Asynchronous Multi-rate Sampled-data Control: an Embedded Model Control Perspective

Asynchronous Multi-rate Sampled-data Control: an Embedded Model Control Perspective

Carlos Perez-Montenegro[1], Luigi Colangelo[1], Jose Pardo[1,2], Alessandro Rizzo[1], Carlo Novara[1]

Abstract— **This study presents an estimation and control method for systems with asynchronous measurements with time-stamp. This approach is developed within the framework of Embedded Model Control (EMC), a control design methodology based on the estimation and the active rejection of the uncertainties affecting the plant to be controlled. The proposed control approach allows us to deal with control applications in which the measurement or the control law are characterized by a variable sampling time. Most notably, the proposed algorithm is capable to manage delays and package dropouts, potentially affecting networked control systems, by leveraging not only the sensors measurement information but also their time-stamp.**

I. INTRODUCTION

The Embedded Model Control (EMC) technique has been studied and used in multiple applications, during the last decades [1], and its potential in precision control problems has been validated especially in space applications [2] and complex nonlinear systems [3]. In this paper, the possibility to extend the EMC theory design framework to problems characterized by asynchronous measurements and commands is explored.

One of the most promising control applications, characterized by asynchronous measurements and commands, is represented by networked control systems (NCS). NCS are endowed with a communication network (CN) connecting the plant and the control unit. More recently, the development of applications relying on wireless communications networks to reduce the wired infrastructure has been gaining more and more attention [4].

As a matter of fact, a CN, between the control unit and the process, may introduce undesired effects, such as delays and loss of information (package dropouts). From the control problem perspective, these issues can lead to instability, unless explicitly accounted for [5], [6]. To this aim, they represent a stimulating within the control community, where multiple estimation and control strategies have been proposed to deal with the delays or the loss of information packages.

One way to tackle such issues typically affecting the NCS is through the use of measures with time-stamps. Control applications making use of time-stamp are typically found in the aerospace or automotive industry, where processes with a large number of systems running in parallel force the Electronic Control Unit (ECU) to execute tasks at different sampling or asynchronous times [6].

In this study, the time-stamp information is leveraged to provide the ECU with all the necessary elements to manage the undesired effects. Specifically, the designed Embedded Model Control (EMC) unit, allows the estimation and the active rejection of the disturbances originated in case of delays and package dropouts, reference generator is not treated here. The EMC technique is a disturbance-rejection-based control methodology. Techniques based on such a principle have been the subject of multiple theoretical studies, in the last few years [7]. The disturbance-rejection-based design has been also applied to the control of asynchronous and multi-rate systems and processes. For instance, in [8], the disturbance rejection framework is applied to a process control, in presence of time-delays and an integrating or unstable dynamics, although the discussion is limited to first-order plus time delay systems.

A. Main Contribution

In this study, the approach devised to solve the asynchronous control problems consists in taking into account at the same time the plant model and all the timing issues; i.e. the dynamics to be controlled, the delays, the asynchronous times, and the packet losses. In turn, such a control strategy implies that the proficient design of an omni-comprehensive estimation and control scheme, fully integrating the time-stamp information within the EMC disturbance-rejection architecture, becomes crucial to ensure the final performance.

Such a holistic approach takes advantage of the disturbance-rejection-based EMC control unit strengths. However, the peculiarities of the control problem requires the basic EMC design framework to be extended, so to make it able to proficiently deal with asynchronous measurements and commands. As a result, in this study, we propose the asynchronous-EMC design. The asynchronous-EMC design makes the typical EMC control unit time-adaptive, and apt to take into account simultaneously the dynamics of the system and the problems that can arise from a variable sampling of measures and commands.

