Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 472 (2023) 108696

FUZZY sets and systems

www.elsevier.com/locate/fss

Fuzzy discretization and control for non-linear, multiple binary input systems

Carlos Ariño, Lluís Mas, Pedro Balaguer *

Systems Engineering and Design Department (ESID), Universitat Jaume I de Castellón, Spain

Received 21 July 2021; received in revised form 28 November 2022; accepted 9 August 2023
Available online 23 August 2023

Dataset link: https://github.com/Mas-T/Binary

Abstract

The control of continuous-time linear systems with binary inputs cannot benefit from existing control design techniques because they are based on continuous control actions. In particular, optimal control problems with binary inputs lead to combinatorial optimization problems, which are difficult to solve. In this article we provide an exact discretization model of the binary continuous-time system that results in a non-linear multiple input controlled system. The non-linear model is then converted into a fuzzy discrete Takagi-Sugeno model, thus allowing the use of optimal control techniques based on LMI design. The modelling of the non-linear model by a discrete Takagi-Sugeno model is a complex process but it can be automatically performed as shown in the article and the code of the application examples.

Keywords: Fuzzy; Binary input; Discretization

1. Introduction

A large class of systems are actuated with binary inputs because either it leads to more economical industrial solutions or the system is inherently binary, like in power electronic converters actuated with ON/OFF switches.

Despite the actuator simplicity and robustness of systems with binary inputs, the design of a suitable control algorithm is more difficult, because existing control design techniques are based on continuous control actions. Furthermore, optimal control problems with binary inputs lead to combinatorial optimization problems, which are difficult to solve, what promotes the use of heuristic optimization algorithms [1].

An exact discretization model of the binary continuous-time system, parametrized by the duty cycle, that is, the ratio between the time the control action is ON and the sampling time, results in a non-linear, multiple input, binary controlled system. As a result, in the control of power electronic systems [2], in order to maintain the linearity of the original continuous-time system, the resulting non-linear model is linearised around an operating point, and the linear

* Corresponding author.
E-mail addresses: arino@uji.es (C. Ariño), mastorreslluis@gmail.com (L. Mas), pbalague@uji.es (P. Balaguer).

https://doi.org/10.1016/j.fss.2023.108696

model is used for control design. The drawback is that the model used for control is just an approximation of the original model, and the control may degrade when the duty cycle changes from the operating point as may happen in transient responses and disturbance rejections.

Another approach to deal with the non-linearity of the model is the use of multiple linearized models [3], where each linearized model is a local approximation of the non-linear one. However, the linearized models still has binary inputs that must be combined to describe the non-linear system, hence the model is not suitable to be used with existing controller design methodologies.

The approach taken in this article is different as we adhere to an exact discretized model of the original system. In this way, the resulting discrete-time model is exact and non-linear. We transform the non-linear model into a fuzzy discrete Takagi-Sugeno model, thus allowing to apply well-known fuzzy control design techniques [4–6]. Furthermore, the parametrization of the binary inputs in terms of the percentage of the period that the control action is ON, transform the binary decision variables into real bounded variables, hence avoiding integer optimization problems that are computationally costly.
Note for instance that in [7] a linear matrix inequality (LMI) approach to mixed-integer model predictive control (MPC) is proposed. As the optimization problem is not convex, the article proposes a convex relaxation that leads to suboptimal solutions.

In order to avoid an over-constrained controller that may led to conservative results, the more general procedures have been used on the LMI based design controller. [8,9]
As a result, the modelling of the non-linear model by a discrete Takagi-Sugeno model is a complex process both theoretically and computationally. However, previous results [10] allow to provide a code implementation that automatically performs the conversion of the non-linear model as a Takagi-Sugeno model, for arbitrary order and number of inputs. Furthermore, the control design based on the discrete Takagi-Sugeno model is also provided. The code can be found in https://github.com/Mas-T/Binary.

The structure of the article is as follows: In section 2 we state the problem and the notation used. Section 3 explains the discretization method. Section 4 presents the transformation of the exact discretized non-linear model into a discrete Takagi-Sugeno model and section 5 shows two application examples of the discretization process. Then, section 6 presents the proposed controller design approach and finally, section 7 shows other two application examples for the controller.

2. Problem statement

Given the system $\dot{x} = Ax + Bu$, with $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{nxn}$ and $B \in \mathbb{R}^{nxr}$, and multiple binary inputs $u \in \mathbb{D}^r$, with $\mathbb{D} = \{0, 1\}$, the main goal of this paper is the exact discretization of that system through the conversion into a discrete Takagi-Sugeno model, so it can be later analysed or controlled.
The continuous-time system is discretized with constant sampling period T and, for each sample k done at time $t = kT$, the i-th control action $u_i(t)$ is triggered to one from the beginning of the period $t = kT$ for a time interval $\mu_i(k)$ and then remaining inactive for the rest of the period, as can be see in Fig. 1. Hence, the discrete control action may be defined as the portion of the period that is triggered $\delta_i(k) = \mu_i(k)/T$, with $\delta_i(k) \leq 1$.
The discretized model obtained is therefore a discrete non-linear system where the control actions are $\delta_i(k)$. In order to systematically perform analysis and controller design, the non-linear model is converted into a Takagi-Sugeno fuzzy model using the non-linear sector technique. This type of non-linear model, being the interpolation of multiple linear models, has a multitude of theoretical developments which allow the design of robust controllers [4].

3. Exact system discretization

First, a system with an unique control action is considered for discretization. Each discrete state x_{k+1} will be a progression from the previous instant x_k, meaning that the state at each inter-period time $\delta(k)T$ within any x_k and x_{k+1} can be represented as [11]:

$$x(kT + \delta(k)T) = e^{AT\delta(k)}x_k + (e^{AT\delta(k)} - I)A^{-1}B \tag{1}$$

Since the state evolution from $kT + \delta(k)T \leq (k+1)T$ to a time $(k+1)T$ can be solved as:

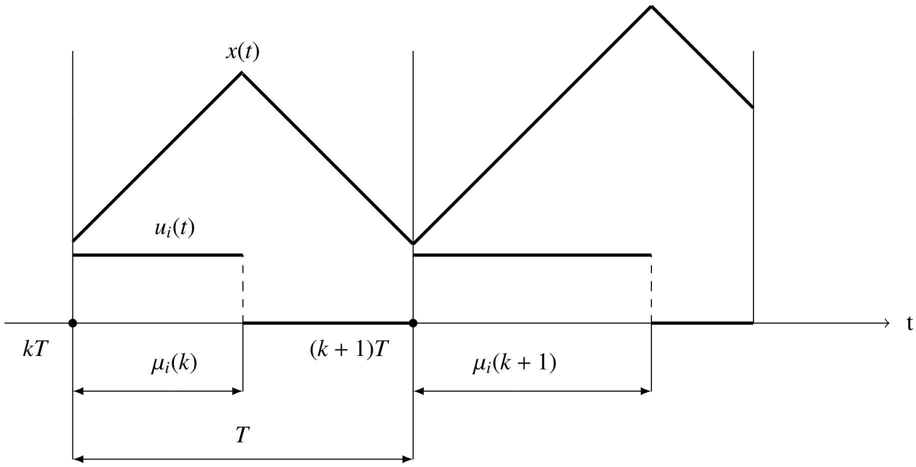

Fig. 1. The continuous-time system is discretized with constant sampling period T. For each sample k done at time $t = kT$, the control action $u_i(t)$ is triggered during time $\mu_i(k)$. Given the percentage of the period that is triggered $\delta_i(k) = \mu_i(k)/T$, the triggering time can also be given as $\mu_i(k) = \delta_i(k)T$. As a result, the control actions are $\delta_i(k)$.

$$x((k+1)T) = e^{AT(1-\delta(k))}x(kT + \delta(k)T) \tag{2}$$

The evolution of space state at instant $T(k+1)$ can be defined as:

$$x_{k+1} = e^{AT}x_k + e^{AT}(I - e^{-AT\delta(k)})A^{-1}B \tag{3}$$

Note that, in order to compute this discrete model, A^{-1} is needed. But its existence is not guaranteed as the system can have an integrator. In that case, the inverse of A does not exits. However, (3) can be computed if some details are taken into account. The procedure is detailed in Section 4.1.

