

## Research article

# Reliable PID dual-rate controller based on LoRaWAN for long-range distributed systems

Salvatore Dello Iacono<sup>1</sup>\*, Alessandro Depari<sup>1</sup>, Paolo Ferrari<sup>1</sup>,  
Alessandra Flammini<sup>1</sup>, Stefano Rinaldi<sup>1</sup>, Emiliano Sisinni<sup>1</sup>

Dept. Information Engineering, University of Brescia, via Branze 38, Brescia (BS), 25123, Italy

## ARTICLE INFO

Dataset link: [Zenodo](https://zenodo.org/record/101570)

## Keywords:

Wireless sensor networks  
Networked control systems  
LoRaWAN  
Dual-rate controller  
PID  
Performance metrics

## ABSTRACT

With the advances in digital communication systems, Network Control Systems (NCS) appeared as a feasible control solution for automation tasks. More recently, the exploitation of ubiquitous Internet connectivity for implementing a NCS has been suggested, paving the way for the Industrial Internet-of-Things paradigm. In particular, moving towards the Control as a Service (CaaS) would allow to benefit from the adaptability and scalability offered by cloud computing. Unfortunately, the needs of automation systems are very different from those dictated by the office and enterprise scenarios. The performance and availability of communication infrastructure used for connecting with the cloud can severely affect the control strategy effectiveness. This study assesses the implementation of a dual-rate controller, enhanced by an appropriate predictor, to facilitate NCS utilizing a potentially unstable LoRaWAN network. The selection of LoRaWAN, a type of Low Power Wide Area Network, is due to its extensive utilization in academic and industrial sectors. Moreover, the standard explicitly delineates backend entities, facilitating novel CaaS business models. A case study of a control system composed of a single LoRaWAN end-node device placed in a local plant to be controlled and connected to a single gateway network is employed to simulate the DR-PID control strategy on a second-order time-continuous system. The simulation model incorporates LoRaWAN characteristics, such as node-initiated transactions, and synthesizes measurable parameters, such as up-link and down-link losses. This model has been employed to assess control performance relative to a reference ideal situation in which communication losses are absent. A figure of merit is defined, enabling the assertion of the suggested approach's superiority over the plain LoRaWAN NCS. Furthermore, the quasi-orthogonality of time and frequency superposed frames with varying Spreading Factors (SF) is proposed as an alternate method to leverage the enhanced noise immunity provided by elevated SF values.

## 1. Introduction

With the spreading of fast and reliable networking systems in the industry and automation field, traditional point-to-point control systems become Networked Control Systems (NCS), meaning that control logic and controllers are no more located in the proximity of the process but distributed across a wider area and connected by a communication channel; feedback loops are closed using networks, that connect sensors, actuators, and controllers. NCS are usually classified as: • centralized: if a single centralized controller uses the communication infrastructure to have full knowledge of the plant; • decentralized: if several, non interacting controllers

\* Corresponding author.

E-mail address: [salvatore.delloiacono@unibs.it](mailto:salvatore.delloiacono@unibs.it) (S. Dello Iacono).

<https://doi.org/10.1016/j.iot.2025.101570>

Received 13 December 2024; Received in revised form 6 March 2025; Accepted 9 March 2025

Available online 15 March 2025

2542-6605/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

use the communication infrastructure to have each one a partial knowledge of the plant; • distributed: if several controllers interact to provide some form of cooperative control [1].

The idea of using the Internet as the communication infrastructure for NCS arose at the beginning of the century. The ubiquitous nature of the Internet can offer advantages in terms of global access to the control functionalities, allowing experts of different companies (e.g., the plant manager and engineers of device manufacturers) to easily access data of interest (possibly using standard tools and protocols) without bounds and distance limits. More recently, the advent of the Industrial Internet-of-things (IIoT) and the Industry 4.0 paradigms further pushes production processes, historically implemented using immutable software tools and hardware devices, to be a part of larger intra and inter-site connected systems. Several benefits are offered when data from many devices on the field are gathered and aggregated in a central location, namely the cloud, where decisions are de facto taken. For instance, scalability and maintainability, since strategies for system-wide performance enhancements can be envisioned, and failures can be detected before compromising the production [2]. Some control applications leverage the cloud for real needs, such as data availability, others for practical reasons (e.g., the Internet is the only data connection), and others for the almost unlimited availability of computational and storage resources. Research activities are ongoing to push to the limit this approach, inheriting technologies that are typical of cloud-based applications (such as virtualization), and leading to the Control as a Service (CaaS) paradigm. According to CaaS, the control functionality is just another *softwarized* service exploiting the cloud infrastructure: it is very interesting because its performance may be comparable or better than using dedicated hardware, which is difficult/expensive to maintain in a long-term [3,4].

This work focuses on a specific application of CaaS to long-range, wide-area distributed systems with low maintenance budgets. Such scenarios are critical from the data transmission point of view, suffering more about power consumption, variable network delays, and missing data. In turn, this makes the control action prone to long settling times, if not to failures. In this paper, two technologies are combined to reach the final distributed control goal over area of several square kilometers, these are: Dual Rate Distributed Control System (DR-DNCS) and LoRaWAN. As it will be discussed in more detail in the following sections, a two-level architecture is applied to a DR-DNCS in order to implement a proportional-integral-derivative (PID) controller with a LoRaWAN network for connecting the local and the remote sites. The main motivation for the experimentation with PIDs is their attractiveness in the industry field because they can be easily designed to accurately meet time or frequency constraints, given the process model, and are already widely used and known in the automation field. On the other hand, LoRaWAN, a member of the Low Power Wide Area Network (LPWAN) family, offers many advantages compared to other wireless technologies, like low-cost, widespread adoption, standardization (and also possibly private) backend.

This research aims to demonstrate the feasibility, limitations, and possible advantages of combining LoRaWAN communication infrastructure with DR-PID control strategy in a Distributed Networked Control System. The study, conducted through simulations described in the manuscript, focuses on verifying the effectiveness of a DR-PID control approach when subject to constraints imposed by low-power wireless communication. The simulations use a simple testbench consisting of a single LoRaWAN node, ideally installed in the (local) processing plant with one sensor and one actuator. Simulations consider the peculiarities of LoRa and LoRaWAN, such as the node-originated communications, duty-cycle limitations, and the use of multiple spreading factors, without the intent of developing a complete LoRaWAN simulator neither to create a generic DR-PID control framework. The first innovative idea is to leverage the lower operating rate of the remote controller level to enable the use of the low-cost (but low-transfer-rate) LoRaWAN interface. This results in gaining the desired long-range. The second innovative idea is to exploit the quasi-orthogonality of LoRaWAN virtual channels to add redundancy and reduce data loss. This provides the needed control reliability. In detail, the novel and original contributions of this paper are:

- the analysis of possible DR-DNCS solutions for long-range, low maintenance budget, scenarios;
- the analysis of the feasibility of a LoRaWAN-based DR-DNCS, exploiting specific LoRa peculiarities;
- the implementation of a prediction strategy to reduce the impact of message losses and its mapping of LoRaWAN virtual channels
- the development of a simulation model taking into account the real behavior of a LoRaWAN connection, including message losses;
- the definition of metrics to evaluate the obtainable performance;
- the simulation of a real-world scenario, to highlight the superior performance of the proposed reliable DR-DNCS approach with respect to a centralized NCS, when the same control functionality is implemented.

The paper is arranged as follows. In Section 2, the proposed approach is detailed, also taking into account state-of-the-art strategies. In Section 3 some concepts about the LoRaWAN standard are described and the dual-rate PID controller is addressed as the considered DR-DNCS. The reference use case is described in Section 4. At the same time, results are provided in Section 5 and a discussion about the proposed approach over the regular NCS approach is in Section 6. Finally, some conclusions are drawn.

## 2. The proposed approach

The main goal of this work is to realize a wide area of distributed control systems with affordable management budgets that can result in multiple contrasting constraints. In this section, an overview of related works is provided, and the proposed use of LoRaWAN for implementing a DR-DNCS is discussed. A schematic representation of the overall architecture is depicted in Fig. 1 where it is possible to see, from the left to the right, the remote control part that communicates with the LoRaWAN infrastructure that is responsible for relaying the messages to the local nodes. On the other hand, the node, composed of the LoRa radio and all

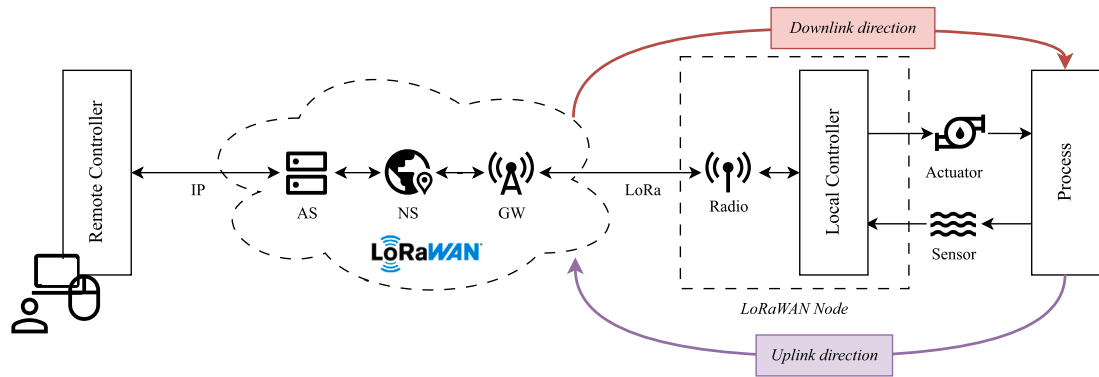


Fig. 1. Block diagram of the proposed approach, showing how the controller is split into remote and local parts connected by means of a LoRaWAN link exchanging uplink and downlink traffic.

the control logic needed to implement the digital control strategy, is responsible for interacting through the actuator and the sensor with the process under control. In the considered scenario, a single LoRaWAN node is placed in the plant, sending sensor data using uplink frames and receiving control actions and predictions using downlink frames to and from the backend accessed through a single Gateway.

