_A$ : The D/A conversion time.
- $d_{B_1}$ : The transmission time from slave to master. After having performed the master to slave communication, the slave answers to the master. This parameter is the time needed for the slave information to reach the master buffers.
- $d_{B_2}$ : The transmission time from master to slave. When a new transmission cycle is started, the master sends information to the slaves. This parameter is the time for the messages to arrive at the slave.

Usually the parameters denoted by  $d_*$  are negligible with respect to the sampling times.

Using the numbers in figure 3, it is possible to analyze the steps followed by a signal from sensor to actuator (round-trip). These steps imply the measurement sampling, the sample transmission through the fieldbus (from slave to master, and viceversa), and the sample processing by the controller in the master device. In more detail, in instant 1 the sensing starts. It finishes after  $d_s$ , and then the sample is written in the slave output buffer. When the master gets the right to use the fieldbus and the slave (the sensor) is chosen to transmit its data, the sample is collected by the bus in 2 and left in the master input buffer after  $d_{B_1}$ . In 3 the sample is picked up by the controller, and after a computation time  $d_c$  the sample is ready in the master output buffer. In 4 (the same bus cycle than in instant 2), the information is transmitted to the slave<sup>1</sup> input buffer arriving after  $d_{B_2}$ . From this buffer the sample will be collected in the sampling period 5

---

<sup>1</sup>This slave could be the same one previously used, or another one.

of the actuator.

### 3. Problem Statement and previous analysis of data

#### 3.1. Problem Statement

Straight afterwards introducing the network used in this application, the problem to be dealt with will be described. The initial scenario is a classical digital control of a continuous process (that is, a single-rate control, with no phase difference among time-scales), being  $T_s$  the sampling period, which is small enough to reach the desired behaviour. When a shared network is assumed for the communication between controller and controlled process (see Figure 1) some new restrictions must be observed. If the network is Profibus DP, some specific performance must be taken into account (see Figure 2). In the considered case, a new problem appears due to the installation configuration; as a consequence of the available bandwidth of the shared resource, the period  $T_B$  of the imposed bus cycle is such that  $T_B > T_s$ . So, there will be some sampled measurement losses (this problem will be later justified).

In addition, it is also necessary to choose the master sampling period  $T_c$ . A logical selection is to select  $T_c > T_B$  in order to assure that every computed control action or group of control actions arrive to the local (slave) part. So, although some samples of the controlled variable are lost because of the limited bandwidth, every control action will be injected.

One of the operation modes in Profibus DP uses a constant bus cycle. Even though a constant and known  $T_B$  can be guaranteed in this condition, a new problem must be considered; as the clocks that set the sample, communication and control instants (i.e.  $T_s$ ,  $T_B$  and  $T_c$ ) are physically implemented in different devices, there is a phase shift in the time scales. The deeper problem is that the  $T_B$  phase shift is never known in advance. When the application is started, this variable takes a constant but unknown value. Obviously, the local and remote clocks (i.e.  $T_s$  and  $T_c$ ) will also present different phases but it is possible to know them in advance. As commented, until the application is not started, the skew of the bus cycle is unknown. So, an on line data analysis is needed to identify the patterns for control purposes. This computation is necessary in order to know in advance when the next measure of the controlled variable will be sensed (the time interval between two measures of this nature -effective samples- is denoted like  $N$ ). In this way, the round-trip will be completed, and a correct group of control actions will be provided. Then, the key concept is the bus period (constant in our specific case) and its skew (time-scale phase). Actually, this last variable is critical, since although the bus period can be determined by the topology when programming the network configuration, the skew is unknown until the application is started. The effective sample will arrive with a certain delay  $\tau$  to complete the round trip. Obviously, the asynchronous pattern  $(N, \tau)$  is fully dependent of the periods and their respective time-scale phases.

Regarding the control design, a dual-rate controller is a natural option. Initially, a control problem that performs in single rate ( $T_s$ )<sup>1</sup> is assumed. If the network is introduced, and assuming a configuration with constant bus cycle ( $T_B$ ), the control system could become unstable if some  $N$  succession appears (say, the worst case). In order to avoid this possible worst case, a larger sampling period could be used, but then it could be unacceptably slow for controlling the original plant. So, a dual-rate controller is proposed in order to guarantee the control signal injection every  $T_s$ , but with measurements every  $N \times T_s$ . The main aim of this control strategy is to approximately achieve the same original single-rate control behaviour despite having  $N$  times less measurements. In order to assure control system stability, a proper analysis in terms of LMIs will be exposed.

---

<sup>1</sup>this period will be denoted later simply as T

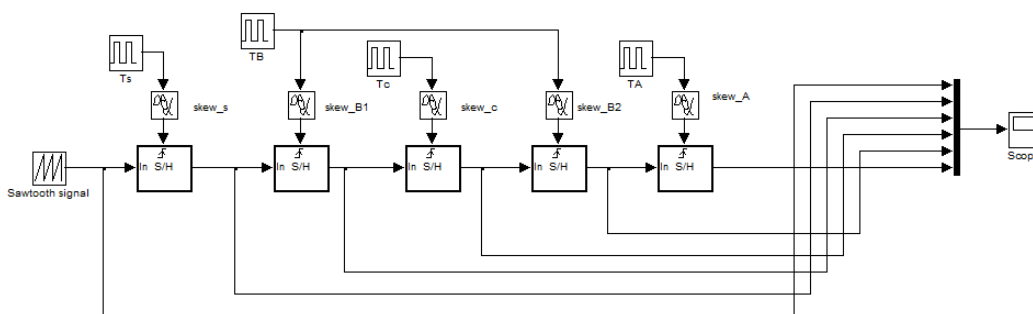


Figure 4. Simulation model

### 3.2. Network-control Setting

First of all, it is necessary to clarify the network set up this work deals with. The setup considers a time-triggered sensor ( $T_s$ ) and an event-based actuator<sup>2</sup>, being both placed at the local side and sharing a local clock (In this case,  $T_A$  is set tending to zero, hence the sample is immediately injected to the process when arriving to the actuator-slave input buffer). A time-stamping procedure, like in Moayedi et al. (2011), enables to measure the round-trip time delays. In this specific case, every offset of the time-scales is computed assuming  $T_s$  as the time reference, that is, the rest of time-scales will be skewed with respect to this sampling scheme (See Figure 3 or Figure 7 for instance). A dual-rate controller, Sala et al. (2009), Salt and Albertos (2005), Salt et al. (2011) with input each  $T_c = NT_s$  and output  $T_s$  is placed at the remote side.

### 3.3. Simulation Model

In order to evaluate different control structures explicitly designed to solve the problems involved in the selected NCS, the next simulation model will be useful. This model has been developed to imitate the behaviour and different events applied to a continuous signal, transmitted through Profibus-DP in a round trip communication. It must include the clock signals and time offsets defined in the previous section. In summary, the model generates the next sequence of events applied over a continuous signal: the signal is captured by a sensor slave with A/D converter, sent through Profibus-DP to a master device, processed by a discrete transfer function in this master device, sent to an actuator slave with D/A converter and, finally, converted into a continuous signal. The communication process becomes itself an irregular discretization, and this is the operation that it is being simulated.