To this point, only one control action has been modelled into its discrete form but, taking into consideration the superposition principle to every action u_i, when each action triggers for a time $\mu_i \le T$ and $i \le r$, the discrete equivalent of the system is:

$$x_{k+1} = e^{AT}x_k + \sum_{i=1}^{r} e^{AT}(I - e^{-AT\delta_i(k)})A^{-1}b_i \tag{4}$$

where b_i is the column i of the matrix B. Since this system is not yet linear dependent of the duty cycle of the control actions, it is artificially introduced multiplying b_i by $\frac{\delta_i(k)}{\delta_i(k)}$ then defining $\Delta_k \in \mathbb{R}^{r\times 1}$, $\Delta_k = (\delta_1(k), \ldots, \delta_r(k))^T$ and rearranging the terms in order to make the system linearly dependant to Δ_k.

$$x_{k+1} = e^{AT}x_k + \sum_{i=1}^{r} e^{AT}\frac{I - e^{-AT\delta_i(k)}}{\delta_i(k)}A^{-1}B_i\Delta_k \tag{5}$$

where B_i is a matrix $\mathbb{R}^{n\times r}$ where the column i is b_i but the rest of elements are 0 and $\delta_i(k)$ is the i-element of the duty cycle Δ at instant k.

Further development of the problem will require the manipulation of the eigenvalues from the state matrix A. To better accommodate this situation, the above expression can be rewritten with the state matrix in a diagonal form where $A = VDV^{-1}$, so all the eigenvalues become isolated within the main diagonal of D to be operated individually later on. The remaining expression from where the next steps will be developed, will be this one:

$$x_{k+1} = e^{AT}x_k + \sum_{i=1}^{r} e^{AT}V\frac{I - e^{-DT\delta_i(k)}}{\delta_i(k)}D^{-1}V^{-1}B_i\Delta_k \tag{6}$$

4. Fuzzification process

In this section, a fuzzification process is performed using non-linearity methodology that obtains an exact Takagi-Sugeno fuzzy model [12]. This fuzzy modelling methodology is based on the non-linear combination of a set of linear models. This combination is done with the membership functions. This method treats every non-linearity of the system as a fuzzy term so the result takes shape as a Takagy-Sugeno model [13].
Since the matrix D already is a diagonal matrix, therefore the non-linear term $\frac{I-e^{-DT\delta_i(k)}}{\delta_i(k)}D^{-1}$ is also diagonal. So this algorithm steps all of terms in the diagonal, one by one, in order to find its independent "sector non-linearity".

4.1. Sector non-linearity on real poles and origin

When D is a real matrix, the first step is straightforward. If λ_j are A eigenvalues disposed in a diagonal form within D, the expression $\frac{I-e^{-DT\delta_i}}{\delta_i}D^{-1}$, found in (6), can be defined in a diagonal form. Then each of the values are λ_j dependant and treated as the non-linear term above mentioned. The resulting matrix will be this one:

$$\frac{I-e^{-DT\delta_i}}{\delta_i}D^{-1} = diag\left(\frac{1-e^{-\lambda_1 T\delta_i}}{\lambda_1\delta_i},\dots,\frac{1-e^{-\lambda_j T\delta_i}}{\lambda_j\delta_i},\dots,\frac{1-e^{-\lambda_n T\delta_i}}{\lambda_n\delta_i}\right) \tag{7}$$

where δ_i for simplification is only referring to one period k (i.e. $\delta_i = \delta_i(k)$). Let us define the j diagonal element of (7) as $\mathcal{D}_j(\delta_i) = \frac{1-e^{-\lambda_j T\delta_i}}{\lambda_j\delta_i}$.
Note that some λ_j can be in origin. This is a well known issue that has to be studied since it is a problem which derives from the non-existence of A^{-1}. This study can be done finding the limit of any $\mathcal{D}_j(\delta_i)$ when $\lambda_j \longrightarrow 0$. It can be proven that:

$$\lim_{\lambda_j\longrightarrow 0}\frac{1-e^{-\lambda_j T\delta_i}}{\lambda_j\delta_i} = 1 \tag{8}$$

Therefore, if there is any $\lambda_j = 0$ then this j-element $\mathcal{D}_j(\delta_i)$ can be substituted by 1.
Now, if λ_j is real and not zero, we have the certainty that the range that each diagonal term $\mathcal{D}_j$ can take, will be always constrained within the values of that term when δ is at its limits. This is know because the expression takes the shape of a monotonically decreasing function and, since the duty cycle δ can only take values from 0 to 1, the extremes for the range in each term are known just by finding those limits.

$$d_{j1} = \lim_{\delta_i\longrightarrow 0}\frac{1-e^{-\lambda_j T\delta_i}}{\lambda_j\delta_i} = 1, \tag{9}$$

$$d_{j2} = \lim_{\delta_i\longrightarrow 1}\frac{1-e^{-\lambda_j T\delta_i}}{\lambda_j\delta_i} = \frac{1-e^{-\lambda_j T}}{\lambda_j} \tag{10}$$

being d_{j1} the upper limit and d_{j2} the lower limit for $\mathcal{D}_j$.
Then, following sector non-linearity Takagi-Sugeno modelling technique, the membership functions for those expressions have to meet the following criteria:

$$\mathcal{D}_j(\delta_i) = \alpha_{j1}(\mathcal{D}_j(\delta_i))d_{j1} + \alpha_{j2}(\mathcal{D}_j(\delta_i))d_{j2} \tag{11}$$

where $\alpha_{j1}(\mathcal{D}_j(\delta_i)) + \alpha_{j2}(\mathcal{D}_j(\delta_i)) = 1$ and $\mathcal{D}_j(\delta_i) = \frac{1-e^{-\lambda_j T\delta_i}}{\lambda_j\delta_i}$
This can easily be rewritten into the following equations in order to find both membership functions:

$$\alpha_{j2}(\mathcal{D}_j(\delta_i)) = \frac{\mathcal{D}_j(\delta_i) - d_{j1}}{(d_{j2}-d_{j1})}, \quad \alpha_{j1} = 1-\alpha_{j2} \tag{12}$$

All the above procedure casts all the information needed to model each of the real eigenvalues into a Takagi-Sugeno model. Later on, it is explained how those membership functions will be used to deliver the completed model.