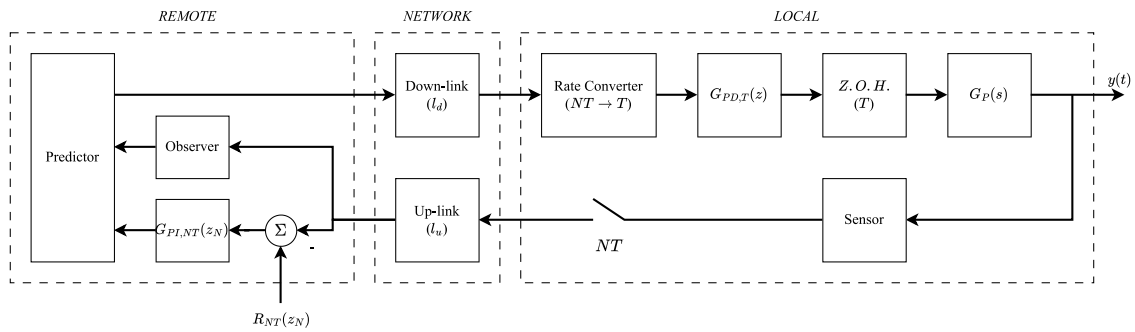
## 2.1. Related works

It is well-known that NCSs suffer from the delay in exchanging feedback and control actions due to the latency and jitter introduced by a heterogeneous infrastructure consisting of a network of networks. A vast literature exists addressing these issues, including surveys highlighting recent advances and open challenges (e.g., to mention a few, consider [5–7]). Additionally, network reliability can be very different depending on the actual communication technology; for instance, if wireless links are considered, it is expected that message losses can occur, degrading the performance of the NCS [8,9].

Despite the higher complexity, a carefully designed distributed approach can mitigate the impact of an unreliable data exchange. An example is the two-level control architecture which consists of a higher level implementing the global control function and a lower level that ensures the plant is under control even if the connectivity is lost. Typically, the solution is configured as a DR-DNCS, in which the higher level is hosted remotely with respect to the plant and is operating slower. In comparison, the lower level is hosted locally and is operating faster to stabilize the plant [10]. Therefore, an advantage is the reduced traffic between the remote and local sides that can occur across a low-bandwidth communication link [11].

There is extensive literature about multirate control systems and their vast use in industrial systems to solve different problems. As an example, to compensate for the inter-sampling period, the authors in [12] designed a control strategy for linear periodic systems based on a multirate event-triggered controller in which trigger checking and control update are performed at different rates. Another common example of the use of multirate control systems, as a solution in the reduction of computational requirements for industrial control systems is reported in [13] where the authors propose a case study based on an industrial print-belt system for multirate repetitive controller capable of perfectly attenuating periodic disturbance and guarantee the desired control performance. Also dual-rate controller, as a specific implementation of multirate controls, are used in conjunction with advanced prediction techniques with the intent of reducing the energy consumption of NCSs where computational capabilities and energy storage are a tight constraint [14] to address. For the sake of simplicity, in this section, we limit the focus to dual rate solutions that address PID controllers. In the PID case, the derivative part deals with the high-frequency components of the control action, whereas the integral part deals with low-frequency components. Thus, PIDs are good candidates for dual-rate implementations, e.g., splitting them into a PI section operating at lower rate, complemented by a PD section operating at a higher rate.

In [15], the authors demonstrated the feasibility of a DR-DNCS implementing a PID controller when the remote and local sides are connected by means of Profibus DP; in this case, the slow section is in the master and the fast section in an intelligent slave. In [16], a delay-dependent DR-DNCS gain-scheduled PID controller is proposed, and a similar approach is adopted in [17], targeting the use of an Ethernet network where other nodes are connected, randomly generating a variable amount of background traffic, thus affecting latency and jitter of communications among the local and remote sections. The advantage of slowing down the rate of the remote controller section when over-the-network traffic must be minimized is underlined in [18], where the DR-DNCS approach is complemented by a prediction stage at the remote site (generally offering high computational and storage capabilities) is used for sending control actions in advance to be used if message losses are experienced. The use of dual-rate PID controller can be seen in the field of autonomous vehicles. As an example, authors in [19] propose a systematic combination of an Extended Kalman Filter (EKF) for step-ahead prediction and dual-rate dynamic controller in a wireless network system where the slow sensing rate is a technical constraint imposed by the camera used to sense the vehicle position, while the fast actuation rate is required to maintain the vehicle path. As a side consideration, dual rate control systems, where the plant output is sampled at a multiple of the control actions, have also been proposed for improving security in cyber-physical systems (CPSs), as shown in [20,21]. Additionally, an



**Fig. 2.** Block diagram of an NCS based on a dual rate controller.

interesting example of the use of dual-rate controllers in the field of cyber-physical systems is given in [22], where a dual-rate strategy is used to defend against sensor deception with an application example of a quadruple tank process.

Several approaches to converting the continuous time transfer functions of a PID controller into its equivalent dual-rate controller have been proposed, evaluating the effectiveness of leveraging different communication infrastructures. In [23], the authors report the discretization of a continuous-time PID controller, assumed to be already designed to meet required performances, with a dual-rate high-order-hold circuits.

As regards the LoRaWAN standard, it is designed to address the needs of classical IoT-like applications, targeting unidirectional traffic (i.e., from the field to the cloud) sporadically exchanged; accordingly, no relevant control applications have been reported so far. However, some efforts have been carried out to demonstrate that proprietary solutions based on the LoRa radio can be used for controlling systems with low dynamics, despite the low throughput, mainly because of the low data rate and stringent duty-cycle constraints for operating in unlicensed bands. As an example, consider [24], where LoRa radios operating in the 2.4 GHz ISM region are considered for controlling smart greenhouses.

## 2.2. DR-DNCS over LoRaWAN

As stated before, LoRaWAN can achieve long-range coverage, but severe limitations are imposed by the LoRa modulation. In particular, when operating in Europe in sub-GHz unlicensed bands, regional parameters limit the transmission duty-cycle  $DC$  (typically  $DC = 1\%$ ). Additionally, the useful  $DataRate$  is constrained in the range  $DataRate \in \{0.3, \dots, 5.5\}$  kbps, if the channel bandwidth is  $B_{UC} = 125$  kHz. Consequently, minimum update intervals must be greater than several tens of seconds.

To the best of the authors' knowledge, this work is the first attempt suggesting the implementation of a DR-DNCS to counteract these well-known limits of a fully standard LoRaWAN infrastructure. The DR approach can effectively reduce the duty-cycle while still ensuring good control capability. In the considered scenario, we assume that remote and local sides have been synchronized in advance (e.g., as in [25]) so that feedback and action are complemented by timestamps, referring to the measurement and actuation instants, respectively. However, even if delays are known and jitter is negligible, packet losses could greatly degrade the behavior of the controlled plant.

As a matter of fact, it is highly probable that throughput is reduced when considering deployments in large, complex, and heterogeneous areas with interference and noise. For this reason, a suitable strategy to increase the reliability of the distributed system must be introduced. In this paper, the authors propose using state-of-the-art predictions of future control actions to mitigate packet losses, combined with a novel mapping of the prediction data on LoRaWAN virtual channels to maintain short update intervals. In detail, it is suggested the use of quasi-orthogonality of messages sent with different spreading factors  $SFs$  (introduced in the next section) for transferring multiple control predictions, and the benefits are demonstrated.

Last, in this work, a simulation model is developed, inspired by the block diagram shown in Fig. 2. It has been kept simple but effective to demonstrate the superior performance of the DR with respect to the fully remote (single rate) PID implementation when real LoRaWAN protocol characteristics are taken into account, and data loss is present. Attention has been paid to consider possible impairments dictated by the LoRaWAN specifications. For instance, the packet loss in the uplink direction causing a missing data transfer also in the opposite direction is taken into account. Additionally, the effects of intentional and unintentional radiators (including other co-located LoRaWAN nodes), has been evaluated changing the total message loss.

### 3. Enabling technologies

### 3.1. LoRa and LoRaWAN

LoRa radios utilize sub-GHz wireless technology for implementing an enhanced version of chirp spread spectrum modulation. This modulation technique encodes multiple bits in a single symbol, which consists of an upchirp with a fixed bandwidth  $B_{UC}$  and a variable time duration  $T_{UC}$ . Specifically, there are  $2^{SF}$  initial chirp frequencies available, where the tunable parameter

Spreading Factor  $SF$  typically ranges from 7 to 12. There is a trade-off between data rate and reliability, because the relationship  $2^{SF} = B_{UC} \cdot T_{UC}$  must be satisfied. Additional details about LoRa can be found in [26]. The advantage of using LoRa is the quasi-orthogonality of frames sent using different  $SFs$  that, in principle, can be successfully detected even if overlapping in time and frequency.

