IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 July 2022, accepted 23 July 2022, date of publication 29 July 2022, date of current version 8 August 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3194911*

RESEARCH ARTICLE

# Predictor-Based Current Limitation Method for a DC Motor-Actuated Upper-Limb Rehabilitation Exoskeleton

**DAVID PONT-ESTEBAN, MIGUEL ÁNGEL SÁNCHEZ-URÁN, ALDO FRANCISCO CONTRERAS-GONZÁLEZ, AND MANUEL FERRE, (Senior Member, IEEE)**
Centro de Automática y Robótica UPM-CSIC, Universidad Politécnica de Madrid, 28006 Madrid, Spain

Corresponding author: Manuel Ferre (m.ferre@upm.es)

This work was supported in part by the "Lightweight Upper Limbs eXosuit for BImanual Task Enhancement/Exoesqueleto Ligero del Tren Superior para Ayuda a Tareas Bimanuales (LUXBIT)" through the Spanish Ministerio de Ciencia, Innovación y Universidades, under Grant RTI2018-094346-BI00; and in part by the "Programa Propio de I+D+i" of the Universidad Politécnica de Madrid.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by UPM Ethical Committee.

**ABSTRACT** Current limitation is crucial to protect DC motors from overheating in overload situations. This consideration is more critical in applications that involve the movement of variable loads, such as exoskeleton applications, where there exists dynamic human-robot interaction. This work presents a software current limitation method based on a continuous DC motor model predictor. By predicting, according to the motor model, the voltages that would make the current match a threshold, the proposed method dynamically saturates the maximum voltages applicable to the motor to maintain the current under the predefined threshold. Using this method, a full-time current limitation can be obtained, but if desired, it also allows current peaks of configurable duration. Only motor parameters need to be known for the application of this algorithm, and it is easily adjustable to different DC motor models. Stand-alone motor tests show an effective limitation of current, achieving up to a 98.04% of maximum current when limited. The method has also been successfully validated in a rehabilitation exoskeleton application with four participants, obtaining an average current of 92.34% of current threshold when limited.

**INDEX TERMS** Current limiters, DC motors, embedded software, exoskeletons, rehabilitation robotics.

## I. INTRODUCTION

Most robots are electromechanical systems that are actuated by electric motors. Every kind of electric motor has operational limits in terms of current, temperature and speed. Surpassing those limits can shorten the life expectancy of the motors or even destroy them.

It is necessary to introduce a current limitation system in every electric motor to protect it from overcurrent. Overcurrent situations can be generated, mainly, in four different scenarios: motor start, abrupt position or speed variations in control references, abrupt variations in external load and application of high external loads. All four of these situations are common in any machine that uses electric motors, and are therefore important to consider when designing a robotic system. In the robotics field, clear exponents of this issue are robots that work in unstructured scenarios (for instance, a rock can be preventing a mobile robot from moving) or robots which involve physical interaction with humans (such as exoskeletons and exosuits).

In regard to the concrete case of exoskeletons, the torque applied by a human in a specific joint can cause an abrupt load variation for the joint actuator or even a too high external load, depending on the dimensioning of the motor and the

The associate editor coordinating the review of this manuscript and approving it for publication was Giulio Reina.

applied torque magnitude. Some exoskeletons include torque or force sensors in the joints, so if the torque applied by the user is excessive, the motor controller can change the position or speed references to reduce the interaction force and, consequently, the motor current.

There exist two general types of applications for exoskeletons in terms of control modes: motion assistance ([1]–[3]) and passive rehabilitation ([4], [5]). Motion assistance relies on impedance/admittance control architectures to adapt to the movements the user performs. During normal motion, these movements are often produced in opposite to the motor, increasing its current even if only for a short period of time. The impedance/admittance controller should be able to rapidly adapt motor speed to favor the movement, but depending on the application and the use the subject makes of the exosuit (for instance, full articular range fast movements), overcurrent situations could arise. Alternatively, passive rehabilitation does not require the use of force sensors. Predefined trajectories are programmed and the user leaves the limb inert. In this case, if for some reason the user impedes the movement of their limb or a disturbance is applied, a motor stall scenario could be provoked and damage the motors. During normal operation, overcurrent situations will generally be acceptable (high current peaks for a short period of time), but aggressive user motion, in terms of acceleration, speed and torque can severely damage the motor if no current limiting method is implemented.

Upper-limb exoskeletons are commonly actuated by DC motors or brushless DC motors (BLDC). Although the latter ones have been most common through last decades to the former ones, DC motors still represent an interesting option for exoskeleton design, as show the devices in [2], [6], [7] and [8]. The common use of DC motors in exoskeletons is exposed in the review carried out in [9] about wearable rehabilitation and assistive devices for the shoulder joint, in which 60 devices were examined. Up to the 20% of the devices were actuated by DC motors, 18.3% by BLDC motors, 18.3% by pneumatic actuators, 5% by servomotors, 11.7% by unspecified-type electric motors and the rest by other systems or the actuation system was unspecified.

Although the limitation of current in electric motors is a problem that has been addressed for decades, in recent years, significant effort is still being made in the design of techniques for limiting current in different kinds of electric motors ( [10]–[14]). In this regard, and according to the authors in [15], there exist two families of current limitation methods in terms of their presence in the normal operation of the motor. On the one hand, interventionist methods actuate only when the current exceeds a certain limit. On the other hand, regulating methods are current controllers that are normally implemented as the inner control loop inside a speed control loop. Speed controller outputs are used as current loop references. Since the current is being controlled, current limits are not exceeded. The main drawback of interventionist methods is that an overshoot beyond the current limit threshold has to occur in order to start limiting the current. In contrast, their main advantage in comparison with regulated methods is that it is only necessary to design and adjust a single control loop, whether it is a speed or a position one. In this way, the motor motion loop is designed independently and is not affected by an inner control loop.

Depending on their implementation, current limitation methods can also be classified in hardware and software ones. On the one hand, hardware methods implement the limitation logic in electronic circuits which control the commutation of the motor's driving circuit's switching semiconductors. Alternatively, software methods act over the voltage command outputted from the speed controller before that command is sent to the motor driver circuit. Their main advantage over hardware ones is that they are much more flexible (easily adaptable to motors with different parameters) and that they can reduce the application cost since they can be implemented in the control application microcontroller.

To the knowledge of the authors, the specific issue of current limitation in exoskeleton applications is not addressed in current literature. However, current limitation methods of different type, according to the aforementioned classification, have been studied in other applications. In [16], a hardware interventionist method was designed to avoid implementing a current controller in a speed control application. Specific current sensing and limiting circuits were proposed. Similarly, in [15] another hardware interventionist method was proposed. The current is compared with a limit threshold and, if limitation is necessary, the limitation circuit controls the switching state of the motor driving circuit, ignoring the speed controller commands. The method in [17] also presents a hardware interventionist method. In this case, the current is allowed to achieve a 150% of the maximum current for a period of 500 ms. Afterwards, the current is limited to the 100% of the maximum. The designed limiting circuit is based on opening the corresponding low side transistor of the H-bridge when current has to be limited, forcing the current to circulate in the high-side and, therefore, rapidly reducing it. This philosophy is followed by many off-the-shelf DC motor drivers with overcurrent protection. The works presented in [18] and [19] show a hardware regulated method for a separately excited DC motor. In both cases, a Hysteresis Current Controller (HCC) is used to implement the inner current control loop. The current is constantly controlled via hardware by comparing the current value with the two thresholds of a hysteresis window, which generates positive or negative pulses that control the switching of the chopper circuit.

As for the software methods, only interventionist methods are of interest in this work since regulated ones simply consist on implementing the aforementioned speed-current cascade control architecture. The following software interventionist methods are based on limiting the voltage applied to the motor if an overcurrent situation arises. The authors in [20] propose an interesting interventionist approach: if the motor current sample $i[k-1] > 0.8i_{\max}$, then, the voltage applied to the motor is $u[k] = 0.98u[k-1]$. Otherwise, the voltage command generated by the speed controller is transferred to the driver circuit, bypassing the current limitation module. The work presented in [21] introduces a method to limit the motor initial current by limiting the voltage variation between consecutive sampling periods. If the voltage to be applied by the speed controller belongs to the allowable range, that voltage is applied to the motor. Otherwise, the voltage corresponding to the maximum allowable variation with respect to the previous sample is applied. In both algorithms, an arbitrary parameter tuning according to the used motor and the current threshold is necessary. However, no general selection method is provided, making it difficult to implement these algorithms over different motors.

In this paper, a software interventionist current limitation method for DC motors via a predictor based on the motor's continuous dynamic model is proposed. The objective is to dynamically saturate the voltage applied to the motor in order to avoid overcurrent situations. The algorithm allows current to surpass the threshold during a configurable period of time. In this way, more torque can be delivered in a controlled way in situations as, for example, motor start.

The remaining of the document is structured as follows. Section II presents the analytical design and analysis of the proposed algorithm along with simulation results, applying the limitation method under different conditions. A comparison with methods [20] and [21] is provided. Section III presents the experimental tests that have been conducted to validate the performance of the proposed method in a real application, which include the implementation of the algorithm over an upper-limb rehabilitation exoskeleton. Finally, Section IV concludes this article.

## II. ALGORITHM DESIGN AND ANALYSIS

The following development is focused on the design of the algorithm for a single DC motor. For simulation, the chosen parameters are the ones of a Maxon DCX22S GB KL 48V, which is the model that authors use for their exoskeletons/exosuits in [5] and [22]. Motor parameters can be found in Table 1 and have been extracted from the datasheet. However, $R$ and $k_e$ have been measured experimentally following the typical procedure exposed in [23], as the precision in those values highly influences the performance of the algorithm, as will be treated later in this section. The method is first designed considering that any desired voltage in the operational range (±24 V) can smoothly and directly be applied to the motor. Further on, the effect of including an H-bridge as the motor driver is analyzed.

### A. ANALYTICAL FORMULATION

In order to limit the motor current, a method consisting of the dynamic saturation of the voltage applied to the motor by means of a simple predictor based on the DC motor's continuous model is proposed. The motor electric model is defined in (1).