4.2. Sector non-linearity on complex poles

If D has complex values, the above procedure cannot be used straightforward. To avoid numerical problems using complex terms in the expression $\mathcal{D}_j(\delta_i)$, block diagonalization is used on D. So each complex-conjugate pair ($\lambda_j = a_j + \mathbf{i}b_j$[1]) "throws" its complex part out of the main diagonal in its adjacent position rearranging both terms into 2x2 matrices as:

$$D_j = \begin{bmatrix} a_j & b_j \\ -b_j & a_j \end{bmatrix} \tag{13}$$

Taking that into account, the equivalent of the equation (6) on real poles, will apply for $\mathcal{D}'_j$ as this:

$$\mathcal{D}'_j = \left(I - e^{D_{j,j+1}T\delta_i}\right) D^{-1}_{j,j+1} \tag{14}$$

Then, the above expression can be simplified into two functions, one for each of both diagonal terms in $\mathcal{D}'_j$.

$$\mathcal{D}'_j = \frac{1}{a_j^2 + b_j^2}\left(\begin{pmatrix} -a_j & b_j \\ -b_j & -a_j \end{pmatrix}\mathcal{D}_j(\delta_i) + \begin{pmatrix} b_j & a_j \\ -a_j & b_j \end{pmatrix}\mathcal{D}_{j+1}(\delta_i)\right) \tag{15}$$

For each of those blocks, the same sector non-linearity can be applied. In this case, the function is not a monotonically decreasing one. The following expression have to be used to find its maximum and minimum values on each of the conjugate eigenvalues sectors:

$$\mathcal{D}_j(\delta_i) = \frac{e^{-a_jT\delta_i}\cos(b_jT\delta_i) - 1}{T\delta_i}, \mathcal{D}_{j+1}(\delta_i) = \frac{e^{-a_jT\delta_i}\sin(b_jT\delta_i)}{T\delta_i} \tag{16}$$

With both of those functions, the maximum and minimum values can be found by scooping δ range with an iterative or analytic procedure. This is necessary due to the oscillatory nature of this function and the uncertainty of both limits being on ends of the range. Once both limits are known, the membership functions α can be solved in a similar way than the previous case. Let d_{j1} be the minimum value for $\mathcal{D}_j(\delta_i)$ and d_{j2} the maximum value, and $d_{j+1,1}$ the minimum value and $d_{j+1,2}$ the maximum value for $\mathcal{D}_{j+1}(\delta_i)$, then the membership functions will be:

$$\begin{aligned} \alpha_{j2}(\mathcal{D}_j(\delta_i)) &= \frac{\mathcal{D}_j(\delta_i) - d_{j1}}{(d_{j2} - d_{j1})}, \quad \alpha_{j1} = 1 - \alpha_{j2} \\ \alpha_{j+1,2}(\mathcal{D}_{j+1}(\delta_i)) &= \frac{\mathcal{D}_{j+1}(\delta_i) - d_{j+1,1}}{(d_{j+1,2} - d_{j+1,1})}, \quad \alpha_{j+1,1} = 1 - \alpha_{j+1,2} \end{aligned} \tag{17}$$

4.3. Final model

Now that every membership function is calculated, along with each of the sector limits, the model can be fully constructed for all of the λ_j as a Takagi-Sugeno fuzzy system.
The final step comes with the construction of a new diagonal matrix $\mathfrak{D}_i \in \mathbb{R}^{n\times n}$ where it takes a Takagi-Sugeno fuzzy form [14] such as:

$$\mathfrak{D}_i = \sum_{l\in\mathcal{B}} \alpha^l(\delta_i)\mathfrak{d}_l \tag{18}$$

where

$$\mathcal{B} = \{l \;/\; l \in \{0,1\}^m, l_{2p-1} + l_{2p} = 1, p = 1\ldots n\} \tag{19}$$

and $\alpha(\delta_i)$ is a vector with all the membership functions:

$$\alpha(\delta_i) = (\alpha_{11}(\delta_i), \alpha_{12}(\delta_i), \alpha_{21}(\delta_i), \alpha_{22}(\delta_i), \ldots, \alpha_{n1}(\delta_i)) \tag{20}$$

Then $\alpha^l(\delta_i)$ is defined as $\prod_{p=1}^{2n} \alpha_p^{l_p}(\delta_i)$ where α_p is the p element of vector α, finally $\mathfrak{d}_l$ is a block diagonal matrix such as:

[1] Being $\mathbf{i} = \sqrt{-1}$.

$$\mathfrak{d}_l = \begin{bmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_n \end{bmatrix} \tag{21}$$

being $D_p = d_{p,1}^{l_{2p-1}} d_{p,2}^{l_{2p}}$ if the p-eigenvalue is real, $D_p = 1$ if the p-eigenvalue is 0 and D_p is a $2x2$ matrix if the eigenvalue p is complex $a_p \pm b_p$ which is defined as:

$$D_p = \frac{1}{a_p^2 + b_p^2} \left(\begin{pmatrix} -a_p & b_p \\ -b_p & -a_p \end{pmatrix} d_{p,1}^{l_{2p-1}} d_{p,2}^{l_{2p}} + \begin{pmatrix} b_p & a_p \\ -a_p & b_p \end{pmatrix} d_{p+1,1}^{l_{2p+1}} d_{p+1,2}^{l_{2p+2}} \right) \tag{22}$$

In that case, D_{p+1} does not exist as columns p and $p+1$ are included in D_p.
Finally, the initial equation (6) can be rewritten as:

$$x_{k+1} = e^{AT} x_k + \sum_{i=1}^{r} e^{AT} V \mathfrak{D}_i V^{-1} B_i \Delta_k \tag{23}$$

Substituting $\mathfrak{D}_i$ by expression (18):

$$x_{k+1} = e^{AT} x_k + \sum_{i=1}^{r} e^{AT} V \left(\sum_{l \in \mathcal{B}} \alpha^l(\delta_i) \mathfrak{d}_l \right) V^{-1} B_i \Delta_k \tag{24}$$

Then, taken into account that all the membership functions α_{ij} are scalar, it can be rearranged as:

$$x_{k+1} = e^{AT} x_k + \sum_{i=1}^{r} \sum_{l \in \mathcal{B}} \alpha^l(\delta_i) e^{AT} V \mathfrak{d}_l V^{-1} B_i \Delta_k \tag{25}$$

Defining $\mathfrak{B}_{l,i} = e^{AT} V \mathfrak{d}_l V^{-1} B_i$, a Takagi-Sugeno model is obtained:

$$x_{k+1} = e^{AT} x_k + \sum_{i=1}^{r} \sum_{l \in \mathcal{B}} \alpha^l(\delta_i) \mathfrak{B}_{l,i} \Delta_k \tag{26}$$

The implementation of this process is showcased on the next examples, where an exact discretization is displayed for various systems with distinct number of control actions and with real and complex eigenvalues.

4.4. Fuzzification example

To better understand how this fuzzification process works, a brief example is exposed:
Suppose a system such as $A \in \mathbb{R}^{2\times 2}$ with two real eigenvalues, that only has one control action is discretized with period $T = 0.3s$.

$$A = \begin{bmatrix} -4 & 5 \\ 2 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} -10 \\ 0 \end{bmatrix} \tag{27}$$

being $A = VDV^{-1}$

$$V = \begin{bmatrix} -0.8798 & -0.8037 \\ 0.4754 & -0.5950 \end{bmatrix}, \quad D = \begin{bmatrix} -6.7016 & 0 \\ 0 & -0.2984 \end{bmatrix} \tag{28}$$

and following the procedure explained above, there's a point where the sector limits can be computed following (10). For this example, the four sector limits are:

$$d_{11} = d_{21} = 1 \tag{29}$$

$$d_{12} = \frac{1 - e^{-\lambda_1 T}}{\lambda_1} = 3.2166 \tag{30}$$

$$d_{22} = \frac{1 - e^{-\lambda_2 T}}{\lambda_2} = 1.0461 \tag{31}$$

and the four membership functions ($\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}$):

$$\begin{aligned} \alpha_{12}(\mathcal{D}_1(\delta)) = \tfrac{\mathcal{D}_j(\delta)-1}{3.2166-1}, \quad \alpha_{j1} = 1-\alpha_{j2} \\ \alpha_{22}(\mathcal{D}_2(\delta)) = \tfrac{\mathcal{D}_{j+1}(\delta)-1}{1.0461-1}, \quad \alpha_{j+1,1} = 1-\alpha_{j+1,2} \end{aligned} \tag{32}$$

with:

$$\mathcal{D}_1(\delta) = \tfrac{1-e^{-\lambda_1 T\delta}}{\lambda_1\delta} = \tfrac{1-e^{2.0105\delta}}{-6.7016\delta} \tag{33}$$

$$\mathcal{D}_2(\delta) = \tfrac{1-e^{-\lambda_2 T\delta}}{\lambda_2\delta} = \tfrac{1-e^{0.0895\delta}}{-0.2984\delta} \tag{34}$$