LoRaWAN is a communication protocol stack built above the LoRa physical layer. It standardizes the media access control (MAC) layer, which is based on pure ALOHA. A star-of-stars topology is adopted; the star center hosts gateways (GWs) that relay wireless frames to and from the backend via an IP-based backbone. In the backend, logical entities known as the Network Server (NS), the Application Server (AS), and the Join Server (JS) manage the network (including adaptive data rate strategy, message deduplication, and confirmed message acknowledgment), facilitate end-user integration and handle security keys, respectively.

Node behavior varies based on its class. Compulsory Class A communication is initiated by the end device, which sends an uplink message to the backend if an event occurs. After the uplink message is sent, two reception windows,  $RX1$  and  $RX2$ , are available, allowing for the reception of acknowledgment or data transfer in the downlink direction. The time interval from the end of the uplink frame and the beginning of the windows,  $T_{RX1}$  and  $T_{RX2} = T_{RX1} + 1$  s, is fixed but configurable. Class B adds Beacon messages for time synchronization and allows additional reception windows, while Class C enables continuous listening. Several comprehensive reviews of the LoRaWAN features are available in literature, as the [27].

### 3.2. The dual-rate PID controller

In this subsection, the description of the PID dual-rate controller and symbols will be reported. Process model identification [28] and PID controller design and optimization [29] are out of the scope of this work.

Being  $G_P(s)$  the transfer function of a continuous process to be controlled, we assume it to be of the second order, as commonly happens in real-world applications (e.g., in the reference use case reported in Section 4):

$$G_P(s) = \frac{G_0}{(1 + s\tau_1)(1 + s\tau_2)}, \quad (1)$$

where  $\tau_1$  and  $\tau_2$  are the two time constants we suppose to be different from each other. The continuous time controller  $G_{PID}(s)$  can be described using the parallel form:

$$G_{PID}(s) = K_{PID} \left( 1 + \frac{1}{T_i s} + T_d s \right), \quad (2)$$

and can be designed using well-known techniques, in order to determine the gain  $K_{PID}$  and the integral and derivative time constants  $T_i$  and  $T_d$ , respectively.

The  $G_{PID}(s)$  can be represented in the discrete time domain and split into two subsystems: a slower component  $G_{PI,NT}(z^N)$  and a faster section  $G_{PD,T}(z)$  sections, where subscripts are adopted for highlighting the dependence from the update interval. The time constants of the initial continuous time controller are preserved, while the gains  $K_{PI}$  and  $K_{PD}$  are tuned to match the desired response. The slower part will be considered the remote part, while the faster component can also be denoted as the local part.

Since different rates are considered for discretizing the signals of interest, it must be noticed that the Z-transform of a stream  $x(kT)$  sampled at the shorter period  $T$  is as in Eq. (3):

$$X_T(z) = \sum_{k=0}^{\infty} x(kT) z^{-k}, \quad (3)$$

while the one of a stream  $x(kNT)$  sampled at the longer period  $NT$  is as in Eq. (4):

$$X_{NT}(z^N) = \sum_{k=0}^{\infty} x(kNT) z^{-kN}. \quad (4)$$

In particular, the resulting transfer function in the  $z^N$ -domain for the remote-side section is shown in Eq. (5):

$$G_{PI,NT}(z^N) = \frac{U_{PI,NT}(z^N)}{E_{PI,NT}(z^N)} = K_{PI} \frac{z^N - \left(1 - \frac{NT}{T_i}\right)}{z^N - 1}. \quad (5)$$

$U_{PI,NT}(z^N)$  is the Z-transform of the control action, and  $E_{PI,NT}(z^N)$  is the Z-transform of the error signal. Indeed, being  $R_{NT}(z^N)$  and  $Y_{NT}(z^N)$  the Z-transform of the reference and feedback signals, respectively, the error signal can be represented by:

$$E_{PI,NT}(z^N) = [R_{NT}(z^N) - Y_{NT}(z^N)]. \quad (6)$$

The update interval on the local side is  $N$  time faster; therefore, a rate conversion is required. Since only step references are considered, the rate converter is a digital Zero Order Hold (Z.O.H.)  $RC_{NT,T}(z)$ :

$$RC_{NT,T}(z) = \frac{U_{PI,T}(z)}{[U_{PI,NT}]^N(z)} = \frac{1 - z^{-N}}{1 - z^{-1}} = 1 + z^{-1} + \dots + z^{-(N-1)}. \quad (7)$$

In Eq. (7),  $U_{PI,T}(z)$  is the Z-transform of the fast-rate input of the local side PD controller, obtained from the slow-rate remote side PI controller output via the expanding operator  $[\cdot]^N$ . In other words,  $[U_{PI,NT}]^N(z)$  is the Z-transform of the upsampled version of the slow-rate PI control action. Formally, it can be assumed:

$$[U_{PI,NT}]^N(z) = \sum_{k=0}^{\infty} \overline{u_{PI,NT}}[k]z^{-k} = 1 + z^{-1} + \dots + z^{-(N-1)}, \quad (8)$$

and:

$$\overline{u_{PI,NT}}[k] = \begin{cases} u_{PI,NT}[k/N] & \text{if } k = iN, i \in \mathbb{Z}, \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Finally, the PD controller operating on the local side is described by Eq. (10):

$$G_{PD,T}(z) = \frac{U_{PD,T}(z)}{U_{PI,T}(z)} = K_{PD} \frac{z \left(1 + \frac{T_d}{T}\right) - \frac{T_d}{T}}{z}, \quad (10)$$

where  $U_{PD,T}(z)$  is the actual action controlling the plant.

Due to the unreliable nature of the wireless communication infrastructure, it is expected that packet losses can occur on both the uplink and downlink directions. Moreover, the LoRaWAN medium access strategy implies that losing an uplink frame also results in missing the transmission opportunity for the downlink one (which carries the control information). In order to overcome the poor effectiveness of message replicas (because of the duty-cycle limitations and the bursty nature of interferences), still minimizing the impact of losses, future control actions are predicted and sent to the local controller. In particular, mimicking the approach described in [18],  $M$  outputs of the PI controller in the remote side (where computation power is not a constrain) are estimated in advance from the corresponding setpoint samples  $r_{NT}$ , which are assumed to be known, and the actual  $y_{NT}$  and the predicted  $\langle y_{NT} \rangle$  plant outputs.

Moving in the time domain, it means that the local PD controller action  $u_{PD,T}$  and  $M$  predictions  $\langle u_{PD,T} \rangle$  are computed also at the remote side according to the Eqs. (11) and (12), respectively. It is assumed that the expanding operator has been applied so  $k$  is the sample index on the remote side, while  $k + j, j = 0..N - 1$  is the corresponding sample index on the local side.

$$u_{PD,T}[k + j] = K_{PD} \left\{ \left(1 + \frac{T_d}{T}\right) \overline{u_{PI,NT}}[k + j] - \left(\frac{T_d}{T}\right) \overline{u_{PI,NT}}[k + j - 1] \right\}, \quad (11)$$

$$\langle u_{PD,T}[k + j + iN] \rangle = K_{PD} \left\{ \left(1 + \frac{T_d}{T}\right) \langle \overline{u_{PI,NT}}[k + j + iN] \rangle - \left(\frac{T_d}{T}\right) \langle \overline{u_{PI,NT}}[k + j - 1 + iN] \rangle \right\}, \quad i = 0..M - 1 \quad (12)$$

Subsequently, a state observer, running at period  $T$ , permits to retrieve the future estimated values of the plant state  $\langle x_T \rangle$  and plant output  $\langle y_T \rangle$ , given the state matrix  $\mathbf{A}$ , the input matrix  $\mathbf{B}$  and the output matrix  $\mathbf{C}$ , as shown in Eq. (13):

$$\begin{cases} \langle x_T[k + j + 1 + i] \rangle = \mathbf{A} \langle x_T[k + j + i] \rangle + \mathbf{B} \langle u_{PD,T}[k + j + iN] \rangle & i = 0..M - 1 \\ \langle y_T[k + j + 1 + i] \rangle = \mathbf{C} \langle x_T[k + j + 1 + i] \rangle & i = 0..M - 1 \end{cases} \quad (13)$$

Finally, the PI controller action  $u_{PI,NT}$  and  $M$  predictions  $\langle u_{PI,NT} \rangle$  are evaluated using the sensed and estimated plant outputs at period  $NT$ ,  $y_{NT}$  and  $\langle y_{NT} \rangle$  respectively, according to the Eqs. (14), (15) and (16):

$$u_{PI,NT}[k] = K_{PI} \left\{ u_{PI,NT}[k - 1] + (r_{NT}[k] - \langle y_{NT}[k] \rangle) - \left(1 - \frac{NT}{T_i}\right) (r_{NT}[k - 1] - y_{NT}[k - 1]) \right\} \quad (14)$$

$$\begin{aligned} \langle u_{PI,NT}[k + i + 1] \rangle = K_{PI} \left\{ u_{PI,NT}[k + i] + (r_{NT}[k + i + 1] - \langle y_{NT}[k + i + 1] \rangle) + \right. \\ \left. - \left(1 - \frac{NT}{T_i}\right) (r_{NT}[k + i] - y_{NT}[k + i]) \right\}, \quad i = 0 \end{aligned} \quad (15)$$

$$\begin{aligned} \langle u_{PI,NT}[k + i + 1] \rangle = K_{PI} \left\{ \langle u_{PI,NT}[k + i] \rangle + (r_{NT}[k + i + 1] - \langle y_{NT}[k + i + 1] \rangle) + \right. \\ \left. - \left(1 - \frac{NT}{T_i}\right) (r_{NT}[k + i] - \langle y_{NT}[k + i] \rangle) \right\}, \quad i = 1..M - 1 \end{aligned} \quad (16)$$

Therefore, accepting to transmit a larger LoRaWAN payload, the  $\langle u_{PI,NT} \rangle$  predictions can be sent along the actual  $u_{PI,NT}$  value so that in the case of a future downlink message loss, the predictions can be used to update the PD controller outputs instead of simply keeping the previous value.

