Observers and Predictors for Wireless Control Loops

Linda Patricia Osuna-Ibarra, David Gómez Gutierrez,
Dave Cavalcanti and Humberto Caballero-Barragan

April 12, 2020

Observers and Predictors for Wireless Time-Sensitive Control Loops

Linda Patricia Osuna Ibarra
Intel
Guadalajara, Mexico
linda.patricia.osuna.ibarra@intel.com

David Gomez Gutierrez
Intel
Guadalajara, Mexico
david.gomez.gutierrez@intel.com

Dave Cavalcanti
Intel
Hillsboro, Oregon, USA
dave.cavalcanti@intel.com

Humberto Caballero Barragan
Intel
Guadalajara, Mexico
humberto.caballero.barragan@intel.com

***Abstract*—The present work deals with diminishing the effects of wireless communication latency in time-sensitive networks. It takes advantage of modern approaches that offer bounded latency and it focuses on methods to cope with the non-deterministic delays introduced by communication latency.**

An observer-predictor scheme module is designed and added in the control loop to prevent the performance degradation when a wireless network is introduced. The goal is to allow the use of same controllers used in wired networks. This module may exploit information provided by the wireless communication devices, such as expected latency, timestamp, and time synchronization along the network.

The proposed module allows the controller to have a performance similar as when in a wired network. This enables wireless networked control in systems with low time constants (i.e. time-sensitive).

Lab experiments are presented in real-time to illustrate the observer-predictor scheme module proposed. Finally, the conclusions and future work are presented.

***Index Terms*—industrial control, wireless time sensitive networking (WTSN), control algorithms, wireless control loops.**

I. Introduction

WIRELESS Time Sensitive Networks (WTSN) [1] are the logical next step in the configuration of factories around the world. Nevertheless, even with the current state of development of the Industry 4.0 the implementation of WTSN is not as wide as it could be. One reason for this is that the requirements for some industrial processes are beyond the capabilities of the current wireless communication protocols.

Different processes with different latency requirements are described in Fig. 1.

When the latency requirements and/or the message delivery guarantees are not met, but both are still within the operational margin, it is necessary to make a re-formulation of the problem. This means that the system has to be treated as discrete-time system, sometimes with delay due to latency communication, and the controller needs to be re-calculated and programmed, in the best cases at least the gains have to be adjusted, by means of calculations or doing an on-line tuning.

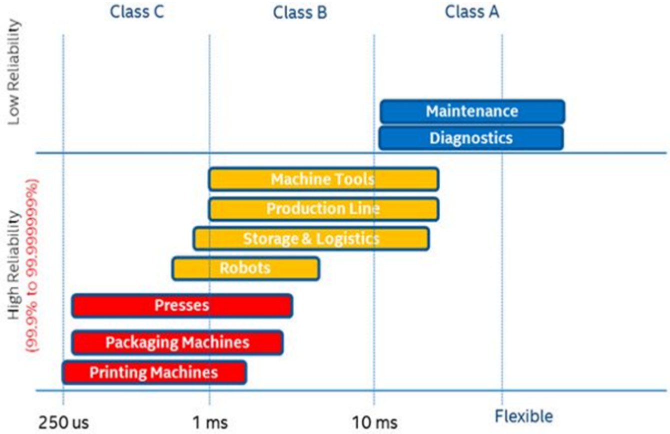


Fig. 1. Latency requirements for different industrial applications.

Industrial standard organizations such as ISA [2], HART [3], WINA [4], and ZigBee [5] have been actively pushing the application of wireless technologies in industrial automation and manufacturing, nevertheless, none of them deal with applications that require low latency, leaving out a lot of interesting systems in the industry with "fast dynamics" or with demanding low latency and high reliability requirements (Class B and C systems from Fig. 1) [6]. 802.11/Wi-Fi and 5G technologies are also being developed to address time-critical requirements from industrial systems. However, even with the faster data rates, lower latencies and higher reliability with next generation 802.11ax and 5G URLLC, guaranteeing extremely low latency and high reliability for the most time-critical applications (e.g. Class C) will still remain an open challenge.

II. Statement of Problem

Many modern industrial processes including systems where control signals are transmitted through networks, are modeled by delay differential equations. In these systems, the time delay can appear in the system state as well as in the control input. The last case is more dangerous for the closed-loop stability if the delay is large enough with respect to the plant dynamics rate and the standard memoryless feedback, i.e. the usual current system state, is used.