Note that in these equations δ is used instead of δ_1 as there is only one control action. Then, the fuzzy form for D on this system will be:

$$\mathfrak{D} = \begin{bmatrix} \alpha_{11}d_{11} + \alpha_{12}d_{12} & 0 \\ 0 & \alpha_{21}d_{21} + \alpha_{22}d_{22} \end{bmatrix} \tag{35}$$

This matrix can also be rewritten, artificially introducing the remaining alphas on each term, taken into account that $\alpha_{i1} + \alpha_{i2} = 1$:

$$\mathfrak{D} = \begin{bmatrix} (\alpha_{11}d_{11} + \alpha_{12}d_{12})(\alpha_{21} + \alpha_{22}) & 0 \\ 0 & (\alpha_{21}d_{21} + \alpha_{22}d_{22})(\alpha_{11} + \alpha_{12}) \end{bmatrix} \tag{36}$$

The vector l is now introduced. On this case l is a vector with four terms one for each membership function α_{ij} where every term l_p represents if the corresponding membership function is in the coefficient. In this case, following (19) the possible values of l are $\mathcal{B} = [(1010), (1001), (0110), (0101)]$ on this example.
Then, the above equation can be rewritten into:

$$\begin{gathered} \mathfrak{D} = \\ \alpha_{11}^1\alpha_{12}^0\alpha_{21}^1\alpha_{22}^0 \begin{bmatrix} d_{11}^1 d_{12}^0 & 0 \\ 0 & d_{21}^1 d_{22}^0 \end{bmatrix} + \alpha_{11}^1\alpha_{12}^0\alpha_{21}^0\alpha_{22}^1 \begin{bmatrix} d_{11}^1 d_{12}^0 & 0 \\ 0 & d_{21}^0 d_{22}^1 \end{bmatrix} \\ +\alpha_{11}^0\alpha_{12}^1\alpha_{21}^1\alpha_{22}^0 \begin{bmatrix} d_{11}^0 d_{12}^1 & 0 \\ 0 & d_{21}^1 d_{22}^0 \end{bmatrix} + \alpha_{11}^0\alpha_{12}^1\alpha_{21}^0\alpha_{22}^1 \begin{bmatrix} d_{11}^0 d_{12}^1 & 0 \\ 0 & d_{21}^0 d_{22}^1 \end{bmatrix} \end{gathered} \tag{37}$$

Note that every block has each element powered to 1 or 0 following the four combinations on l.
In other words, terms $\mathfrak{d}_l$ can be defined as:

$$\mathfrak{d}_{1010} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathfrak{d}_{1001} = \begin{bmatrix} 1 & 0 \\ 0 & 1.0461 \end{bmatrix}, \quad \mathfrak{d}_{0110} = \begin{bmatrix} 3.2166 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathfrak{d}_{0101} = \begin{bmatrix} 3.2166 & 0 \\ 0 & 1.0461 \end{bmatrix} \tag{38}$$

Then, $\mathfrak{B}_l = e^{AT} V \mathfrak{d}_l V^{-1} B$ can be calculated. As there is only one control action B is used instead of B_1.

$$\mathfrak{B}_{1010} = \begin{bmatrix} -4.6320 \\ -2.4377 \end{bmatrix}, \quad \mathfrak{B}_{1001} = \begin{bmatrix} -4.8099 \\ -2.5693 \end{bmatrix}, \quad \mathfrak{B}_{0110} = \begin{bmatrix} -6.3481 \\ -1.5104 \end{bmatrix}, \quad \mathfrak{B}_{0101} = \begin{bmatrix} -6.5260 \\ -1.6421 \end{bmatrix} \tag{39}$$

Finally, the Takagi-Sugeno fuzzy model is obtained with:

$$x_{k+1} = e^{AT} x_k + \sum_{l\in\mathcal{B}} \alpha^l(\delta)\mathfrak{B}_l \Delta_k \tag{40}$$

For simplicity, this example only takes an order two system with one control action into account. Extending it to a multiple actions is straightforward. More complex cases occur when the system order is increased and complex eigenvalues do exist in it. The reader can check the github repository if there is interest on these cases. Also detailed explanation of this notation can be reed in [14].

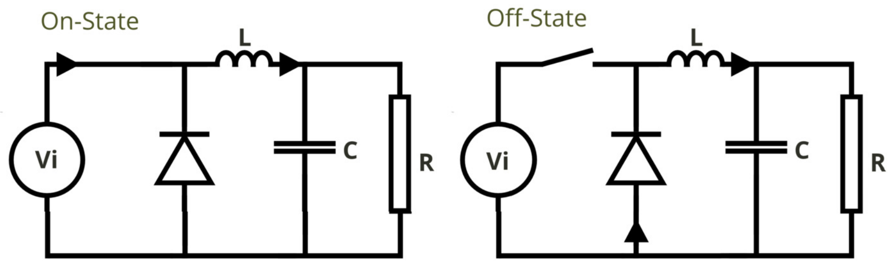

Fig. 2. Buck converter, with its binary controller (a switch).

5. Exactly discretized, Takagi-Sugeno fuzzy model examples

In this section, the full fuzzification process is applied to two example systems that need to be exactly discretized and converted into Takagi-Sugeno models.
The continuous system is transformed into a discrete system using the general procedure explained in Section 3 and then finally transformed into a fuzzy system following the steps on Section 4.
Once the system has taken shape of a Takagi-Sugeno fuzzy model, it will allow the use of controller techniques based on LMIs [4,6]. These last procedures are further explained on next sections 6 and 7.

5.1. Example: buck converter

The process above explained has on the buck converter one great example.
This circuit, shown in *Fig. 2*, opens and closes its switch to give and output voltage that is a fraction of the input given [2]. This reduction on voltage is done by switching on periodically the full input for a time $\mu \leq T$ and cutting that input for the rest of the time, being T the full duration of those periods.
During the time that the input is triggered, the output voltage rises while charging the capacitor and, when its shut down, that same capacitor keeps supplying the output while uncharging.
If a constant trigger time $\mu_k < T$ is maintained through every k period on the input voltage, when stabilizes, the output would oscillate near the average between the time that it has been on and off so $V_{avg} = V_i \frac{\mu_k}{T} + V_{zero}(1 - \frac{\mu_k}{T})$. Since $\frac{\mu_k}{T} = \Delta_k$ and the second part of the expression is always null, the above expression can be simplified as $V_{avg} = V_i \Delta_k$, only on this case where the activation triggers with a regular frequency.
Note that the output has been mentioned as an average since it will always have some ripple due to the on/off cycle and the measured value is not the average value it is some value near the minimum.
For this specific scenario, some values are assigned to each of the parts on this circuit:
$L = 100\, mH, \quad C = 1\, mF, \quad R = 10\,\Omega, \quad V_i = 200\ V$
The state and input matrices that define such system are:

$$A = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, \ B = \begin{bmatrix} \frac{V_i}{L} \\ 0 \end{bmatrix} \tag{41}$$

If a 50 V average output is wanted, then a constant $\Delta = 0.25$ would stabilize around that value since the input is 200 V and it will be bucked down to a quarter of it. This can be seen in *Fig. 3*, where there is a clear ripple around the 50 V marker instead of rising to match the 200 V input. Note that, since the system has a fixed delta and it is not controlled, the response from an uncharged system overshoots the 50 V marker to almost reach 60 V and then it stabilizes on the target value.
The remaining ripple obviously oscillates with the same frequency as the switch cycles, in this case, $T = 0.01s$. Now, to prove that the above algorithm to discretize that system matches the continuous response, the result of that discretization, for a time T that matches the trigger frequency, is overlapped with dots over the continuous response displaying a perfect coincidence.
Further analysis of this system, out from the main goal of this study, is done later on in Section 7, where a controller is implemented.