**TABLE 1.** Selected motor model parameters.

| Sym. | Description | Value |
|---|---|---|
| $R$ | Resistance | $18\ \Omega$ |
| $L$ | Inductance | 0.881 mH |
| $k_e$ | Electric constant | 0.0359 V/(rad/s) |
| $k_p$ | Torque constant | 0.0359 N/A |
| $J$ | Inertia | $0.2941\ \mathrm{kg}\cdot\mathrm{m}^2$ |
| $f$ | Friction coefficient | $0.6299\ \mathrm{N}\cdot\mathrm{m}/(\mathrm{rad/s})$ |
| $n$ | Reduction ratio | 794 |
| $i_{\mathrm{sat}}$ | Maximum current | 0.4 A |

$$u(t) = Ri(t) + L\frac{di(t)}{dt} + nk_e\omega(t) \tag{1}$$

where $u(t)$ is the voltage applied to the motor, $i(t)$ is the stator current, $\omega(t)$ is the angular speed at the output of the motor reducer and $R$, $L$ and $k_e$ are the motor's resistance, inductance and speed constant, respectively. The dynamic equation of the movement of the motor's shaft is given by (2),

$$nk_p i(t) = f\omega(t) + J\dot{\omega}(t) \tag{2}$$

where $J$ is the lumped motor and reducer moment of inertia, $f$ is the friction coefficient and $k_p$ is the motor torque constant. Although the computation of the saturation values will be done discretely, it is not possible to use the discrete motor model for the predictor since the system's electrical time constant ($L/R$, in the range of 50 $\mu$s) is around 20 times less than the algorithm execution period in the implementation $T_{\mathrm{ctrl}} = 1$ ms.

The algorithm computes the minimum and maximum voltages that can be applied to the motor given its present state such that the motor current variation along a prediction horizon $t_{\mathrm{ph}}$ is maintained under a determined threshold. The maximum current variation during a prediction period allowable by the algorithm depends on the motor speed and current in the instant in which the prediction is made. Every time a prediction is made, those control actions which minimize or maximize the current, in the allowable range, are computed. The prediction horizon has been set to $t_{\mathrm{ph}} = 5L/R$, where $L/R$ is the electrical time constant of the motor. This election of $t_{\mathrm{ph}}$ assures that, a time $t_{\mathrm{ph}}$ after a prediction is made, the current will have achieved the 99.33% of its steady-state value. Therefore, given this prediction horizon, the saturation voltages computed by the algorithm are the ones that, if applied for every state of the motor, ensure the steady-state current due to the application of these voltages is the motor's nominal current, in both directions. Note that, far from having to wait for a controller to act over the speed of the motor, our method directly saturates the voltage applied to the motor and the saturation is made effective in the current period, so there is no delay in current limitation. Fig. 1 qualitatively shows the algorithm temporal diagram.

In order to obtain the temporal expression of the current evolution, the Laplace transform has to be applied to (1). The

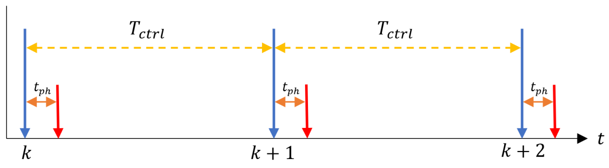


**FIGURE 1.** Temporal diagram of the current limitation algorithm.

superposition principle can be used given that the system has the input $u(t)$ and the virtual input $\omega(t)$. The speed $\omega(t)$ is in fact a state variable and not an input, but, in terms of the system evolution and given that it is measurable, speed will be considered as an input in order to design the predictor.

During the prediction horizon $t_{\text{ph}}$, the voltage $u(t)$ applied to the motor is a constant step of amplitude $U$, but it cannot be assured that the speed $\omega(t)$ will be constant during that period. For this reason, the speed virtual input is modelled as a step signal whose amplitude is the average speed, $\omega_{\text{avg}}$, during the prediction horizon. The estimation $\hat{\omega}_{\text{avg}}[k]$ is made considering a uniformly accelerated motion during the prediction horizon. The acceleration is estimated with the speed measured in the current and previous control periods.

$$\hat{\omega}_{\text{avg}}[k] = \frac{1}{2}\left(2\omega[k] + \frac{\omega[k]-\omega[k-1]}{T_{\text{ctrl}}}t_{\text{ph}}\right) \qquad (3)$$

The computation of $\hat{\omega}_{\text{avg}}[k]$ is an approximation. During the prediction horizon, the motor speed varies according to load applied to the motor. However, the effects produced by load variations over the motor have much slower dynamics than the prediction/control period $T_{\text{ctrl}} = 1$ ms, so considering an average speed in the prediction is reasonable. Additionally, even though a very abrupt change in $\omega(t)$ due to an external disturbance was produced immediately after the prediction in instant $k$ is computed, the algorithm would compensate it in the $k+1$ prediction. A deeper analysis of the effect that the error estimating $\hat{\omega}_{\text{avg}}[k]$ generates over current limitation is provided in Section II-E. Under these conditions, splitting (1) for both inputs according the superposition principle, and solving both differential equations, the currents due to each of the inputs are obtained through expressions (4) and (5).

$$i_u(t) = \frac{U}{R}\left(1 - e^{-\frac{R}{L}t}\right) + C_u e^{-\frac{R}{L}t} \qquad (4)$$

$$i_\omega(t) = -\frac{\omega_{\text{avg}} n k_e}{R}\left(1 - e^{-\frac{R}{L}t}\right) + C_\omega e^{-\frac{R}{L}t} \qquad (5)$$

$C_u$ and $C_\omega$ are constants dependent on the initial conditions. Applying the superposition principle

$$i(t) = i_u(t) + i_\omega(t) = \frac{U - \omega_{\text{avg}} n k_e}{R}\left(1 - e^{-\frac{R}{L}t}\right) + Ce^{-\frac{R}{L}t} \qquad (6)$$

with $C = C_u + C_\omega$. Considering the initial conditions $i(0) = i_0$ and $i'(0) = \left(U - Ri_0 - \omega_{\text{avg}} k_e\right)/L$, this last one obtained from (1), the following expression of the temporal evolution of the motor current is obtained

$$i(t) = \frac{U - \omega_{\text{avg}} n k_e}{R}\left(1 - e^{-\frac{R}{L}t}\right) + i_0 e^{-\frac{R}{L}t} \qquad (7)$$

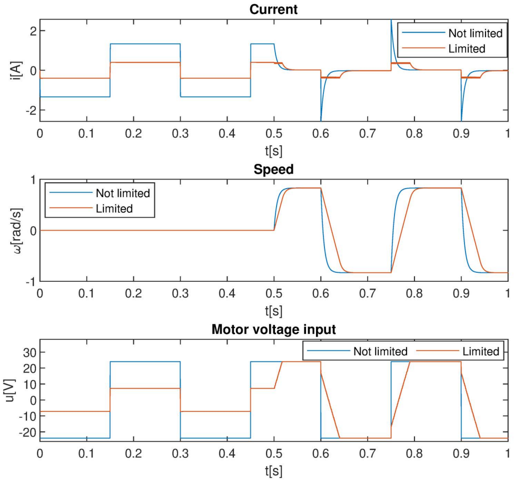


**FIGURE 2.** Simulation of current, speed and input voltage evolution without current limitation and with the model-based limiter. Motor is stalled until $t = 0.5$ s. From then on, the rotor can move freely.

From (7), and using $\hat{\omega}_{\text{avg}}$ to estimate $\omega_{\text{avg}}$, the voltage saturation thresholds are obtained

$$\begin{cases} u_1 = \frac{R\left(i_{\text{sat}} - i_0 e^{-(R/L)t}\right)}{1 - e^{-(R/L)t}} + \hat{\omega}_{\text{avg}} n k_e \\ u_2 = \frac{-R\left(i_{\text{sat}} - i_0 e^{-(R/L)t}\right)}{1 - e^{-(R/L)t}} + \hat{\omega}_{\text{avg}} n k_e \\ u_{\max} = \max(u_1, u_2) \\ u_{\min} = \min(u_1, u_2) \end{cases} \qquad (8)$$

where $i_{\text{sat}}$ is the desired saturation value for the motor current and $i_0$ the current at the moment the predictor is executed.

In order to initially evaluate the performance of the proposed current limitation method, a simulation has been performed. The simulation step has been set to 1 $\mu$s, the period for the execution of the current limitation algorithms $T_{\text{ctrl}}$ has been set to 1 ms and the prediction horizon $t_{\text{ph}}$ has been set to $t_{\text{ph}} = 5L/R = 245$ $\mu$s. A voltage input of ±24 V steps of a frequency of 3.33 Hz and a duty cycle of 50% has been used to aggressively excite the motor and generate overcurrent. The simulation combines two different overcurrent situations that have to be addressed. Initially, the motor is stalled, the rotor cannot move and overcurrent condition is maintained in time. After 0.5 s, the motor can move freely with no load. In this situation, the abrupt voltage variations in the input generate high current peaks.

Fig. 2 shows the current and speed evolution of the motor when the ±24 V steps are applied with and without the use of the proposed current limitation algorithm. In the stall situation (from $t = 0$ s to $t = 0.5$ s) if the current is not limited, it reaches an absolute value of approximately 1.33 A. However, with the proposed algorithm, the absolute value of current under those conditions decreases to the saturation level of ±0.4 A. In regard to the situation when the rotor can move freely (from $t = 0.5$ s to $t = 1$ s), when the current is not limited, current peaks of an amplitude of 2.57 A appear when the input voltage polarity is inverted. This phenomenon is caused by the induced voltage in the motor due to the high angular speed, according to the model in (1).

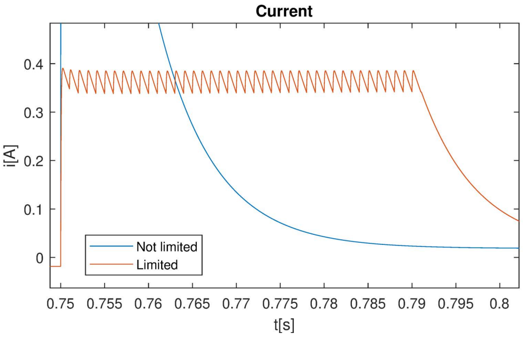


**FIGURE 3. Current ripple produced when applying the model-based predictor algorithm.**

When the model-based limiter is applied, these peaks are suppressed, maintaining the maximum allowed current while the overcurrent conditions are held. This is accomplished by dynamically saturating the voltage applied to the motor. However, the elimination of these peaks, which in most applications are unharmful for the motor due to their short duration, increases the time the motor takes to achieve the maximum speed.

Special attention should be paid to the current ripple produced when limiting the current in the free-rotor situation (Fig. 3). This ripple is caused by motor speed $\omega(t)$ diverging from the estimation $\omega_{\mathrm{avg}}[k]$ given by (3) after the prediction horizon $t_{\mathrm{ph}}$ has elapsed. The speed continues evolving which, given a voltage $u[k]$ that is maintained during the control period $T_{\mathrm{ctrl}}$, makes the current to slightly vary. The maximum current value of 0.4 A is not reached because the supposition of an average speed $\omega_{\mathrm{avg}}[k]$ along $t_{\mathrm{ph}}$ does not exactly correspond with the real evolution of $\omega(t)$ during that prediction horizon. The imprecision generated by these two phenomena is aggravated as the time between predictions increases.