Matlab/Simulink is frequently used in many academic and research fields. It has become a "de facto" standard in the automatic control knowledge area, Bucher and Balemi (2006), Sánchez et al. (2005). This is the tool that has been chosen to develop the proposed simulation model of Profibus-DP showed in Figure 4. The main idea is to use 'Sample & Hold' blocks to simulate the beginning and end of each task in the sequence of operations. A sawtooth signal was used as an input in order to validate this application. A specific case enables to validate this simulation model: a configuration with parameters  $T_c = T_a = T_s = 50 \text{ ms}$  and  $T_B = 130 \text{ ms}$  with some

<sup>2</sup>Actually, this is a usual configuration in this kind of installations.



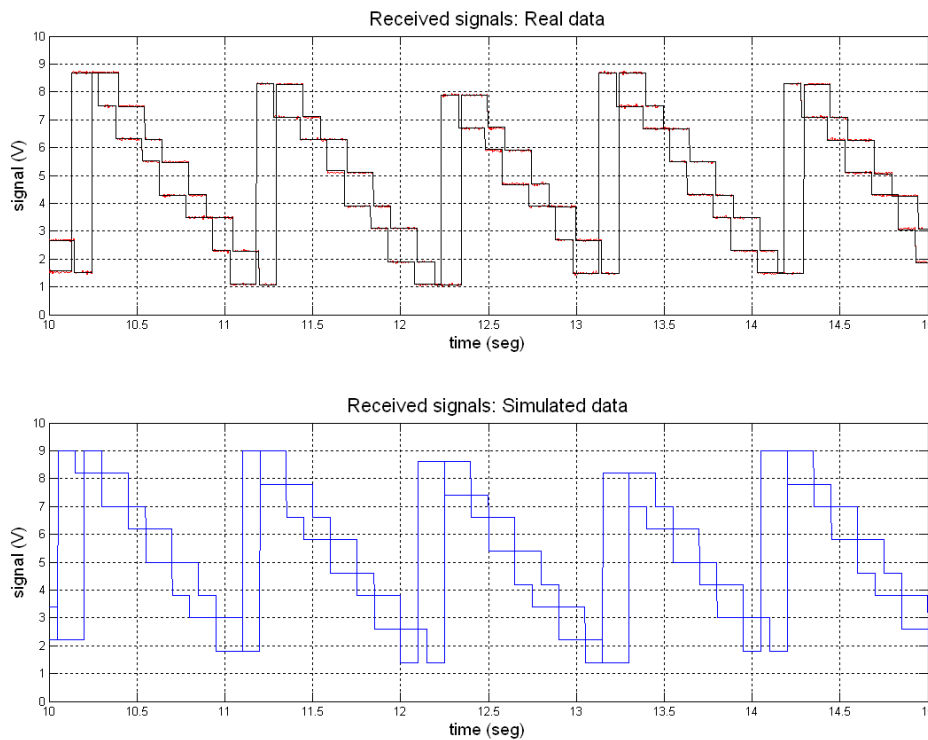


Figure 5. Data obtained

offsets. Figures 5 and 6 show the comparison between real and simulated data. A continuous sawtooth has been transmitted through a real Profibus-DP installation. Upper graphics in figure 5 are the discrete signals received in the controller (remote side) and in the actuator (local side). Lower graphics are the corresponding signals transmitted through the simulation model. As it can be seen the travel through the bus (real or simulated) becomes an irregular sampling of the sawtooth signal. The length of the steps depends on the combination of periods and time offsets in this particular configuration. The sequence of these steps is not the same in real and simulation data because they depend on the particular values of the clock (time offsets) that can be arbitrarily chosen in the simulation model but not on the real bus. Nevertheless, the values of the step lengths and its frequency depend on the periods that can be set in both real and simulated systems. This can be shown by the histogram of the step lengths in figure 6. The position of each bin in the histograms depends on the length of the step and its height defines the number of times that this particular length appears in the whole signal. Real and simulated histograms show the same information proving the accuracy of the simulation model and it can be used in the development of an appropriated control structure to be used in these conditions.

### 3.4. Previous results

First of all, the context must be explained. As it was said, a time-triggered operation mode for sensor and controller, and an event-driven one for actuator was assumed. In this kind of Profibus-DP schemes the sensor is fixed rate operating. The bus rate sets the use of the bandwidth, being the bus clock independent of the rest of clocks. In order to save controller sampling losses, the controller sampling rate will be  $n$  times slower than the bus rate (being  $n$  an integer number). Remember, sensor and actuator share the same clock at the local side, but the controller is

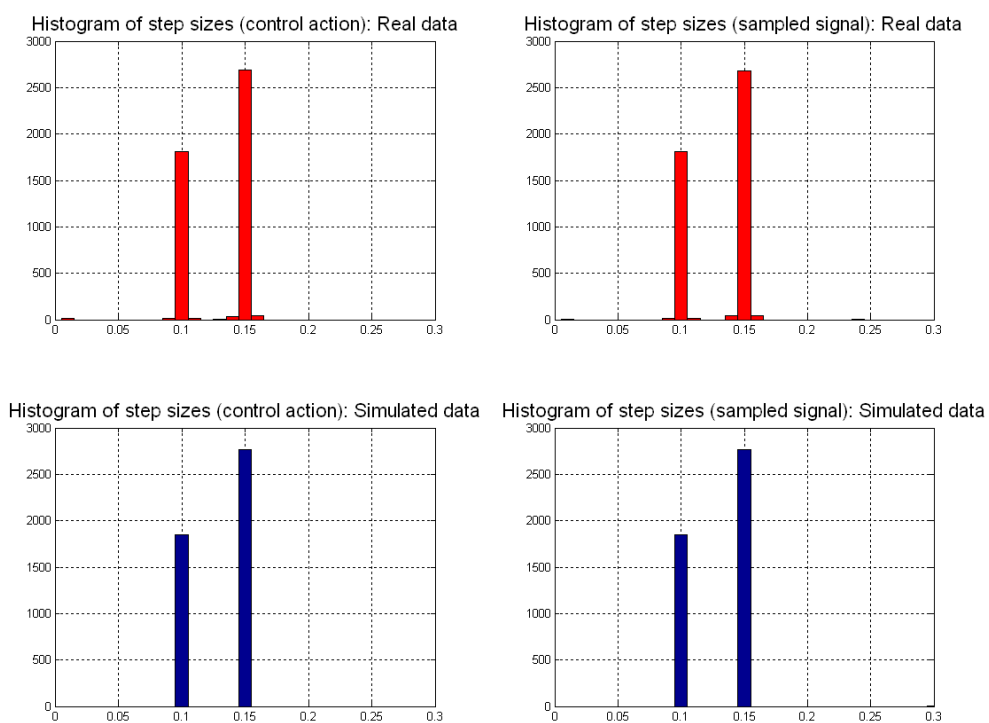


Figure 6. Histograms comparison

located at the remote side with a different clock. In order to synchronize these clocks some procedures can be used, but they usually imply higher network traffic.