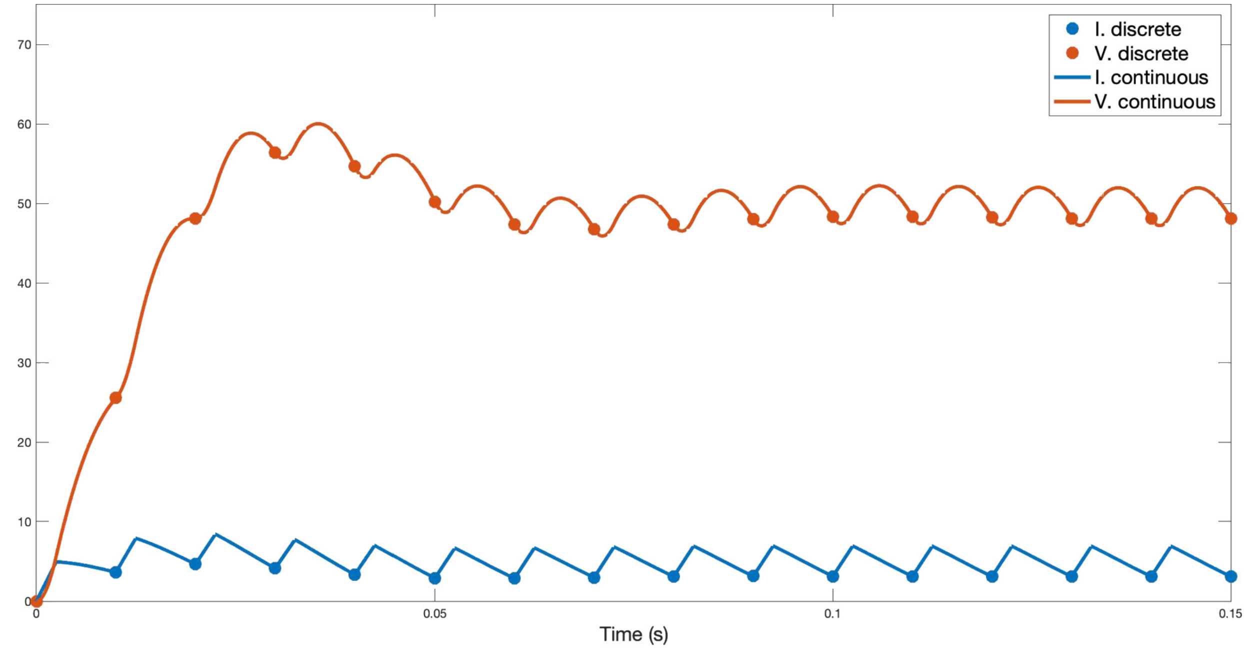

Fig. 3. Buck converter, discrete-continuous overlap.

5.2. *Example: multiple input binary system*

Now, the buck converter example only shows part of the potential for the discretization algorithm. As said at the beginning of this text, the goal for this process is to be able to discretize a system with independence of the type of poles or the quantity of inputs.
For instance, the following system is exposed:

$$A = \begin{bmatrix} -5 & -9 & -5 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{42}$$

This system has not only two complex poles as the buck converter does, but two complex poles, a pole in the real axis and one in the origin itself. Also, more than one control action applies to it, as can be seen on the number of columns for the B matrix. On this case, each of the columns, from now on B_i, can trigger with independent deltas, but the process to find the equivalent discretized system remains the same as explained on sections 2 and 4.
In this example a $\Delta_1 = 0.25$ is applied to B_1 and $\Delta_2 = 0.50$ applies to B_2, giving the continuous free response of the system shown in *Fig. 4*. As with the Buck converter example, the discretized response of that system is also shown in the same figure, displaying the expected coincidence marked with dot symbols all along the continuous response.

6. Controller design

Although the main subject of this study has been already exposed, this section has been added in order to fully comprehend the nature of this discretized model and to give support if the target system was to be controlled.
Since the remaining model has a TS fuzzy discrete form, the techniques explained on *"Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach"* by Kazuo Tanaka and Hua O. Wang [4], can be implemented straightforward into our system. For instance, if a stable fuzzy controller has to be implemented on that system, the following process, which is based on LMI conditions [15], can be used to do so:
The first step, the state matrix and the input matrix for the discrete system $x_{k+1} = \mathfrak{A}x_k + \mathfrak{B}u$ have to be identified. It is already known that:

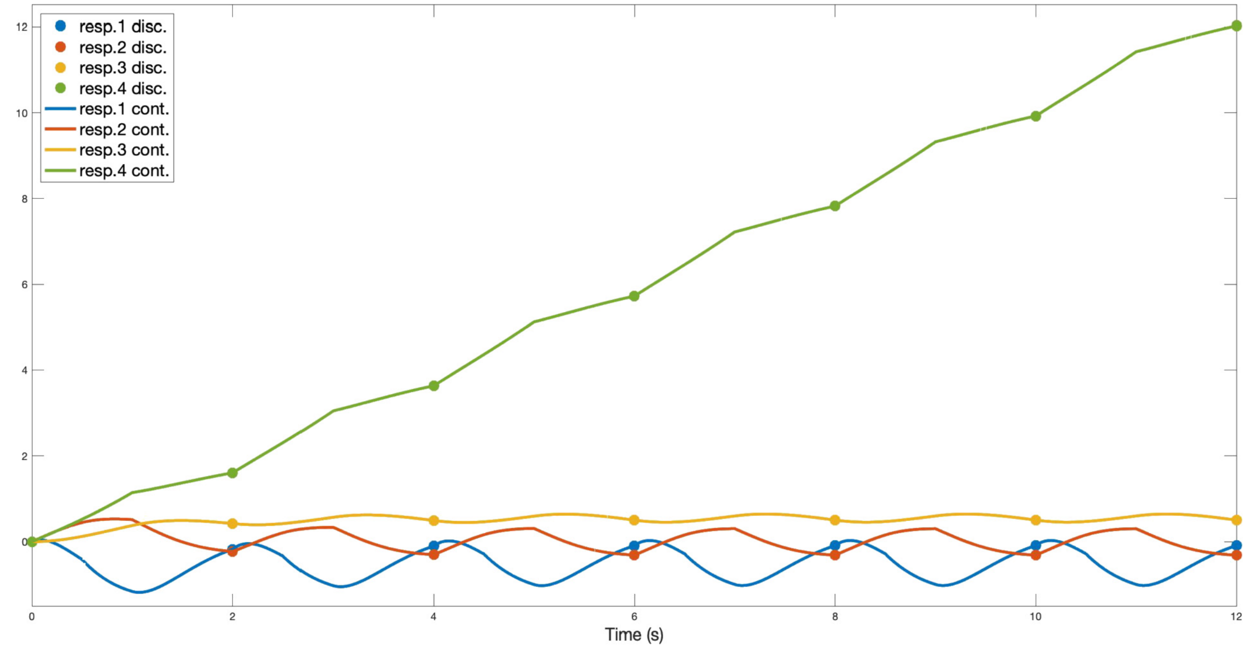

Fig. 4. Overlap of responses with various control actions.