The remainder of this paper is organized as follows. Section II shows the general architecture of the asynchronous problem, while in Section III the theoretical and methodological approach towards the asynchronous-EMC design are introduced. Section IV and Section V detail the two applicative case-studies developed in this research. Finally,

[1]Carlos Perez-Montenegro, Jose Pardo, Luigi Colangelo, Alessandro Rizzo, and Carlo Novara are with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy {carlos.perez,luigi.colangelo, alessandro.rizzo, carlo novara}@polito.it

[2]Jose Pardo is with Escuela Tcnica Superior de Ingenieros Industriales, Universidad Politcnica de Madrid, Madrid, Spain {jm.pardo}@alumnos.upm.es

A.R. acknowledges financial support by Compagnia di San Paolo, Torino, Italy

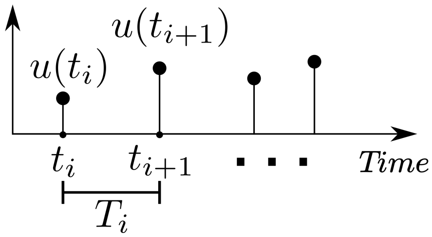

Fig. 1. The variable-sampling concept: sensor measurements rationale.

Section VI draws the conclusions and the main implications of the study and the presented results.

II. Problem Statement

In this study, the control unit receives its measurements asynchronously, and the command is provided asynchronously by the control law (CL), as well. The continuous state-space representation of the system can be written in a general form as

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

where the state vector $\mathbf{x}(t) = [\mathbf{x}_c(t) \;\; \mathbf{x}_d(t)]^T$ is divided in: (i) the controllable part $\mathbf{x}_c(t)$, and (ii) the disturbance dynamics $\mathbf{x}_d(t)$. Given the measurement asynchronous behaviour, Figure 1 sketches the sensor measurements rationale, in which both the value of the measurements and the time interval between each two consecutive measurement instances are variable. Being the value of the interval time-variable, its value is defined by the asynchronous sampling of the application. It is worth to notice that, although the time-variable measurement behaviour, the model always describes a causal relationship between the command $\mathbf{u}(t_i)$ and the measurements $\mathbf{y}(t_o)$. Where t_i and t_o are the sampling moments respectively In this context, there are mainly three timing issues that are addressed in this study:

- periodicity is not guaranteed;
- delay of the measures and commands;
- loss of information (Package dropouts).

Other problems that may arise, such as packet exchange, are not addressed here.

III. Embedded Model Control

In this section, general guidelines of the EMC technique [1] are presented. Specifically, this Section will focus and detail the development of an interval time adaptive Embedded Model (EM), matching the main dynamical system behavior, to be directly coded and real-time executed in the control unit. The EM typically consists of by two interconnected modules: (i) the controllable dynamics and (ii) the disturbance dynamics. The controllable dynamics is usually a proper input-output nominal model of the plant, accounting for the dynamics controllable by the command. The disturbance dynamics block models the disturbances acting on the plant; namely the command-independent unknown effects, the external disturbances, and the parametric uncertainties with respect to the nominal plant model. This disturbance dynamics is supposed to be observable from the plant measurement.

The asynchronous and variable-sampling scenario defined in this study relies on the following set of assumptions:

- the plant is an LTI system, although the result can be extended to non-linear plants through Jacobian local linearization or feedback linearization [9];
- the EM, namely the core of the estimation unit, runs asynchronously with respect to the sensors and their measurements;
- a time-stamp is always associated with each measurement sent to the control unit input.

Concerning the third assumption, being the sampling of the plant not deterministic, the interval time between two successive samples T_i is defined as $T_i = t_{i+1} - t_i$ (see Figure 1), where t_i is the sampling time-stamp of the ith measurement. From the previously mentioned assumptions, the strategy to tackle the control problem consists of an asynchronous implementation of the EMC design principles, in LTI systems, leveraging a system dynamics in discrete representation. As a result, this asynchronous-EMC control unit will result to be adaptive with respect to the interval-time T_i. In the following, the asynchronous-EMC design is introduced, together with its adapting capabilities according to the time-interval input.

A. Embedded Model

The EM is defined in the sampled state-space via the states $\mathbf{x}(t_i)$, whose subsets are $\mathbf{x}_c(t_i)$, referring to the controllable states, and $\mathbf{x}_d(t_i)$, referring to the disturbance dynamics. Moreover, $\mathbf{u}(t_i)$ and $\mathbf{w}(t_i)$ are the model command and the disturbance driving noise, respectively. Finally, the discrete-time state derivation can be obtained using the forward difference approximation of the derivative, i.e.,

$$\dot{\mathbf{x}}(t_i) \cong [\mathbf{x}(t_i + T_i) - \mathbf{x}(t_i)]/T_i. \tag{2}$$