#### 4. The reference use case

##### 4.1. The plant: two coupled water tanks

As stated before, due to its limited throughput, LoRaWAN could be used to control systems with comparable slow transient and possibly stable behavior. Without losing generality, in this work the management of coupled water tanks [30], that exploits a



**Table 1**The possible uplink and downlink frames duration and payload length depending on the  $SF$  and  $M$  parameters.

| Uplink frame payload   |     |            |                 |
|------------------------|-----|------------|-----------------|
| $SF$                   |     | Length [B] | Time on Air [s] |
| 7                      |     | 16         | 0.067           |
| 8                      |     | 16         | 0.123           |
| 9                      |     | 16         | 0.226           |
| Downlink frame payload |     |            |                 |
| $SF$                   | $M$ | Length [B] | Time on Air [s] |
| 7                      | 0   | 16         | 0.067           |
| 7                      | 3   | 40         | 0.103           |
| 7                      | 5   | 56         | 0.128           |
| 7                      | 10  | 96         | 0.185           |
| 7                      | 20  | 176        | 0.302           |
| 8                      | 1   | 24         | 0.144           |
| 9                      | 0   | 16         | 0.226           |

Class A end device transmitting sensor data in the uplink direction and receiving control actions and predictions in the downlink one, is considered. For example, the use of the IoT paradigm has already been proposed for addressing the needs of innovative, low budget, aquaculture, and fish farming solutions [31]. In such an application scenario, it is common to deal with coupled tanks arrangement [32]. In natural water flow systems, a basic control strategy involves a circulation pump that introduces and discharges natural supply water. Nowadays, it is also expected to have recirculating systems [33], that minimize the problem of polluting the water at the cost of increased complexity due to the additional physical and biological filtering actions. However, similar models are also useful for describing the management of hydroelectric power plant basins, artificial snow-making, and/or irrigation systems.

The problem considered here is the series piping of two non-interacting tanks with uniform cross-sectional area  $A_1 = 50 \text{ m}^2$  and  $A_2 = 70 \text{ m}^2$ , filled with a constant density liquid that flows through linear resistances  $R_1 = R_2 = 10 \text{ s m}^{-2}$  (laminar range). The transfer function relating the second tank level to the first tank input flow is a second-order function  $G_P(s)$ :

$$G_P(s) = \frac{G_0}{(1 + s\tau_1)(1 + s\tau_2)} = \frac{10}{(1 + s \cdot 500)(1 + s \cdot 700)}, \quad (17)$$

where  $\tau_1 = R_1 \cdot A_1 = 500 \text{ s}$  and  $\tau_2 = R_2 \cdot A_2 = 700 \text{ s}$  are the two time constants and  $G_0 = R_2$  is the gain, respectively [34].

The slowest tolerable sampling period  $NT_{max}$  has been evaluated according to:

$$NT_{max} = \min \left[ \min(\tau_1, \tau_2), \frac{1}{f} \cdot s.t. \frac{G_0}{G_P(2\pi i f)} = 100 \right] = 50 \text{ s}. \quad (18)$$

As stated in Eq. (18), the necessary sample period is 50 s; however, to provide a sampling time margin and, more importantly, to put more stress on the LoRaWAN link, the sampling period has been set to  $NT = 30 \text{ s}$ . In fact, while taking the duty-cycle constraint into consideration, 30 s represents a lower bound for the message' over the air time.

It is well-known that higher  $SFs$  can offer better noise immunity and/or larger coverage due to the additional processing gain, at the cost of lower data rate. Accordingly, the payload must be limited to satisfy the duty-cycle constraint. In particular, it means that when operating in the EU868 region with a 1% duty-cycle and bandwidth  $B_{UC} = 125 \text{ kHz}$ , if  $SF = 7$ , the maximum user payload length is 174 B, that decreases to 85 B for  $SF = 8$ , down to 31 B for  $SF = 9$ . On the other hand, higher  $SF$  values are not supported at all. In this work, the  $RX1$  window has been selected for downlink reception, which utilizes the same  $SF$  as the uplink frame. Nevertheless, the long sampling period  $NT$  of the regulated plant (30 s) would also allow operation in  $RX2$ , with frame parameters being predetermined in advance, e.g. when  $T_{RX1} = 1 \text{ s}$  and thus  $T_{RX2} = 2 \text{ s}$ , as often happens.

The payload is split into multiple fields; the first one is the timestamp (coded as a 8 B-long epoch time), related to the sense or actuation time instants depending if it is dealing with uplink or the downlink frame; successive field(s) are the feedback/control values (encoded as 8 B-long double-precision floating point numbers). Data is encapsulated in the LoRaWAN frame as depicted in Fig. 3, where the header and trailer blocks are representative of all the message fields specified by the LoRaWAN protocol and that cannot be filled with application specific data. The actual frame duration, also considering the number of repetitions  $M$ , is reported in Table 1. Considering a regional limitation of  $DC = 1\%$ , then the maximum message length should have a time on air lower than 300 ms. For this reason, it has to be highlighted that the case  $M = 20$  does not satisfy the duty-cycle limitation, and it is mentioned only for comparison purposes only, as reported in Section 5.

#### 4.2. The controller

The role of the controller is to modify the input flow to obtain the second tank height imposed by the set point. In the proposed solution, the control action is provided by a LoRaWAN downlink frame; for this reason, it is received after the fixed interval  $T_{RX1}$  or  $T_{RX2}$  (being  $T_{RX1} < T_{RX2}$ ) from the end of the corresponding uplink frame. Therefore, it is assumed the actual plant model is subject to a delay, as in the following:

$$G_P(s) \exp(-sT), \quad (19)$$

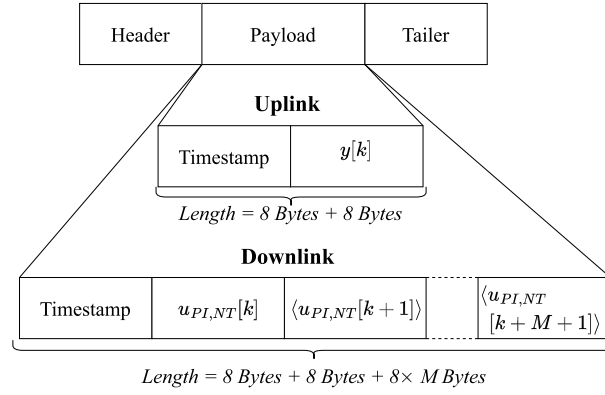


Fig. 3. LoRaWAN custom payload encapsulation.

Table 2

PID controllers implementations and parameters.

| Controller |                                                                                                   | Parameters                                                                                    |
|------------|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| CT-PID     | $K_p \left( 1 + \frac{1}{T_i} \frac{1}{s} + T_d \frac{s}{1 + \frac{T_d}{N}s} \right)$             | $K_p = 3.7940$<br>$T_i = 543.4164$<br>$T_d = 174.1937$<br>$N = 2.8550$                        |
| DR-PID     | Eq. (5)                                                                                           | $K_{PI} = 0.8334$<br>$T_i = 584.4444$<br>$NT = 30\text{ s}$                                   |
|            | Eq. (10)                                                                                          | $K_{PD} = 0.5685$<br>$T_d = 146.1111$<br>$T = 3\text{ s}$                                     |
| SR-PID     | $K_p \left( 1 + \frac{1}{T_i} \frac{T_s}{z-1} + T_d \frac{z-1}{T_s + (z-1)\frac{T_s}{N}} \right)$ | $K_p = 5.8145$<br>$T_i = 690.8515$<br>$T_d = 103.2732$<br>$N = 3.3047$<br>$T_s = 30\text{ s}$ |

where  $T \geq T_{RX2} + T_{oA_{DLmax}} > T_{RX1} + T_{oA_{DLmax}}$  and  $T_{oA_{DLmax}}$  is the longest downlink frame air duration. As already reported in 4.1  $NT = 30\text{ s}$ , and assuming  $T = 3\text{ s}$ , it means that  $N = 10$ .