Next, some experience with time scales is introduced. Some authors claim about the deterministic behaviour of Profibus-DP, Lee et al. (2002), Alves and Tovar (2007). However, as the experience will show, even with a fixed bus period, an irregular time scheme appears (incomplete measure information and time-varying transmission delay). In fact, some specific conditions will appear when considering the sensor time-scale as the time reference to compute time-offsets, and the following values:

- $T_s = 0.1$
- $T_B = 0.13$  with a time-offset of 20%
- $T_c = 0.3$  with a time-offset of 80%
- The event-driven actuator sampling rate is tending to infinite (that is, practically, a negligible sampling period).

These magnitudes have been considered because  $T_B$  is fixed by the Profibus-DP application when the number of slaves and the distance among different slaves and master are defined (a constant cycle mode was used). When this value is known, a good choice is a  $T_c$  greater than  $T_B$  in order to guarantee that the sampling measure got by the controller is the most recent sampling at the sensor side. With this condition the master is less time busy (computational cost) as well. The results depicted in the figure 7 were obtained using the simulation model. As it is usual in practical installations, the sensor is sampling every  $T_s$ , but due to appearing different time-scales and offsets among bus, controller and actuator, some samples are lost. In figure 7 the marks 'x' shows the lost measures as a result of not being picked up by the bus. In this case, the bus will collect a new sampling when accessing the sensor output buffer. The marks '\*' represents

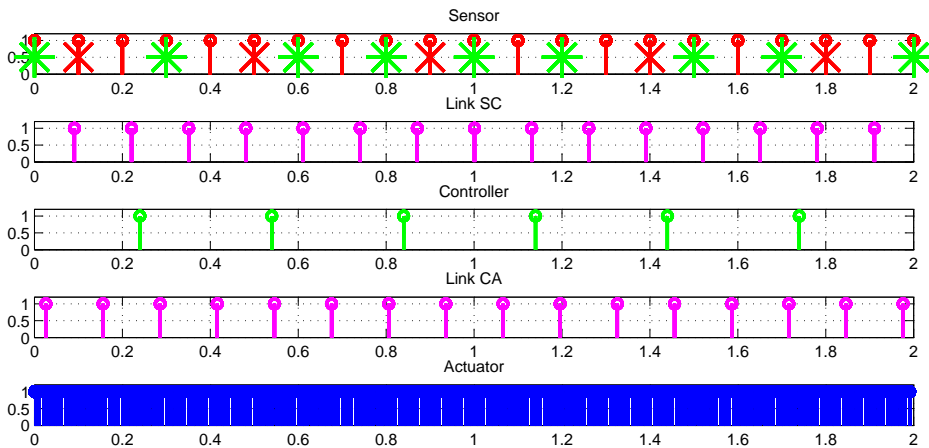


Figure 7. Time schedules.

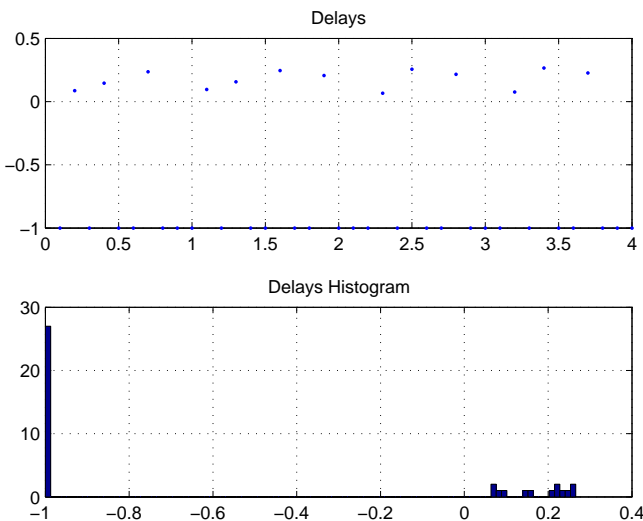


Figure 8. Delay interval of sensor samplings.

the transmitted samplings that are replaced by new ones before the controller accesses its input buffer; this is a worse case than the previous one because these losses consume bandwidth. So, as it can be seen, a multirate control scheme with  $N = 2, 3, 4$  appears (being  $N$  the so-called multiplicity of the multirate control system). A time-varying transmission delay is also observed. Figure 8 points out the delay interval of the sensor samplings that complete the round trip. In this figure the losses are marked with a '-1'. As shown, a complex problem is introduced, since an irregular sequence of  $N$  with time-varying delays  $\tau$  must be faced.

### 3.5. Statistical Analysis and Experimental Application

When the networked control is performing, an important issue is to know in advance the next  $N$ , that is, the antiquity of the next sampling, and its associated round-trip time delay  $\tau$ . In order to do it, a previous off-line execution is required to collect some samplings, considering the

Identified patterns:	Relat. Freq.	NEXT VALUE				TIME DELAYS:				
		1	2	3	4					
3-3-	0.08	0.00	0.00	0.00	1.00	0.21				
3-4-	0.31	0.00	1.00	0.00	0.00	0.07	0.08	0.09	0.10	
4-2-	0.31	0.00	0.00	1.00	0.00	0.15	0.16	0.26	0.27	
3-3-4-2-3-	0.08	0.00	0.00	0.00	1.00	0.22				
3-3-4-2-3-4-2-3-	0.07	0.00	0.00	0.00	1.00	0.23				
2-3-4-2-3-4-2-3-4-2-3-	0.07	0.00	0.00	1.00	0.00	0.25				
3-3-4-2-3-4-2-3-4-2-3-	0.07	0.00	0.00	0.00	1.00	0.24				

Figure 9. Case study. Statistical analysis.

random feature of the offset of the time-scale of the bus clock (however, local and remote clock skews are fixed). So, from this data collection, a conditioned-probability statistical analysis can be helpful for the engineer in order to determine the next  $N$  and  $\tau$  values with some accuracy. Figure 9 depicts this analysis for the time setup described in the previous section. The results show a quite reliable prediction from a short data collection<sup>1</sup>. This operation can be performed on the remote side, where the controller is located.

In Figure 9, the sequences shown in the column "identified patterns" are the different recurrent behaviour patterns with their relative frequency of appearance found in the data collection. Every prediction of  $N$  and  $\tau$  is included in the columns at the right of every pattern. So, after  $N = 2$ , a  $N = 3$  with probability 100% is expected to be appeared with the delays 0.15, 0.16, 0.26, 0.27 following a uniform distribution function. After  $N = 4$ , a  $N = 2$  with probability 100% will appear with delays 0.07, 0.08, 0.09 and 0.10 following a uniform distribution as well. When one  $N = 3$  is reached, the following  $N$  will be  $N = 3$  or  $N = 4$  with different set of delays. In order to decide between the former or the latter option, the key is to identify the past values of  $N$  assuming the identified patterns. For instance, if the pattern was 2-3-4-2-3-4-2-3-4-2-3, the predicted  $N$  will be  $N = 3$  for sure, with a delay of 0.25.

As commented, this previous performance analysis is needed in some real applications. Although reachable, the synchronization between remote and local clocks is quite complicated. In addition, the bus clock offset is absolutely unknown, even in the constant cycle (synchronous) bus mode. So, one solution is performing the off-line data collection in such a way that the data are sent by the local side at every bus cycle when the last round-trip is completed. Then, the remote side stores these data to compute the proper analysis. In this way, the different sequences of couples  $(N, \tau)$  and their associated probabilities are calculated, being able to be applied in the experimental process control execution when the steady state has been reached.