Consequently, the asynchronous-EM state equations are defined in terms of the continuous state space representation $\{A, B, C\}$ and D as:

$$\begin{aligned} \mathbf{x}(t_i + T_i) &= (T_i A + I)\mathbf{x}(t_i) + T_i B\mathbf{u}(t_i) + G(t_i)\mathbf{w}(t_i), \\ \mathbf{x}(t_i + T_i) &= A_i\mathbf{x}(t_i) + B_i\mathbf{u}(t_i) + G_i\mathbf{w}(t_i), \end{aligned} \tag{3}$$

where A_i, B_i, and G_i are adaptable with respect to the time interval as:

$$\begin{aligned} \mathbf{x}(t_i) &= \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_d \end{bmatrix}(t_i), \quad A_i = \begin{bmatrix} A_{ci} & H_{ci} \\ 0 & A_{di} \end{bmatrix}, \\ B_i &= \begin{bmatrix} B_{ci} \\ 0 \end{bmatrix}(t_i), \quad G_i = \begin{bmatrix} G_{ci} \\ G_{di} \end{bmatrix}, \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned} \tag{4}$$

In (4), A_{ci} and A_{di} are the state matrices for the controllable and the disturbances states, respectively, while H_{ci} is the interconnection matrix between the disturbances and the controllable states. In addition, B_{ci} relates the inputs to the controllable states, whereas the noises affect the controllable dynamics via G_{ci} and the disturbances via G_{di}. The eigenvalues of estimation and control can be placed as desired by a proper selection of the design parameters. For estimator and CL design, it is necessary to reorganize the equation (3), so that it is possible to highlight the disturbances as,

$$\mathbf{x}(t_i + T_i) = A_i \mathbf{x}(t_i) + B_i \mathbf{u}_c(t_i) + \mathbf{d}(t_i). \tag{5}$$

The disturbance vector $\mathbf{d}(t_i)$ includes all the components that are rejected by the CL, this is obtained from (4) and defined as:

$$\mathbf{d}(t_i) = H_{ic}\mathbf{x}_d(t_i) + G_d \mathbf{w}(t_i). \tag{6}$$

The noise vector $\mathbf{w}(t_i)$ is assumed to consist of by arbitrary, bounded, unpredictable, command-independent signals. Therefore, for every component of the noise vector, the noise prediction is the null element $\hat{\mathbf{w}}(t_i) = 0$.

Taking into account the output $\mathbf{y}_m(t_i)$ and the measurements $\mathbf{z}_m(t_i)$, the following asynchronous-EM output equations hold:

$$\begin{aligned} \mathbf{y}_m(t_i) &= C_i \mathbf{x}(t_i), \\ \mathbf{z}_m(t_i) &= F_i \mathbf{x}(t_i), \end{aligned} \tag{7}$$

where

$$C_i = \begin{bmatrix} C_{ci} & C_{di} \\ 0 & A_{di} \end{bmatrix}, \quad F_i = \begin{bmatrix} F_{ci} & 0 \end{bmatrix}. \tag{8}$$

We observe that a model with variable sampling must take into account not only the classical quantization errors in amplitude, due to the digital converters, but also the resolution in time of the counters used to obtain the time variable t_i.

The EM is executed in an asynchronous fashion within the ECU. On the other side, the EM, together with any input-output devices (e.g. DAC, ADC, counters), as well as a frequency representation of the neglected dynamics and uncertainties, are considered jointly to make the so-called extended plant. Therefore, the extended plant is defined from the digital command $\mathbf{u}(t_i)$ to the digital measurement $\mathbf{y}(t_i)$, namely it includes all the disturbances and the neglected dynamics present within the system or the process to be controlled as well as due to the operating environment and the scenario conditions.

Being the command input in the extended plant and in the EM the same [1], an asynchronous simulation error may be defined as the plant and the model outputs differences, i.e.:

$$\mathbf{e}(t_i) = \mathbf{y}(t_i) - \mathbf{y}_m(t_i), \tag{9}$$

where the EM output is denoted as $\mathbf{y}_m(t_i)$. The operator request is defined as a target: this is the input of the reference generator and this system has as output the command $\overline{\mathbf{u}}(t_i)$ and the reference states $\overline{\mathbf{x}}(t_i)$ that respect the system dynamic restrictions. The reference dynamics $\underline{\mathbf{x}}(t_i)$ will be restricted to the controllable part of (4) and provides the reference command $\underline{\mathbf{u}}(t_i)$. Consequently, based on the reference, the control requirements are defined trough the output tracking error, viz.:

$$\underline{\mathbf{e}}(t_i) = \underline{\mathbf{y}}(t_i) - \mathbf{y}(t_i). \tag{10}$$

For the sake of brevity, the reference generator design is not reported here, yet in all the test it was included and accounted for.