The methods proposed in [20] and [21] have been simulated to compare the performance offered by the proposed predictor method. Fig. 4 presents the same simulation as Fig. 2, but in this case, the current limitation method proposed in [20], introduced in Section I and defined by (9) is used.

$$\begin{aligned}&\text{if}\ \ |i[k-1]| > 0.8 i_{\mathrm{sat}},\\ &\text{then}\ \ u[k] = a \cdot u[k-1]\end{aligned} \tag{9}$$

In the original research in [20], the parameter $a$ was set to $a = 0.98$. However, with the control period set to $T_{\mathrm{ctrl}} = 1$ ms, the algorithm does not limit the current at all. Via simulation, it was proven that $a = 0.85$ offers a much better performance, so that value has been used in the following simulation. This fact demonstrates the dependence that parameter $a$ has on the execution frequency of the limitation algorithm. In the stall situation, as soon as the current is under the limitation threshold, the applied voltage is the one determined by the predefined input, ±24 V in this case, what provokes a current peak to be produced again, generating ripple. In order to avoid this phenomenon, a security time after current limitation could be added (i.e., keep saturating the voltage for a certain time although the current is under the threshold). However, this would not be a good solution because, in case that the situation that has generated the current raising is still standing, once the security time is out, the current would surpass again the threshold; furthermore, in case that the cause of the overcurrent has vanished, the motor would have been working for a certain period of time below the maximum safe performance region.

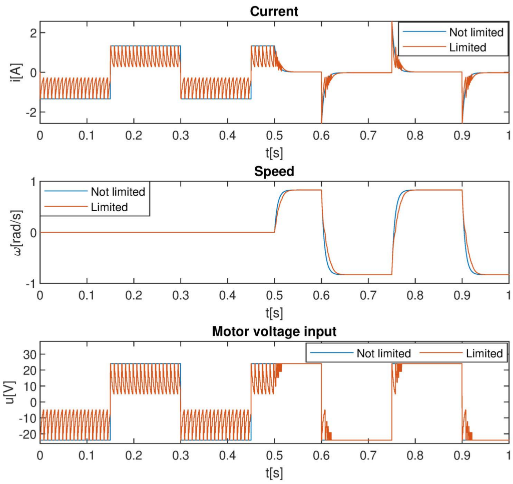


**FIGURE 4. Simulation of current, speed and input voltage evolution without current limitation and with the limiter defined by (9). Motor is stalled until $t = 0.5$ s. From then on, the rotor can move freely.**

The factor $a$ determines the rate at which the current is limited, so the performance of this algorithm highly depends on that factor. When the motor can move freely, this algorithm does not produce an effective current limitation, since high current peaks are still achieved. In overload conditions, the current is non-smoothly limited, reducing the average current value to 0.700 A, but widely surpassing the threshold of 0.4 A. Hence, this method can be useful, to some extent and not sufficiently accurately for many applications, to reduce overcurrent in stall conditions, but the model-based predictor limits current in stall and free-rotor situations much more effectively and precisely.

The current limitation behavior of the algorithm proposed in [21] is presented in Fig. 5. The algorithm is described by (10).

$$\begin{aligned}&\text{if}\ \ u[k] - u[k-1] > \dot{V}_{\mathrm{max}}\\ &\text{then}\ \ u[k] = u[k-1] + \dot{V}_{\mathrm{max}}\\ &\text{else}\ \ \text{if}\ u[k-1] - u[k] > \dot{V}_{\mathrm{max}}\\ &\text{then}\ \ u[k] = u[k-1] - \dot{V}_{\mathrm{max}}\end{aligned} \tag{10}$$

Two different values of the maximum allowable variation in motor voltage have been selected: $\dot{V}_{\mathrm{max}} = 1$ V and $\dot{V}_{\mathrm{max}} = 0.5$V. This algorithm is isolated from the motor state, as the only data it evaluates are the applied voltages in the current cycle and in the previous one. With this method, the saturation of the voltage variation is produced independently of the interaction between the motor and the environment. This method can be useful for motor startup

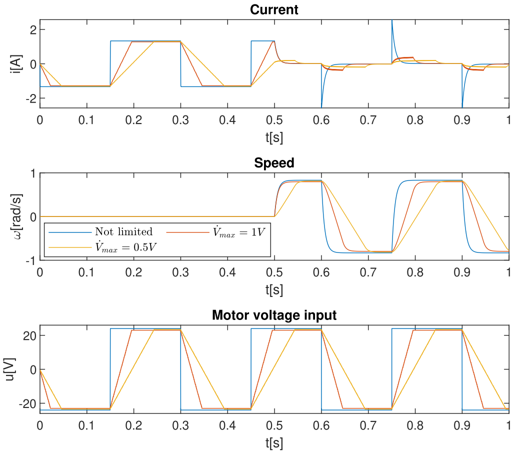


**FIGURE 5. Simulation of current, speed and input voltage evolution without current limitation and with the limiter defined by (10). Motor is stalled until $t = 0.5$ s. From then on, the rotor can move freely.**

(Fig. 5) since, in that situation the overcurrent is caused by a fast voltage variation (which is exactly what this algorithm limits). However, when an overload is applied, this method does not provide effective current limitation since, in order to avoid overcurrent under these conditions, voltage should be drastically reduced in absolute value. Additionally, the value of the maximum voltage variation $\dot{V}_{\max}$ is arbitrary and may vary among different motors and the maximum current allowable in startup.

The qualitative behavior observed in the simulations with the three methods (including the two configurations used for the one in [21]) is quantitatively presented in Table 2 and Table 3. On the one hand, data in Table 2 present the total time during the simulation that the current has surpassed $i_{\text{sat}}$, and, for that time, the average percentage of power and absolute value of current in relation to what the perfect saturation would be: $P_s = i_{\text{sat}}^2 R$ and $|i_s| = i_{\text{sat}}$. As data show, the proposed method exceeds the saturation current between 9 and 8 times less time than in other methods. Additionally, even when the threshold is exceeded, power and current surpass very slightly the limits, always below 1%, while the other methods surpass by several times the maximum thresholds. On the other hand, Table 3 presents the time during which the current is saturated and the average percentage of power and current for that time in relation with $P_s$ and $|i_s|$, for the four simulations. These data allow to evaluate how close to the ideal saturation situation each of the algorithms can perform. The proposed method, when current is limited offers an average 98.06% of $i_{\text{sat}}$, what delivers an average 96.34% of $P_s$.

Observing the metrics separated in stall and free rotor conditions in Table 2 and Table 3, it is remarkable that the method in [21] can eliminate the current peaks due to abrupt voltage variations. However, the current during saturation is very low, around 30% of $i_{\text{sat}}$, so the motor is safe, but it is very far from delivering the full power that it can do in the safe region. In the stall condition, methods [20] and [21] offer a very poor performance.

**TABLE 2. Performance metrics: current over $i_{\text{sat}}$.**

| | Total time (0 s to 1 s) | | |
|---|---|---|---|
| **Method** | **Sat. time [s]** | **Power%** | **Current%** |
| **New** | 0.054 | 100.61 | 100.30 |
| **[20]** | 0.450 | 489.69 | 204.97 |
| **[21] (0.5 V)** | 0.410 | 795.65 | 272.70 |
| **[21] (1 V)** | 0.456 | 882.71 | 291.55 |
| | **Stall (0 s to 0.5 s)** | | |
| **Method** | **Sat. time [s]** | **Power%** | **Current%** |
| **New** | 0.054 | 100.61 | 100.30 |
| **[20]** | 0.401 | 459.64 | 201.60 |
| **[21] (0.5 V)** | 0.410 | 795.65 | 272.70 |
| **[21] (1 V)** | 0.448 | 890.99 | 293.29 |
| | **Free (0.5 s to 1 s)** | | |
| **Method** | **Sat. time [s]** | **Power%** | **Current%** |
| **New** | 0.000 | 100.00 | 100.00 |
| **[20]** | 0.049 | 735.04 | 232.42 |
| **[21] (0.5 V)** | 0.000 | 100.00 | 100.00 |
| **[21] (1 V)** | 0.008 | 398.35 | 189.28 |

**TABLE 3. Performance metrics: current is saturated.**

| | Total time (0 s to 1 s) | | |
|---|---|---|---|
| **Method** | **Sat. time [s]** | **Power%** | **Current%** |
| **New** | 0.641 | 96.34 | 98.06 |
| **[20]** | 0.504 | 317.67 | 161.73 |
| **[21] (0.5 V)** | 1.000 | 335.42 | 131.53 |
| **[21] (1 V)** | 1.000 | 414.18 | 149.94 |
| | **Stall (0 s to 0.5 s)** | | |
| **Method** | **Sat. time [s]** | **Power%** | **Current%** |
| **New** | 0.500 | 100 | 99.99 |
| **[20]** | 0.450 | 309.68 | 162.66 |
| **[21] (0.5 V)** | 0.500 | 658.4 | 232.47 |
| **[21] (1 V)** | 0.500 | 802.40 | 268.30 |
| | **Free (0.5 s to 1 s)** | | |
| **Method** | **Sat. time [s]** | **Power%** | **Current%** |
| **New** | 0.141 | 83.37 | 91.22 |
| **[20]** | 0.054 | 384.25 | 153.98 |
| **[21] (0.5 V)** | 0.500 | 12.43 | 30.59 |
| **[21] (1 V)** | 0.500 | 25.96 | 31.58 |

From this point on, a deeper analysis of the proposed method is addressed, leading to the experimental validation. As it has proven to outperform the methods presented in [20] and [21] it is unnecessary to continue comparing the proposed algorithm with the two aforementioned ones throughout the rest of the paper.

### B. RESPONSE ACCELERATION

The algorithm presented in the previous subsection limits the current to the configurable threshold $i_{\text{sat}}$ in all situations. However, it is desirable to allow high current peaks for a short period of time, which can be produced in motor start-up or when changing rotation direction. If current peaks are allowed, the motor will be protected against stall or similar conditions, but will be able to deliver full torque during short periods of time. DC motors normally have a startup current around 6 to 9 times greater than the nominal one. However, these peaks are not harmful. The problem caused by the overcurrent is that it produces a temperature increase that can damage the motor. A startup peak of current in the order of milliseconds is not harmful since it does not generate

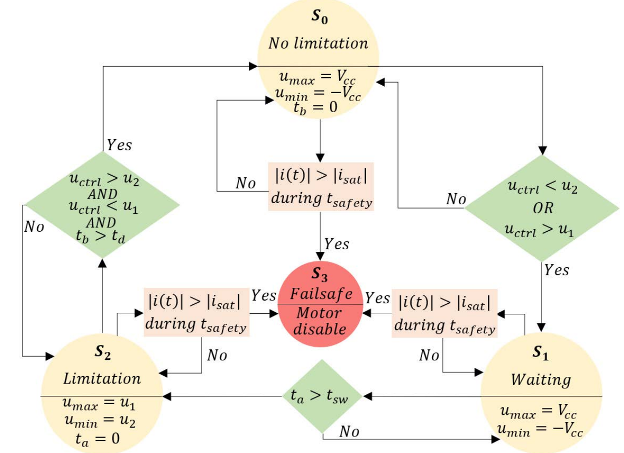


**FIGURE 6.** State machine that allows current peaks of a configurable duration $t_{sw}$ to be produced.

a noticeable increase in the motor temperature, given that standard motor thermal constants are in the order of at least 10 seconds. However, the startup current peaks can cause overheating problems if the motor is started and stopped (or speed reversed) at a frequency such that those peaks generate a neat increase in motor temperature, so these situations should also be avoided.