#### 4. Control Design

In this section, the dual-rate controller design is explained. Firstly, the so-called "nominal" controller is introduced, that is, the controller under a delay-free context (the desired control system behaviour). Different cases for  $N$  can be treated ( $N$  is including in a finite set  $\mathcal{N}$  of positive integers). The goal is to design a structure in which the output is produced at an  $N$  times faster rate than the input, that is, a dual-rate controller with one input each  $NT$  ( $N \in \mathcal{N}$ ,  $T \equiv T_s$ ) and  $N_f$  outputs ( $N_f \leq N$ ) ( $N_f$  depends on the round trip delay<sup>2</sup>). Then, a gain scheduling approach is arranged to modify, in a very easy way, the controller parameters in order to take into account the expected round trip delay.

<sup>1</sup>Actually, a small difference is obtained when the window of data is shifted.

<sup>2</sup>The next restriction is assumed: the round trip delay cannot be greater than the following  $NT$ . If this case appears, the best option is to perform another execution with a different bus time-scale offset.

#### 4.1. A dual-rate controller proposal

The proposed controller structure is a dual-rate PID one, Sala et al. (2009). In this chosen dual-rate PID controller structure  $k$  denotes the iterations at period  $NT$ , and  $e$  is the error signal. The input to the controller is a slow sensor input  $y(kNT)$ , and a setpoint  $r(kNT)$ , forming an error  $e(kNT) = r(kNT) - y(kNT)$ . Then, the basic control actions (proportional  $v_p$ , derivative  $v_d$ , and integral  $v_i$ ) are generated by:

$$v_p(kNT) = K_p e(kNT), \quad (1a)$$

$$v_i((k+1)NT) = v_i(kNT) + K_i e(kNT), \quad (1b)$$

$$v_d(kNT) = K_d(e(kNT) - e((k-1)NT)) + f \cdot v_d((k-1)NT) \quad (1c)$$

$$u(kNT) = v_p(kNT) + v_i(kNT) + v_d(kNT) \quad (1d)$$

The controller parameters are  $K_p$ ,  $K_i$ ,  $K_d$ , i.e., the proportional, integral and derivative gains, respectively,  $u$  describes the complete PID control action, and  $f$  is a derivative noise-filter pole (whose inclusion is standard in PID control, Astrom and Hagglund (1995)). It must be noted that the nominal dual-rate PID controller gains  $\theta_{nom} = \theta(0)$  (that is,  $K_p(0)$ ,  $K_i(0)$ ,  $K_d(0)$ ,  $f(0)$ ) can be designed by means of classical procedures such as directly in discrete-time with, say, root-locus or pole-placement criteria, Ogata (1987), by optimizing the time response, either manually (PID regulators are in many cases fine-tuned by hand with tuning rules-of-thumb or tuning charts, Astrom and Hagglund (1995)) or with toolboxes such as Matlab® Nonlinear Control Design Blockset, and obtained from some discretization method (Euler, bilinear) of continuous PID ones Salt et al. (2006). Afterwards, such parameters will be adjusted via the scheduling approach above mentioned, assuming the round trip delays.

These equations give rise to a second order realization which, if the regulator were a single-rate PID would be:

$$\begin{aligned} \psi(k+1) &= \begin{pmatrix} 1 & 0 \\ 0 & f \end{pmatrix} \psi(k) + \begin{pmatrix} K_i \\ 1-f \end{pmatrix} e(k) \\ u(k) &= (1 - K_d) \psi(k) + (K_p + K_d) e(k) \end{aligned} \quad (2)$$

where the state vector is  $\psi(k) = (v_i(k), \mu(k))^T$  composed by the integral basic action component plus an intermediate dummy variable  $\mu$  needed to compute the derivative component. Next, this variable is introduced formally:

$$v_d(z) = \left[ K_d \frac{1-z^{-1}}{1-fz^{-1}} \right] e(z) = K_d \left[ 1 - \frac{(1-f)z^{-1}}{1-fz^{-1}} \right] e(z)$$

where  $z$  is the discrete-time variable related to the sampling period  $NT$ . The dummy variable  $\mu$  is defined as:

$$\mu(z) = \frac{(1-f)z^{-1}}{1-fz^{-1}} e(z)$$

From this definition it is possible to consider:

$$v_d(z) = K_d e(z) - K_d \mu(z)$$

When implementing a dual-rate version of the above PID, the decision remains on how to design the faster control actions based on the above computed basic control actions. One possibility is assuming that the proportional and integral actions are generated at the slow period and the derivative one, which is related with anticipation and high-frequency behaviour, is concentrated in the first sample. For more details, see Sala et al. (2009). The first option leads to a lifted representation at global period  $NT$  yielding:

$$\psi(k+1) = \begin{pmatrix} 1 & 0 \\ 0 & f(\tau(k)) \end{pmatrix} \psi(k) + \begin{pmatrix} K_i(\tau(k)) \\ 1 - f(\tau(k)) \end{pmatrix} e(k) \quad (3a)$$

$$U_k(\tau(k)) = \begin{pmatrix} u_1(\tau(k)) \\ u_2(\tau(k)) \\ \vdots \\ u_N(\tau(k)) \end{pmatrix} = \begin{pmatrix} 1 & -K_d(\tau(k)) \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \end{pmatrix} \psi(k) + \begin{pmatrix} K_p(\tau(k)) + K_d(\tau(k)) \\ K_p(\tau(k)) \\ \vdots \\ K_p(\tau(k)) \end{pmatrix} e(k) \quad (3b)$$

where  $\tau(k)$  is the current network delay (affecting every PID controller gain, as will be later detailed in section 4.2), and the actions  $u_i$  for  $i \geq 2$  are applied at their respective trigger times ( $KNT + iT$ ) under no network delay (note, the first action  $u_1$  depends on the delay). A subset of the  $N$  expected control actions inside the period  $NT$  could not be applied if the network delay  $\tau$  is larger than the corresponding trigger time for each one of the actions included in the subset. So, note that the number of rows in (3b) depends on the values of  $N$  and  $\tau$ , that is, the number of fast control updates required, which is denoted by  $N_f$  in this work.

Once the lifted realisation of the controller is available, the closed-loop matrices needed in the stability analysis (see subsection 4.3) are easily computed. The procedure discussed in the following subsection may be applied to schedule the parameters  $K_p$ ,  $K_i$ ,  $K_d$  and  $f$  as a function of the measured network delay.

Note that, even if the structure (3) might seem complicated, the advantage of the approach is that the interpretation of the adjustable parameters  $K_p$ ,  $K_i$ ,  $K_d$  and  $f$  coincide with that of standard PID regulators and, hence, many of the single-rate rules of thumb or tuning rules may be used.