B. Asynchronous Extended State Observer

The typical EMC extended state observer (ESO) is a linear dynamic feedback, built by leveraging the model error $\mathbf{e}(t_i)$ in (9). In short, the output-to-state feedback closing the ESO loop holds:

$$\mathbf{w}(t) = \mathbf{L}(t)(\mathbf{y}(t_i) - \hat{\mathbf{y}}_m(t)) = \mathbf{L}(t)\mathbf{e}(t_i). \tag{11}$$

As a matter of fact, the gains $\mathbf{L}(t)$ are dynamic and adaptive in the aynchronous-EMC ESO, although their systematic and general tuning procedure complies with [1]. To sum up, the ESO output-to-state feedback may be defined in two cases:

- in case $\dim(\mathbf{y}(t)) > \dim(\mathbf{w}(t))$, the observer is derived as:
$$A_m(t_i) = A_{mi} = A_i(t) - G_i \mathbf{L}(t) C_i; \tag{12}$$
- in case $\dim(\mathbf{y}(t)) < \dim(\mathbf{w}(t))$, to ensure its stability, the ESO cannot have a static feedback. Conversely, a dynamic output-to-state feedback, defined through the gains β and $\mathbf{M}_i$, must be introduced with:
$$A_{mi} = \begin{bmatrix} A_i - G_i L_i C_i & G_i M_{fi} \\ -C_i & 1 - \beta_i \end{bmatrix}. \tag{13}$$

We remark that the characteristic polynomial $P(\lambda_o)$, as well as the observer eigenvalues $\lambda(A_{mi})$ are fixed in order to set a desired frequency behavior, regardless of the time-interval.

the observer eigenvalues are function of the time interval T_i. Thus, the stability of the model introduces, in the minimum sampling interval, a condition to be met so to guarantee the correct dynamic equivalence of the system, i.e.:

$$T_i < \frac{1}{2 f_{\text{Estimation}}}, \quad \forall T_i, \tag{14}$$

where the frequency $f_{\text{Estimation}}$ is defined by the bandwidth of the dynamics to be estimated.

C. Control Law (CL)

Concerning the CL, the generic command of the asynchronous-EMC controller $\mathbf{u}(t_i)$ encompasses three terms: (i) a state feedback, (ii) a feed-forward term represented by the nominal command $\underline{\mathbf{u}}(t_i)$, and (iii) the disturbance rejection term. Thus, the adaptive CL command $\mathbf{u}(t_i)$ is given by:

$$\mathbf{u}(t_i) = \underline{\mathbf{u}}(t_i) + K(t_i)\underline{\mathbf{e}}_c(t_i) - M_i \mathbf{x}_d(i) \tag{15}$$

The feedback gains $K(t_i)$ are calculated aiming to set the frequency response as desired . This implies that, as in the extended state observer, the control eigenvalues $\lambda_c(A_c - B_c K(t_i))$ are calculated at each time-step (see (14)).

The tracking error, defining the state feedback term of the CL (equation (15)), is defined as:

$$\underline{\mathbf{e}}_c(t) = \underline{\mathbf{x}}(t) - (\mathbf{x}_c(t) + Q_i \mathbf{x}_d(t)). \tag{16}$$

Hence, the error propagation can be written as:

$$\begin{aligned}\underline{\mathbf{e}}_c(t+T_i) = A_{ci} + B_{ci}(\underline{\mathbf{u}}(t) - \mathbf{u}(t)) + \\ - (G_{ci} + Q_i G_{di})\mathbf{w}(t) + \\ + (A_{ci}Q_i - H_{ci} - Q_i A_{di})\mathbf{x}_d(t).\end{aligned} \tag{17}$$