$$x_{k+1} = e^{AT} x_k + \sum_{i=1}^{r} \sum_{l \in \mathcal{B}} \alpha^l(\delta_i) \mathfrak{B}_{l,i} \Delta_k \tag{43}$$

So the u dependant part will be the input matrix and the non dependant part, the state matrix. Like this:

$$\mathfrak{A} = e^{AT}, \quad \mathfrak{B} = \sum_{i=1}^{r} \sum_{l \in \mathcal{B}} \alpha^l(\delta_i) \mathfrak{B}_{l,i} \tag{44}$$

Then, following the simultaneous Lyapunov stability conditions, if there is an existing $P \in \mathbb{R}^{nxn}$ such as:

$$P > 0, \quad \mathfrak{A}^T P \mathfrak{A} - \lambda^2 P < 0. \tag{45}$$

The system will be stable with a decay rate of λ.
This condition is not guaranteed, and the ultimate goal is to stabilize the system. For that, an state feedback controller is proposed following the PDC structure [4]:

$$\Delta_k = \mathfrak{F} x_k = \sum_{i=1}^{r} \sum_{s \in \mathcal{B}} \alpha^s(\delta_i) F_{s,i} x_k \tag{46}$$

If the controller turns the system quadratically stable with λ decay rate, the following Matrix Inequality must hold [15]:

$$\sum_{i=1}^{r} \sum_{l \in \mathcal{B}} \sum_{s \in \mathcal{B}} \alpha^s(\delta_i) \alpha^l(\delta_i) (X\{\mathfrak{A} - \mathfrak{B}_{l,i} F_{s,i}\}^T X^{-1} \{\mathfrak{A} - \mathfrak{B}_{l,i} F_{s,i}\} X - \lambda^2 X) < 0 \tag{47}$$

Where $X = P^{-1}$.
A new matrix $M_{s,i} = F_{s,i} X$ is defined, following Tanaka and Wang guidelines, so that for $X > 0$ applies to the equation as $F_{s,i} = M_{s,i} X^{-1}$. Then the inequality can be rewritten, by applying Schur complement, as:

$$X > 0 \tag{48}$$

$$\sum_{i=1}^{r} \sum_{l \in \mathcal{B}} \sum_{s \in \mathcal{B}} \alpha^s(\delta_i) \alpha^l(\delta_i) \begin{bmatrix} \lambda^2 X & \left(\mathfrak{A} X - \mathfrak{B}_{l,i} M_{s,i}\right)^T \\ \mathfrak{A} X - \mathfrak{B}_{l,i} M_{s,i} & X \end{bmatrix} > 0 \tag{49}$$

Note that (49) can be rewritten as an homogeneous matrix polynomial in the form:

$$\sum_{i=1}^{r}\sum_{l\in\mathcal{B}^2}\alpha^s(\delta_i)G_{l,i}>0,\ \mathcal{B}^2=\{l\ /\ l\in\{0,1\}^m, l_p+l_{n+p}=2,\ p=1...n\} \tag{50}$$

This is a matrix polynomial expression on the positive variables $\alpha(\delta_i)$. So in order to prove (50) is positive, it is enough checking if all its different coefficients are positive. Less conservative results can be obtained applying the Polya theorem [8] and non-quadratic Lyapunov functions following [16–18]. Therefore checking if all coefficients $G_{l,i}$ are positive definite is a sufficient condition that ensures stability with a decay rate of λ.

$$X>0,\ G_{l,i}>0 \tag{51}$$

If there exist an X and $M_{s,i}$ so the above LMI holds, the state feedback gain is defined, as stated before, as $F_{s,i}=M_{s,i}X^{-1}$.
With this process, the input gain control is fully defined but the final result could be not a realizable one since those LMI do not take into account the restrictions that system may have.
For instance, it is know that Δ can't reach values above 1 on the binary system since, by definition $\Delta=\frac{t}{T}$ and that would mean than the control action takes place for more time than the actual period T, which is not possible.
To avoid that situation, the control input can be constrained if the initial condition $x(0)$ is known. To do so, the following conditions can be added:

$$\begin{gathered}\begin{bmatrix}1 & x(0)^T\\ x(0) & X\end{bmatrix}\geq 0,\\ \begin{bmatrix}X & M_{s,i}i^T\\ M_{s,i} & I\end{bmatrix}\geq 0\end{gathered} \tag{52}$$

Then, if the LMIs hold, the input will be constrained to $||\Delta_k||\leq 1$.

6.1. Note on controller implementation

Obtaining a controller this way, will cast a solution where the state feedback is $\Delta_k=\sum_{i=1}^{r}\sum_{s\in\mathcal{B}}\alpha^s(\delta_i)F_{s,i}x_k$ as shown in (46). As also previously stated, $\alpha^s(\delta_i))$ is the vector with all the controller membership functions which are δ_i dependant.
This means that, in order to know the required Δ for a given gain on the system for a moment k, all the individual δ_i have to be known previously, thus giving no other choice to implement this controller than using an iterative procedure for each of the periods involved on any analysis.

6.2. Linear controller design

The issues presented above, lead us to considerer the design of a simple linear controller for the Fuzzy model:

$$\Delta_k=Fx_k \tag{53}$$

In this case, the Matrix Inequality that ensures quadratic decay rate is:

$$\sum_{i=1}^{r}\sum_{l\in\mathcal{B}}\alpha^l(\delta_i)(X\{\mathfrak{A}-\mathfrak{B}_{l,i}F\}^TX^{-1}\{\mathfrak{A}-\mathfrak{B}_{l,i}F\}X-\lambda^2X)<0 \tag{54}$$

And a LMI that ensures decay are:

$$X>0 \tag{55}$$

$$\begin{bmatrix}\lambda^2X & \left(\mathfrak{A}X-\mathfrak{B}_{l,i}M\right)^T\\ \mathfrak{A}X-\mathfrak{B}_{l,i}M & X\end{bmatrix}>0 \tag{56}$$

for all $l\in\mathcal{B}$ and $i=1\ldots r$.
If this linear controller exist, for the required decay rate λ, it will be preferable than the fuzzy controller, because both controllers have the same performance but also the Linear controller has simpler implementation.