The use of predictors for systems with delay in the control input has been vastly studied through the years. One of the first recognized predictors was the one proposed by Smith in 1957 [7], with this predictor it is possible to stabilize systems with delay in closed-loop, nevertheless, in 1983 it is shown in [8] that Smith's predictor can only be applied in Time-Delay Systems (TDS) which are stable in open-loop. Around those years, the reduction method transformation is introduced [8]–[12]. This method consists in making a transformation based on the state and the integrals over the past control in order to get a delay-free system allowing the design of a delay-free control law.

Having that in mind, it is important to develop control strategies that enables the use wireless technologies to be implemented. In this paper it is proposed a strategy to deal with delays in the control loop, which are caused mainly by the communication latency between the sensors, actuators and the controller [13].

Nonlinear mechanical systems constitute a good case of study since most of the industrial robots belong to this class of systems. Also, robots are popular because they serve various practical purposes. [14]

The control of systems with delay is a current challenge for the control community [15]. Delays complicate the direct implementation of control techniques, because the introduction of a delay can down-perform the controller or even destabilize the system when delays are not taken into account.

III. Proposed Scheme

Different efforts to enable the implementation of wireless networks in industrial environments, such as improved Wi-Fi or 5G [1], have focused on improving the communication by reducing the latency and increasing the reliability of the wireless network to apply it as a replacement of a wired network, with all other elements of the control system remaining the same. Given that current wireless technologies have relatively lower link speeds and link reliabilities w.r.t. their wired counterparts, this replacement approach is only applicable to systems with large time constants (slow dynamics). The proposed observer-predictor solution complements the WTSN technology in order to extend the "replacement approach" to systems with smaller time constants (faster dynamics), allowing the controller to have a comparable performance to that of a wired network based control without having to change this previously established controller.

With the objective of easing the migration from wired networking to WTSN-enabled, a module composed by an observer and a predictor is proposed.

- **The observer**: It uses the measured variables (anything that comes from sensor, like temperature, speed, position, etc.) to infer the internal state of the system. In essence, the observer is a mathematical copy of the system in the sense that it is a virtual system with the same input-output behavior. If there are disturbances inputs that are not know a priori, an Unknown Input Observers (UIO) for state estimation can be used. In this way the disturbances can be estimated and assuming the estimation as valid for a time window it can be used in the predictor as input along with the control inputs.
- **The predictor**: Since there is latency in the communication links for the sensing-actuating loop, then the observer is in fact a delayed synchronized copy of the system. That is, the state of the observer is approximately the same as the state of the system at time $t - \tau_s$, where τ_s is the sensing-link random latency. If there is an additional latency in the actuation loop τ_a, in order to enable the controller to apply the appropriate correction action, the predictor evolves the observer for $\tau_s + \tau_a$ (using the historic of control actions, to predict what the state of the system would be at the moment the actuation is to be applied, and compute the control action accordingly.

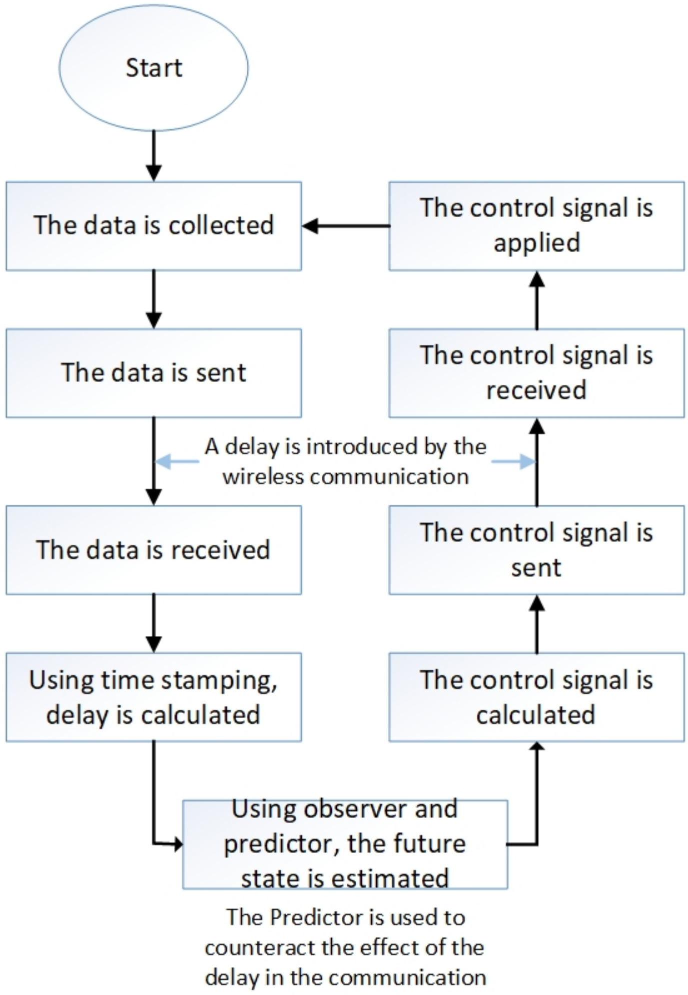


Fig. 2. Flowchart of the closed-loop control process including observer-predictor scheme.