Finally, The disturbance rejection is ensured through the matrices Q_i and M_i. Q_i allows The CL to account for the disturbances not entering the EM at the command level. On the other hand, M_i properly selects the disturbance channels within $\mathbf{u}(t_i)$ to ensure their direct rejection. Hence, assuming that the disturbances are observable from the system outputs, the asynchronous-EMC CL is able to handle the plant disturbances regardless of the state of the system dynamics they affect. Those two matrices are obtained solving (18), coming from (17), namely:

$$\begin{bmatrix} H_{ci} + Q_i A_{di} \\ 0 \end{bmatrix} = \begin{bmatrix} A_{ci} & B_{ci} \\ F_{ci} & 0 \end{bmatrix} \begin{bmatrix} Q_i \\ M_i \end{bmatrix}. \tag{18}$$

In short, (18) is based on the system state matrices defining the EM dynamics for a zero input noise.

In the following, the designed asynchronous-EMC control unit will be applied to two case-studies.

IV. Case-study 1: Asynchronous-EMC for a Rotary System

The case-study presented in this Section is a rotational motion system, endowed with an encoder, as a sensor. The system is characterized by an inertia J. The command action $\Omega(t_i)$ is synchronized with the angle $\theta(t_i)$.

The measurement sampling and the command action are not executed with a constant sampling time; a control strategy adopted, for instance, for the control of systems like injection engines. As a result, the concepts outlined in Section III are applied in designing the asynchronous-EMC controller. Nevertheless, since the measurement has a constant variation, namely $\delta\theta$, regardless of the time interval T_i, a special attention was paid to the design of the estimation unit of our asynchronous-EMC, as detailed below.

A. Rotary System Modelling

Applying the principles recalled in Section III, the continuous-time state vector, encompassing the controllable $\mathbf{x}_c$ and the disturbance $\mathbf{x}_d$ states can be written as:

$$\mathbf{x}_c(t) = \begin{bmatrix} \theta \\ \omega \end{bmatrix}(t), \quad \mathbf{x}_d(t) = \begin{bmatrix} d_m \end{bmatrix}(t), \tag{19}$$

where $\omega(t)$ is the angular rate and $d_m(t)$ is the disturbance state that summarizes all the neglected dynamics and parametric uncertainties, with respect to the nominal plant model. Hence, the continuous-time plant can be modelled as:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 \\ \frac{1}{J} \\ 0 \end{bmatrix} \\ G &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\ C &= F = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{20}$$

B. The Asynchronous-EMC Extended State Observer (ESO)

To build the ESO, we started by defining the angle model error $\tilde{\theta}(t_i)$ leveraging the measurement according to (9), viz.:

$$\tilde{\theta}_m(t_i) = \breve{\theta}(t_i) - \hat{\theta}(t_i) \tag{21}$$

where $\hat{\theta}(t_i)$ is the estimated angle, and $\breve{\theta}(t_i)$ is the measured angle. Therefore, converting (20) into discrete-time via a one-step forward Euler discretisation algorithm, the controllable states of the rotary system model results to be modelled as:

$$\begin{aligned} \hat{\theta}(t_{i+1}) &= \hat{\theta}(t_i) + T_i\hat{\omega}(t_i), \\ \hat{\omega}(t_{i+1}) &= \hat{\omega}(t_i) + \hat{\Omega}(t_i). \end{aligned} \tag{22}$$

thus:

$$\hat{\Omega}(t_i) = \left(d_m(t_i) + l_0\tilde{\theta}(t_i) + m_0 p(t_i)\right)/T_i^2, \tag{23}$$

being l_0 and m_0 the adjustable gains of the ESO, to be suitably tuned towards the disturbance estimation performance, once ensured its closed-loop stability. The controllable part of the ESO is completed by the disturbance estimation dynamics:

$$d_m(t_{i+1}) = d_m(t_i) + l_1\tilde{\theta}(t_i), \tag{24}$$

also characterized by a tuneable gain, l_1, to be set consistently with those belonging to the controllable states. In turn, such design choice let the ESO input-output structure to properly fit the real physics of the plat to be controlled. On the other side, such design structure of the ESO implies that the noise estimator, providing the output-to-state feedback to the observer, has a dynamic structure [1], as depicted in (13). Therefore, we have:

$$p(t_{i+1}) = \tilde{\theta}(t_i) + p(t_i) + \beta_i p(t_i) \tag{25}$$

where $p(t_i)$ is the state of the dynamics feedback, whose effect is adjusted via the tuning of the parameter B_0, together with those belonging to the controllable and disturbance states. To sum up, the state-space matrices of the ESO in closed-loop are

$$A_i = \begin{bmatrix} 1 & T_i & 0 & 0 \\ -\frac{l_0}{T_i^2} & 1 & \frac{1}{T_i^2} & \frac{m_0}{T_i^2} \\ -l_1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1-\beta_i \end{bmatrix}, \quad B_{pi} = \begin{bmatrix} 0 \\ T_i \\ 0 \\ 0 \end{bmatrix} \tag{26}$$

The four design gains (l_0, l_1, β_i, and m_0) were tuned so to enforce the ESO performance. To this aim, the discrete eigenvalues were defined in terms of their complementary eigenvalues, $\lambda = \gamma + 1$, so to simplify the tuning procedure. As a result, to set the desired eigenvalues set of the ESO, via a pole placement methodology, the ESO closed-loop gains were computed according to the interval of time of input, by writing the the characteristic polynomial:

$$\begin{aligned} P(\gamma) = &\gamma^4 + \beta_i\gamma^3 + l_0\gamma^2/Ti + \\ &(l_1 + m_0 + \beta l_0)\gamma/T_i + \beta_i l_1/T_i \end{aligned} \tag{27}$$

As a matter of fact, the gains l_0, l_1, β_0, and m_0 were considered as adaptive, with respect to the time, and they were calculated at each time/sampling-step, so to guarantee a constant response in frequency. To this aim, the continuous eigenvalues were maintained, while the discrete ones regularly adapted.

To conclude, let us notice that, to tune the asynchronous-EMC, the bandwidth of the system is regularly selected in continuous time in order to keep the system stable. Conversely, the discrete eigenvalues of the asynchronous-EMC ESO and of the CL are computed at each sampling step. In the limit case in which the maximum limit frequency, selected in continuous time, does not have a corresponding discrete time eigenvalue, the discrete time eigenvalue is set to be the maximum possible. Of course, this causes a deterioration in the control performances.

C. Simulation Results

To test the devised asynchronous-EMC control unit and methodology, a variable-time Matlab/Simulink simulator was developed. Such a simulator was specifically endowed with an asynchronous execution module that implements the capability to use the measurement time-stamp. The latter is determined with a time, thus making the temporal quantization also a source of potential unknown disturbances. The simulation parameters for the plant and the processing unit are listed in Table I.

TABLE I
SIMULATOR PARAMETERS

Parameter	Symbol	Value	Units
System inertia	J	0.01	kgm^2
Encoder Angle	$\delta\theta$	52.4	mrad
Timer Base	δt	0.1	m s
Timer number of bits	n_{bits}	16	–

In the simulation trial here presented, the rotary system receives a input step reference, in angular velocity, to be tracked by the control system. In this scenario, the estimated angular rate suitably matches the simulated one, up to a higher frequency band. The angle error is shown in the Figure 2, in which it can be seen that the estimation error $\tilde{\theta}_m(t_i)$ is negligible, compared to the encoder angle $\delta\theta$. From this result, we can infer that the design of the asynchronous-EMC ESO, especially concerning the disturbance estimation dynamics, fits the physics of the system. In addition, the dynamic design of the noise estimator is able to ensure both closed-loop stability of the ESO, and a consistent convergence of the estimated states towards their values. Moreover, the asynchronous-EMC CL effectively drives the plant state toward the desired target value, to be tracked.

Figure 3 shows the position of the control eigenvalue variation in discrete-time. We observe how, according to our design guideline for the asynchronous-EMC unit, the eigenvalues change depends on the time interval, so to maintain the frequency response. Specifically, Figure 3 shows that, at the beginning of the control action, the time intervals are longer, due to the low speed of the rotary system at its onset. In this period of time, the eigenvalues are closer to the limit. Then, once the angular rate increases, the time interval is reduced accordingly, and the discrete time eigenvalues are modified and shift towards the origin of the unit circle.