7. Controller implementation examples

7.1. Case 1

This section proposes the control of the Buck converter described on Section 5.1, which will be forced to have a voltage drop to 25 V from an initial voltage of 50 V. It seems to be obvious that, since the input is 200 V, the initial $\Delta(0)$ of that buck converter is operating at $\Delta = 0.25$ and, when stabilized after the drop, the remaining 25 V would be an indicator that the final Δ has settled at $\Delta = 0.125$. But it is not true for the discretized model. Because the measure of the voltage is done after the period T. This $\Delta = 0.125$ is a well know result where output voltage is assumed to be constant during the inter-period time. In order to obtain the final operation point Δ_f and x_f we follow the well known procedure for the steady state in discrete linear system applied to this Takagi-Sugeno model. That is:

$$x_f = \mathfrak{A} x_f + \mathfrak{B} \Delta_f \tag{57}$$

$$x_f = (I - \mathfrak{A})^{-1} \mathfrak{B} \Delta_f \tag{58}$$

$$y_f = C(I - \mathfrak{A})^{-1} \mathfrak{B} \Delta_f \tag{59}$$

$$\Delta_f = (C(I - \mathfrak{A})^{-1} \mathfrak{B})^{-1} y_f \tag{60}$$

Therefore, if $y_f = 25V$ then $\Delta_f = 0.1326$ and $x_f = [1.4941\ 25]^T$. The controller is calculated following the procedure of Section 6 with decay rate $\lambda = 0.8$ and without constraints on the control action. The controller gains are:

$$F_{1010} = [-0.0937\ \ -0.0019],\ F_{0110} = [-0.0463\ \ -0.000867] \tag{61}$$

$$F_{1001} = [-0.0705\ \ -0.0014],\ F_{0101} = [-0.0401\ \ -0.000781] \tag{62}$$

Finally de applied control action is:

$$\Delta_k = \mathfrak{F}(x_k - x_f) + \Delta_f \tag{63}$$

To better compare the result of this fuzzy controller from other well known approaches, the system has also been modelled into a discrete average model [19] and then controlled using the same LMI technique to achieve a unique controller gain for the same system $F = [-0.0743\ \ -0.0016]$. For this model the obtained steady state is $\Delta_f = 0.125$ and $x_f = [2.5\ 25]^T$.

Next, Fig. 5 represents the system response with a period of time $T = 0.01s$. On this figure, the buck converter controlled using the fuzzy controller is shown in blue, the system response controlled with the average model controller, is displayed in red.

It can clearly be seen on Fig. 5 that the fuzzy controller deploys a faster response than the other controller. And there is not a steady state error for the fuzzy controller.

This is due to the average model based controller has been design over an approximation then the stability and performance design $\lambda = 0.8$ are not guaranteed, as the average model may differ too much from the real system. In fact, if the period is reduced to $T = 0.001s$ then the average controller archives the performance design $\lambda = 0.8$. In order to avoid the steady state error an integrator can be added to the control design.

7.2. Case 2

On this final section, the same system that has been shown on previous sections is used. In this case, in order to test in a practical implementation an integrator is added to both controllers. This avoids the necessity of x_f, Δ_f and avoid the steady state error over disturbances (Fig. 6). Moreover the average controller can reach the reference, which was really complicated due to the difficulties for obtaining Δ_f. The design has been done following the well known techniques for discrete linear systems to our Takagi-Sugeno fuzzy model. That is:

$$u_k = \mathfrak{F} x_k + \mathfrak{Fi} \sum_{i=1}^{k} (y_{ref} - C x_i) T \tag{64}$$

Where $\mathfrak{F} = \sum_{s \in \mathcal{B}} \alpha^s(\delta) F_s$, and $\mathfrak{Fi} = \sum_{s \in \mathcal{B}} \alpha^s(\delta) Fi_s$ being Fi_s the fuzzy gains for integral error. In order to obtain the gains, an integral state $I_{k+1} = I_k + (y_{ref} - C x_k)$ is add to fuzzy system definition:

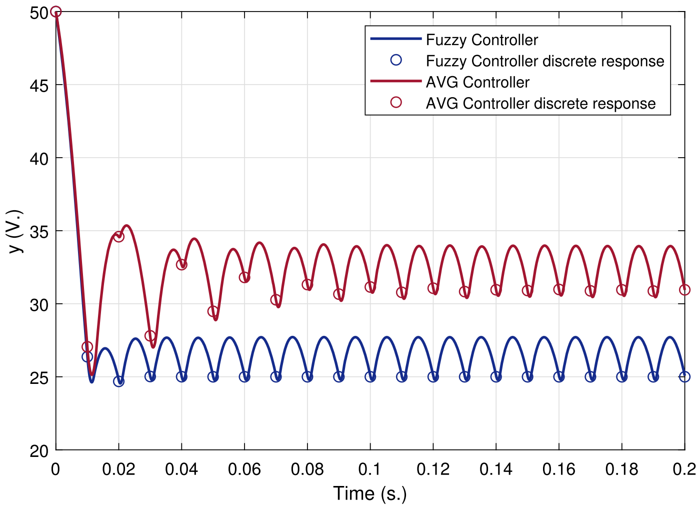

Fig. 5. Controlled responses, case 1. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

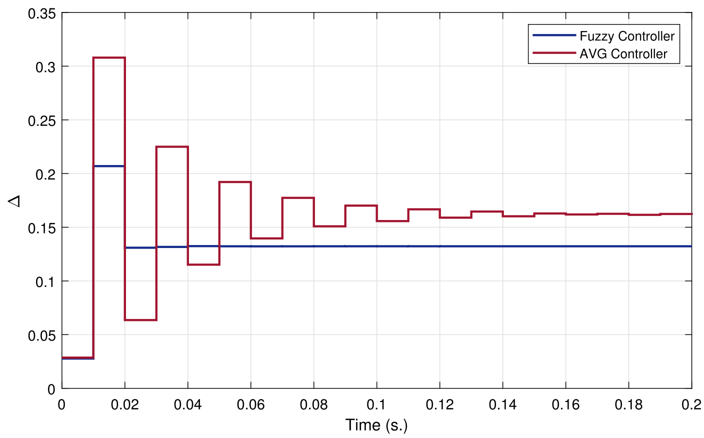

Fig. 6. Evolution of the control action (duty cycle Δ), case 1.

$$\hat{x}_{k+1} = \begin{pmatrix} \mathfrak{A} & 0 \\ -C & 1 \end{pmatrix} \hat{x}_k + \begin{pmatrix} \mathfrak{B} \\ 0 \end{pmatrix} u_k + \begin{pmatrix} 0 \\ 1 \end{pmatrix} y_{ref} \tag{65}$$

Note that, multiplying by T is unnecessary as it can be included in the controller gain. Then following Section 6 a Fuzzy controller can be computed. For a decay rate of $\lambda = 0.9$, obtained control gains are:

$$F_{1010} = [-0.0953 \ -0.0126 \ 0.0104], \ F_{0110} = [-0.0461 \ -0.0058 \ 0.0050] \tag{66}$$

$$F_{1001} = [-0.0712 \ -0.0092 \ 0.0077], \ F_{0101} = [-0.0402 \ -0.0051 \ 0.0044] \tag{67}$$

In this example, as in general, the implementation of fuzzy controller can require quite a few resources, a linear controller is also designed for the fuzzy model following Section 6.2. As in previous design, parameter λ is set to 0.9 the LMIs holds and linear controller is find with gain $F = [-0.0573 \ -0.0063 \ 0.0058]$. The same procedure is done for the avg. model $F_{avg} = [-0.0761 \ -0.0086 \ 0.0072]$. Fig. 7 shows the output responses with the same colour basis used on the previous figure. The red line represents the avg. controller, the blue line fuzzy controller and now the yellow is the linear controller for the fuzzy model. As it can be seen, the response with the controller based on an averaged model, has quite more settling time in comparison with the fuzzy modelled. In fact the avg. controller do not holds the decay rate design condition. This is due to the system approximation is not valid for this discretized time of $T = 0.01$. But in this case a zero steady state error is obtained (Fig. 8). Moreover changing λ to 0.8 makes the

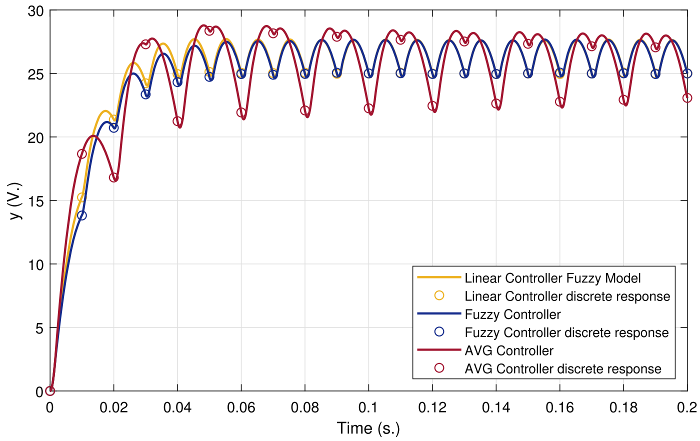

Fig. 7. Output Controlled responses, case 2.

