HOSM control under quantization and saturation constraints: Zig-Zag design solutions

Leonardo Amet, Malek Ghanes and Jean-Pierre Barbot

Abstract—In many experimental systems, discrete and bounded actuators implement control laws with sampling, quantization and saturation problems. This paper is dedicated to only the last two in the context of the implementation of a Super-Twisting sliding mode control. A new control design, called “Zig-Zag sliding mode control”, is proposed. Issues of quantization and saturation problems are respectively investigated directly and implicitly by the proposed control. The main contribution of the proposed method consists in having a faster convergence and well performances even when the saturation of the actuators is decreased up to a certain limit in which other methods fail to converge. Simulation results of the proposed method compared to the results of traditional implementations highlight the well founded Zig-Zag design.

I. INTRODUCTION

It is well known that sliding mode techniques provide very good properties such as robustness against a certain class of perturbations and parametric incertitude, as well as finite time convergence of the switching function $s$ to zero [1], [2], [3]. The main disadvantage of this technique is a phenomenon called “chattering”, which consists of high-frequency oscillations around the sliding surface [4]. This behavior may become difficult, even impossible, the implementation of this technique in certain systems, such as mechanical ones. It can also excite non modeled high frequency modes that could destabilize the system under control. Higher order sliding mode techniques retain the excellent properties of classical sliding modes, but minimize the “chattering” effect [1], [5].

Moreover, real systems such as A/D-D/A converters, power electronic converters and actuators in general introduce problems such as discretization, quantization and/or saturation. Because of these limitations, the implementation of continuous control techniques may degrade its performances [6], [7], [8], [9].

In this paper the problems of quantization and saturation are addressed. Implicitly, the sampling frequency is considered fast enough. Some results dealing with this topic for linear systems and nonlinear systems can be found in [7], [10], [11] and [12], [13] respectively. However, for our best knowledge, both quantization and saturation problems are not treated in the case of higher order sliding mode control.

From this point of view, a new way to implement the Super-Twisting algorithm under saturation and quantization constraints is proposed. Our method is presented in the basis of a very simple example which is representative of a wide range of industrial applications.

The present work is organized as follows. The problem statement of this work is presented in Section II. Section III introduces the proposed Zig-Zag sliding mode controller. Simulation results and a comparative study of different quantization techniques are illustrated in section IV. Finally, some concluding remarks and future researches are drawn in the last section.

II. PROBLEM STATEMENT

In order to illustrate the advantage of our method we will consider a system found in a large number of electromechanical applications : an RLE load, driven at first by an ideal actuator, and by a real industrial actuator then. Its state space model is as follows:

$$\frac{di}{dt} = \frac{R_i}{L} - \frac{E}{L} + \frac{u}{L}$$

(1)

This model could represent, for example, the armature circuit of a DC electric motor (see [2]). It is well known that the current of such a load can be controlled by a Super-Twisting controller. In fact, the following switching function

$$s = i_{\text{ref}} - i$$

(2)

has relative degree one, which is deduced owing to the explicit presence of the control $u$ in its first time derivative:

$$\dot{s} = \frac{ds}{dt} = \frac{di_{\text{ref}}}{dt} + \frac{R_i}{L} + \frac{E}{L} - \frac{u}{L}$$

(3)

Under this condition, the control may be performed by a Super-Twisting algorithm [1], which is described by the following equations:

$$u = u_1 + u_2$$

(4)

where:

$$\dot{u}_1 = \alpha \text{sign}(s)$$

(5)

and

$$u_2 = \lambda |s|^\rho \text{sign}(s)$$

(6)

with $\rho < 1$.

We show in figure 1 the current of an RLE load controlled by a Super-Twisting algorithm. The parameters used for this simulation are shown in table I and will be taken as reference for all the simulations run in this paper. The reference signal is a sinusoid of 100 Hz and 2A of amplitude. The controller parameters and the initial condition were set to achieve convergence at 25 ms.

Unfortunately sources are not ideal. In the case of linear ones we must deal with saturations, but in power
TABLE I: Reference parameters

<table>
<thead>
<tr>
<th>RLE Load</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1Ω</td>
</tr>
<tr>
<td>L</td>
<td>50mH</td>
</tr>
<tr>
<td>E</td>
<td>20V</td>
</tr>
<tr>
<td>ST controller</td>
<td></td>
</tr>
<tr>
<td>α</td>
<td>$210^5$</td>
</tr>
<tr>
<td>λ</td>
<td>10</td>
</tr>
<tr>
<td>ρ</td>
<td>0.5</td>
</tr>
<tr>
<td>S₀</td>
<td>20</td>
</tr>
</tbody>
</table>

Fig. 1: Super-Twisting current control

applications we usually find sources whose output can take two or more fixed voltage levels, so we have saturation and quantization constraints. This is for example the case of multilevel converters [14], [15]. In this work we will consider systems with linearly quantized symmetric outputs. Such kind of systems may be classified as with odd and even number of levels. Before the description of the proposed Zig-Zag control, some definitions have to be done.

