Research on Biomedical Engineering
https://doi.org/10.1007/s42600-020-00089-1

ORIGINAL ARTICLE

Identifying the knee joint angular position under neuromuscular electrical stimulation via long short-term memory neural networks

Héber H. Arcolezi[1,2] · Willian R. B. M. Nunes[3] · Selene Cerna[1,2] · Rafael A. de Araujo[2] · Marcelo Augusto Assunção Sanches[2] · Marcelo Carvalho Minhoto Teixeira[2] · Aparecido Augusto de Carvalho[2]

Received: 6 February 2020 / Accepted: 27 August 2020

Abstract

Purpose Recurrent neural networks (RNNs) offer a promising opportunity for identifying nonlinear systems. This paper investigates the effectiveness of the long short-term memory (LSTM) RNN architecture in the specific task of identifying the knee joint angular position under neuromuscular electrical stimulation (NMES). The standard RNN model referred to as SimpleRNN and the well-known multilayer perceptron (MLP) are used for comparison purposes.

Methods Data from seven healthy and two paraplegic volunteers were experimentally acquired. These data were adequately scaled, encoded using three timestep values (1, 5, and 10), and divided into training, validation, and testing sets. These models were mainly evaluated using the root mean square error (RMSE) and training time metrics.

Results The three NN models demonstrated very good fitting to data for all volunteers. The LSTM presented smaller RMSE for most of the individuals. This is even more notable when using 5 and 10 timesteps achieving half and one-third of the error from MLP and half of the error from the SimpleRNN. This higher utility comes with a substantial time-utility trade-off.

Conclusion The results in this paper show that the LSTM worths deeper investigation to design control-oriented models to knee joint stimulation in closed-loop systems. Even though the LSTM takes more time for training due to a more complex architecture, time and computational costs could be increased if achieving better modeling of systems. Rather than mathematically modeling this system with several unique parameters per individual, the use of NNs is encouraged in this task where there exist high nonlinearities and time-varying parameters.

Keywords Neuromuscular electrical stimulation · Spinal cord injury · Long short-term memory · Knee joint · Nonlinear system identification

Introduction

Context of the problem

It is well-known that spinal cord injury (SCI), which may be caused by traumatic reasons (e.g., road accidents, day-to-day falls, sports injuries) or nontraumatic ones (e.g., tumors or diseases that destroy the neurological tissue of the spinal cord), can cause partial or total paralysis under the level of injury. Due to lack of movements, SCI can lead to muscle atrophies and spasms, inability to complete daily or occupational activities, a lot of pain to its individuals, loss of sexual function, and much more (Grin et al. 2009; Law and Shields 2007; Cattagni et al. 2018).

In this context, several studies have shown that neuromuscular electrical stimulation (NMES) and functional electrical stimulation (FES) can produce good results in rehabilitation treatments of spinal cord–injured patients. Many social and health benefits were reported in the literature, for example, the maintenance and recovery of muscle's strength, prevention of flaccidity and hypotrophy (classic signs of muscle inactivity), and by offering a higher expectation of life with more quality. On the use of electrical stimulation, the strength

✉ Héber H. Arcolezi
hh.arcolezi@unesp.br; heber.hwang_arcolezi@univ-fcomte.fr

1 Femto-ST Institute, University Bourgogne Franche-Comté, UBFC, CNRS, Belfort 90000, France

2 Department of Electrical Engineering, UNESP – University Estadual Paulista, Campus of Ilha Solteira, Ilha Solteira, SP 15385-000, Brazil

3 Department of Electrical Engineering, UTFPR - Federal University of Technology, Apucarana 86800, Paraná, Brazil

Published online: 04 September 2020

of muscle contraction is controlled with electrical current pulses by changing the pulses amplitude, width, or frequency. Moreover, numerous investigations for NMES and FES applications to knee joint control in SCI patients have been reported in the literature consisting of two main lines, controlling and modeling (Ferrarin and Pedotti 2000; Lynch and Popovic 2008).