In this way, the observer-predictor module may provide an estimation of the state of the system between samples and overcome (up to some extend) delays in the communications. Thus, allowing the controller to actuate at higher frequencies, similarly as in the wired network. A flowchart depicting the complete closed-loop control including the proposed idea is shown in Fig. 2.

IV. Mathematical Background

A. Observer

An observer is a mathematical structure that combines sensor output and plant excitation signals with models of the plant and sensor. An observer provides feedback signals that are superior to the sensor output alone.

The most obvious approach to estimating the state of a known system

$$\dot{x}(t) = f(x, u), \tag{1}$$

where x is the state of the system, and $f(x, u)$ is a function that depends on the values of the state x and the control signal u, is to create a copy

$$\dot{\hat{x}} = f(\hat{x}, u), \tag{2}$$

whose state provides an estimate $\hat{x}(t)$ of the original system's state x(t). (We know the control input u(t), so we can apply it to the copy as well as to the original system.) The simplicity of this method has a downside, however. If the initial state of the copy do not match exactly, the error evolves according to

$$\dot{e_0} = \dot{\hat{x}} - \dot{x}. \tag{3}$$

To deal with this, a correction function is introduced such that

$$\dot{\hat{x}} = f(\hat{x}, u) + g(e_0), \tag{4}$$

and by designing a suited $g(e_0)$ function that depends on the observation error, this error can be taken to zero to guarantee to have an stable convergent observer, [16].

B. Predictor

Consider the following nonlinear system:

$$\dot{x}(t) = f(x(t), u(t - \tau)), \tag{5}$$

where $x \in \mathbb{R}^{2n}$ is the state, $u \in \mathbb{R}^n$ is the control input, τ is a known scalar constant and $f : \mathbb{R}^{2n} \times \mathbb{R}^n \to \mathbb{R}^{2n}$ is a continuous function vector field that satisfies $f(0, 0) = 0$.

The predictor $\xi(t) = x(t+\tau)$ for the system (5) is designed as follows [17]

$$\xi(t) = x(t) + \int_{t-\tau}^{t} f(\xi(\theta), u(\theta))d\theta, \tag{6}$$

where $\xi \in \mathbb{R}^{2n}$ is the predictor state.
Now, taking the time derivative of (6) along trajectories of (5), the delay-free system reads as

$$\dot{\xi}(t) = f(\xi(t), u(t)). \tag{7}$$

The controller u can be designed using system (7).

V. Observer-Predictor Scheme

For the proposed idea to be feasible, the following requirements must be fulfilled by the wireless network.

Requirements:

1) The wireless network must ensure proper synchronization between the different devices.
2) The maximum expected latency has to be known with a reliability $\geq 99\%$.

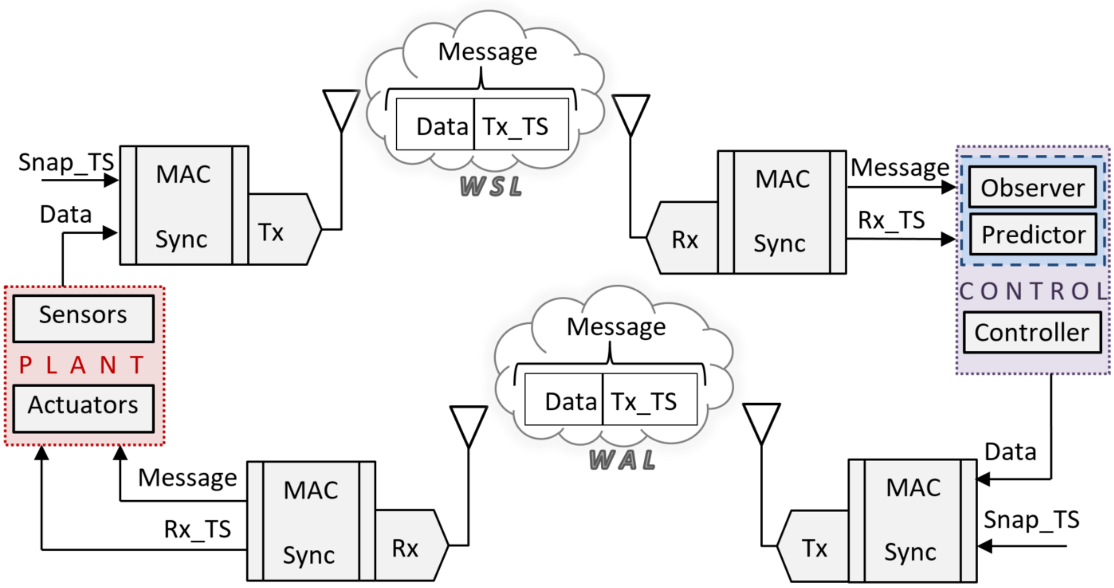


Fig. 3. Wireless communication scheme with Observer-Predictor module integrated (blue highlighted).