Considering the scope of the paper, not focused on finding an optimal control strategy, it has been decided to identify the  $G_{PID}(s)$  parameters leveraging on the *PID Tuner* tool available in the *Matlab Control System Toolbox* from Mathworks. The *PID Tuner* toolbox automatically provides an initial solution for the controller parameters trading off performance and reliability. The user can examine the controller performance in both time and frequency domains and interactively refine the controller's performance, e.g., to adapt loop bandwidth and phase margin or to improve the reference tracking or disturbance rejection.

In Table 2 the PID controllers expressions and parameters have been reported. The table reports three purposely designed controllers:

- CT-PID: continuous time controller;
- DR-PID: the dual rate PID controller where the derivative function has a sampling period of  $T = 3\text{ s}$  while the integrative part of the controller has  $NT = 30\text{ s}$ ;
- SR-PID: a single rate discrete time PID controller, designed to obtain an identical step response to the CT-PID controller with a sampling time  $T_s = NT = 30\text{ s}$ , as the controller should be implemented with an NCS structure using the LoRaWAN.

The three different implementations of the PID controllers have an identical step response by design, in the absence of transmission losses. The CT-PID controller response (black line) in the time domain, when a step input is applied (dotted red line), is reported in Fig. 4. The same picture reports the difference between the step responses of the DR-PID (blue line) and the SR-PID (red line) controllers against the CT-PID controller. In this comparison, the SR-PID is implemented as a simple discrete time PID controller with a single sampling time equivalent to  $T$ . Moreover, this comparison does not consider the presence of disturbances on the channel; on this basis, the effects of the observer and predictor of the DR-PID are not considered. It could be noted that, the three controller's parameters assure a system step response with a steady state difference lower than 1%, and a transient response with a maximum error of about 30%. Both the step responses present an average error of 1.3%. Moreover, it can be highlighted that the DR-PID and the SR-PID responses perfectly overlap.



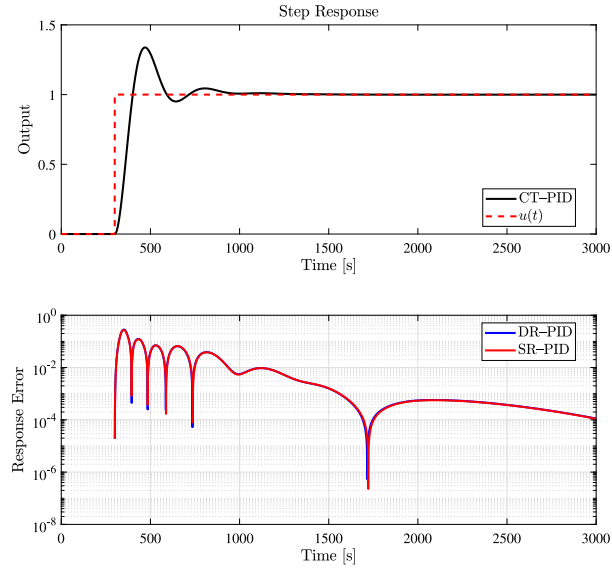


Fig. 4. Step response of the continuous time PID (CT-PID) and response errors of Single Rate (SR-PID) and Dual Rate (DR-PID) implementations with respect to the CT-PID.

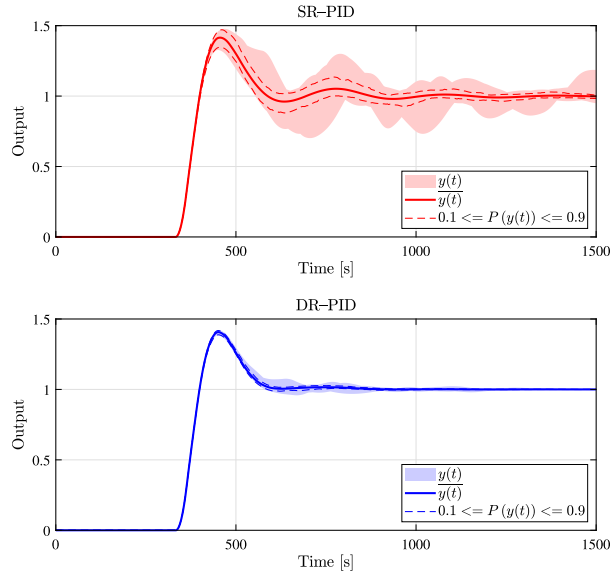


Fig. 5. Comparison of the DR-PID and SR-PID, both considered in the case of ten predictions ( $M = 10$ ), under interferences with  $l_u = 20\%$  and  $l_d = 20\%$ . The colored regions represent the maximum and minimum values while the dashed lines represent 10% percentile and 90% percentile.

## 5. Results

Superiority of the DR-PID over the reference SR-PID, in presence of interferences degrading the LoRaWAN communication channel, is clearly shown in Fig. 5. In particular, the SR-PID used in the comparison is the same as previously described and employed in the comparison with the CT-PID. DR-PID and SR-PID comparison has been carried out considering the case of ten predictions ( $M = 10$ ) for both the DR-PID and the SR-PID; further consideration and comparison on the effect on the presence of different predictions will be given in the following sections. In this work,  $l_u$  and  $l_d$  represents the uplink and downlink losses, respectively. These two parameters synthesize the overall losses of the messages across the LoRaWAN communication, including the message path across the communication infrastructure connecting the back-end devices and services with the GW and the end user. In fact, these links principally consist of fast and reliable wired connections, leveraging by transport protocols such as TCP, which allow for retransmission whenever needed. Moreover, their influence in terms of bandwidth and latency has a minor impact

on the overall transmission rate of the LoRaWAN infrastructure, as already demonstrated in the literature (e.g., [35,36]). Referring to the previously cited Fig. 5, the particular case where  $l_u = l_d = 20\%$  has been considered.

The figure reports the mean response  $\bar{y}(t)$  calculated across 100 simulations under the previously described loss conditions; moreover, the colored regions represent the maximum and minimum values while the dashed lines represent 10% percentile and 90% percentile of the 100 simulation runs. As it is possible to observe, even though the two control systems under ideal channel conditions present an almost identical response, it is clear that even with these small loss conditions, the dual-rate PID approach assures a more consistent systems behavior.

For completeness, an exhaustive evaluation of obtainable performance has been carried out as reported in the following Section 5.2, leveraging on the *ERMS* figure of merit introduced in Section 5.1.

### 5.1. The figure of merit

In order to evaluate the deviation from the SR-PID reference (step) response in ideal conditions, many different metrics have been analyzed.<sup>1</sup> However, the most concise and accurate information is provided by the root mean square of the response error *ERMS*, representing in a single value the overall error level across the observed interval of interest. In particular, different time windows  $W_i$ ,  $i = 1..5$  have been identified, as depicted in Fig. 6. With the aim of ensuring proper time alignment of the different simulation runs, without losing generality, beginning  $t_0$  of the first window  $W_1$  is marked by the actual response crossing the threshold  $TH_{t_0} = 10\%$  of the steady state value for the first time. Similarly, the ending  $t_1$  of the first window  $W_1$  is marked by the response crossing the threshold  $TH_{t_1} = 105\%$ . Thus, windows  $W_i$ ,  $i = 1..4$  can be defined by Eq. (20):

$$\begin{aligned} W_i &= t_i - t_{i-1}, \quad i = 1..4, \\ t_i &= t_{i-1} + T_W, \quad i = 1..4, \\ T_W &= t_1 - t_0 \end{aligned} \quad (20)$$

and the corresponding *ERMS* <sub>$i$</sub> ,  $i = 1..4$  is computed as in Eq. (21):

$$ERMS_i = \sqrt{\frac{\sum_{k=t_{i-1}}^{t_i} (\text{DR-PID}[k] - \text{SR-PID}[k])^2}{T_W}}, \quad i = 1..4. \quad (21)$$

Finally, the last window  $W_5$  starts at  $t_4$  and lasts until the end of the simulation  $t_{END}$ , so that Eq. (22) holds:

$$ERMS_5 = \sqrt{\frac{\sum_{k=t_4}^{t_{END}} (\text{DR-PID}[k] - \text{SR-PID}[k])^2}{t_{END} - t_4}}. \quad (22)$$

### 5.2. The DR-PID performance

Different channel conditions have been accounted for changing the uplink and downlink losses in the set  $l_u, l_d \in \{10, 20, 30\}\%$  when  $SF = 7$ . Therefore the total loss  $l_t$  can be evaluated as in Eq. (23):

$$l_t = 1 - (1 - l_u)(1 - l_d). \quad (23)$$

Indeed, losing the uplink message also causes the loss of the downlink one. A subset of relevant statistics regarding *ERMS* <sub>$i$</sub> , with  $i = 1..4$ , is resumed in Table 3, whereas *ERMS*<sub>5</sub> is reported in Table 4. These results address a population of  $N_{SIM} = 1000$  simulation runs, each one ending at  $t_{END} = 3000$  s, when the uplink/downlink losses are the same, resulting in  $l_t \in \{19, 36, 51\}\%$ .

It must be noticed that the *SF* is not very relevant for the analysis since we are not considering the Signal to Noise Ratio (SNR) of the channel but the loss values. Consequently, provided that the duty-cycle constraint is satisfied, the same results also apply for higher *SF* values.