Motivations and objectives

An efficient mapping describing the relationship between the muscular model and stimulation parameters is a very strict requirement for developing powerful feedback control techniques, motivating the main focus of this paper.

Indeed, identifying and modeling nonlinear systems are very important to predict the future behavior of a dynamic system and have been applied in most areas of science. Many real-world systems exhibit nonlinear behaviors. In this context, there exist mathematical models describing systems using linear and nonlinear differential equations. This setting depends on multiple parameters from the physics and mechanics of systems. On the other side, there are black-box models, which are based on direct modeling based on input-output data. Black-box models are well-recommended where the physics of the system in consideration is too difficult or require too much expertise. For example, in this case, muscular behavior presents several time-varying properties and high nonlinearities.

In the context of black-box modeling, recurrent neural networks (RNNs) offer a promising opportunity for identifying and modeling dynamic nonlinear systems. RNNs have the advantage of a self-feedback connection, which allows maintaining long-term memories. For instance, the output of dynamic systems does not depend only on its current inputs but on the previous behavior of the system, which makes RNNs ideal to solve this task.

Even though standard RNNs are very effective for complex tasks, there is the problem of vanishing and exploding gradients due to the big computation of gradients over many timesteps. One of the most popular and applicable RNNs is the long short-term memory (LSTM) architecture. It was introduced by Hochreiter and Schmidhuber (1997) and is well-known by its intrinsic characteristics of maintaining long-state memories, avoiding the vanishing and exploding gradient problem. This model has been recently applied for nonlinear system identification, as we discuss it in a later subsection ("Related work"). And, it has been successfully applied in many other tasks in both academy and industry.

In Arcolezi et al. (2020), our group presented a novel control-based approach to human lower limb rehabilitation. The approach assumes a well-mapped relationship (model) between pulse width and knee angular position. Next, an off-line optimization fine-tunes the parameters of a closed-loop controller, namely RISE (Robust Integral of the Sign of the Error), using this model. In the end, rather than using an empirical tuning (the common approach using this controller), the rehabilitation procedure is retaken with good parameters.

With these elements in mind, in this paper, we aim to extend the analysis to the particular task of identifying the knee joint angular position under NMES. The main goal is to achieve better control-oriented models to improve closed-loop techniques to this system. Hence, we propose to investigate the LSTM architecture for this task. To the best of the authors' knowledge, the LSTM has never been implemented to identify this system's dynamics. To validate the effectiveness of this model, we acquired experimental data from seven healthy and two SCI individuals. Furthermore, we compared the LSTM with the *SimpleRNN* and with a multilayer perceptron (MLP) as a nonlinear autoregressive exogenous (NARX) model. Without loss of generality, we refer to the standard RNN model as *SimpleRNN* in the rest of this paper.

The motivation to investigate the LSTM for this task emerges from the fact that NNs can learn intrinsic nonlinear relationships of dynamic systems. As more data becomes available throughout rehabilitation sessions, the more insights these models are able to learn. On the other hand, one can describe the behavior of a dynamic system as special time-series problems, in which RNNs, such as the LSTM model, are successfully applied. Lastly, training NN structures is easier in comparison with mathematically modeling the knee joint dynamics and executing tests for identifying parameters of each individual. And, there is high power for computation and powerful tools to deal with big data nowadays.

Related work

The adequate identification of the knee joint angle under NMES is one main key for developing powerful closed-loop control techniques. System identification techniques are mainly divided into linear and nonlinear models. Common modeling techniques, such as transfer functions and state-space, are in the class of linear approaches. On the other hand, there exist many nonlinear modeling techniques, for instance, nonlinear autoregressive exogenous model, Volterra series, Hammerstein and Wiener models, and neural networks (Aguirre 2015).