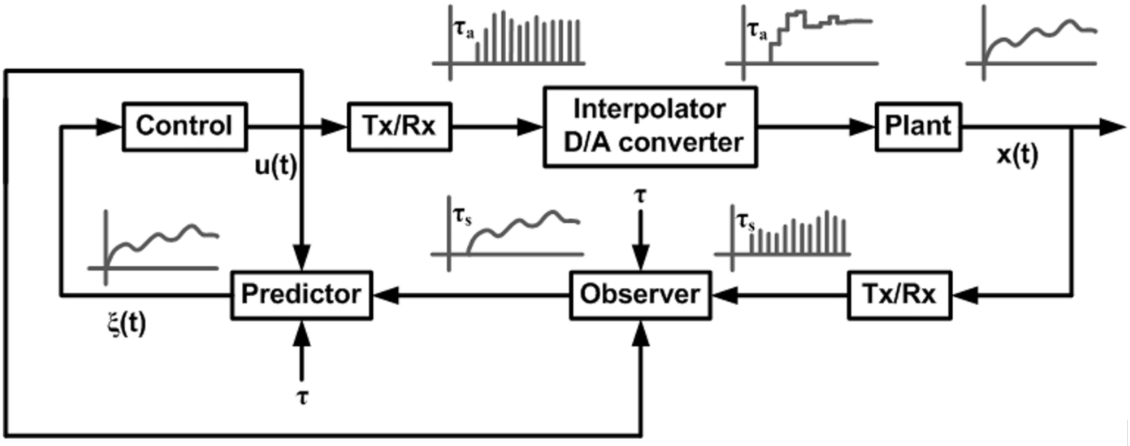


Fig. 4. Closed-loop control scheme with predictor-observer module integrated.

3) The transmitter/receiver (Tx/Rx) should be, preferably, able to time-stamp (TS) the signals when they are sent/received (otherwise the Tx/Rx must provide the necessary services to synchronize an external RTC, which will be used for timestamping).

*T*he Requirements 1 and 2 are fulfilled by the WTSN capabilities defined in the IEEE 802.11-2012 standard. [18].

The observer-predictor module is meant to be located logically between the WTSN receiver and the controller, as shown in Fig. 3 highlighted in blue. This module can be either integrated with the controller, i.e. in the same CPU (as a SW process), DSP (as a SW/FW process) or MCU (as a FW process), or it can be implemented in a separate dedicated computing element. Furthermore, the proposed solution relies on the feature of message time-stamping (TS) made by the transmitter Tx, that stamp (Tx_TS) is sent along the data through the Wireless Sensing Link (WSL) / Wireless Actuation Link (WAL). Once the message is received, the receiver Rx time-stamps (Rx_TS) the information and sends it to the control/plant, where the data is recovered and the latency (e.g. the communication delay) is calculated using the time-stamps. The calculated latency is latter used in the proposed algorithm. Nevertheless, the delay could also be calculated by means of some algorithms as [19]. As noted before, if the wireless communication devices do not provide time-stamping services, these must provide services to synchronize an external real-time clock, which would be used to generate the timestamps.