In order to accelerate the response allowing short-duration current peaks, a simple state machine that controls when the current must be limited has been designed (Fig. 6). The proposed current limiting algorithm in (8) is executed in every control period. However, in state $S_0$, the current is not limited, establishing the voltage saturation value at the DC bus tension $\pm V_{cc}$. If the voltage to be applied to the motor (computed by the application's corresponding controller) is beyond the limits $u_{max}$ and $u_{min}$ computed by the model-based predictor algorithm, the state machine switches to $S_1$, a state in which the current is allowed to evolve freely during the predefined time interval $t_{sw}$. After the period $t_{sw}$ has elapsed, if the applied voltage is beyond the limits, the state switches to $S_2$, a state in which the voltage applied to the motor is saturated according to the values $u_{min}$ and $u_{max}$ computed by the predictor. When the voltage to be applied returns to the allowable values provided by the predictor, and a period $t_d$ has elapsed (in order to avoid several consecutive current peaks), the state switches back to $S_0$. Additionally, a safety system has been added so that, if the current surpasses the saturation values during a period $t_{safety}$, the motor outputs are disabled.

Fig. 7 presents the simulation results of the comparison between no current limitation protocol application and the use of the model-based predictor algorithm with the state-machine inclusion. As was shown in the simulations presented in Section II-A, the first 0.5 s correspond to a rotor in stall situation, and from $t = 0.5$ s to $t = 1$ s, the rotor can move freely. Input voltage has been generated again as $\pm 24$ V pulses. The temporal parameters have been set to $T_{ctrl} = 1$ ms and $t_{sw} = 4$ ms. The worst-case duration of the allowed current peaks is $(t_{sw} + 1) = 5$ ms. This situation arises when an intense perturbation is applied to the motor just after the current limitation algorithm is executed. As seen in Fig. 7, once the waiting time $t_{sw}$ has elapsed, the applied voltage corresponds to the saturation values computed by the algorithm and, consequently, the motor current is limited to the threshold. In regard to the motor speed, a comparison between allowing current peaks in current limitation or not is shown in Fig. 8. It is clear that allowing short-duration current peaks is beneficial for reducing the response time of the motor speed, while protecting the motor.

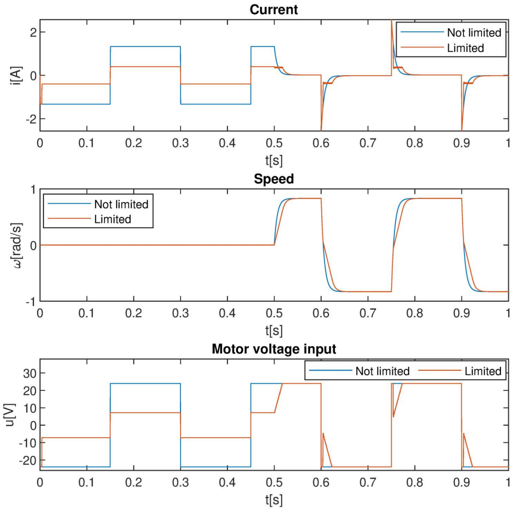


**FIGURE 7.** Simulation of current, speed and input voltage evolution without current limitation and with the model-based limiter along with the state machine that allows short-duration current peaks. Motor is stalled until $t = 0.5s$. From then on, the rotor can move freely.

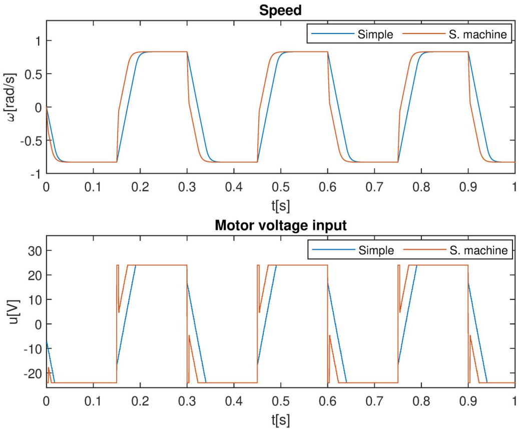


**FIGURE 8.** Simulation of speed evolution with free rotor movement applying the limitation algorithm in its simple version and when allowing current peaks according to the state machine in Fig. 6.

### C. IMPLEMENTATION OVER H-BRIDGE

In the previous analysis, it was supposed that a power supply capable of generating a continuous voltage between $\pm 24$ V was available. However, in the real implementation this is not feasible and, therefore, it is necessary resorting to the use of an H-bridge to drive the DC motor. A simulation to evaluate

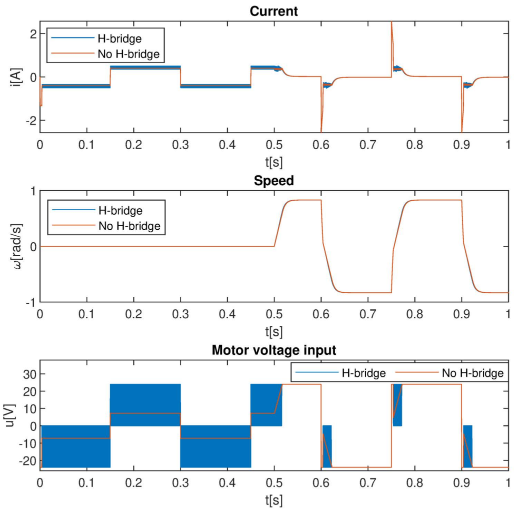


**FIGURE 9. Simulation of current, speed and input voltage evolution with and without an H-bridge. Model-based predictor current limitation algorithm allowing short-duration current peaks is used. Motor is stalled until $t = 0.5s$. From then on, the rotor can move freely.**

the performance of the algorithm with the inclusion of an H-bridge has been conducted. The switching frequency has been set to $f_{\mathrm{PWM}} = 40$ kHz, which is the frequency at which the PWM will be generated in the physical implementation. Fig. 9 shows the evolution of current and speed, along with the voltage applied to the motor, with and without the inclusion of the H-bridge. In both cases, the state machine that allows short-duration current peaks has been incorporated.

When using an H-bridge, the voltage applied to the motor can only take the values $-V_{\mathrm{cc}}$, 0 V and $+V_{\mathrm{cc}}$. Whether or not the H-bridge is used, the evolution of the current presents a ripple around the saturation threshold $i_{\mathrm{sat}} = 0.4$ A. With the H-bridge, this ripple has an amplitude of 0.07 A with the rotor in stall and 0.11 A when he rotor moves freely. The ripple introduced by the inclusion of the H-bridge is added to the one generated by the variation of $\omega(t)$ during the control period, as explained in Section II-A. The average value of the current in the interval form 0.15 s to 0.3 s is 0.426 A. As can be seen, the inclusion of the H-bridge generates a slight overshoot of the current threshold. In some applications this small overshoot might be critical. However, this phenomenon can be easily compensated by marginally decreasing $i_{\mathrm{sat}}$ in the algorithm.

### *D. INFLUENCE OF ERROR IN MOTOR PARAMETERS ESTIMATION*

The proposed algorithm is model-based. In the previous analyses, it was considered that exact motor model parameters where known. However, in practice, there will be errors in parameters' values estimations.

The suitability of the saturation voltages computed by the current limitation algorithm depends on the error between the estimated motor parameters and the real ones. For this analysis, the relevant parameters are the resistance, the inductance and the speed constant, and their real values and tolerances are as follows: $R = R_n \pm 5\%$, $L = L_n \pm 30\%$ and $k_e = k_{\mathrm{en}} \pm 5\%$, with $R_n = 18\ \Omega$, $L_n = 0.881$ mH and $k_{\mathrm{en}} = 0.0359$ V/(rad/s), according to Table 1. As $R$ and $k_e$ have been measured, it is reasonable to consider their tolerances around 5%, in a conservative manner. The tolerance of the inductance has been considered of 30% as it is also a conservative value of coils' typical tolerance. To obtain the maximum error between the motor current saturation threshold $i_{\mathrm{sat}}$ and the current generated by the application of the saturation voltages obtained by the current limitation algorithm (8), the error function

$$e(t_{\mathrm{ph}}) = \left|i(t_{\mathrm{ph}}) - i_n(t_{\mathrm{ph}})\right| \tag{11}$$

has to be maximized, being $i_n(t_{\mathrm{ph}}) = i_{\mathrm{sat}}$ the current in the prediction horizon if the estimations of motor parameters coincided with real ones. As aforementioned, the prediction horizon is set to five times the electrical time constant of the estimated motor model, $t_{\mathrm{ph}} = 5L_n/R_n$. Taking the expression of $u_1$ in (8) and substituting it in (11), (12) is obtained. As here we want to study the influence of model parameter estimation error, it has been considered for this analysis that $\hat{\omega}_{\mathrm{avg}} = \omega_{\mathrm{avg}}$, leaving the study of speed estimation error for Section II-E.

$$\begin{aligned} e\left(\frac{5L_n}{R_n}\right) &= \left|\frac{R_n\left(1-e^{-\frac{5RL_n}{LR_n}}\right)}{R\left(1-e^{-5}\right)}\left(i_{\mathrm{sat}} - i_0e^{-5}\right)\right. \\ &\quad \left. + \frac{\omega_{\mathrm{avg}}}{R}n\left(k_{\mathrm{en}} - k_e\right)\left(1-e^{-\frac{5RL_n}{LR_n}}\right) + i_0\left(e^{-\frac{5RL_n}{LR_n}}\right) - i_{\mathrm{sat}}\right| \end{aligned} \tag{12}$$

Given that $i_{\mathrm{sat}} \gg i_0e^{-5}$ and $i_0e^{-\frac{5RL_n}{LR_n}} \approx 0$, (12) can simplify to (13) as follows

$$\begin{aligned} e\left(\frac{5L_n}{R_n}\right) = &\left|\frac{R_n\left(1-e^{-\frac{5RL_n}{LR_n}}\right)}{R\left(1-e^{-5}\right)}i_{\mathrm{sat}}\right. \\ &\quad \left. + \frac{\omega_{\mathrm{avg}}}{R}n\left(k_{\mathrm{en}} - k_e\right)\left(1-e^{-\frac{5RL_n}{LR_n}}\right) - i_{\mathrm{sat}}\right| \end{aligned} \tag{13}$$