In the context of NNs, which is the main focus of this paper, the LSTM architecture has been recently introduced for system identification. Ogunmolu et al. (2016) presented a comparison of deep dynamic neural networks for nonlinear system identification. Wang (2017) introduced the LSTM structure as a new concept for nonlinear system identification. The author introduced a new adaptive-learning approach for a convex-based LSTM to achieve fast learning. Hirose and Tajima (2017) applied the LSTM model for modeling rolling friction in physical systems. Gonzalez and Yu (2018)

compared the LSTM and MLP architectures for modeling nonlinear systems. Wang et al. (2018) applied the LSTM for activity recognition. Their identification model was applied to a flexible lower extremity exoskeleton robot to identify activities such as ascending/descending stairs, sitting down, and standing up. Similarly, Ordóñez and Roggen (2016) combined the LSTM with convolutional NNs for activity recognition from wearable sensors.

In our application scenario, one can find different mathematical models of electrically stimulated knee joints. For instance, Riener et al. (1996) proposed a biomechanical model of this system. To acquire the individual's parameters, the authors performed specific procedures such as anthropometric measurements, a passive pendulum test, and specific open-loop stimulation experiments. Ferrarin and Pedotti (2000) modeled the dynamics between electrical stimulus and joint torque. The authors modeled the dynamics of the lower limb with a nonlinear second-order model. They identified the parameters of each individual with free pendular movements. And they measured the active torque applying FES to the quadriceps muscles. Law and Shields (2007) proposed three mathematical models of human paralyzed muscles, one linear, and two nonlinear models. Lynch et al. (2011) presented an upgraded version for the model introduced in (Ferrarin and Pedotti 2000). The authors included nonideal conditions to the nominal torque such as fatigue, tremor, and spasms. Benoussaad et al. (2009) proposed to identify the parameters of the physiological-based model of the quadriceps-shank under FES.

On the other hand, there are also black-box models. For example, Previdi (2002) investigated the problem of identifying nonlinear models for the functional electrical stimulation (FES) process. The author introduced NARX models identified from input-output data, considering polynomial and neural network structures. Kamaruddin et al. (2016) presented an improvement of the quadriceps muscle model under FES by using a counterfeit smart procedure named backpropagation neural network nonlinear autoregressive. Next, Ghani et al. (2016) presented a continuation of the previous work. The authors implemented neural networks in the nonlinear autoregressive moving average model for modeling the quadriceps under NMES.

Theoretical background

System identification via neural networks

Neural networks are a set of algorithms inspired by biological neural networks. These algorithms are based on the interconnection of several units (neurons) to transmit signals, which are normally structured into three or more layers, input, hidden(s), and output. In the literature, NNs are considered universal function approximators by mapping from an input space to the output one.

More specifically, given an input space U and an output space Y where an operator F describe the system to be identified, the aim is finding a function $\widehat{Y}$ to approximate F. From the Stone-Weierstrass theorem, it is known that a continuous and bounded function F can be uniformly approximated by a polynomial function $\widehat{Y}$. Furthermore, the universal approximation theorem states that there exists a combination of hyperparameters to an NN that allows it to learn any nonlinear function (Narendra and Parthasarathy 1990; Chen et al. 1990; Narendra and Mukhopadhyay 1997; Haykin 2009).

Generally, NNs can be divided into two large classes, feedforward and recurrent neural networks. A simple MLP from the big class of feedforward NNs is characterized by the unidirectional flow of computation. For example, considering U as the input layer, H as hidden layers, and Y the output layer, its flow of information is $U \rightarrow H \rightarrow Y$. The training of the NN consists of adjusting each connection between neurons from one layer to the other, which is composed of weight ω_{ij} and bias b_i values activated by nonlinear functions $f(.)$. During training, a loss function is used to measure the error between the desired $y(k)$ and the actual value outputted from the NN $\widehat{y}(k)$, in which this error is back-propagated through the network in order to update the weights of each module and so "learn."