The control scheme of the whole wireless loop is depicted in Fig. 4. The sensed (measured) signals from the plant are sampled (digitized), time-stamped, and encapsulated alongside the timestamp as a sensing message, and provided to the WSL wireless transmitter to be sent to the controller. The delivery of these messages over the WSL present a random delay τ_s (due to multiple factors like packet-errors over the wireless channel, non-deterministic access to the wireless channel, etc.). Note that a similar random delay effect τ_a exists on the WAL, i.e. when the controller sends actuation messages to the plant over the WAL. Once received at the control side, the sensing messages are analyzed to estimate the delay τ_s based on the time-stamps (taken with Snap$_{TS}$), and provide it to the observer. The observer uses the sensing data to correct its state (note that the observer's state is in continuous time but with the delay τ_s remaining). In order to deal with both delays introduced by the WSL and WAL, a predictor is used to forecast the signal value at $\tau_s + \tau_{a,MAX}$) into the future (where $\tau_{a,MAX}$ is the maximum expected latency for the WAL). The output of the predictor is fed to the control which calculates the control/actuation signal. The actuation signal is encapsulated alongside the time-stamp in an actuation message, which is provided to the WAL transmitter, to be transmitted back to the plant through the WAL. The delivery of the messages over the WAL will present a random delay τ_a, which is guaranteed by the WTSN to be bounded. When the actuation messages are received at the plant, they are actuated accordingly to the time-stamps, by knowing when the control signals were sent it is possible to know when they must be applied since the delay to be compensated in the predictor is given by design (Equivalently, instead of time-stamping the actuation signal with the sending time (Tx_TS), it could be time-stamped with the time for which the actuation signal was computed and similarly wait for its time to be applied). Actuation signals are transformed to continuous time (e.g. with a zero order hold (ZOH)) and fed to the plant, closing the loop.

This scheme is only possible considering the assumptions of known dynamics of the plant and estimated time-delay (by time-stamping the signals or with algorithms). As it can be deduced by Figure 3, the observer is used to provide signals for the controller between each received sample (a sort of interpolation). For a dynamics given plant dynamics, the observer algorithm is as follows:

If sensing signal is received

Observer error is calculated

$$error\ e_0 = sensed\ measurements - observer\ state$$

$e_0 = x(k) - \hat{x}(t)$,

Observer state dynamics is corrected based on the error

$\dot{\hat{x}} = \hat{x} + f(u) + g(e_0)$,

Else

Observer state is calculated based on previous state

$\dot{\hat{x}} = \hat{x} + f(u)$.

(8)

To deal with the delays introduced by the transmitter, a predictor is designed. The predictor dynamics is governed by (7) with $\xi(t)$ the predicted state; the tracking error e(t) is calculated with the predicted state instead of the observed delayed state $e(t) = predicted\ state - reference$.

VI. Application to Mechanical Systems

Consider the mathematical model o a mechanical system obtained by means of the Euler-Lagrange formulation

$$\begin{aligned}\dot{x}_1(t) =& x_2(t)\\ \dot{x}_2(t) =& M(x_1)^{-1}(-C(x_1,x_2)x_2(t) - G(x_1) + u(t-\tau)),\end{aligned} \tag{9}$$

where $x_1(t) \in \mathbb{R}^n$ is the angular position vector, $x_2(t) \in \mathbb{R}^n$ is the angular velocity vector, $M(x_1) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(x_1, x_2) \in \mathbb{R}^{n\times n}$ is the matrix of Coriolis and centrifugal forces, $G(x_1) \in \mathbb{R}^n$ is the vector of gravitational forces, $u(t) \in \mathbb{R}^n$ is the vector of control input and τ is the delay in the control input, introduce for example by Wireless communication. Also, consider the Assumptions:

Assumption 1: State vector $x(t-D) = \begin{bmatrix} x_1(t-D)\\ x_2(t-D)\end{bmatrix}$ is available.

Assumption 2: System (9) is fully actuated and matrix $M(x_1)$ has full rank. Also, its inverse $M(x_1)^{-1}$ exists $\forall x_1$. The objective is to design an observer-predictor for the nonlinear system and reduce the time-delay effect. Having a desired reference for the state of the system and considering a delay in the control loop, the observer-predictor block designed should allow the controller (that has been designed under no-delay circumstances) to be able to track said reference with no changes required in the controller structure, nor the gains or any other parameter.

In order to ease the treatment of the delay present in the loop (which can be in the sensor link, actuation link, or both), it can be manipulated to be represented as a delay on the control input. In order to make this, and under Assumption 1, the following change of variable is introduced:

$$\bar{x}(t) = \begin{bmatrix}\bar{x}_1(t)\\ \bar{x}_2(t)\end{bmatrix} = x(t-D) = \begin{bmatrix} x_1(t-D)\\ x_2(t-D)\end{bmatrix}, \tag{10}$$

where D is the resulting delay present in the control-loop, which is $D = \tau_a + \tau_s$. Therefore, taking the time derivative of (10) along of trajectories (9), reads as

$$\dot{\bar{x}}(t) = f(\bar{x}(t), \tau(t-D)), \tag{11}$$

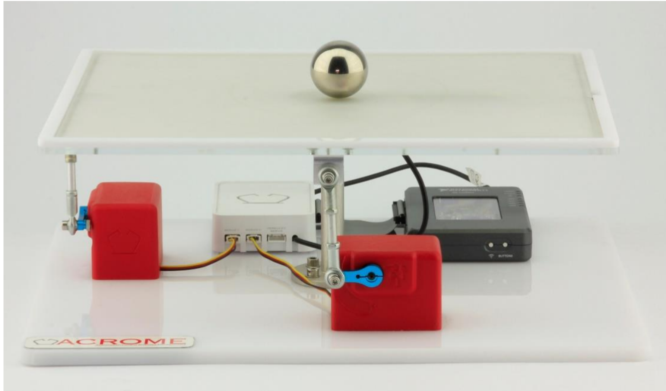


Fig. 5. Ball Balancing Table by ACROME.