In order to maximize the error, it is clear that $L = L|_{\mathrm{min}}$, $k_e = k_e|_{\mathrm{min}}$ and $\omega_{\mathrm{avg}} = \omega_{\mathrm{max}}$. The value of $\omega_{\mathrm{max}}$ is the maximum speed that could be achieved such that the motor current is $i_{\mathrm{sat}}$. Substituting the saturation current in (1) and reordering terms, (14) is obtained.

$$\omega_{\mathrm{max}} = \frac{u_{\mathrm{max}} - Ri_{\mathrm{sat}} - L\frac{di(t)}{dt}}{nk_e} \tag{14}$$

In order to accelerate the motor and achieve maximum speed, positive current must be applied. In that situation, the derivative of the current is positive. Once the maximum speed with maximum current is achieved, it is not possible that the derivative of the current is negative, since that would imply that the motor was previously in a situation in which the current was above $i_{\text{sat}}$. Hence, the maximum speed in (14) is achieved when $\frac{di(t)}{dt} = 0$. In other words, if $\omega > \frac{u_{\max} - Ri_{\text{sat}}}{nk_e}$, then, the actual motor current is under $i_{\text{sat}}$, which is a situation in which the current does not have to be limited and, therefore, it is not a case of study of this analysis. Consequently, the maximum speed achievable with a maximum current is given by (15).

$$\omega_{\max} = \frac{u_{\max} - Ri_{\text{sat}}}{nk_e} \tag{15}$$

Unlike the other parameters, the value of $R$ that contributes to error maximization cannot be obtained via first-glance exploration of expression (13). Hence, the error function in (13) has been plotted for the whole range of $R = R_n \pm 5\%$ (with steps of 0.0036 Ω) to find the maximum error value. From this analysis, the maximum error for current limitation is $e(t_{\text{ph}})_{\max} = 72$ mA, and it is achieved at maximum speed, as explained before. In this situation, the 66.67% of the error is due to the term related to the motor speed $\omega$ and error in estimating $k_e$. However, the scenarios in which current limitation is more frequently necessary, involve high loads coupled to the motor, which imply low speeds, so under those circumstances, the error will be drastically lower (33.33% if the motor is in stall, which leads to an error of 24 mA). The exact same result is obtained if negative current and speed are considered for the analysis from (11) to (15) and the saturation value $u_2$ of (8) is used in the analysis.

Fig. 10 represents the same simulation as in previous subsections in order to illustrate the behavior of the limitation algorithm in the presence of parameter uncertainty. In the first simulation, motor parameters have been set such that the parameter estimations are the worst-case-error ones. In contrast, in the second simulation, the motor parameters have been set to the nominal ones, so no estimation error influences the performance of the algorithm. The H-bridge has been included in both simulations. As shown in Fig. 10, although the maximum theoretical error obtainable is 72 mA, the error in normal motor operation due to uncertainties in model parameters is under 30 mA. Note that the difference in the maximum achieved speeds for both cases is due to the 5% variation of $k_e$.

### E. INFLUENCE OF ERROR IN SPEED ESTIMATION

First, an analytical expression of the current error generated by an error in the estimation $\hat{\omega}_{\text{avg}}[k]$ is derived. Then, an application-specific bounds for the speed estimation error is provided.

The voltage saturation values are given by the first and second equation in (8). Substituting each of those equations in (7), the currents generated by the application of the saturation voltages $u_1$ and $u_2$ are obtained.

$$i(u_1) = i_{\text{sat}} + \frac{nk_e\left(1 - e^{-\frac{R}{L}t}\right)}{R}\left(\hat{\omega}_{\text{avg}} - \omega_{\text{avg}}\right) \tag{16}$$

$$i(u_2) = -i_{\text{sat}} + \frac{nk_e\left(1 - e^{-\frac{R}{L}t}\right)}{R}\left(\hat{\omega}_{\text{avg}} - \omega_{\text{avg}}\right) \tag{17}$$

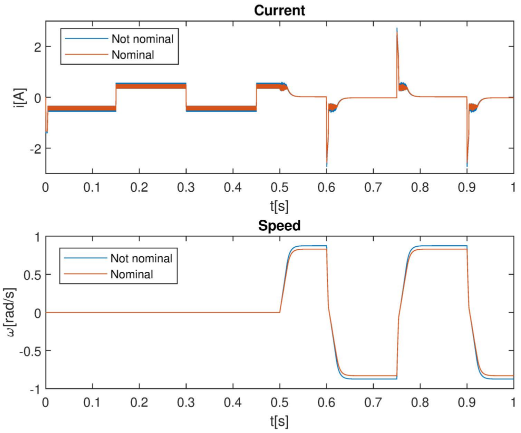


**FIGURE 10. Evolution of current and speed when real motor parameters coincide with nominal ones and when parameter estimations are worst-case ones.**

The voltages $u_1$ and $u_2$ were computed such that the current in the prediction horizon $t_{\text{ph}}$ reaches $\pm i_{\text{sat}}$. The current errors generated by the error in speed estimation when saturation voltages are applied ($\tilde{i}^{u_1}, \tilde{i}^{u_2}$) are given by

$$\tilde{i}^{u_1}\left(t_{\text{ph}}\right) = i(u_1) - i_{\text{sat}} \approx \frac{nk_e}{R}\left(\hat{\omega}_{\text{avg}} - \omega_{\text{avg}}\right) \tag{18}$$

$$\tilde{i}^{u_2}\left(t_{\text{ph}}\right) = i(u_2) + i_{\text{sat}} \approx \frac{nk_e}{R}\left(\hat{\omega}_{\text{avg}} - \omega_{\text{avg}}\right) \tag{19}$$

The obtained errors are equal. According to equations (18) and (19), the lower the term $\frac{nke}{R}$ is, the lower the current error due to the error in speed estimation will be. However, the bigger the reduction ratio $n$ is, the more inertia the motor will have and the lower the speed estimation error will be, since motor speed will vary more slowly. For the motor used in this study, $\frac{nke}{R} = 1.584\ \frac{\text{V·s}}{\Omega\text{·rad}}$.

The upper bounds for the speed estimation error depends on both the specific motor and the application. The error will depend on how far from the hypothesis of a uniformly accelerated motion during the prediction horizon the real evolution of the motor speed is. On the one hand, the higher the maximum motor torque is, the larger perturbation is needed to generate a change on its speed, so the smaller the current limitation error due to an imprecise estimation will be. On the other hand, the lower the frequency of the perturbation is, the slower the speed variations will be and the more similar to (3) the estimation of the speed will be. In this regard, the application of a step perturbation just after the prediction is made is the worst-case situation for speed estimation, since a whole control period has to elapse before the unexpected modification of the speed is considered by the algorithm.

In regard to upper-limb speed requirements, according to [24], the elbow flexion and extension movement bandwidth for ADLs (activities of daily living) is in the range of 0.05 Hz, and 0.5 Hz. In the experiments performed in that work, the elbow speed in this kind of tasks achieved a maximum speed of 150°/s, which is a 120% of the elbow speed in ADLs [25]. As for the shoulder, less strict speed requirements are

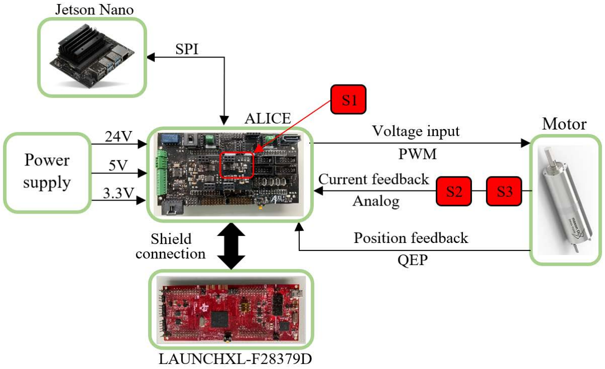


**FIGURE 11. System electronics architecture for the validation of the current limitation algorithm.**

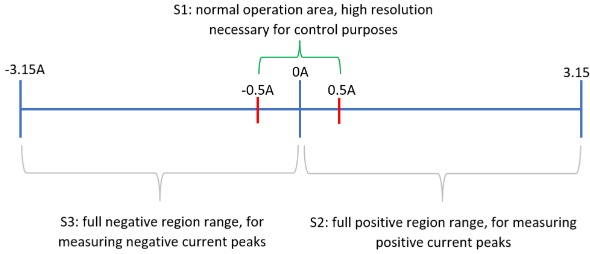

S1: normal operation area, high resolution necessary for control purposes
-3.15A
0A
3.15A
-0.5A
0.5A
S3: full negative region range, for measuring negative current peaks
S2: full positive region range, for measuring positive current peaks

**FIGURE 12. Current ranges measured by each of the three current sensors. Sensor S1 is specially designed to measure current in the safe operation range. Sensors S2 and S3 allow measuring current peaks.**

involved but, as worst-case, similar bandwidth and articular speed can be considered for the ADLs as both articulations are in the same kinematic chain. In rehabilitation therapies, the upper-limb speed and bandwidth are considerably below the maximum ADL ones, and the speeds may increase according to the evolution of the subject.

To evaluate the effects of external disturbances in speed estimation error, a simulation has been conducted. Dynamic and kinematic requirements of upper-limb rehabilitation are not too demanding in comparison to other applications so, with the objective to prove that the algorithm can be generalized to applications with more demanding requirements, the perturbation has been modelled as a 100 Hz square wave of an amplitude of 200 N. It is important to note that the disturbances were applied just after the algorithm predictions were made which, as aforementioned, is the worst-case moment of application. The simulation parameters are the same as in previous simulations, with the exception that the rotor is not stalled the first 0.5 s, as we need the motor to move. The average speed estimation error for all the prediction horizons is $\tilde{\bar{\omega}}_{\text{avg}} = 0.006$ rad/s and the maximum one is $\tilde{\omega}_{\text{avg}}\big|_{\max} = 0.013$ rad/s. Evaluating equations (18) or (19) with those values, the current limitation errors due to speed estimation ones are $\tilde{\bar{i}}(t_{\text{ph}}) = 9.5$ mA and $\tilde{i}(t_{\text{ph}}) = 21$ mA, which represent the 2.38% and the 5.25% of $i_{\text{sat}}$.

## III. EXPERIMENTAL VALIDATION

The model-based predictor current limitation algorithm is implemented over a F28379D Texas Instruments microcontroller. The microcontroller communicates via SPI with a Jetson Nano, which writes all the received data into a file for offline processing. A 24 V power supply is used to power the MAX14870 motor driver, and 5 V and 3.3 V power supplies are used to power the current sensors and the microcontroller, respectively. A diagram of the electronics set-up is shown in Fig 11.