#### 4.2. Controller gain scheduling approach

In the proposed approach, a plant  $\mathcal{P}$  will receive a set of control actions which depend on the round-trip network delay and the period between two consecutive measures  $N(k)T$  of the controlled variable; in addition, the controller will have adjustable parameters  $\theta$ , hence it can be represented by  $C(N, \tau, \theta)$ . In this way, the control action will be obtained from  $u = C(N, \tau, \theta) \cdot e$ , being  $e = S(r - y)$ , where  $S$  denotes a time-driven sampling operation at instants correlated with a set of  $N(k) \in \mathcal{N}$ . As the timing information  $N$  and  $\tau$  can be predicted at the remote controller side, the actual gain scheduling can be carried out at this side (as commented in section 3.5) where the computation power is installed.

The aim of this approach is to introduce a controller retuning law  $\theta(N, \tau)$  in order to keep the nominal (single rate  $T$ , no-delay) control performance despite  $\tau$  and  $N$  variations.

In this work, the closed-loop poles will be chosen as a sensible criterion to define control performance. So, a so-called *performance vector*  $\pi(N, \tau, \theta)$  will take the form

$$\pi(N, \tau, \theta) = \text{Eig}(Q(N, \tau, \theta)) \quad (4)$$

where  $\theta$  are the design parameters (it includes the PID controller parameters),  $Q$  represents the closed-loop operator  $Sy = Q(Sr)$ , and the operator  $Eig$  represents the eigenvalue computation of the resulting state-space matrix. In principle, a time-invariant nature for  $\tau$  and  $N$  will be considered for design (otherwise, poles do not exist). Later on, time-variant stability analysis will be used to check the obtained results (see section 4.3).

Thus, the nominal situation to be kept will be defined by  $\pi(N^*, 0, \theta_{nom})$ , being  $\theta_{nom}$  the nominal controller parameters designed for a delay-free case and a specific  $N^{*1}$ . Every dual-rate PID controller based on a different  $N$  is retuned with approximately the same single-rate closed loop performance, and they are smoothly retuned by following the scheduling approach in order to cope with time-varying network delays. If a delay  $\tau(k)$  appears, the values for the performance vector will change with respect to  $\pi(N, 0, \theta_{nom})$ . The goal of the gain scheduling proposal is retuning  $\theta$  according to the delay,  $\theta(N(k), \tau(k))$ , in order to make this difference as small as possible. So, the following linear scheduling law is proposed:

$$\theta(N(k), \tau(k)) = \theta_{nom} + M_g(N(k))\tau(k) \quad (5)$$

where  $M_g(N(k))$  is denoted as *scheduling vector* for each possible value of  $N$  at each iteration  $k$ , which is deduced after solving a least-square problem on the minimization of the first-order Taylor term of  $\|\pi(N, \tau, \theta) - \pi(N, 0, \theta_{nom})\|$  (details are omitted for brevity, more information in Sala et al. (2009)). The result takes the form

$$M_g(N(k)) = (\Delta^T W^T W \Delta)^{-1} W^T \Delta^T \delta_\tau \quad (6)$$

being  $W$  a weighting filter (to give priority to dominant poles),  $\delta_\tau = \frac{\partial \pi}{\partial \tau}$  evaluated at  $\tau = 0$  (and some  $N$  value),  $\theta = \theta_{nom}$ , and  $\Delta$  is the vector of derivatives with respect to parameters  $\theta$  evaluated at the same nominal point.

### 4.3. Stability analysis

From a lifted framework Khargonekar et al. (1985), the feedback connection of a generic controller and a process can be expressed as

$$\begin{pmatrix} \psi_{k+1} \\ x_{k+1} \end{pmatrix} = \begin{pmatrix} A_C & -B_C C_P \\ B_P C_C & A_P - B_P D_C C_P \end{pmatrix} \begin{pmatrix} \psi_k \\ x_k \end{pmatrix} = \bar{A}_{cl} \bar{x}(kNT) \quad (7)$$

where  $\Sigma_C = \{A_C, B_C, C_C, D_C\}$  and  $\Sigma_P = \{A_P, B_P, C_P, D_P\}$  are respectively the lifted discrete state-space representation for the controller and the process, and the augmented state  $\bar{x}(kNT) \equiv \bar{x}$  contains both the controller states,  $\psi$ , and the plant state variables,  $x$  (for more details, see for example Albertos and Sala (2004), Sala et al. (2009)).

Due to the specific network characteristics, in this case  $N$  and  $\tau$  can vary from metaperiod to metaperiod,  $N(k)T$ . Then, the equation (7) will take this form:

$$\bar{x}(k+1) = \bar{A}_{cl}(N, \tau) \bar{x}(k) \quad (8)$$

representing a discrete LTV system.

---

<sup>1</sup>In this work  $N^* = 1$  was selected.

If the  $(N, \tau)$  sequence is known, that is, the variation of the couple  $(N, \tau)$  along the time, a proper stability analysis of the closed-loop system can be carried out. In the present work, considering an average decay rate  $\alpha$  (being  $0 < \alpha < 1$ ) as a performance objective, a multiobjective analysis can be formulated due to the next reasons:

- $N$  can take different values as shown in Figure 7. Concretely,  $N = 2, 3, 4$ . So, for each value of  $N$ , a different network state which is able to reach some performance objective  $\alpha_N$  can be taken into account<sup>1</sup>.
- different transitions can appear between each pair of states. In general, the total number of unidirectional transitions (that is, from one state to another state, but not considering the opposite situation which implies going from the latter state to the former one) will be  $\frac{n_0^2 - n_0}{2}$ , being  $n_0$  the total number of states. Then, every possible transition  $i$  ( $i = 1, 2, \dots, \frac{n_0^2 - n_0}{2}$ ) between each pair of states will be analysed. In the present study,  $n_0 = 3$ , and 3 are also the transitions to be taken into account as depicted in Figure 9<sup>2</sup>.
- as shown in Figure 9, each value of  $N$  presents a set of network delays,  $\Theta_N$ , being  $\Theta_N$  polytopic.

In summary, every  $N$  can be defined by a performance objective  $\alpha_N$  and a set of network delays  $\Theta_N$ . Since for a same  $N$ ,  $\Theta_N$  and  $\alpha_N$  can vary depending on the transition  $i$ , then  $\Theta_N(i)$ ,  $\alpha_N(i)$ . The multiobjective study will analyze stability for every possible transition between each pair of states. In the present study, the transitions to be taken into account will be:

- transition  $i = 1$ , between  $N = 2$  and  $N = 3$
- transition  $i = 2$ , between  $N = 4$  and  $N = 2$
- transition  $i = 3$ , between  $N = 3$  and  $N = 4$ .