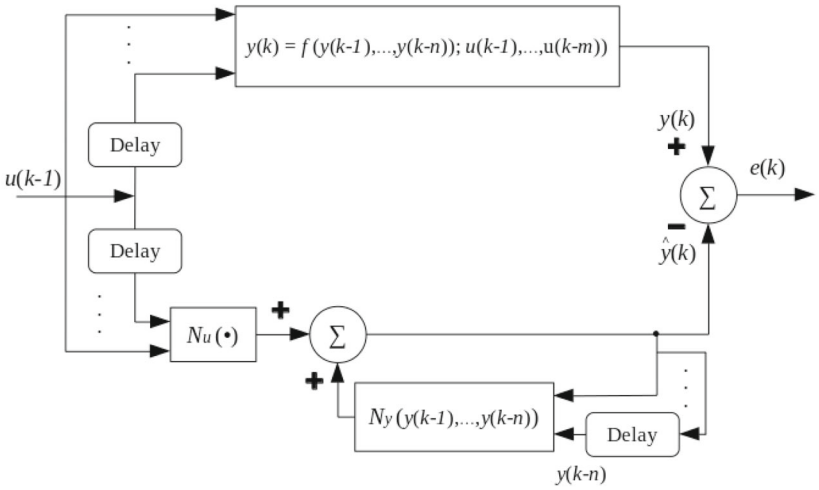
Fundamentally, building black-box models rely on acquiring appropriated data about the system. That is, acquiring data covering the whole operating range of interest is considered the bottom line to achieve success with the developed models. Hence, the concept of modeling systems with input-output data using NNs concerns about searching for the best combination of weights and other hyperparameters of the NN itself (Wang 2017; Gonzalez and Yu 2018; Haykin 2009).

Without loss of generality, consider the following single-input and single-output discrete system structured as:

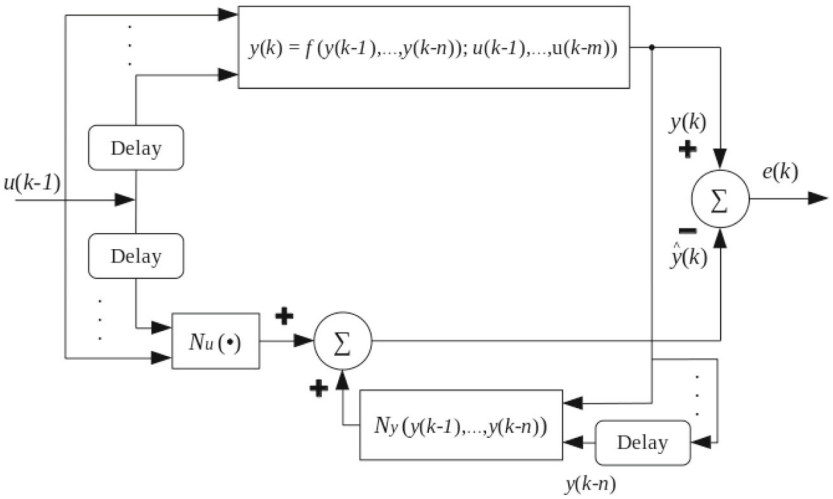
$$y(k) = f[y(k-1), \cdots, y(k-n); u(k-1), \cdots, u(k-m)] \quad (1)$$

where $f(.)$ is an unknown nonlinear difference equation representing the system dynamics; u and y are measurable scalar input and output, respectively; and m and n are the maximum lags for the systems' output and input. Extended versions for multi-input and multi-output cases are also possible, but it is out of the scope of this paper.

To identify the discrete-time system in Eq. (1), one can use two major types of identification structures presented in the literature. The parallel and the series-parallel identification model (Wang 2017; Gonzalez and Yu 2018; Narendra and Parthasarathy 1990). The first structure depends on past inputs of the system and the outputs of the NN model. The second arrangement uses both past inputs and systems' outputs. Figure 1 illustrates both parallel and



(a) Parallel structure for system identification.



(b) Series-parallel structures for system identification.