Three current sensors have been used to implement the current limitation algorithm. These three sensors monitor different current ranges, as shown in Fig. 12. The three sensors are based in the Allegro ACS723 current sensor IC, along with the appropriate signal conditioning circuitry (amplification stage). Sensors S2 and S3 are Sparkfun SEN-14544 sensors, and their potentiometers have experimentally been adjusted to cover the specified current ranges shown in Fig.12. Sensor S1 is based on the SEN-14544, but the signal conditioning circuit components have been specifically selected to match the 0 V to 3 V output of the sensor to approximately the safe motor current operation range (from −0.5 A to 0.5 A), eliminating the need of potentiometer adjustments. In this way, the measurement obtained by the ADC is of maximum resolution in relation to the current range that is needed to be measured. However, in order to allow short-duration current peaks, the motor current will go beyond the aforementioned safe operation region. To measure those currents, sensors S2 and S3 are used, and they respectively measure the full positive and negative current ranges, as described in Fig. 12. When the motor current is in the normal operation region, measurements of S1 are used. If the current surpasses those limits, the measurements of either S2 or S3 are considered.

The three sensors output voltages that are contained in the range from 0 V to 5 V. However, the microcontroller ADCs work in the range from 0 V to 3 V. With the objective of protecting the microcontroller's ADC modules, very simple clipping circuits have been added between the sensors' outputs and the ADC inputs. These circuits consist of an operational amplifier used as a comparator which controls a relay that switches between the sensor output and the 3V maximum admissible voltage.

The three current sensors have been calibrated using a current source. For S1, currents between -0.5 A and 0.5 A with steps of 0.1 A were applied. One thousand samples were collected for each current, the average of the samples for each current was computed and a regression curve was obtained in Matlab. The procedures for sensors S2 and S3 were the same, but the applied currents were 0 A to 3 A with steps of 0.3 A and -3 A to 0 A with steps of 0.3 A, respectively. The calibration curves for all the sensors resulted in first order polynomials with the following $R^2$ factors: $R^2_{S1} = 0.99995$, $R^2_{S2} = 0.99998$ and $R^2_{S3} = 0.99999$.

The exact same values of all the parameters in simulation have been used in the real implementation tests.

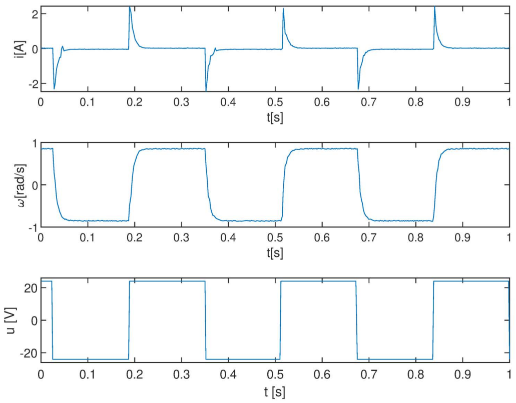


**FIGURE 13.** Current, speed and applied voltage without current limitation, free movement.

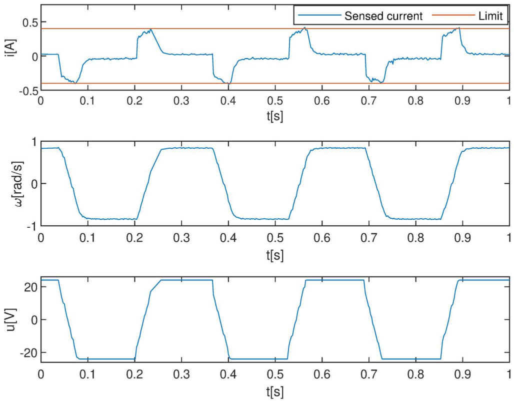


**FIGURE 14.** Current, speed and applied voltage with full-time current limitation, free movement.

### A. STAND-ALONE MOTOR VALIDATION TESTS

The current limitation algorithm has been implemented over a stand-alone Maxon DCX22S GB KL 48V with a 794 to 1 reducer. The behavior of the motor in stall and in free rotor movement have been evaluated in separate experiments due to physical set-up requirements.

Experiments with motor in free movement are presented first. Each experiment took the duration of 1s and the motor was powered with a 3.33 Hz ±24 V square steps. Motor behavior with no current limitation is shown in Fig. 13. As it can be seen, the ±24 V steps are directly applied to the motor, producing high current peaks. The maximum values of the current peaks are not all equal among them, and those are not exactly the same as the ones obtained in simulation (2.57 A). This is due to the fact that, although the algorithm is executed at 1 kHz, data are transmitted to the Jetson Nano at 400 Hz, while the current surpasses the absolute value of 2.32 A (minimum peak value) only for around 0.8 ms. Therefore, the maximum real current peaks of 2.57 A are difficult to detect and transmit at that rate.

Fig. 14 presents the results of applying the limitation algorithm without allowing current peaks. Current peaks are suppressed and the current is maintained around the ±0.4 A threshold. When short-duration current peaks are allowed, the motor behavior is the one presented in Fig. 15. In contrast to the current peaks in Fig. 13, in this case the durations are shorter, of around 4 ms, as this is the time interval during which current limitation is not performed. A decrease in the acceleration is produced when the voltage is saturated.

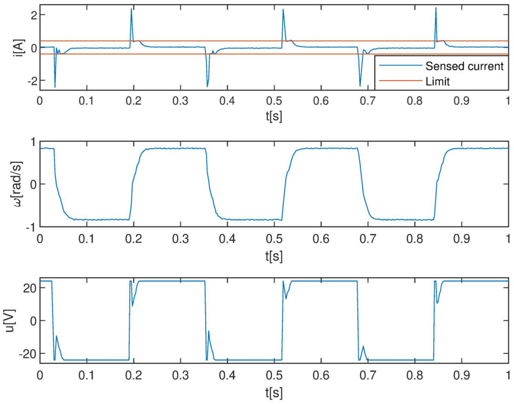


**FIGURE 15.** Current, speed and applied voltage with current limitation and allowable current peaks, free movement.

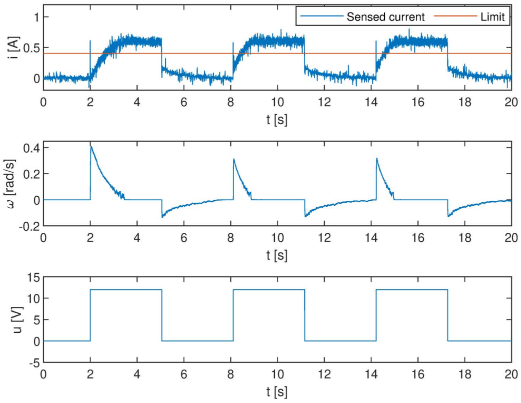


**FIGURE 16.** Current, speed and applied voltage without current limitation, stall condition.

In regard to the experiments with the motor in stall condition, the high torque delivered by the motor when powering it with 24 V makes it hard to maintain its shaft still. For this reason, these tests have been performed powering the motor with ±12 V steps, which reduces the motor torque to the half. These experimental tests additionally differ from the stall condition simulation in that the physical set-up of the motor allows a degree of slack in the motor shaft movement. The motor shaft has been clamped such that the motor can rotate a few degrees before it completely stops. In order to overcome that issue, for these experiments, the experimental time has been set to 20 s and the voltage excitation to pulses of 12 V and 0 V of 3 seconds of duration, with an initial 0 V excitation during the 2 first seconds of each experiment.

Fig. 16 represents a trial of the stall condition experiment in which the current has not been limited. When the 12 V pulse is applied, a current peak is produced, in the same way

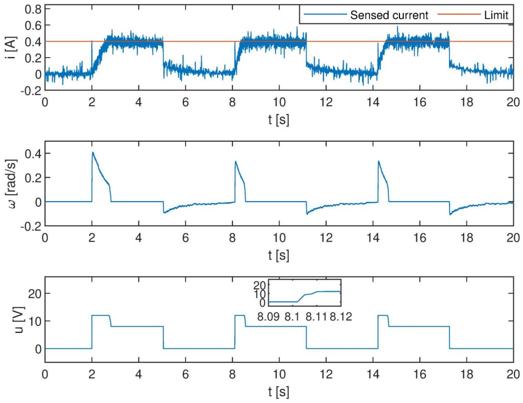


**FIGURE 17.** Current, speed and applied voltage with full-time current limitation, stall condition.

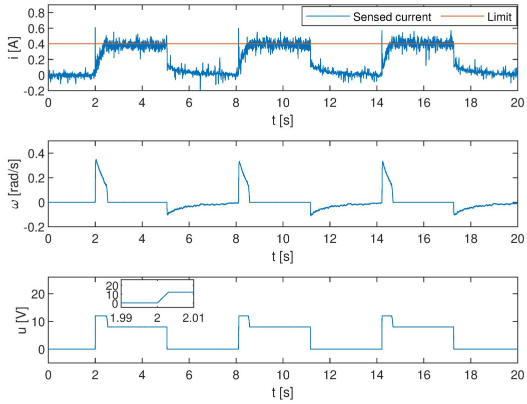


**FIGURE 18.** Current, speed and applied voltage with current limitation and allowable current peaks, stall condition.

as in the free movement experiment. However, note that the amplitude of this peak is lower than in simulation because the applied voltage is the half. The motor current settles around 0.6 A when the motor is stalled, which corresponds to what one would expect to obtain combining the dynamic equation in (1) with the model parameters in Table 1. The results of applying the current limitation algorithm without allowing current pulses are presented in Fig. 17. In this case, both the current peaks produced when the motor starts to move and the current when the motor is stalled are around the 0.4 A predefined limit. Finally, the behavior of applying the current limitation algorithm allowing short-duration current peaks is shown in Fig. 18. Here, current is not limited when the motor starts, during the short duration of the peak, but it is limited when the motor is in stall. The global behavior is very similar between both experiments, but the detailed voltage plots in Fig. 17 and Fig. 18 show how voltage is limited in Fig. 17 in order to avoid current peaks, whereas in Fig. 18 the 12 V steps are applied directly, as current peaks are allowed.

Table 4 compares the percentage of power and absolute value of current in relation to saturation conditions ($P_s = i_{\text{sat}}^2 R$, $|i_s| = i_{\text{sat}}$) obtained in these experimental tests with the values obtained in the previous simulations in Section II-A (Table 3). The values in the table only compute the instants in which the current is limited by the algorithm. As observed in Table 4, the current is over-limited in each of the experimental tests compared with each analogous experiment in simulation. However, it is remarkable that the differences in current between real and simulated tests are always under 4%, in both versions of the algorithm. The current over-limitation is more evident when limiting current peaks (87.38% and 88.09% in both full-time and peak-allowing methods, respectively). In the overload condition tests, $|i_s|$ is less than a 4% below $i_{\text{sat}}$ for the experimental tests. In regard to the average power, the behavior is obviously similar to the $|i_s|$ one, with a maximum difference between simulated and experimental tests of 6.19%.