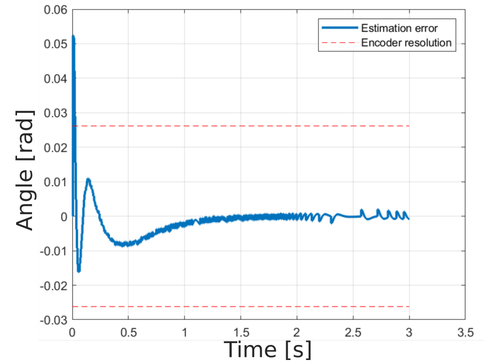

Fig. 2. Case-study 1 - Rotary system with encoder: Angle estimation error.

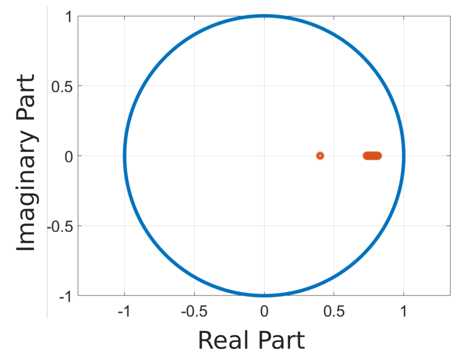

Fig. 3. Case-study 1 - Rotary system with encoder: discrete-time poles.

V. CASE-STUDY 2: ASYNCHRONOUS-EMC FOR NCS

The second case-study developed to test the asynchronous-EMC control unit design is based on a NCS, namely a control system with a CN between the plant and the control unit. Also in this case, the designed control unit includes an ESO and a CL, and was applied to a real experimental platform. The objective of this test was to validate the asynchronous-EMC methodology in a complex real-time system, primarily based on: (i) the control unit execution on a remote server, plus (ii) an extension of the typical hardware-in-the-loop (HIL) test scenario, including a CN at the interface between the control unit and the plant to be controlled. Such a specific test methodology was defined cloud-in-the-loop (CIL). The implemented CIL test setup relied upon: (i) a remote server, hosting the asynchronous-EMC control unit processing, and (ii) a further computing platform enabling the plant-server communication through a properly developed application programming interface (API). More in detail, the plant was implemented in a Matlab/Simulink high-fidelity simulator, while the real-time simulation execution was guaranteed through the Real-Time Pacer toolbox [10]. Moreover, the test setup leverages the Internet to connect the plant and the simulator, while a second level Matlab S-function was defined to interact with the API, through a TCP/IP using MQTT (message queuing telemetry transport) packages; a light-weight machine-to-machine protocol typically adopted for small sensors and mobile devices, also effective in case of high-latency.

A. Cloud-in-the-loop (CIL) Results

In this test scenario, the measurement to be employed by the asynchronous-EMC ECU must include the time-stamp. To this aim, the time interval T_i is calculated on the remote server. The asynchronous-EMC unit tries to execute the control with a constant period of T_{con}. Nevertheless, due to the CN between the ECU and the plant, and its conditions, a constant delay t_d is always experimented, thus making the period of execution of the tasks not guaranteed. The parameters for the plant and processing within the CIL tests setup are listed in Table II.

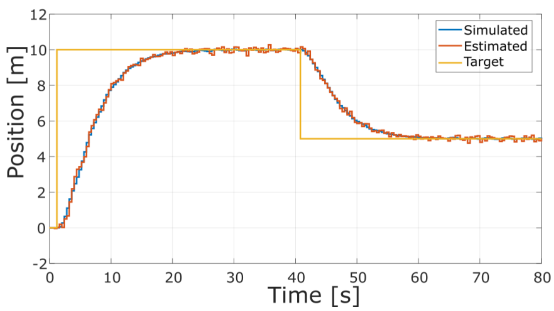

Fig. 4. Case-study 2 - Asynchronous-EMC for NCS: Simulated plant position (blue), the estimated one (red), and the set position target (orange)

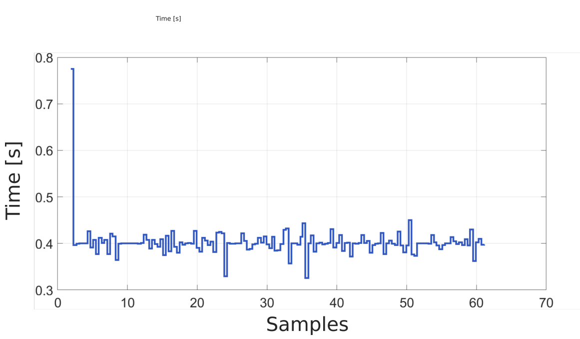

Fig. 5. Case-study 2 - Asynchronous-EMC for NCS: time interval T_i.