where

$$f(\bar{x}(t), \tau(t-D)) = \begin{bmatrix} \bar{x}_2(t)\\ M(\bar{x}_1)^{-1}(-C(\bar{x}_1,\bar{x}_2)\bar{x}_2(t) - G(\bar{x}_1) + \tau(t-D))\end{bmatrix}. \tag{12}$$

Now, the predictor can be designed as

$$\xi(t) = \bar{x}(t+D) = \bar{x}(t) + \int_{t-D}^{t} f(\xi(\theta), \tau(\theta))d\theta, \tag{13}$$

where

$$f(\xi(\theta), \tau(\theta)) = \begin{bmatrix} \bar{\xi}_2(\theta)\\ M(\bar{\xi}_1(\theta))^{-1}(-C(\bar{\xi}_1(\theta), \bar{\xi}_2(\theta))\bar{\xi}_2(\theta) - G(\bar{\xi}_1(\theta)) + \tau(\theta))\end{bmatrix}, \tag{14}$$

with $\xi(t) = \begin{bmatrix}\bar{\xi}_1(t)\\ \bar{\xi}_2(t)\end{bmatrix}$.

Likewise, the observer is designed given the dynamics of the system as:

$$\begin{aligned}&\text{if signal is received}\\ &e(t) = \begin{bmatrix} x_1(k)\\ x_2(k)\end{bmatrix} - \begin{bmatrix}\hat{x}_1(t)\\ \hat{x}_2(t)\end{bmatrix},\\ &\dot{\hat{x}}_1 = \hat{x}_2 + g_1(e(t)),\\ &\dot{\hat{x}}_2 = f(u(t)) + g_2(e(t)),\\ &\text{else}\\ &\dot{\hat{x}}_1 = \hat{x}_2,\\ &\dot{\hat{x}}_2 = f(u(t)),\end{aligned} \tag{15}$$

A. Simulation Results

In order to test the proposed scheme, a simulation was made using as test bench the ball-balancing table shown in Fig. 5.

The Acrome Ball Balancing Table (BBT) [20] is the system used to test the network communications and control algorithms designed to mitigate the effects of communication delays. Even when it is not an industrial cell, it is a good platform to test the algorithms because:

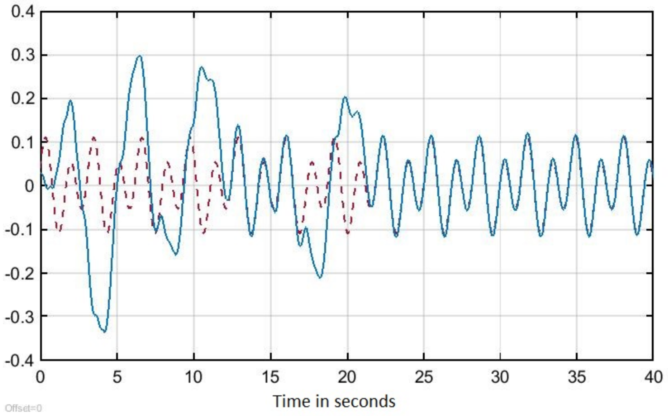


Fig. 6. Tracking of reference in simulation. The dashed red line is the reference and the solid blue line is the tracking performance. For $t < 20$ the observer-predictor scheme was turned off.

- It has mechanical dynamics (which is present in many industrial plants).
- It has electrical dynamics (due to the servomotors).
- It includes wireless sensing and actuation links.

The network consists of 3 user. The control computer is in charge of generating the control signals in order to follow trajectories and perform routines. On the other hand, two small single-board computers are in charge of sensing the position of the ball and controlling the motors of the platform respectively.