**TABLE 4.** Performance metrics: real implementations vs. simulation, when current is saturated.

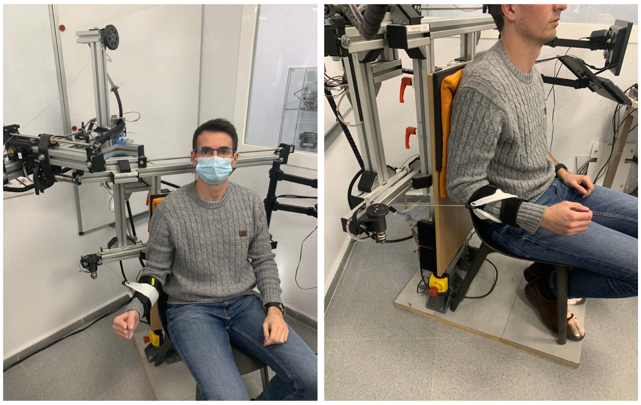
| | Limitation | Rotor | Power% | Current% |
|---|---|---|---|---|
| EXP | Full | Free | 77.18 | 87.38 |
| | | Stall | 94.51 | 96.55 |
| | Peaks allowed | Free | 82.87 | 88.09 |
| | | Stall | 97.47 | 98.04 |
| SIMU | Full | Free | 83.37 | 91.22 |
| | | Stall | 100 | 99.99 |
| | Peaks allowed | Free | 84.36 | 91.38 |
| | | Stall | 100.06 | 100.04 |

## B. EXOSKELETON VALIDATION TESTS

The algorithm has been implemented over the cable-driven rehabilitation upper-limb exoskeleton ExoFlex [5]. The objective of this experiment is to validate the current limitation method developed in this work in a real application, in which the load over the motor can vary arbitrarily since there exists human-robot interaction. Four healthy subjects have participated in the experiments. Currently, ExoFlex (Fig. 19) is oriented to passive rehabilitation, where a position control loop is implemented and the exoskeleton assists the user in performing predefined motion trajectories. A supertwisting SMC has been used in order to robustly control the movement of the limb. The actuators that generate articular motion in this exoskeleton are the Maxon DCX22S GB KL 48V, the motor that has been used along this work. Aluminum pulleys of 5 cm of diameter have been coupled to the motor shafts. Nylon-covered steel cables are attached to the corresponding actuator pulley on one side, and to the corresponding user articulation on the other one (Fig. 19). The movement of the actuator produces the winding or unwinding of the transmission cable over the pulley in order to generate articular motion.

All experimental procedures were carried out in accordance with the Declaration of Helsinki on research involving human subjects and approved by the ethical committee of the Universidad Politécnica de Madrid. The subjects were informed about the experiment and gave their consent to participate in it.

A shoulder rotation exercise has been chosen for these experiments. In the start position, the user has their shoulder at 0 degrees (hand aiming at the front), while the final position is set to around 80 degrees. Four concatenations



(a) (b)

**FIGURE 19.** Subject performing a shoulder rotation exercise with ExoFlex.

**TABLE 5.** Performance metrics under current limitation: rehabilitation exoskeleton tests.

| Max. speed [rad/s] | Power% | Current% |
|---|---|---|
| **0.2** | 86.88 | 92.46 |
| **0.4** | 84.64 | 91.13 |
| **0.6** | 88.49 | 93.23 |
| **0.8** | 87.04 | 92.52 |
| **Average** | 86.76 | 92.34 |

of two trapezoidal speed profiles are used to generate speed and position references (the first profile generates the opening movement of the rotation, and the second one allows the subject to return the arm to the original position). As proven in Section II-E, the error in speed estimation has a low impact in the performance of the current limitation algorithm. However, it is interesting evaluating the algorithm performance at different speeds in a real exoskeleton rehabilitation application. The maximum speeds for the four profiles have been set to 0.2 rad/s, 0.4 rad/s, 0.6 rad/s and 0.8 rad/s (maximum motor speed is 0.83 rad/s, as seen in previous tests and simulations). The angle to be swept by the motor is set to 10 rad for the four profiles and the acceleration is set to 0.2 rad/s$^2$. The current limitation that has been implemented for these tests does not allow current peaks and the current limitation value has been set to 100 mA, in order not to make the subjects apply too much force to make the limitation condition take place.

Subjects were asked to arbitrarily generate a force opposing to the motor movement in order to stop it twice: the first one, in a relatively progressive way at some moment of the second third of the opening phase, and the second one, in a more aggressive way in the last third of this phase. Likewise, in the closing phase, they were asked to try to accelerate the movement pulling the cable, at some moment of their choice. Each of the four subjects performed the test at one of the four different speeds. Using different subjects for different trajectories has the objective of introducing more variability in the validation process, given that each subject will apply their force in a different way, generating different disturbances at different moments of the trajectory.

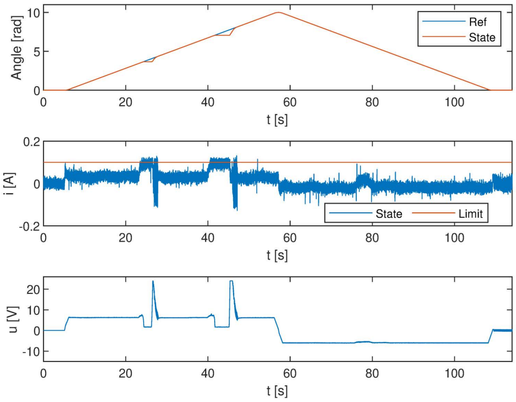


**FIGURE 20.** Motor position, current and applied voltage with full time current limitation, $\omega_{\max} = 0.2$ rad/s.

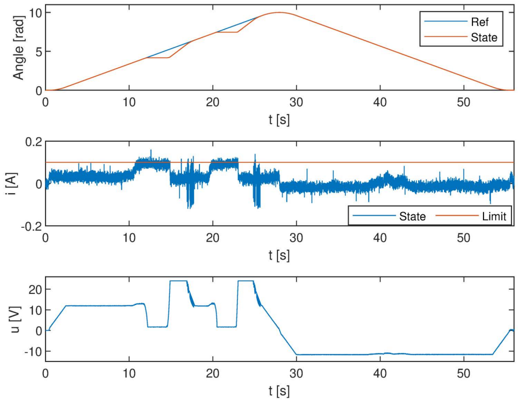


**FIGURE 21.** Motor position, current and applied voltage with full time current limitation, $\omega = 0.4$ rad/s.

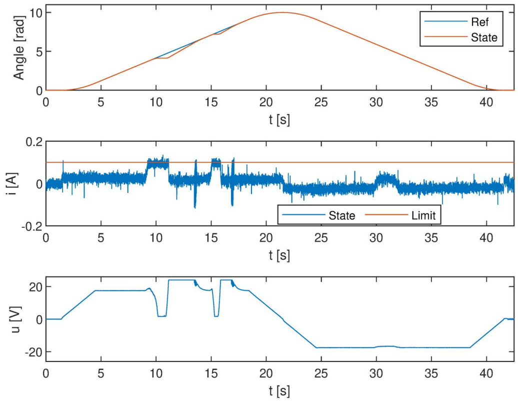


**FIGURE 22.** Motor position, current and applied voltage with full time current limitation, $\omega = 0.6$ rad/s.

As can be seen in Fig. 20 to Fig. 23 at the beginning of the movement, the motor follows the predefined trajectory while the current is below the current threshold. When users apply force to oppose to the movement, current reaches the limitation value of 0.1 A and it is maintained around that value. The motor stops its movement but it applies the force given by the current threshold value to the limb. As soon as the users reduce the resistance force, the current decreases and the motor keeps following the trajectory reference,

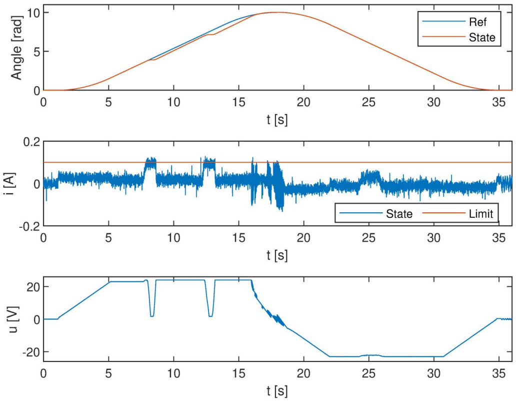


**FIGURE 23.** Motor position, current and applied voltage with full time current limitation, $\omega = 0.8$ rad/s.

as commanded by the position controller. Note that this behavior is completely expected, since what the algorithm does is to protect the motor from overcurrent, allowing it to deliver the torque generated by $i_{\text{sat}}$. If motor movement implies unsafe operation, the algorithm should prevent motor movement while exerting torque, as it is doing. In the closing phase (descending part of the trajectory profile) the pulling force applied by the subjects is not enough to make the current reach the limitation threshold, but a decrease in the absolute value for the current is observed. Table 5 shows the percentage of both power $P_s$ and absolute value of current $|i_s|$ in relation to ideal maximum current conditions, considering in this case $i_{\text{sat}} = 0.1$ A. No significant variations in performance depending on the motion speed are noticeable, as it was expected. The average percentage of delivered power when current is limited is 86.76%, while the average current under those conditions is 92.34%. In this way, the performance of the proposed current limitation method is validated for exoskeleton rehabilitation applications, which inherently involve interaction with humans.

## IV. CONCLUSION

A software current limitation method based on a continuous DC motor model predictor has been designed, simulated and experimentally validated. The algorithm has proven to be robust to external disturbances in both simulation and experimental tests. It can completely limit the current or it also can allow short-duration current peaks to be produced, which are harmless for the motor and accelerate its response. These two possibilities make the algorithm suitable for a very wide range of applications. Both the current limit and the temporal width of the allowed pulses can be easily configured.

The algorithm has proven to effectively limit the current with minimal motor underuse, achieving an average 92.34% of the saturation current when it is limited in tests involving a rehabilitation exoskeleton. Similar performance has been observed when testing the motor in stand-alone conditions, obtaining results between 87.38% (limiting current peaks generated by rapid voltage variations) and 98.04% (in overload conditions). Not only does the algorithm protect the motor, but it also maintains the current very proximal to the established limit. The proposed method works properly using an H-bridge driver and is robust to conservative bounded uncertainty in motor model parameters and external load variations. The parameter values used in the real implementations for the tests were exactly the same ones used in simulation.