In this CIL test-case, the plant consists of a simple mass m, in which an axis is controlled by the force exerted by an actuator whose power noise is P_{na}, whereas its position is retrieved via a sensor whose power noise P_{ns}. The general behaviour of the vehicle position state is shown in Figure 4, depicting the system position. Indeed, Figure 4 highlights how the cloud ESO estimated trend is consistent with the simulated one, up to a high frequency band, thus underlining the effective tuning of the ESO eigenvalues. In fact, the system state is shown to be able to track the target position. Hence, the disturbance-rejection-based CL, employed in our asynchronous-EMC, succeeds in preventing that low-frequency errors, due to external disturbances and parametric uncertainties, affect the steady-state behaviour of the system to be controlled.

Figure 5 shows sampling time-interval T_i evolution, during the CIL test. As it can be seen, due to limitations introduced by the CN, the time-interval not always results to be $40\,\text{ms}$.

TABLE II
CLOUD-IN-THE-LOOP TESTS PARAMETERS

Parameter	Symbol	Value	Units
Mass	m	50	kg
Actuator noise power	P_{na}	100μ	W
Sensor noise Power	P_{ns}	$100m$	W
Constant delay	t_d	0.83	s
Nominal Control time	T_{con}	$40m$	s

VI. CONCLUSIONS

This study presents an estimation and control method for systems with asynchronous measurements with time-stamp. This approach is developed within the framework of Embedded Model Control (EMC); a disturbance-rejection-based control design methodology. The asynchronous-EMC control unit makes the typical EMC control unit time-adaptive, and apt to deal with control applications in which either the measurements, or the Control Law action, or even both of them have an asynchronous execution. The proposed algorithm is capable to manage the problems linked to a variable sampling of measures and commands (e.g. asynchronous sampling, delays and package dropouts). The developed asynchronous-EMC methodology was tested and validated in two case-studies: (i) an asynchronous control for a rotational mechanical system, and (ii) Networked Control System in real-time. In both cases, the designed control unit was proven to be able to yield the demanded control performances. This technique can be used in several systems that do not guarantee sampling such as combustion engines or that involve communication networks in the control loop.

REFERENCES

[1] E. Canuto, "Embedded Model Control: Outline of the theory," ISA Transactions, vol. 46, pp. 363–377, jun 2007.
[2] E. Canuto, Carlo Novara, Luca Massotti, Donato Carlucci, and Carlos Perez-Montenegro, Spacecraft Dynamics and Control: The Embedded Model Control Approach. Butterworth-Heinemann, 1st ed., 2018.
[3] M. A. Lotufo, L. Colangelo, C. Perez-Montenegro, E. Canuto, and C. Novara, "UAV quadrotor attitude control: An ADRC-EMC combined approach," Control Engineering Practice, vol. 84, pp. 13–22, mar 2019.
[4] X.-M. Zhang, Q.-L. Han, and X. Yu, "Survey on Recent Advances in Networked Control Systems," IEEE Transactions on Industrial Informatics, vol. 12, pp. 1740–1752, oct 2016.
[5] Y.-G. Sun and S.-Y. Qin, "Stability and Stabilization of Networked Control Systems with Bounded Packet Dropout," Acta Automatica Sinica, vol. 37, pp. 113–118, jan 2011.
[6] A. Rabello and A. Bhaya, "Stability of asynchronous dynamical systems with rate constraints and applications," IEE Proceedings - Control Theory and Applications, vol. 150, pp. 546–550, sep 2003.
[7] Y. Huang and W. Xue, "Active disturbance rejection control: Methodology and theoretical analysis," ISA Transactions, vol. 53, pp. 963–976, jul 2014.
[8] Q. Zheng and Z. Gao, "Predictive active disturbance rejection control for processes with time delay," ISA Transactions, vol. 53, pp. 873–881, jul 2014.
[9] J.-J. E. Slotine and W. Li, Applied nonlinear control, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.
[10] G. Vallabha, "Real-time pacer for simulink," Published in Matlab Central repository, 2016.