## 6. Discussion and alternatives

Other than simply defining the number of predictions as a function of the consecutive losses, in this work  $M$  is evaluated as a trade-off between the effective improvements of the control action with respect to the LoRaWAN constraints. It is evident that when  $M = 0$ , message losses greatly degrade the behavior of the control system, especially during the transient (e.g., consider *ERMS* <sub>$i$</sub> ,  $i = 1..3$ ), preventing the use of LoRaWAN for control applications in noisy channels. On the other hand, sending predictions, i.e.,  $M \neq 0$ , requires larger payloads and message durations that, in turn, increase the probability of collisions with other uplink messages and result in higher power consumption. The latter point could be less relevant if the power supply from the grid is available, as for the actuator in the controller, but can be significant for the plant sensor, possibly battery powered (despite the time spent for receiving is upper bounded by the duty-cycle constrain). For all these reasons, the choice of the  $M$  value must be carefully considered.

<sup>1</sup> A resume of all the metrics is publicly available at [doi:10.5281/zenodo.14833837](https://doi.org/10.5281/zenodo.14833837).

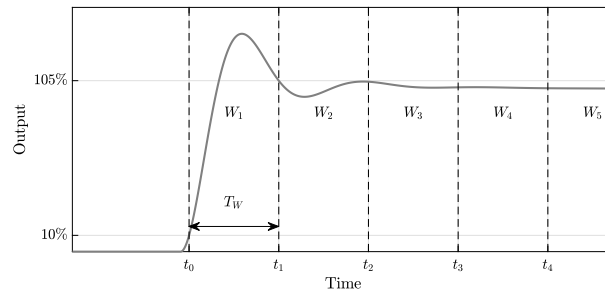


Fig. 6. Time windows  $W_i, i = 1..5$  used for computing the  $ERMS_i$ .

Table 3

$ERMS$  values for different windows  $W_i, i = 1..4$  with  $M \in \{0, 3, 5, 10, 20\}$  and increasing loss levels  $l_i$ , under the condition  $l_u = l_d$ . For each window  $\mu, \sigma, Q_{10}, Q_{90}$  are the average, standard deviation, 10% percentile, 90% percentile statistics calculated over 1000 simulation runs.

| M  | $l_i$ | $ERMS_1$ |          |          |          | $ERMS_2$ |          |          |          | $ERMS_3$ |          |          |          | $ERMS_4$ |          |          |          |
|----|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|    |       | $\mu$    | $\sigma$ | $Q_{10}$ | $Q_{90}$ | $\mu$    | $\sigma$ | $Q_{10}$ | $Q_{90}$ | $\mu$    | $\sigma$ | $Q_{10}$ | $Q_{90}$ | $\mu$    | $\sigma$ | $Q_{10}$ | $Q_{90}$ |
| 0  | 19.0  | 0.492    | 0.193    | 0.373    | 0.666    | 0.132    | 0.116    | 0.069    | 0.225    | 0.049    | 0.033    | 0.027    | 0.074    | 0.019    | 0.015    | 0.007    | 0.030    |
| 0  | 36.0  | 0.690    | 0.416    | 0.380    | 1.221    | 0.359    | 0.506    | 0.078    | 0.826    | 0.138    | 0.247    | 0.032    | 0.269    | 0.057    | 0.149    | 0.010    | 0.106    |
| 0  | 51.0  | 0.999    | 0.741    | 0.441    | 1.857    | 1.103    | 1.870    | 0.115    | 2.605    | 0.878    | 2.474    | 0.047    | 1.930    | 0.587    | 1.715    | 0.019    | 1.250    |
| 3  | 19.0  | 0.367    | 0.010    | 0.355    | 0.374    | 0.073    | 0.008    | 0.064    | 0.080    | 0.037    | 0.003    | 0.034    | 0.039    | 0.014    | 0.001    | 0.013    | 0.015    |
| 3  | 36.0  | 0.371    | 0.066    | 0.350    | 0.374    | 0.075    | 0.060    | 0.052    | 0.079    | 0.040    | 0.025    | 0.030    | 0.043    | 0.015    | 0.009    | 0.012    | 0.016    |
| 3  | 51.0  | 0.413    | 0.245    | 0.347    | 0.464    | 0.141    | 0.361    | 0.047    | 0.232    | 0.080    | 0.263    | 0.026    | 0.116    | 0.046    | 0.260    | 0.011    | 0.057    |
| 5  | 19.0  | 0.367    | 0.007    | 0.357    | 0.374    | 0.073    | 0.008    | 0.063    | 0.080    | 0.037    | 0.003    | 0.034    | 0.040    | 0.014    | 0.001    | 0.013    | 0.015    |
| 5  | 36.0  | 0.362    | 0.015    | 0.351    | 0.373    | 0.070    | 0.038    | 0.054    | 0.078    | 0.037    | 0.013    | 0.031    | 0.042    | 0.014    | 0.005    | 0.012    | 0.016    |
| 5  | 51.0  | 0.369    | 0.080    | 0.346    | 0.371    | 0.112    | 0.272    | 0.047    | 0.131    | 0.068    | 0.472    | 0.027    | 0.064    | 0.036    | 0.406    | 0.011    | 0.033    |
| 10 | 19.0  | 0.360    | 0.007    | 0.351    | 0.366    | 0.073    | 0.007    | 0.063    | 0.079    | 0.036    | 0.003    | 0.034    | 0.039    | 0.014    | 0.001    | 0.013    | 0.015    |
| 10 | 36.0  | 0.355    | 0.009    | 0.344    | 0.366    | 0.068    | 0.011    | 0.054    | 0.079    | 0.036    | 0.006    | 0.031    | 0.042    | 0.014    | 0.002    | 0.012    | 0.016    |
| 10 | 51.0  | 0.351    | 0.048    | 0.339    | 0.361    | 0.069    | 0.038    | 0.050    | 0.080    | 0.036    | 0.019    | 0.027    | 0.044    | 0.015    | 0.011    | 0.011    | 0.021    |
| 20 | 19.0  | 0.360    | 0.007    | 0.352    | 0.366    | 0.073    | 0.007    | 0.063    | 0.080    | 0.037    | 0.003    | 0.034    | 0.040    | 0.014    | 0.001    | 0.013    | 0.015    |
| 20 | 36.0  | 0.354    | 0.009    | 0.343    | 0.366    | 0.067    | 0.012    | 0.052    | 0.078    | 0.036    | 0.006    | 0.030    | 0.041    | 0.014    | 0.002    | 0.012    | 0.016    |
| 20 | 51.0  | 0.350    | 0.009    | 0.340    | 0.361    | 0.067    | 0.015    | 0.049    | 0.078    | 0.036    | 0.012    | 0.027    | 0.046    | 0.015    | 0.010    | 0.011    | 0.020    |

Table 4

$ERMS$  values in the  $W_5$  with  $M \in \{0, 3, 5, 10, 20\}$  and increasing loss levels  $l_i$ .  $\mu, \sigma, Q_{10}, Q_{90}$  are the average, standard deviation, 10% percentile, 90% percentile statistics calculated over 1000 simulation runs.

| M  | $l_i$ | $ERMS_5$ |          |          |          |
|----|-------|----------|----------|----------|----------|
|    |       | $\mu$    | $\sigma$ | $Q_{10}$ | $Q_{90}$ |
| 0  | 19.0  | 0.004    | 0.005    | 0.001    | 0.009    |
| 0  | 36.0  | 0.013    | 0.028    | 0.002    | 0.027    |
| 0  | 51.0  | 0.215    | 0.763    | 0.005    | 0.388    |
| 3  | 19.0  | 0.002    | 0.000    | 0.002    | 0.002    |
| 3  | 36.0  | 0.002    | 0.003    | 0.002    | 0.003    |
| 3  | 51.0  | 0.010    | 0.047    | 0.002    | 0.018    |
| 5  | 19.0  | 0.002    | 0.000    | 0.002    | 0.002    |
| 5  | 36.0  | 0.002    | 0.001    | 0.002    | 0.002    |
| 5  | 51.0  | 0.009    | 0.089    | 0.002    | 0.010    |
| 10 | 19.0  | 0.002    | 0.000    | 0.002    | 0.002    |
| 10 | 36.0  | 0.002    | 0.001    | 0.002    | 0.002    |
| 10 | 51.0  | 0.003    | 0.003    | 0.002    | 0.004    |
| 20 | 19.0  | 0.002    | 0.000    | 0.002    | 0.002    |
| 20 | 36.0  | 0.002    | 0.001    | 0.002    | 0.002    |
| 20 | 51.0  | 0.003    | 0.002    | 0.002    | 0.004    |

### 6.1. On the choice of the prediction number $M$

The advantages offered by increasing the number of predictions  $M$  is evaluated by means of the metric  $\eta_i(a, b)$ , as reported in Eq. (24).

$$\eta_i(a, b) = 1 - \frac{\mu_{ERMS_i@M=b}}{\mu_{ERMS_i@M=a}}, i = 1..5. \quad (24)$$

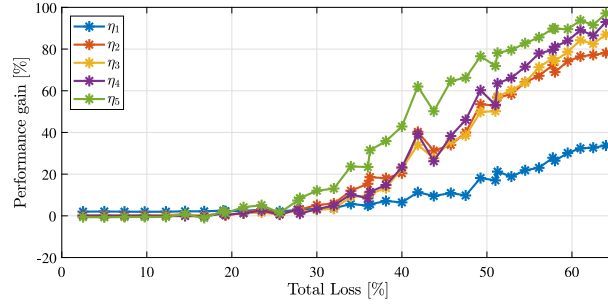


Fig. 7. The performance gain  $\eta_i(3, 10)$ ,  $i = 1..5$  as function of the total loss  $l_t$ .