An indirect use of this method in exoskeleton applications is that, by adjusting the current limitation threshold, the maximum force that the exoskeleton can apply to the user is bounded. Using the proposed method, without a force or torque sensor, the force exerted by the motors to the users' limbs is indirectly limited, increasing the safety of the system. In this regard, future work involves performing force/impedance control based on generating suitable maximum current profiles such that, by dynamically limiting the current, the motor torque is also limited.

The proposed method represents a good alternative to hardware current limiting systems. Its application is not dependent on the DC motor nor the used driver, since model parameters can easily be adjusted via software. Additionally, the cost of the motor driving system is reduced, since specific hardware for current limitation is not required. The main limitation of the proposed method is that motor parameters have to be known as precisely as possible. However, this is not a too restrictive requisite, since motor parameters can be experimentally measured with a reasonably low tolerance. As shown in the parameter uncertainty analysis, 5% tolerances in $R$ and $k_e$ are assumable by the method.

Finally, it should be noted that if the current peaks produced when the motor starts moving are required to be limited, the algorithm should be executed at a high frequency, around 1 kHz (as in the conducted study), as the current dynamics are fast. The higher the execution frequency of the algorithm is, the faster it will react to external load variations. However, if a less strict current limitation is required, for example, just in order to avoid overcurrent in quasi-stall conditions, the algorithm can be executed at much lower frequencies.

## REFERENCES

[1] C. Jarrett and A. J. McDaid, "Robust control of a cable-driven soft exoskeleton joint for intrinsic human–robot interaction," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 7, pp. 976–986, Jul. 2017.

[2] L. H. Blumenschein, C. G. McDonald, and M. K. O'Malley, "A cable-based series elastic actuator with conduit sensor for wearable exoskeletons," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6687–6693.

[3] K.-Y. Wu, Y.-Y. Su, Y.-L. Yu, K.-Y. Lin, and C.-C. Lan, "Series elastic actuation of an elbow rehabilitation exoskeleton with axis misalignment adaptation," in *Proc. Int. Conf. Rehabil. Robot. (ICORR)*, Jul. 2017, pp. 567–572.

[4] S. Lessard, P. Pansodtee, A. Robbins, J. M. Trombadore, S. Kurniawan, and M. Teodorescu, "A soft exosuit for flexible upper-extremity rehabilitation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 8, pp. 1604–1617, Aug. 2018.

[5] I.-M. Alguacil-Diego, A. Cuesta-Gómez, A.-F. Contreras-González, D. Pont-Esteban, D. Cantalejo-Escobar, M. Á. Sánchez-Urán, and M. Ferre, "Validation of a hybrid exoskeleton for upper limb rehabilitation. A preliminary study," *Sensors*, vol. 21, no. 21, p. 7342, Nov. 2021.

[6] A. Riani, T. Madani, A. Benallegue, and K. Djouani, "Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton," *Control Eng. Pract.*, vol. 75, pp. 108–117, Jun. 2018.

[7] B. K. Dinh, M. Xiloyannis, C. W. Antuvan, L. Cappello, and L. Masia, "Hierarchical cascade controller for assistance modulation in a soft wearable arm exoskeleton," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1786–1793, Jul. 2017.

[8] S. Lessard, P. Pansodtee, A. Robbins, L. B. Baltaxe-Admony, J. M. Trombadore, M. Teodorescu, A. Agogino, and S. Kurniawan, "CRUX: A compliant robotic upper-extremity exosuit for lightweight, portable, multi-joint muscular augmentation," in *Proc. Int. Conf. Rehabil. Robot. (ICORR)*, Jul. 2017, pp. 1633–1638.

[9] H. M. F. Vatan, S. Nefti-Meziani, S. Davis, Z. Saffari, and H. El-Hussieny, "A review: A comprehensive review of soft and rigid wearable rehabilitation and assistive devices with a focus on the shoulder joint," *J. Intell. Robotic Syst.*, vol. 102, no. 1, pp. 1–24, May 2021.

[10] M. S. A. Rahim, M. M. Azizan, M. I. Yusoff, M. H. Mat, N. I. B. Ahmad, N. F. B. Fadzail, and S. B. M. Esa, "Determination of soft starter firing angle performance to mitigate motor high inrush current using current limitation method," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 767, Feb. 2020, Art. no. 012025.

[11] A. G. Yepes and J. Doval-Gandoy, "Overmodulation method with adaptive *x-y* current limitation for five-phase induction motor drives," *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2240–2251, Mar. 2021.

[12] C. Lee, J. Lee, and I. G. Jang, "Shape optimization-based design investigation of the switched reluctance motors regarding the target torque and current limitation," *Struct. Multidisciplinary Optim.*, vol. 64, no. 2, pp. 859–870, 2021.

[13] G. D. K. Amesimenu, K.-C. Chang, T.-W. Sung, H.-C. Wang, G. Shyirambere, K.-C. Chu, and T.-L. Hsu, "Study of reduction of inrush current on a DC series motor with a low-cost soft start system for advanced process tools," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.* Cham, Switzerland: Springer, 2020, pp. 586–597.

[14] Z. Wen, G. Valente, A. Formentini, L. Papini, C. Gerada, and P. Zanchetta, "A novel current limitation technique exploiting the maximum capability of power electronic inverter and bearingless machine," *IEEE Trans. Ind. Appl.*, vol. 57, no. 6, pp. 7012–7023, Nov. 2021.

[15] K. R. Zelenka and T. H. Barton, "A fast-acting current limit for a DC motor drive," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 5, pp. 798–804, Sep. 1986.

[16] A. M. O. El Zawawi and H. A. Ashour, "A fast acting current limit for a PC-based DC drive," in *Proc. 8th Medit. Electrotech. Conf. Ind. Appl. Power Syst., Comput. Sci. Telecommun. (MELECON)*, Mar. 1996, pp. 332–336.

[17] B. Chen and C. Qiu, "A current-limiting control drive circuit for DC motor start-up and blocking," in *Proc. Int. Conf. Robot. Control Eng.*, Apr. 2021, pp. 64–69.

[18] M. George, "Speed control of separately excited DC motor," *Amer. J. Appl. Sci.*, vol. 5, no. 3, pp. 227–233, Mar. 2008.

[19] M. George, K. P. Basu, and A. T. W. Chiat, "Model reference controlled separately excited DC motor," *Neural Comput. Appl.*, vol. 19, no. 3, pp. 343–351, Apr. 2010.

[20] M. Minkova, D. Minkov, J. L. Rodgerson, and R. G. Harley, "Current limitation in the adaptive neural speed control of a DC motor," in *Proc. IEEE AFRICON*, Sep. 1996, pp. 837–842.

[21] M. Arifujjaman, "Performance and reliability comparison of grid connected small wind turbine systems," Ph.D. dissertation, Memorial Univ. Newfoundland, St. John's, Newfoundland, CA, USA, Aug. 2010.

[22] D. Pont, A. F. Contreras, J. L. Samper, F. J. Sáez, M. Ferre, M. A. Sánchez, R. Ruiz, and A. García, "Exoflex: An upper-limb cable-driven exosuit," in *Proc. Iberian Robot. Conf.* Cham, Switzerland: Springer, 2019, pp. 417–428.

[23] Control Technology Corp. *Measuring Motor Parameters*. Accessed: Feb. 23, 2022. [Online]. Available: https://support.controltechnologycorp.com/motorParameters.pdf

[24] N. Lotti, M. Xiloyannis, F. Missiroli, C. Bokranz, D. Chiaradia, A. Frisoli, R. Riener, and L. Masia, "Myoelectric or force control? A comparative study on a soft arm exosuit," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1363–1379, Jun. 2022.

[25] M. A. Buckley, A. Yardley, G. R. Johnson, and D. A. Cams, "Dynamics of the upper limb during performance of the tasks of everyday living—A review of the current knowledge base," *Proc. Inst. Mech. Eng., H, J. Eng. Med.*, vol. 210, no. 4, pp. 241–247, Dec. 1996.

**DAVID PONT-ESTEBAN** was born in Zaragoza, Spain. He received the B.S. degree in electronics and automation from the Universidad de Zaragoza, Zaragoza, in 2017, and the M.S. degree in automation and robotics from the Universidad Politécnica de Madrid, Spain, in 2018. He is currently pursuing the Ph.D. degree in automation and robotics. Since 2018, he has been a Research Engineer with the Centre for Automation and Robotics (CAR) UPM-CSIC. His research interests include exoskeletons, control engineering, signal processing, and embedded electronics. He received the Extraordinary Award for his B.S. degree.

**MIGUEL ÁNGEL SÁNCHEZ-URÁN** received the Laurea degree in control engineering and electronics and the Ph.D. degree (Hons.) in electronics from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1992 and 1998, respectively. From 1992 to 2001, he was with the Research Department, Telefónica, S. A. He is currently a Professor with the Department of Electrical and Electronic Engineering, Automatic Control, and Applied Physics, UPM, and a member of the Centre for Automation and Robotics UPM-CSIC. He has been the author of several articles on human interfaces for teleoperation. His research interests include automatic control and advanced telerobotics, haptics, test techniques, and signal processing. He is working on different projects and developments in these areas. He is also a member of the Study Committee SC-D1 (Materials and Emerging Test Techniques) of CIGRE.

**ALDO FRANCISCO CONTRERAS-GONZÁLEZ** was born in Saltillo, Mexico. He received the B.S. degree in mechatronics engineering from the Technological Institute of Saltillo, Mexico, in 2011, and the M.S. degree in mechatronics engineering from the Technological Institute of Nuevo León, Mexico, in 2016. He is currently pursuing the Ph.D. degree in automation and robotics with the Universidad Politécnica de Madrid, Spain. From 2011 to 2014, he held the position of Robotics and Welding Engineer at John Deere Industries and was a Member of the Leaders of the Department of Robotics, John Deere Mexico. From 2014 to 2016, he served as an Assistant Professor at the Autonomous University of Coahuila. His research interests include rehabilitation robotics, flexible and wearable applications, and development of human motion sensors in the field of robotics.

**MANUEL FERRE** (Senior Member, IEEE) received the Laurea degree in control engineering and electronics and the Ph.D. degree in automation and robotics from the Universidad Politécnica de Madrid (UPM), in 1992 and 1997, respectively. He worked as a Postdoctoral Researcher at the Human-Machine System Laboratory, Massachusetts Institute of Technology (MIT). He is currently a Full Professor with UPM. He has participated and coordinated several research projects in robotics and automatic control at national and international programs. He has four patents of haptic devices and stereoscopic video cameras. He has authored more than 150 publications. His research interests include automatic control, advanced telerobotics, exoskeleton, and haptics. He is a member of the EuroHaptics Society, euRobotics, and CEA. He has served as the chair for several committees of national and international societies. He is an Editor of the Springer Series on Touch and Haptic System.