Fig. 1 System identification architectures. **a** Parallel structure for system identification. **b** Series-parallel structures for system identification

series-parallel architectures, respectively, which are mathematically described as:

$$\widehat{y}(k) = F\left[\widehat{y}(k-1), \cdots, \widehat{y}(k-n); u(k-1), \cdots, u(k-m)\right] \quad (2)$$

$$\widehat{y}(k) = F[y(k-1), \cdots, y(k-n); u(k-1), \cdots, u(k-m)] \quad (3)$$

where $\widehat{y}$ is the model output, y is the real system output, F is the model structure, and m and n are the regression order for the input and output. Both m and n parameters are chosen before the identification process, which indicates how many past steps (*timesteps*) of input and output will be used in the modeling part.

It is noteworthy that F is a general autoregression model structure (simplified NARX without the error term), which one can directly use machine learning models, e.g., the well-known MLP. Besides the regression terms, RNNs have the advantage of maintaining hidden states to keep memories as explained in the next subsection. In this paper, we design an LSTM model to approximate the nonlinear mapping function *f*(.) in Eq. (1) using the series-parallel structure in Eq. (3) with three timestep values $m = n = [1, 5, 10]$. Additionally, we compare this model with a SimpleRNN and an MLP NARX model as baselines.

Recurrent neural networks

RNN is the class of NNs that manipulate sequential data. This setting has at least one feedback connection that provides the ability to use contextual information when mapping between input and output sequences. In its topology, any neuron can be connected to any other, and even with itself. Figure 2 illustrates an unrolled RNN, where each neuron receives input u, produces an output y, and sends it back to itself. At each time step k, a recurrent neuron receives the input and its own output from the previous time step, which is known as the hidden state $h(k)$ (Gulli and Pal 2017; Haykin 2009). In contrast with a unidirectional flow of information from feedforward NNs,

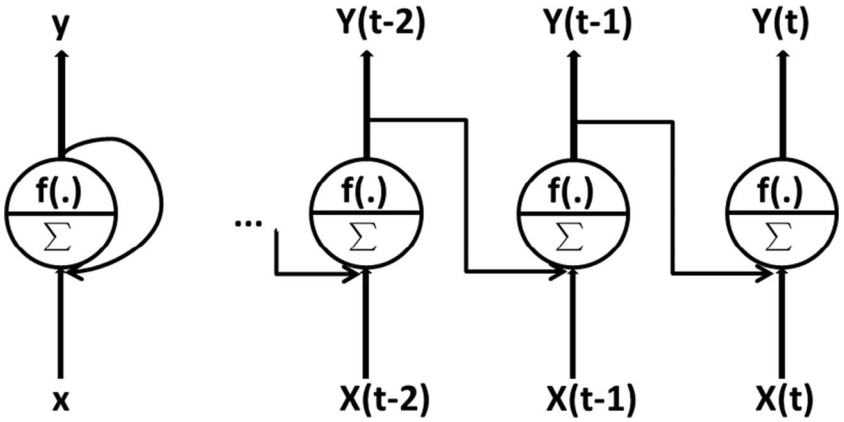


Fig. 2 An unrolled recurrent neural network with self-feedback connection (hidden state)

the self-feedback connection allows a chain-like flow of information for RNNs as $U \leftrightarrows H \leftrightarrows Y$.

The output $\hat{y}(k)$ and hidden state $h(k)$ of RNNs are determined through the mathematical formulation as:

$$h(k) = \tanh(W_{uh}u_k + W_{hh}h_{k-1} + b_h) \tag{4}$$

$$\hat{y}(k) = W_{hy}h_k + b_y \tag{5}$$

where W_{uh}, W_{hh}, and W_{hy} are the weight matrices of input-to-hidden, hidden-to-hidden, and hidden-to-output, respectively; b_h and b_y are the bias terms; and tan$h$ is the hyperbolic tangent function.