The equations that describe the dynamics of the BBT for the mechanical part are:

$$\begin{aligned}\ddot{x} =& \frac{m_b g r_b^2 r_M}{(m_b r_b^2 + j_b)Lx} sin(\vartheta_x),\\ \ddot{y} =& \frac{m_b g r_b^2 r_M}{(m_b r_b^2 + j_b)Ly} sin(\vartheta_y),\end{aligned} \tag{16}$$

Where x, y are the ball position in the x and y axes, $m_b = 0.26\ kg$, $r_b = 0.02\ m$, $j_b = 0.0000416\ kg * m^2$, $r_M = 0.0245\ m$, $L_x = 0.134\ m$, $L_y = 0.168\ m$ are the mass of the ball, its radius, its inertia moment, the length of the arm between the motor and the plate, and the dimensions of the table in the x and y axes respectively, $\vartheta_x, \vartheta_y$are the angles of the motors, used as control variables. The transfer function for the motors is:

$$G_M(s) = \frac{100}{0.01s + 1}. \tag{17}$$

During the simulations, the sensing latency was considered variable from 10 ms to 15 ms maximum with a high reliability, the actuation latency was considered fixed at 15 ms. The sampling rate was considered as well 15 ms, resulting in a total variable delay between 25 and 30 ms. The reference to be tracked is a signal composed by sine and cosine. Consider that for first 20 seconds of simulation the observer-predictor scheme was turned-off.

The simulation results are shown in Fig. 6, where the dashed red line is the reference to be tracked and the blue solid line is the tracking performance. As stated previously, the first 20 seconds simulation (first half) is the performance under delay **without** predictor. The later half (for $20 \leq t \leq 40$) is the performance under delay with the observer-predictor scheme turned on. It can be seen that the performance is taken back to a desirable behaviour unlike the first half of the simulation.

B. Real-Time Application Results

In order to test the proposed scheme with the Ball Balancing Table System, the flow shown in 7 was implemented in Simulink (green blocks) to link through UDP the controller and the observer-predictor scheme with the BBT System (red blocks).

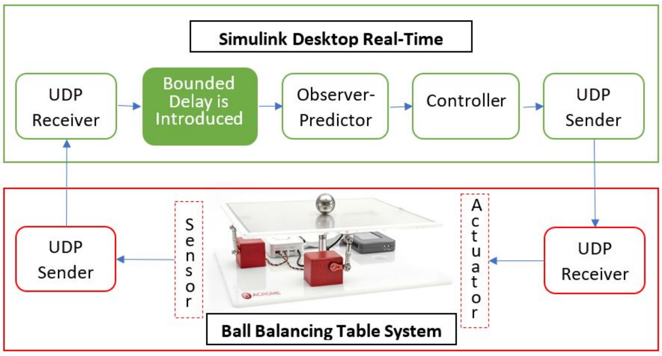


Fig. 7. Ball balancing table real-time implementation.

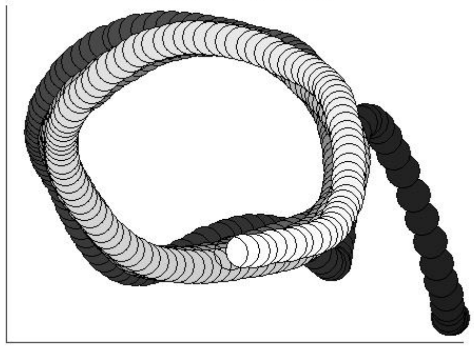
In order to test the performance under delay, a bounded-fixed delay of 20 ms was introduced in the software (solid green block). The controller was programmed to track a circle reference in the table. It is important to remark that the same controller designed for the system with no delay is used in all the three runs.

Figure 8 shows three different runs performed by the system. The gradient color of the ball that goes from black to white represents the evolution through the run time of the trajectory-tracking performance:

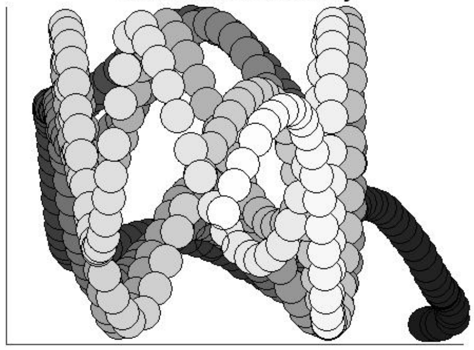
- **a)** The first run was made under normal circumstances, with no delay introduced (solid green block turned-off) and no observer-predictor (there is no need of it). It can be seen that the ball follows the programmed circular movement.
- **b)** In the second run, the 20 ms delay was introduced but without using the observer-predictor block, the effect can be seen, since the controller is unable to perform and the circular movement is not achieved.
- **c)** The third run shows the effect of the observer-predictor scheme. The effect of the 20 ms delay is greatly diminished, and the controller is now able to perform the circular movement with the ball.