So, considering the next quadratic Lyapunov function

$$V(\bar{x}) = \bar{x}^T Q \bar{x} \quad Q > 0 \quad (9)$$

which will be shown to decrease in average<sup>3</sup>, that is

$$E[V(x(k+1))] \leq \alpha_N(i)^2 E[V(x(k))] \quad (10)$$

then, replacing the closed-loop equations in (9), the Lyapunov decrease condition (10) can be written as the following linear matrix inequality:

$$\bar{A}_{cl}(N, \vartheta_N)^T Q_i \bar{A}_{cl}(N, \vartheta_N) - \alpha_N^2(i) Q_i, \quad \forall \vartheta_N \in \Theta_N(i), \quad \forall N \in \text{transition } i, \quad i = 1, \dots, \frac{n_0^2 - n_0}{2} \quad (11)$$

where  $\vartheta_N$  is a dummy parameter ranging in the set  $\Theta_N(i)$  where, for each  $N$ , the time-varying parameter  $\tau$  is assumed to take values in, and matrix  $Q_i$  (a different one for each transition  $i$ ) is composed of decision variables to be found by a semi-definite programming solver. As  $\bar{A}_{cl}$  is an affine function of  $\tau$ , and  $\Theta_N(i)$  is polytopic, then (11) can be checked with a finite number of LMIs, which can be solved by widely-known methods Boyd et al. (1994), Sturm (1999).

Note that the use of the shared Lyapunov function in (11) implies somehow a conservative study, since the analysis is able to prove stability of the networked control system not only for

<sup>1</sup>Note,  $\alpha_N$  is the decay rate  $\alpha$  for the considered  $N$ .

<sup>2</sup>In this case, the total number of transitions coincide with  $\frac{n_0^2 - n_0}{2} = 3$ , but the network behaviour could involve a lower number of them.

<sup>3</sup>Denoting  $E[\cdot]$  as the statistical expectation,  $E[V(x)]$  will tend to zero hence the state will converge to zero with probability one.



unidirectional transitions but also for any mixture of the considered network states (that is, (11) analyses both bidirectional transitions and variations in the delay inside the same  $N$ ).

## 5. Application example: control of a crane platform

In this section, a test-bed Profibus-DP environment is used to implement the proposed NCS (see figure 10), which includes the following devices:

- the plant to be controlled, that is, an industrial crane platform equipped with three cc motors (to actuate each axis: X, Y, Z) and five encoders (to sense the three axis and two different angles). These encoders are the local sensors and they perform their activity each  $T_s$ . The motors are controlled by an analog signal in the range  $\pm 1V$ . The encoders provide a position measurement of 1V/m. In this application only the X axis is controlled<sup>1</sup>.
- a local PLC which is connected to the platform by means of a DAQ board, with restricted computation possibilities,
- two PLCs and one computer working as interference nodes in order to introduce different load scenarios,
- the wire shared by the previous and other devices to connect them to Profibus-DP.
- a remote computer where the controller performing each  $T_c$  is implemented.

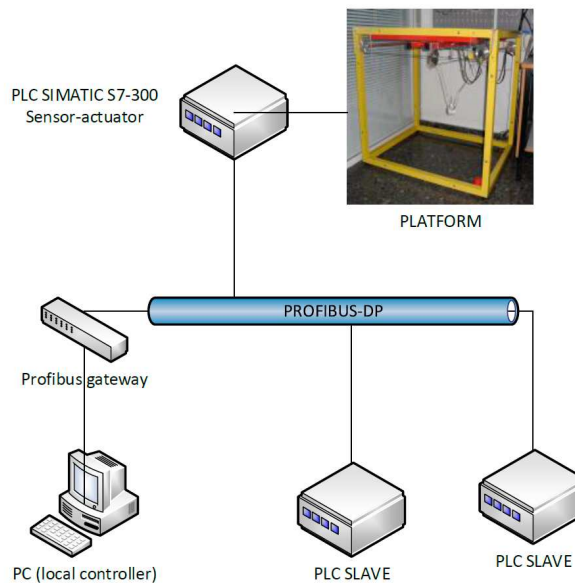


Figure 10. Test-bed Profibus-DP environment

The specific network configuration will be the explained one in section 3 and subsection 3.5, whose probability histograms, parameters and performance have been widely described. As it was already commented, the objective of the methodology of this paper is to design an NCS with a dual-rate controller, and to analyse its stability using LMI tools. The experimental data is used to run a simulation from these practical time-scale values. Hence, the section will be divided into a first *a priori* control design phase with LMI and simulation studies and, later, an experimental verification of the applicability of the obtained results.

<sup>1</sup>Details on the crane characteristics can be obtained at <http://www.inteco.com.pl> (3D crane apparatus)

### 5.1. Model-based design and simulation study

In this section, the following DC motor plant of the X axis of the industrial crane will be used as the controlled plant at the local side:

$$\mathcal{P}(s) = \frac{6.3}{s(s + 17.7)}$$

The main aim of this example is to show the possibilities of the dual-rate PID controller when it is retuned according to the round-trip network delays  $\tau$  and the multiplicity  $N$ . Remember that a time triggered controller is used but an event-based control action updating is assumed. So, the number of control actions updated on  $NT$  will depend on the delay.

First of all, the dual-rate PID controller choice is justified. Then, the LMI analysis is performed, comparing the nominal (non-scheduled) controller with the scheduled one. Finally, time response for both controllers is simulated by means of a simulation application.

Controller choice: A comparison, for the no-delay case, between single-rate PD regulators ( $N = 1$ ), and some dual-rate PD regulators is carried out.

The output sampling time for the dual-rate case will be  $NT$ , being  $T = 0.1s$  and  $N \in \mathcal{N}$ . As explained, to carry out the study the couple  $(N, \tau)$  must be known, which can be accurately determined by means of a previous experience.

Since the motor plant has an integrator, a suitable control can be achieved with a PD controller.

From an initial continuous root-locus design, the single-rate PD is fine-tuned by hand to achieve a response as fast as possible. This is obtained from a bilinear discretisation of:

$$K_p(1 + K_d \frac{s}{f_s + 1})$$

Different sets of values of  $K_p, K_d, f$  were obtained for different  $NT$  and different number of control periods  $N_f$  (for different delays). For every  $N$  a dual-rate PD realization (3) must be considered with the different values of

$$K_p, K_i = 0, K_d, f$$

Moreover, if a fine tune is desired, some test with the viable cases of  $N_f$  for each  $N$  must be carried out. These controllers are the so-called *nominal controllers* for each case.

Table 1. Nominal Controllers Parameters

Case $N, N_f$	Parameters $K_p, K_d$	$f$
2,2	3,2.2	0.1
3,2	5.5, 2.6	0.1
3,3	6.0, 2.6	0.1
4,2	6.0, 2.8	0.1

From the nominal parameters, the gain scheduling approach achieves a delay-dependent law. To do it, firstly, the parameters  $\delta_\tau$  and  $\Delta$  in (6) must be obtained (steps omitted for brevity, see Sala et al. (2009) for details). Then, considering  $W$  as the identity matrix, the least-squares solution produced the scheduling laws shown in table 2 as a function of the measured network delay  $\tau(k)$ .

Non-significant differences were reached for different  $N_f$  in the case of  $N = 3$ .