Long short-term memory

The LSTM NN is an important variant of RNNs. This model is capable of learning long-term dependencies, and it overcame the problem of vanishing and exploding gradients that SimpleRNNs have. This particular problem makes the gradient extremely unstable to deal if a NN runs for a long time. The LSTM model introduced by Hochreiter and Schmidhuber (1997) can efficiently learn to maintain information over several time intervals without suffering from such a problem. The essential key behind LSTM success is its memory cell, which divides its states in long-term state $c(k)$ and short-term state $h(k)$. The LSTM cell has a principal layer $g(k)$ and three nonlinear gating units attached, the input $i(k)$, forget $f(k)$, and output $o(k)$ gates, which allows truncating gradients protecting and controlling the long-term state. Compared with a SimpleRNN, the LSTM model differs the summation units in the hidden layer(s) by memory blocks, for instance, see Fig. 3, which illustrates the inside schematic architecture of LSTM.

Initially, both input $x(k)$ and the previous short-term state $h(k-1)$ are fed to four different and fully connected layers. The first layer computes the internal hidden state $g(k)$, using $x(k)$ and $h(k-1)$, and partially store $g(k)$ in the long-term state. The other three layers are nonlinear gating units, which use logistic activation function outputting values between 0 for "close the gate" and 1 for "open the gate." The first one named forget gate is able to preserve long-term states for as long as it is needed, i.e., it controls what to erase from the long-term state. The second one is the input gate, which regulates the flow of new values by recognizing important inputs, i.e., it controls which part of $g(k)$ is able to be added to the long-term state. Lastly, the output gate defines how much of the long-term state should be read and output for both short-term state $h(k)$ and output $y(k)$.

The mathematical formulation is as follows (Géron 2017; Gulli and Pal 2017; Gonzalez and Yu 2018).

$$i(k) = \sigma\left(W_{xi}^T x(k) + W_{hi}^T h(k-1) + b_i\right) \tag{6}$$

$$f(k) = \sigma\left(W_{xf}^T x(k) + W_{hf}^T h(k-1) + b_f\right) \tag{7}$$

$$o(k) = \sigma\left(W_{xo}^T x(k) + W_{ho}^T h(k-1) + b_o\right) \tag{8}$$

$$g(k) = \tanh\left(W_{xg}^T x(k) + W_{hg}^T h(k-1) + b_g\right) \tag{9}$$

$$c(k) = f(k) \otimes c(k-1) \otimes g(k) \tag{10}$$

$$y(k) = o(k) \otimes \tanh(c(k)) \tag{11}$$

where $\otimes$ means an element-wise multiplication; W_{xi}, W_{xf}, W_{xo}, and W_{xg} are the weight matrices of each four layers (input, forget, output, and internal hidden state) for their connection to the input vector $x(k)$; W_{hi}, W_{hf}, W_{ho}, and W_{hg} are the weight matrices of the same four layers for their connection to the previous short-term state $h(k-1)$; and b_f, b_g, b_i, and b_o are the bias terms of each of the previous four layers. The main objective of nonlinear system modeling with LSTM NN is to update the weights between each layer with the input and the short-term state, such that the output $y(k)$ in Eq. (11) converges to the system output given by Eq. (1).

Methods

Test platform

All experiments were performed at the Instrumentation and Biomedical Engineering Laboratory ("Laboratório de