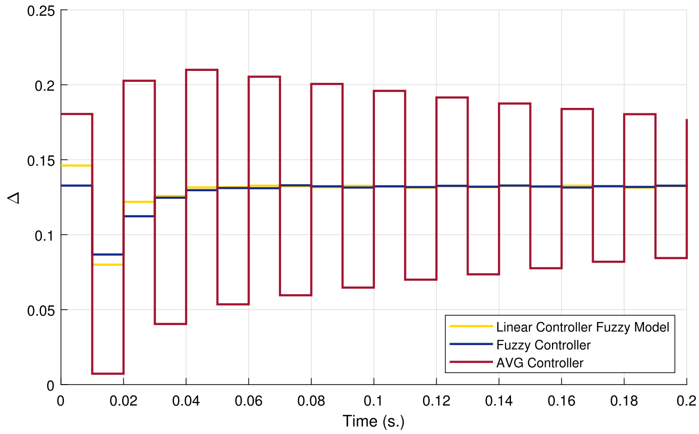

Fig. 8. Evolution of the control action (duty cycle Δ), case 2.

avg. controller response unstable. On the other hand the linear controller for the fuzzy model and the fuzzy controller present the same dynamic behaviour, this is the expected as both hold the LMIs with the same design parameters for the fuzzy model, which is an exact representation of the nonlinear one. Therefore, the linear controller will be the most appropriate since it controls the system properly and its implementation is much simpler.

8. Conclusions

A novel method to transform a non-linear, multiple input, binary controlled system, into a fuzzy discrete Takagi-Sugeno model has been presented. This model is an exact representation of the original non-linear system. The model is a Takagi-Sugeno fuzzy model, so all the previous developments in control for this kind of system can be used. In the final sections, some of them have been presented showing that the use of controller techniques based on LMI can archive better results than other well known techniques.

Although the process to convert such system into a Takagi-Sugeno model and the develop of controllers for this system may seem tedious or complex using the instructions above, further material has been upload to a repository which allows an easy way to convert any system that meets the criteria here exposed.

9. Resources

All the examples shown above, along with the code necessary to be implemented can be freely downloaded from:
Github repositori: https://github.com/Mas-T/Binary
It may be necessary to add third party add-ons to Matlab, such as Yalmip and Mosek, to fully run the code.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data is available at: https://github.com/Mas-T/Binary

References

[1] J.R. Salvador, T. Alamo, D.R. Ramirez, D. Muñoz de la Peña, Model predictive control of partially fading memory systems with binary inputs, J. Process Control 64 (2018) 141–151, https://doi.org/10.1016/J.JPROCONT.2018.02.006.

[2] D.W. Hart, Power Electronics, vol. 32, McGraw-Hill, New York, 2011.

[3] N.N. Nandola, S. Bhartiya, A multiple model approach for predictive control of nonlinear hybrid systems, J. Process Control 18 (2) (2008) 131–148, https://doi.org/10.1016/J.JPROCONT.2007.07.003.

[4] K. Tanaka, H.O. Wang, Fuzzy Control Systems Design and Analysis: a Linear Matrix Inequality Approach, John Wiley & Sons, 2004.

[5] X.-P. Xie, Z.-W. Liu, X.-L. Zhu, An efficient approach for reducing the conservatism of LMI-based stability conditions for continuous-time T-S fuzzy systems, Fuzzy Sets Syst. 263 (2015) 71–81, https://doi.org/10.1016/j.fss.2014.05.020, theme: Automatic Control. https://www.sciencedirect.com/science/article/pii/S0165011414002644.

[6] Z. Lian, Y. He, M. Wu, Stability and stabilization for delayed fuzzy systems via reciprocally convex matrix inequality, Fuzzy Sets Syst. 402 (2021) 124–141, https://doi.org/10.1016/j.fss.2019.12.008, theme: Control Engineering. https://www.sciencedirect.com/science/article/pii/S0165011419305263.

[7] I. Nodozi, M. Rahmani, LMI-based robust mixed-integer model predictive control for hybrid systems, Int. J. Control 93 (10) (2019) 2336–2345, https://doi.org/10.1080/00207179.2018.1556808, https://www.tandfonline.com/doi/abs/10.1080/00207179.2018.1556808.

[8] A. Sala, C. Arino, Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: applications of polya's theorem, Fuzzy Sets Syst. 158 (24) (2007) 2671–2686.

[9] C. Ariño, A. Sala, Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure, Eng. Appl. Artif. Intell. 20 (8) (2007) 1036–1046.

[10] H.O. Wang, K. Tanaka, M. Griffin, An Analytical Framework of Fuzzy Modeling and Control of Nonlinear Systems: Stability and Design Issues, Proceedings of 1995 American Control Conference-ACC'95, vol. 3, IEEE, 1995, pp. 2272–2276.

[11] R.A. DeCarlo, Linear Systems: A State Variable Approach with Numerical Implementation, Prentice-Hall, Inc., 1989.

[12] H.O. Wang, K. Tanaka, M.F. Griffin, An approach to fuzzy control of nonlinear systems: stability and design issues, IEEE Trans. Fuzzy Syst. 4 (1) (1996) 14–23.

[13] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 1 (1985) 116–132.

[14] C. Arino, A. Querol, A. Sala, Shape-independent model predictive control for Takagi-Sugeno fuzzy systems, Eng. Appl. Artif. Intell. 65 (2017) 493–505.

[15] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, SIAM, 1994.

[16] T. Guerra, W. Perruquetti, Non-quadratic stabilisation of discrete Takagi-Sugeno fuzzy models, in: 10th IEEE International Conference on Fuzzy Systems. (Cat. No. 01CH37297), vol. 3, 2001, pp. 1271–1274, vol. 2.

[17] A. Kruszewski, R. Wang, T.M. Guerra, Nonquadratic stabilization conditions for a class of uncertain nonlinear discrete time TS fuzzy models: a new approach, IEEE Trans. Autom. Control 53 (2) (2008) 606–611, https://doi.org/10.1109/TAC.2007.914278.

[18] R. Márquez, T.M. Guerra, M. Bernal, A. Kruszewski, A non-quadratic Lyapunov functional for H_∞ control of nonlinear systems via Takagi–Sugeno models, J. Franklin Inst. 353 (4) (2016) 781–796, https://doi.org/10.1016/j.jfranklin.2016.01.004, https://www.sciencedirect.com/science/article/pii/S0016003216000168.

[19] D. Yan, C. Yang, L. Hang, Y. He, P. Luo, L. Shen, P. Zeng, Review of general modeling approaches of power converters, Chin. J. Electr. Eng. 7 (1) (2021) 27–36.