Figure 11 depicts the control system output for the nominal (without gain scheduling) controller and for the scheduled one considering the proposed NCS setup. Simulation results show

Table 2. Gain scheduling law for different  $N$ 

Case $N$	$M_g(N)$
2	$(K_p K_d f)^T = (-31.2675 \ -6.6734 \ -0.5248)^T \tau(k)$
3	$(K_p K_d f)^T = (15.2497 \ -28.0576 \ -0.4504)^T \tau(k)$
4	$(K_p K_d f)^T = (6.4476 \ -2.0771 \ 0.4677)^T \tau(k)$

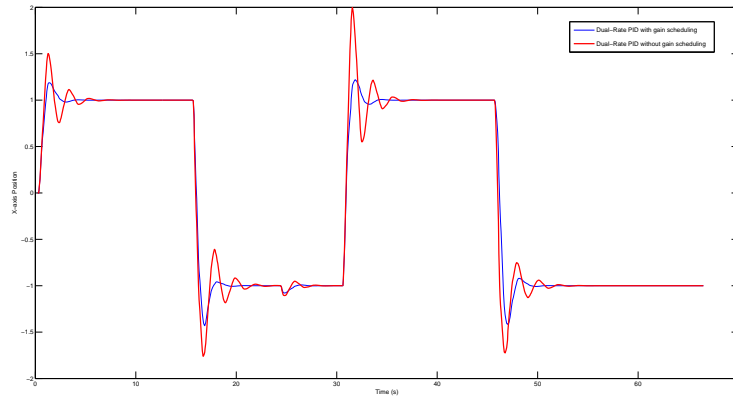


Figure 11. Study Case with PID dual-rate controller with and without gain scheduling.

a worse performance when the gain scheduling is not assumed (for example, the settling time is increased up to 30% and the maximum overshoot up to 70%).

Stability analysis: The main goal of the gain scheduling approach (section 4.2) is to try to keep the nominal performance under  $N$  and  $\tau$  variations. The procedure is based on closed-loop poles, which is an optimistic estimate of the actual performance for varying  $(N, \tau)$ . Then, in order to assess the stability of the setup in time-varying delays, the LMI conditions in (11) has been carried out computing the closed-loop realization for the next situations:

- transition between  $N = 2$  and  $N = 3$  (transition  $i = 1$ ): for  $N=2$  the parameter space is  $\Theta_2(1) = [0.07, 0.08, 0.09, 0.10]$ , and for  $N = 3$ ,  $\Theta_3(1) = [0.15, 0.16, 0.26, 0.27]$ .
- transition between  $N = 4$  and  $N = 2$  (transition  $i = 2$ ): for  $N=4$  the parameter space is  $\Theta_4(2) = [0.21, 0.22, 0.23, 0.24]$ , and for  $N = 2$ ,  $\Theta_2(2) = \Theta_2(1)$ .
- transition between  $N = 3$  and  $N = 4$  (transition  $i = 3$ ): for  $N=4$  the parameter space is  $\Theta_4(3) = [0.15, 0.16, 0.25, 0.26, 0.27]$ , and for  $N = 3$ ,  $\Theta_3(3) = \Theta_3(1)$ .

According to (11), each transition between each pair of states is independently analysed (remember, different  $Q_i$  are used).

As well-known, results of a multi-objective analysis can be illustrated by means of a Pareto front Sawaragi et al. (1985). So, figure 12 summarizes the different analysis carried out for each transition. These analysis are developed by setting the decay-rate of one objective and optimizing the other's one. The figure reveals stability for every transition, since the decay-rates are always less than 1. In addition, the scheduled approach clearly outperforms the non-scheduled, nominal one for the transitions  $i = 2$  and  $i = 3$ , since better (lower) decay-rates are always obtained in the scheduled proposal. However, both approaches behave similarly in the transition  $i = 1$ . This fact can be explained by remembering how the gain scheduling approach works. As commented, the approach modifies the controller gains according to the delay. And, the shorter the delay is,

the less the difference between scheduled and non-scheduled gains will be. Next, let us check the mean delay  $\tau_N^{\text{mean}}(i)$  for every state in every transition. For the transition  $i = 1$ , when  $N = 2$ ,  $\tau_2^{\text{mean}}(1) = 0.085$ , and when  $N = 3$ ,  $\tau_3^{\text{mean}}(1) = 0.21$ , which implies the lowest mean delays. With respect to the transition  $i = 1$ , the transition  $i = 2$  increments the mean delay when introducing  $N = 4$  ( $\tau_4^{\text{mean}}(2) = 0.225$ ). Finally, the transition  $i = 3$  includes the highest mean delays when using  $N = 3$  and  $N = 4$ . So, the exposed mean delay values corroborate the decay-rate tendency.

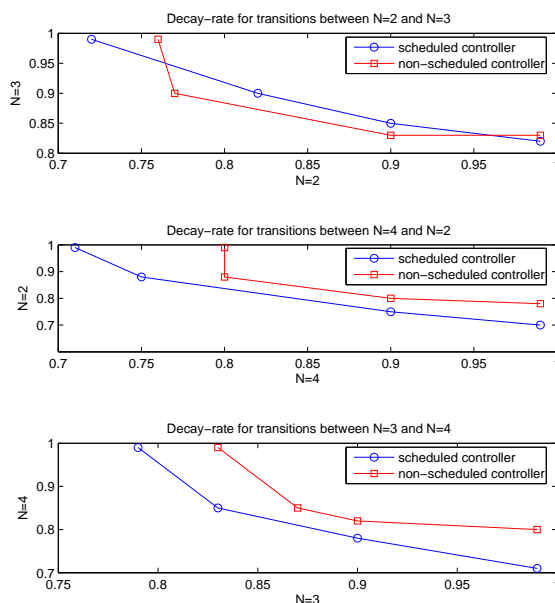


Figure 12. LMI decay-rate. Comparison for each transition.

## 6. Conclusions

A gain scheduling multirate PID control has been analysed and designed over a Profibus-DP installation. Although a Profibus-DP network can be used in industrial environments assuming a deterministic behaviour, some randomness appears (even with the option of synchronous bus) due to concerns such as the lack of synchronization among clocks, the appearance of time-varying delays, the existence of bandwidth limitations, and so on. Dual rate control techniques are proposed to assure a good performance, saving network resources. An LMI based stability analysis is required to ensure feasibility for different sensor measure losses and round trip time delays. In future works, the non-constant bus cycle will be faced.

## Acknowledgements

The authors are grateful to the financial support of the Spanish Ministry of Economy and Competitiveness (Research Grant TEC2012-31506). We would like to express as well our gratitude to both P. Tormos and A. Lova with the Department of Statistics of the Universitat Politècnica de Valencia for their support and help in this work.