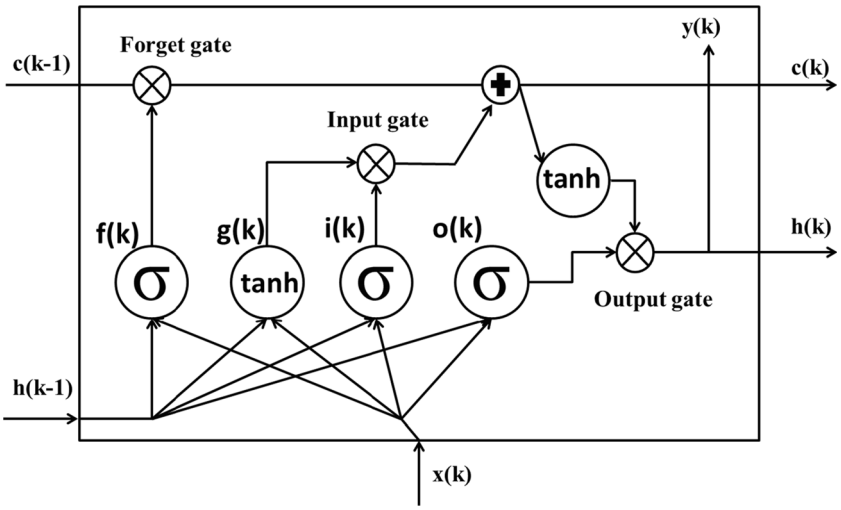


Fig. 3 The inside schematic architecture of an LSTM cell

Instrumentação e Engenharia Biomédica—LIEB") at the São Paulo State University (UNESP)—Ilha Solteira—using the apparatus illustrated in Fig. 4.

The complete apparatus consists of the following: an instrumented chair composed by one electrogoniometer NIP 01517.0001 (Lynx®, São Paulo, Brazil), one gyroscope LPR510AL (ST Microelectronics®, Switzerland), two triaxial accelerometers MMA7341 (Freescale®, USA), one NI myRIO controller (National Instruments®, USA) to operate in real time, a user interface developed in the LabVIEW® student version edition, and one current-based neuromuscular electrical stimulator developed at LIEB by Sanches (2013). It consists of eight independent channels, in which it generates current, rectangular, biphasic, and balanced waveforms. It has the capacity to supply currents with amplitudes of up to 140 mA, with a 1-mA step, pulse width (PW) from 0 to 500 μs, with a step of 1 μs, and frequency (F) from 20 to 300 Hz, with a step of 1 Hz. The stimulator has a current-based output allowing its use in closed-loop control (Arcolezi et al. 2019, 2020; Nunes et al. 2019; Gaino et al. 2020; Teodoro et al. 2020). The output stage of this stimulator is current, which is a great advantage for its use in control, as it makes it possible to control and predict the amount of current applied to the muscle, even if the coupling resistance and tissue impedance change.

We control the stimulation intensity by setting the pulse amplitude to the quadriceps and controlling the PW. In this study, we fixed the following parameters: stimulus frequency at 25 Hz (constant frequency trains (CFT) technique), pulse amplitude at 80 mA for healthy individuals, and 120 mA for

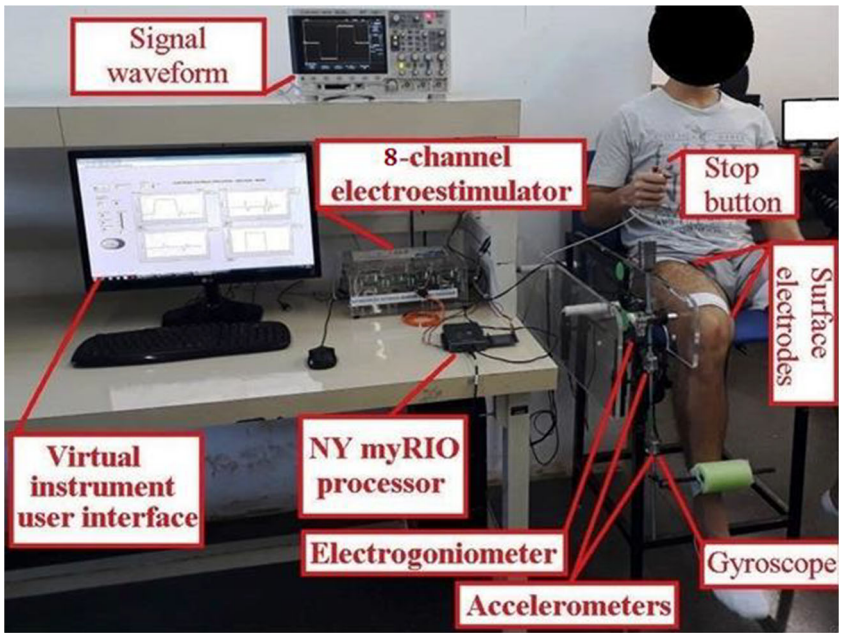


Fig. 4 Test platform for electrical stimulation experiments

paraplegic ones. This difference in pulse amplitude occurred due to insufficient contractions using amplitude below 120 mA for these paraplegic individuals and their respective muscular atrophy conditions. Lastly, we used surface electrodes with rectangular self-adhesive CARCI 50-mm × 90-mm settings.