This metric compares the  $i$ th  $\mu_{ERMS_i}$ , namely the mean *ERMS* value as previously described at the  $i$ th window calculated across 1000 simulations, in the case  $M = b$  with respect to the case  $M = a$ . Because the  $\eta_i(a, b)$  is calculated within the same step response time window, this represents the performance in terms of average error of two different prediction strategies. Its value towards the unity implies an increase in performance, meaning that the  $b$ -scenario obtains an error compared to the reference case quite smaller compared to the  $a$ -case. On the other hand, when the parameter assumes value towards zero the performance of the compared cases are quite similar and no performance improvement is obtained.

Analyzing data in Tables 3 and 4, it is evident that, in the simulated scenarios, the improvement reaches a plateau for  $M = 10$ , since  $\eta_i(10, 20)$ ,  $i = 1..5$  becomes very small. Accordingly, the subsequent analysis has been limited to  $\eta_i(3, 10)$ , as shown in Fig. 7, where additional total loss  $l_t$  values have been accounted. As expected, the improvement is higher when the integral action of the controller, performed remotely, is more relevant (i.e.,  $i \neq 1$ ). As it is possible to observe both from Table 3 and Fig. 7, in the case of low loss values, namely  $l_t < 25\%$ , the impact of additional predictions ( $M > 0$ ) is negligible; for both cases  $M = 3$  and  $M = 10$  the average RMS error value is in the same order of magnitude. As a consequence, it can also be stated that, in the presence of moderate interference (e.g.,  $l_t < 25\%$ ), a good trade-off between the message duration and the control reliability is offered by  $M = 3$ , whereas in very noisy scenarios the use of  $M = 10$  could represent a better solution.

## 6.2. Exploiting LoRa peculiarities

As already affirmed in Section 3.1, the higher the *SF*, the lower the data rate but the better the noisy immunity, and the higher the coverage. Moreover, transmissions occurring with different *SFs* are quasi-orthogonal, possibly allowing for the superposition in time and frequency of these messages, similar to virtual channels. Despite it being out of the paper's scope, authors have already demonstrated that enhanced nodes and gateways, capable of receiving superposed frames having different *SFs*, can be devised and implemented (e.g., as in [37]). Indeed, the overlapping frames sent using diverse *SFs* allow parallelization of message repetition, providing some form of redundancy. Therefore, from the data rate point of view, a trade-off exists, suggesting sending a larger number of predictions with lower *SF* values and vice versa. On the other hand, considering the noise immunity point of view, a lower number of predictions can be sent over higher *SFs*, still ensuring that actuation would arrive at the local control node within the expected time frame. What is proposed is to take advantage of both aspects transmitting concurrently messages that use different *SFs*.

In Fig. 8, the obtainable performance increase due to the use of *SF* multiplexing is reported as a function of the total losses  $l_t$ . In particular, the metrics  $\gamma_i$  are calculated as in Eq. (25), where the average value  $\mu_{ERMS_i}@SF7$  (when all the  $M = 3$  predictions are sent using *SF* = 7), is considered as the performance baseline against the average value  $\mu_{ERMS_i}@SF789$  (when the spreading factor - prediction pairs are  $(SF, M) = \{(7, 3), (8, 1), (9, 0)\}$ , respectively).

$$\gamma_i = 1 - \frac{\mu_{ERMS_i}@SF789}{\mu_{ERMS_i}@SF7}, i = 1..5. \quad (25)$$

This comparison generally reports a decrease in the  $ERMS_i$ , in the case of the same high total losses ( $l_t$ ), when transmissions leveraging different *SF* values are employed. Also, in this case, the increase in performance is lower for  $\gamma_1$ , calculated on the first window  $W_1$ , where the step response presents the maximum slew rate. Moving towards the end of the transient response ( $W_4$  and  $W_5$ ), where the remote integrative control has the major impact on the system response error reduction, the use of multiple *SF* decreases the *ERMS* value of about 60% starting from  $l_t$  greater than 35%.

These results, somehow similar to the performance gain obtainable moving from  $M = 3$  to  $M = 10$  when using the sole *SF* = 7, confirm that exploiting the LoRa peculiarity of *SFs* quasi-orthogonality could be an effective alternative approach for preserving the reliability of the control, especially when the losses are high. On the other hand, it must be highlighted that the overall amount and the accuracy of predictions available can be actually lower than in Section 6.1, when losing consecutive messages, e.g., causing the  $\gamma_2$  to be negative (at least for lower  $l_t$  values).

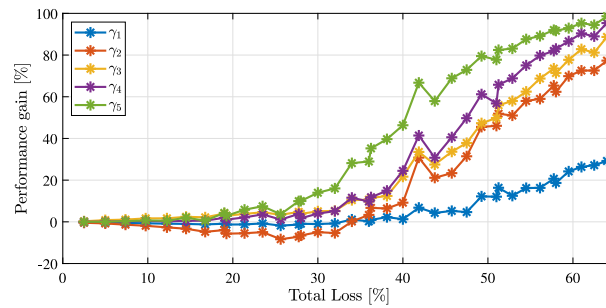


Fig. 8. The performance gain  $\gamma_i, i = 1..5$  as function of the total loss  $l_t$ .

## 7. Conclusion

In this work, the use of a PID-based DR-NCS is suggested to allow for a CaaS architecture suitable for LoRaWAN communication infrastructure. Although LoRaWAN is designed for applications that require sporadic transmissions in the uplink direction, the wide coverage it offers, complemented by the hardware low-cost and the well-defined backend, make it attractive for the CaaS paradigm. We demonstrated that the low transmission rate can be somehow mitigated by the dual-rate strategy, implementing a PI control on the remote side and a PD control on the local side. Additionally, adding a prediction stage allows increasing the reliability against message losses by means of future control actions sent in advance. The purposely designed simulator, implemented in Simulink, permitted to quantitatively evaluate the performance improvement in terms of average error in different intervals of the overall system step response. In particular,  $M = 3$  predictions are enough for a moderately noisy environment. Finally, it has also been demonstrated that simultaneous transmission of messages sent using different SF values allows to obtain similar performance again, but with the advantage of a better noise immunity of higher SFs, due to the additional processing gain.

## CRedit authorship contribution statement

**Salvatore Dello Iacono:** Writing – review & editing, Writing – original draft, Visualization, Software, Formal analysis, Data curation, Conceptualization. **Alessandro Depari:** Writing – review & editing, Writing – original draft, Validation, Formal analysis, Conceptualization. **Paolo Ferrari:** Writing – original draft, Investigation, Formal analysis. **Alessandra Flammini:** Supervision, Funding acquisition. **Stefano Rinaldi:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Emiliano Sisinni:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Salvatore Dello Iacono reports financial support was provided by Government of Italy Ministry of Education University and Research. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This study was carried out within the MOST Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) - Missione 4, Componente 2, Investimento 1.4 - D.D. 1033 17/06/2022, CN00000023), Spoke 5 “Light Vehicle and Active Mobility”. This manuscript reflects only the authors’ views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

## Data availability

The original data is available at [Zenodo](https://zenodo.org/).