## References

- Albertos, P., and Sala, A., *Multivariable Control Systems: An Engineering Approach*, Springer (2004).
- Alves, M., and Tovar, E. (2007), “Real-time communications over wired/wireless PROFIBUS networks supporting inter-cell mobility,” *Computer Networks*, 51(11), 2994–3012.
- Astrom, K., and Hagglund, T. (1995), “PID Controllers: Theory, Design, and Tuning,” *Instrument Society of America*, pp. 80–81.
- Bauer, P.H., Sichitiu, M.L., and Premaratne, K. (1999), “Closing the loop through communication networks: The case of an integrator plant and multiple controllers,” in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, Vol. 3, IEEE, pp. 2180–2185.
- Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Society for Industrial Mathematics (1994).
- Bucher, R., and Balemi, S. (2006), “Rapid controller prototyping with Matlab/Simulink and Linux,” *Control Engineering Practice*, 14(2), 185–192.
- Casanova, V., and Salt, J. (2003), “Multirate control implementation for an integrated communication and control system,” *Control engineering practice*, 11(11), 1335–1348.
- Chang Lee, K., Lee, S., and Hee Lee, H. (2004), “Implementation and PID tuning of network-based control systems via Profibus polling network,” *Computer Standards & Interfaces*, 26(3), 229–240.
- Chow, M.Y., and Tipsuwan, Y. (2001), “Network-based control systems: a tutorial,” in *Industrial Electronics Society, 2001. IECON’01. The 27th Annual Conference of the IEEE*, Vol. 3, IEEE, pp. 1593–1602.
- Cuenca, A., and Salt, J. (2012), “RST controller design for a non-uniform multi-rate control system,” *Journal of Process Control*.
- Cuenca, Á., Salt, J., and Albertos, P. (2007), “Implementation of algebraic controllers for non-conventional sampled-data systems,” *Real-Time Systems*, 35(1), 59–89.
- Cuenca, Á., Salt, J., Sala, A., and Pizá, R. (2011), “A delay-dependent dual-rate PID controller over an ethernet network,” *Industrial Informatics, IEEE Transactions on*, 7(1), 18–29.
- Cuenca, Á., Pizá, R., Salt, J., and Sala, A. (2012), “Linear Matrix Inequalities in Multirate Control over Networks,” *Mathematical Problems in Engineering*, 2012.
- Halevi, Y., and Ray, A. (1988), “Integrated communication and control systems. I- Analysis,” *ASME, Trans., J. Dynamic Syst., Meas. and Control*, 110, 367–373.
- Khan, S., Goodall, R.M., and Dixon, R. (2012), “Non-uniform sampling strategies for digital control,” *International Journal of Systems Science*, (ahead-of-print), 1–21.
- Khargonekar, P., Poolla, K., and Tannenbaum, A. (1985), “Robust control of linear time-invariant plants using periodic compensation,” *Automatic Control, IEEE Transactions on*, 30(11), 1088–1096.
- Lall, S., and Dullerud, G. (2001), “An LMI solution to the robust synthesis problem for multi-rate sampled-data systems,” *Automatica*, 37(12), 1909–1922.
- Lee, K., Lee, S., and Lee, M. (2003), “Remote fuzzy logic control of networked control system via Profibus-DP,” 50(4), 784–792.
- Lee, S., Lee, K.C., Lee, M.H., and Harashima, F. (2002), “Integration of mobile vehicles for automated material handling using Profibus and IEEE 802.11 networks,” *Industrial Electronics, IEEE Transactions on*, 49(3), 693–701.
- Lian, F.L., Moyne, J., and Tilbury, D. (2002), “Network design consideration for distributed control systems,” *Control Systems Technology, IEEE Transactions on*, 10(2), 297–307.
- Lian, F.L., Moyne, J.R., and Tilbury, D.M. (2001), “Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet,” *Control Systems, IEEE*, 21(1), 66–83.
- Lin, J., Fei, S., and Gao, Z. (2013), “Control discrete-time switched singular systems with state delays under asynchronous switching,” *International Journal of Systems Science*, 44(6), 1089–

- 1101.
- Liou, L.W., and Ray, A. (1991), "A stochastic regulator for integrated communication and control systems: Part I-Formulation of control law," *ASME Journal of Dynamic Systems, Measurement and Control*, 113(4), 604–611.
- Lorand, C., and Bauer, P.H. (2006), "On synchronization errors in networked feedback systems," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(10), 2306–2317.
- Moayedi, M., Foo, Y.K., and Soh, Y.C. (2011), "Filtering for networked control systems with single/multiple measurement packets subject to multiple-step measurement delays and multiple packet dropouts," *International Journal of Systems Science*, 42(3), 335–348.
- Ogata, K., *Discrete-time control systems*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA (1987).
- Peñarrocha, I., Sanchis, R., and Romero, J.A. (2012), "State estimator for multisensor systems with irregular sampling and time-varying delays," *International Journal of Systems Science*, 43(8), 1441–1453.
- Piza, R., Salt, J., Sala, A., and Cuenca, Á. (To be published 2013), "Hierarchical Triple-Maglev Dual-Rate Control Over a Profibus-DP Network," *Control Systems Technology, IEEE Transactions on*.
- Ray, A., and Halevi, Y. (1988), "Integrated communication and control systems: Part II-Design considerations," *ASME Journal of Dynamic Systems, Measurement, and Control*, 110(4), 374–381.
- Ray, A. (1989), "Introduction to networking for integrated control systems," *Control Systems Magazine, IEEE*, 9(1), 76–79.
- Sala, A., Cuenca, A., and Salt, J. (2009), "A retunable PID multi-rate controller for a networked control system," *Inform. Sci.*, 179(14), 2390–2402.
- Salt, J., and Albertos, P. (2005), "Model-Based Multirate Controllers Design," *Control Systems Technology, IEEE Transactions on*, 13(6), 988–997.
- Salt, J., Cuenca, A., Casanova, V., and Mascaros, V. (2006), "A PID Dual Rate Controller Implementation over a Networked Control System," *Control Applications, 2006 IEEE International Conference on*, pp. 1343–1349.
- Salt, J., Sala, A., and Albertos, P. (2011), "A Transfer-Function Approach to Dual-Rate Controller Design for Unstable and Non-Minimum-Phase Plants," *Control Systems Technology, IEEE Transactions on*, 19(5), 1186–1194.
- Sánchez, J., Esquembre, F., Martín, C., Dormido, S., Dormido-Canto, S., Canto, R., Pastor, R., and Urquía, A. (2005), "Easy java simulations: an open-source tool to develop interactive virtual laboratories using matlab/simulink," *International Journal of Engineering Education*, 21(5), 798.
- Sawaragi, Y., Nakayama, H., and Tanino, T., *Theory of multiobjective optimization*, Academic Press (1985).
- Schickhuber, G., and McCarthy, O. (1997), "Distributed fieldbus and control network systems," *Computing & Control Engineering Journal*, 8(1), 21–32.
- Specification, P., "Normative Parts of PROFIBUS-FMS,-DP,-PA according to the European Standard EN 50 170," (1998).
- Sturm, J. (1999), "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, 11(1), 625–653.
- Tipsuwan, Y., and Chow, M. (2003), "Control methodologies in networked control systems," *Control Eng. Practice*, 11(10), 1099–1111.
- Vitturi, S. (2004), "Stochastic model of the Profibus DP cycle time," in *Science, Measurement and Technology, IEE Proceedings-*, Vol. 151, IET, pp. 335–342.
- Weigmann, J., and Kilian, G., *Decentralization with Profibus-DP: architecture and fundamentals, configuration and use with Simatic S7*, John Wiley & Sons, Inc. (2000).
- Xie, L., Ozkul, S., Sawant, M., Shieh, L., Tsai, J., and Tsai, C. (2013), "Multi-rate digital redesign

- of cascaded and dynamic output feedback systems,” *International Journal of Systems Science*, (ahead-of-print), 1–12.
- Yang, T. (2006), “Networked control system: a brief survey,” *IEE Proc. Control Theory Appl.*, 153(4), 403–412.