- We define the quantization error bound $a$ as follows:

$$ a = \frac{U_{\text{max}}}{N - 1} $$

where $U_{\text{max}}$ is the saturation value and $N$ is an integer representing the number of levels of the actuator.

- We recall the floor and ceiling functions, noted as $\lfloor x \rfloor$ and $\lceil x \rceil$, respectively:

$$ \lfloor x \rfloor = \max \{ i \in \mathbb{Z} | i \leq x \} $$

$$ \lceil x \rceil = \min \{ j \in \mathbb{Z} | j \geq x \} $$

- At last, we recall the round function, based on definitions (8) and (9):

$$ [x] = \begin{cases} 
  \lfloor x - 0.5 \rfloor & \text{if } x < 0 \\
  \lceil x + 0.5 \rceil & \text{if } x \geq 0 
\end{cases} $$

In what follows, $u_{\text{ST}}$ is the input given by the Super-Twisting algorithm (see equation (4)) and $u_{\text{ZZ}}$ the input obtained with Zig-Zag one. This approach consists of four steps, presented in the next section, which take into account the saturation and quantization constraints.

III. PROPOSED QUANTIZATION LAWS (ODD AND EVEN CASES)

A. Systems with odd number of levels

The output of such systems can be expressed as:

$$ u_{\text{odd}} = ma $$

with $m$ is an even integer given by

$$ m = 2k_{\text{odd}} $$

where $k_{\text{odd}} \in \mathbb{Z}$ is the actual output level, which is between $N_{\text{min}} = -\frac{N-1}{2}$ and $N_{\text{max}} = \frac{N-1}{2}$.

B. Systems with even number of levels

In this case, the output is given by the following expression:

$$ u_{\text{even}} = na $$

with $n$ and odd integer given by

$$ n = 2k_{\text{even}} - \text{sign}(k_{\text{even}}) $$

where $k_{\text{even}} \in \mathbb{Z} - \{0\}$ is the actual output level. It can take values between $N_{\text{min}} = -\frac{N}{2}$ and $N_{\text{max}} = \frac{N}{2}$. The presence of the sign function in equation (14) allows the resulting $n$ to be symmetric with respect to zero.

We describe now the four steps needed to perform the Zig-Zag algorithm.

- **Normalization:** in this step $u_{\text{ST}}$ is normalized with respect to $2a$:

$$ \bar{u}_{\text{ST}} = \frac{u_{\text{ST}}}{2a} $$

This operation is useful in the following step.

- **Map to integer numbers (Zig-Zag design):** in this step the normalized output $\bar{u}_{\text{ST}}$ is mapped to integer values which represent the output levels of the actuator.

It is clear that a classic quantization of the Super-Twisting algorithm, given by

$$ u_q = 2a \lceil \bar{u}_{\text{ST}} \rceil $$

degrades its performances as it will be shown in simulation. To overcome this problem, a different quantization is proposed.

Our strategy cancels the quantization error by introducing the term $\text{sign}(s)$, where $s$ is the switching manifold of the Super-Twisting algorithm. By doing this we introduce some chattering over the variable being controlled.

For systems with odd number of levels the proposed map is the following:

$$ m' = \begin{cases} 
  \frac{1}{2} (\lfloor \bar{u}_{\text{ST}} \rfloor + \lceil \bar{u}_{\text{ST}} \rceil) + \text{sign}(s) & \text{if } \bar{u}_{\text{ST}} \not\in \mathbb{Z} \\
  \bar{u}_{\text{ST}} & \text{if } \bar{u}_{\text{ST}} \in \mathbb{Z} 
\end{cases} $$

(16)

In the case of systems with even number of levels the map is as follows:

$$ n' = \begin{cases} 
  \lceil \bar{u}_{\text{ST}} \rceil + \frac{1}{2} \text{sign}(s) & \text{if } \bar{u}_{\text{ST}} \not\in \mathbb{Z} \\
  \bar{u}_{\text{ST}} & \text{if } \bar{u}_{\text{ST}} \in \mathbb{Z} 
\end{cases} $$

(17)
For the sake of simplicity of notation we introduce the variable $k_{ZZ}'$, defined as:

$$k_{ZZ}' = \begin{cases} 2m' & \text{for odd number of levels} \\ 2n' & \text{for even number of levels} \end{cases}$$

(18)