## References

- [1] X. Ge, F. Yang, Q.-L. Han, Distributed networked control systems: A brief overview, *Inform. Sci.* 380 (2017) 117–131, <http://dx.doi.org/10.1016/j.ins.2015.07.047>.
- [2] P. Skarin, W. Tärneberg, K.-E. Årzén, M. Kihl, Control-over-the-cloud: A performance study for cloud-native, critical control systems, in: 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing, UCC, 2020, pp. 57–66, <http://dx.doi.org/10.1109/UCC48980.2020.00025>.
- [3] O. Givvehchi, J. Imtiaz, H. Trsek, J. Jasperneite, Control-as-a-service from the cloud: A case study for using virtualized PLCs, in: 2014 10th IEEE Workshop on Factory Communication Systems, WFCS 2014, 2014, pp. 1–4, <http://dx.doi.org/10.1109/WFCS.2014.6837587>.
- [4] H. Esen, M. Adachi, D. Bernardini, A. Bemporad, D. Rost, J. Knodel, Control as a service (caas): cloud-based software architecture for automotive control applications, in: Proceedings of the Second International Workshop on the Swarm At the Edge of the Cloud, in: CPS Week '15, ACM, 2015, <http://dx.doi.org/10.1145/2756755.2756758>.
- [5] D. Zhang, P. Shi, Q.-G. Wang, L. Yu, Analysis and synthesis of networked control systems: A survey of recent advances and challenges, *ISA Trans.* 66 (2017) 376–392, <http://dx.doi.org/10.1016/j.isatra.2016.09.026>.
- [6] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, C. Peng, Networked control systems: a survey of trends and techniques, *IEEE/CAA J. Autom. Sin.* 7 (1) (2020) 1–17, <http://dx.doi.org/10.1109/JAS.2019.1911651>.
- [7] N. Vatanski, J.-P. Georges, C. Aubrun, E. Rondeau, S.-L. Jämsä-Jounela, Networked control with delay measurement and estimation, *Control Eng. Pract.* 17 (2) (2009) 231–244, <http://dx.doi.org/10.1016/j.conengprac.2008.07.004>.
- [8] A. Ulusoy, O. Gurbuz, A. Onat, Wireless model-based predictive networked control system over cooperative wireless network, *IEEE Trans. Ind. Inform.* 7 (1) (2011) 41–51, <http://dx.doi.org/10.1109/TII.2010.2089059>.
- [9] S.X. Ding, P. Zhang, S. Yin, E.L. Ding, An integrated design framework of fault-tolerant wireless networked control systems for industrial automatic control applications, *IEEE Trans. Ind. Inform.* 9 (1) (2013) 462–471, <http://dx.doi.org/10.1109/TII.2012.2214390>.
- [10] L. Yang, S.-H. Yang, Multirate control in internet-based control systems, *IEEE Trans. Syst. Man Cybern. C (Appl. Rev.)* 37 (2) (2007) 185–192, <http://dx.doi.org/10.1109/TSMCC.2006.886996>.
- [11] J. Salt, Á. Cuenca, F. Palau, S. Dormido, A multirate control strategy to the slow sensors problem: an interactive simulation tool for controller assisted design, *Sensors (Basel)* 14 (3) (2014) 4086–4110, <http://dx.doi.org/10.3390/s140304086>.
- [12] M. Ghodrati, H.J. Marquez, A multirate approach to event-triggered control of periodic sampled-data systems, *IEEE Trans. Control. Netw. Syst.* (2024) 1–12, <http://dx.doi.org/10.1109/tcms.2024.3395848>.
- [13] L. Aarnoudse, K. Cox, S. Koekebakker, T. Oomen, Multirate repetitive control for an industrial print-belt system, *Mechatronics* 100 (2024) 103187, <http://dx.doi.org/10.1016/j.mechatronics.2024.103187>.
- [14] A. González, Á. Cuenca, J. Salt, J. Jacobs, Robust stability analysis of an energy-efficient control in a networked control system with application to unmanned ground vehicles, *Inform. Sci.* 578 (2021) 64–84, <http://dx.doi.org/10.1016/j.ins.2021.07.016>.
- [15] J. Salt, Á. Cuenca, V. Casanova, V. Mascaro, A PID dual rate controller implementation over a networked control system, in: 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006, pp. 1343–1349, <http://dx.doi.org/10.1109/CACSD-CCA-ISIC.2006.4776837>.
- [16] A. Sala, Á. Cuenca, J. Salt, A retunable PID multi-rate controller for a networked control system, *Inform. Sci.* 179 (14) (2009) 2390–2402, <http://dx.doi.org/10.1016/j.ins.2009.02.017>.
- [17] Á. Cuenca, J. Salt, A. Sala, R. Piza, A delay-dependent dual-rate PID controller over an ethernet network, *IEEE Trans. Ind. Inform.* 7 (1) (2011) 18–29, <http://dx.doi.org/10.1109/TII.2010.2085007>.
- [18] Á. Cuenca, J. Alcaina, J. Salt, V. Casanova, R. Piza, A packet-based dual-rate PID control strategy for a slow-rate sensing networked control system, *ISA Trans.* 76 (2018) 155–166, <http://dx.doi.org/10.1016/j.isatra.2018.02.022>.
- [19] Á. Cuenca, W. Zhan, J. Salt, J. Alcaina, C. Tang, M. Tomizuka, A remote control strategy for an autonomous vehicle with slow sensor using Kalman filtering and dual-rate control, *Sensors* 19 (13) (2019) 2983, <http://dx.doi.org/10.3390/s19132983>.
- [20] N.H. Hirzallah, P.G. Voulgaris, N. Hovakimyan, On the estimation of signal attacks: A dual rate SD control framework, in: 2019 18th European Control Conference, ECC, 2019, pp. 4380–4385, <http://dx.doi.org/10.23919/ECC.2019.8796181>.
- [21] M. Naghnaeian, N. Hirzallah, P.G. Voulgaris, Dual rate control for security in cyber-physical systems, in: 2015 54th IEEE Conference on Decision and Control, CDC, 2015, pp. 1415–1420, <http://dx.doi.org/10.1109/CDC.2015.7402409>.
- [22] I. Bessa, C. Trapiello, V. Puig, R.M. Palhares, Dual-rate control framework with safe watermarking against deception attacks, *IEEE Trans. Syst. Man Cybern.: Syst.* 52 (12) (2022) 7494–7506, <http://dx.doi.org/10.1109/tsmc.2022.3160791>.
- [23] J. Tornero, M. Tomizuka, C. Camina, E. Ballester, R. Piza, Design of dual-rate PID controllers, in: Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01) (Cat. No.01CH37204), 2001, pp. 859–865, <http://dx.doi.org/10.1109/CCA.2001.973977>.
- [24] C. Bua, D. Adami, S. Giordano, GymHydro: An innovative modular small-scale smart agriculture system for hydroponic greenhouses, *Electron. (Basel)* 13 (7) (2024) 1366, <http://dx.doi.org/10.3390/electronics13071366>.
- [25] L. Beltramelli, A. Mahmood, P. Österberg, M. Gidlund, P. Ferrari, E. Sisinni, Energy efficiency of slotted LoRaWAN communication with out-of-band synchronization, *IEEE Trans. Instrum. Meas.* 70 (2021) 1–11, <http://dx.doi.org/10.1109/TIM.2021.3051238>.
- [26] M. Chiani, A. Elzanaty, On the LoRa modulation for IoT: Waveform properties and spectral analysis, *IEEE Internet Things J.* 6 (5) (2019) 8463–8470, <http://dx.doi.org/10.1109/JIOT.2019.2919151>.
- [27] M. Jouhari, N. Saeed, M.-S. Alouini, E.M. Amhoud, A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges, *IEEE Commun. Surv. Tutor.* 25 (3) (2023) 1841–1876, <http://dx.doi.org/10.1109/COMST.2023.3274934>.
- [28] K.J. Keesman, System identification: An introduction, Springer London, 2011, <http://dx.doi.org/10.1007/978-0-85729-522-4>.
- [29] M.A. Johnson, M.H. Moradi, PID Control: New Identification and Design Methods, Springer London, 2005, <http://dx.doi.org/10.1007/1-84628-148-2>.
- [30] N.A. Selamat, F.S. Daud, H.I. Jaafar, N.H. Shamsudin, Comparison of LQR and PID controller tuning using PSO for coupled tank system, in: 2015 IEEE 11th International Colloquium on Signal Processing & Its Applications, CSPA, 2015, pp. 46–51, <http://dx.doi.org/10.1109/CSPA.2015.7225616>.
- [31] C. Wang, Z. Li, T. Wang, X. Xu, X. Zhang, D. Li, Intelligent fish farm—the future of aquaculture, *Aquac. Int.* 29 (6) (2021) 2681–2711, <http://dx.doi.org/10.1007/s10499-021-00773-8>.
- [32] A. Angani, C.-B. Lee, S.-M. Lee, K.J. Shin, Realization of eel fish farm with artificial intelligence part 3: 5g based mobile remote control, in: 2019 IEEE International Conference on Architecture, Construction, Environment and Hydraulics, ICACEH, 2019, pp. 101–104, <http://dx.doi.org/10.1109/ICACEH48424.2019.9041938>.
- [33] M. Ruff-Salis, A. Petit-Boix, G. Villalba, D. Sanjuan-Delmás, F. Parada, M. Ercilla-Montserrat, V. Arcas-Pilz, J. Muñoz-Liesa, J. Rieradevall, X. Gabarrell, Recirculating water and nutrients in urban agriculture: An opportunity towards environmental sustainability and water use efficiency? *J. Clean. Prod.* 261 (2020) 121213, <http://dx.doi.org/10.1016/j.jclepro.2020.121213>.
- [34] D.R. Coughanowr, S. LeBlanc, *Process Systems Analysis and Control*, third ed., McGraw Hill Higher Education, Maidenhead, England, 2008.
- [35] G. Stanco, A. Botta, F. Frattini, U. Giordano, G. Ventre, On the performance of IoT LPWAN technologies: the case of sigfox, LoRaWAN and NB-IoT, in: ICC 2022 – IEEE International Conference on Communications, 2022, pp. 2096–2101, <http://dx.doi.org/10.1109/ICC45855.2022.9839078>.
- [36] D. Fernandes Carvalho, P. Ferrari, E. Sisinni, A. Depari, S. Rinaldi, M. Pasetti, D. Silva, A test methodology for evaluating architectural delays of lorawan implementations, *Pervasive Mob. Comput.* 56 (2019) 1–17, <http://dx.doi.org/10.1016/j.pmcj.2019.03.002>.
- [37] E. Sisinni, P. Ferrari, P. Bellagente, A. Depari, A. Flammini, M. Pasetti, S. Rinaldi, Simultaneous frame transmission for critical LoRaWAN communications, in: 2022 IEEE 18th International Conference on Factory Communication Systems, WFCS, 2022, pp. 1–4, <http://dx.doi.org/10.1109/WFCS53837.2022.9779180>.