VII. Conclusions and Future Work

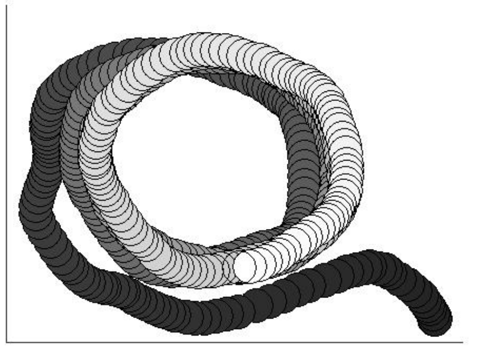
The use of the proposed observer-predictor scheme was introduced with the aim of diminishing the effects of the delay due to communication latency in time sensitive networks. The



(a) The ball is tracking a circle *without delay* in the loop.



(b) The ball is trying to track a circle *with delay* in the loop



(c) The ball is tracking a circle *with delay and predictor* in the loop.

Fig. 8. Lab-testing results of observer-predictor scheme applied to ball balancing table system.

effectiveness of the proposed scheme, was tested in both, simulation and real-time application. The observer-predictor approach provided an effective way to ease the effect of delays, which enables the application of different communication technologies in a wider range of applications, expanding the reach of current wireless solutions. By easing the effects of the delay the requirement of the communication network can be relaxed, allowing to serve more users, also, it offers the advantage of maintaining the performance of the controllers preciously programmed without considering delays. As future work, the use of AI techniques is proposed to do the prediction, with the goal of develop an observer-predictor that is general and applicable to any system without the need of obtain the model of the specific plant to control.

References

[1] Dave Cavalcanti, Javier Perez-Ramirez, Mohammad Mamunur Rashid, Juan Fang, Mikhail Galeev and Kevin B. Stanton, “Extending Accurate Time Distribution and Timeliness Capabilities over the Air to Enable Future Wireless Industrial Automation Systems”, Proceedings of the IEEE, Vol. 107(6), June 2019.
[2] ISA100. [Online]. Available:https://www.isa.org/isa100/.
[3] WirelessHART. [Online]. Available:http://en.hartcomm.org/.
[4] Wireless Industrial Networking Alliance. [Online]. Available:http://www.wina.org
[5] ZigBee Alliance. [Online]. Available: http://www.zigbee.org
[6] S. A. Ashraf and I. Aktas and E. Eriksson and K. W. Helmersson and J. Ansari, “Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G”, 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation, Berlin, 2016, pp. 1-8.
[7] O. J. M. Smith, “Closer control of loops with dead time”, Chemical Engineering Progress, 53(5):217–219, 1957.
[8] T. Furukawa and E. Shimemura, “Predictive control for systems with time delay”, International Journal of Control, 37(2):399–412, 1983.
[9] A. Manitius and A. W. Olbrot, “Finite spectrum assignment problem for systems with delays”, IEEE Transactions on Automatic Control, 24(4):541–552, 1979.
[10] W. Kwon and A. Pearson, “Feedback stabilization of linear systems with delayed control”, IEEE Transactions on Automatic Control, 25(2):266–269, 1980.
[11] Z. Artstein, “Linear system with delayed controls: A reduction”, IEEE Transactions on Automatic Control, 27(4):869–879, 1882.
[12] Y. A. Fiagbedzi and A. E. Pearson, “Feedback stabilization of linear autonomous time lag systems”, IEEE Transactions on Automatic Control, 31(9):847–855, 1986.
[13] E. Fridman, “Introduction to Time-Delay Systems:Analysis and Control”, Birkhäuser, London, UK, 2014.
[14] Michael Decker and Martin Fischer and Ingrid Ott, “Service Robotics and Human Labor: A first technology assessment of substitution and cooperation”, Robotics and Autonomous Systems, 87:348 - 354, 2017.
[15] J-P. Richard, “Time-delay systems: an overview of some recent advances and open problems”, Automatica, 39(10):1667–1694, 2003.
[16] G. Ellis, “Observers in Control Systems: A Practical Guide”, ACADEMIC PRESS, New York, USA, 2002.
[17] N. Bekiaris-Liberis and M. Krstic, “Predictor-Feedback Stabilization of Multi-Input Nonlinear Systems”, IEEE Transactions on Automatic Control, vol. 62(2), pp. 516–531, 2017.
[18] IEEE-802.11-2012. [Online]. Available:https://standards.ieee.org/standard/802_11-2012.html
[19] G. Zheng, A. Polyakov and A. Levant, “Delay estimation via sliding mode for nonlinear time-delay systems”, Automatica, 89, 2018, pp. 266-273.
[20] ACROME Automation Control Robotics Mechatronics [Online]. Available:https://www.acrome.net/ball-balancing-table