- **Introduction of saturation constraints:** the quantified control $k_{ZZ}'$ must respect the limits of the actuator, then a saturation is imposed at this step:

$$k_{ZZ} = \begin{cases} k_{ZZ}' & \text{if } -N + 1 < k_{ZZ}' < N - 1 \\ -N + 1 & \text{if } k_{ZZ}' < -N + 1 \\ N - 1 & \text{if } k_{ZZ}' > N - 1 \end{cases}$$

(19)

- **Denormalization:** the real output value of the Multilevel System is determined as follows:

$$u_{ZZ} = a_k ZZ$$

(20)

**C. Example of Zig-Zag quantization**

Assume two identical systems, one of them controlled by a Super-Twisting algorithm and the other one by a Zig-Zag twisting algorithm. Suppose also that the sliding mode is established in both of them, i.e., $s = 0$. Under these assumptions it is possible to use the concept of equivalent control [2], [16], i.e., a fictitious continuous control $u_{eq}$ which forces $s = 0$. By replacing equation (6) in (4), we can rewrite the last one as:

$$u_{ST} = u_1 + \lambda |s|^p \text{sign}(s)$$

(21)

Given that $u_{ST}$ is continuous and that $s = 0$:

$$u_{ST} = u_1 = u_{eq}$$

(22)

Note that $u_{eq}$ is the same for both systems as this does not depend on the technique used but on condition $s = 0$. Suppose that $u_{eq}$ is as shown in figure 2 and that the “quantized actuator” in which the Zig-Zag implementation is performed does not saturate so as the sliding mode is not lost. Now we describe the steps to obtain the

Zig-Zag implementation. In the normalization step $u_{ST}$ is normalized with respect to twice the normalization error bound as shown in figure 3. The two first terms of equation (16) maps $u_{ST}$ to the mean value of the levels in which $u_{ST}$ is “contained”. This is represented by the blue line shown in figure 3: when $u_{ST}$ is between 0 and 1 the mean value is 0.5, when $u_{ST}$ is between 1 and 2 it is 1.5 and when $u_{ST}$ is between 2 and 3 it is 2.5. Now the last term is taken into account: $\frac{1}{2} \text{sign}(s)$. It depends on $s$ and “decides” which level $m'$ is assigned to.

The saturation step does not affect the quantified control because this is in between the limits of the actuator. Finally, the output of the actuator is determined by the denormalization step. The Zig-Zag control is shown in figure 4. The green zones represent the high frequency switching given by the sign function.

Note that the Zig-Zag control could be locally seen as a classical sliding mode control. In fact, by making the change of variables

$$u' = u_{ZZ} - a(\lfloor \bar{u}_{ST} \rfloor + \lfloor \bar{u}_{ST} \rfloor)$$

in the odd number of levels case, and

$$u' = u_{ZZ} - 2a \lfloor \bar{u}_{ST} \rfloor$$

in the even number of levels one, $u'$ becomes

$$u' = a \text{sign}(s).$$

(23)

in both cases.

It is shown in figure 5 a typical phase portrait of the Zig-Zag twisting algorithm. This algorithm is called “Zig-Zag twisting” because it converges to the origin “twisting” around it as the Super-Twisting algorithm does, but in a zig-zag manner.

Now, in order to highlight the benefits of the Zig-Zag design in the next sections, it will be compared in simulations to usual implementation techniques, such
IV. COMPARATIVE STUDY BASED ON SIMULATION RESULTS

In this section we will run two simulations in order to compare the Zig-Zag technique with continuous Super-Twisting, classic quantization and multilevel PWM methods. Saturation is present in the four implementations. The parameters of the load and Super-Twisting controller are those used in section II. The only difference between simulations remains in the parameters of the multilevel converter.

A. Case 1

In this case the number of levels is $N = 5$ and the saturation voltage $U_{\text{max}}$ is 130 V. Results are shown in figures 6a to 6d. It can be seen that, in this case, convergence is achieved only with the Zig-Zag implementation. Even the Super-Twisting algorithm fails to converge.

B. Case 2

Now, the number of levels is still $N = 5$ but the saturation level is modified to $U_{\text{max}} = 150$ V. Results are shown in figures 7a to 7d. In this case the four implementations provide convergence of the load current to its reference. As it can be seen, the Zig-Zag solution provides a convergence which is about five times faster.

V. CONCLUSION AND PERSPECTIVES

In this paper we have proposed a new way to deal with quantization and saturation problems for the special case of a Super-Twisting controller. Simulation results highlighted the performances of the proposed method. Our ongoing works will focus on generalizations of the Zig-Zag method to other HOSM algorithms and on proof of convergence in the generalized case with analytical conditions.

REFERENCES


R. W. Brockett and D. Liberzon, “Quantized feedback stabi-