Volunteers

In this study, seven able-bodied individuals (male, aged 22–28) labeled as H1–H7 and two male SCI individuals labeled as P1 (32 years old, injury level L4–L5, injury time 9 years) and P2 (43 years old, injury level C6, injury time 17 years) participated as volunteers. Prior to the experiments, written informed consent was obtained from all participants. The study with human beings was authorized by a research ethics committee (CAAE: 79219317.2.1001.5402) at UNESP.

The inclusion criteria for individuals were male or female people equal to or older than 18 years old. We selected healthy individuals with no lesion or congenital disease that could compromise the lower limb movements. And, we only considered paraplegic individuals with SCI. As the exclusion criterion, we did not consider pregnant women and people with cardiovascular diseases. Furthermore, we excluded from the analysis the data from healthy individuals who presented fear or discomfort during the tests by voluntarily moving or holding the movement of the leg.

Stimulation protocol for data acquisition

In order to guarantee the comfort of each individual, the chair backrest and knee joint position are adjusted. Each individual has a different knee angular position in the resting condition. The angular position in this condition was measured and taken as an offset during the experimental protocol. Next, to achieve an adequate positioning of the surface electrodes, a muscle analysis is made to find the motor point. The electrophysiological procedure for identifying the motor point consists of mapping the muscle surface using a stimulation electrode to identify the skin area above the muscle, where the motor threshold is the lowest for a given electrical current, which is the skin area most responsive to electrical stimulation (Gobbo et al. 2014). After this procedure, the electrodes can be positioned, properly allowing the neuromuscular electrical stimulation to maximize the evoked voltage, minimizing the intensity of the injected current and the level of discomfort to the volunteer.

After the motor point has been mapped, a test to evaluate the pulse width and the respective angular extension is performed. The test consists of obtaining the minimum $\rho_{\min}$ and maximum $\rho_{\max}$ pulse width corresponding to the extension of the lower limb $\theta_{\min}$ and $\theta_{\max}$, respectively. The values of $\theta_{\min} = 10°$ and $\theta_{\max} = 40°$ were stipulated for this application. Note that we could adopt other values of lower limb extension, but we consider that it was a suitable value for gait control application (Nunes et al. 2019). Figure 5 shows the angular position corresponding to the pulse width assessment test for patient H1. Note that muscle behavior is nonlinear and uncertain for the relationship between the input (pulse width) and the output (angular position) of the system. It is worth mentioning that this extension range can vary in an open-loop operation. Applying the same pulse width value (from the evaluation test), different lower limb extension positions, can be obtained and not belonging to the set $\Theta = \{\theta_{\min}, \theta_{\max}\}$. This uncertainty is due to the dynamic system being nonlinear and time-varying (Nunes et al. 2019).

Then, a 1-min stimulation test is performed. The test performs the system identification procedure by randomly applying pulse width values belonging to the set of values mapped to each individual. The electrical pulse width random value is constant for a random time between 4 and 7 s. Consequently, a new test has randomness in the domain of the pulse width of the electrical stimulus as well as in the time of each stimulus. In this work, the power of muscle activation by electrical stimulation in paraplegic individuals was greater. Before performing the tests, these individuals were not submitted to a rehabilitation research program involving daily electrically stimulated exercise of their lower limbs. Consequently, when there is high stimulation intensity, there is only partial recruitment of synergistic motor units and there is co-activation of antagonists (Doucet et al. 2012). Unfortunately, this is a disadvantage of conventional single-electrode stimulation, whose increased stimulation intensity will lead to increased muscle fatigue (Laubacher et al. 2017; Maffiuletti 2010). To minimize early fatigue in paraplegic individuals (Gregory et al. 2007), the total test time was reduced to 40 s. Figure 6 illustrates the stimulation test from individual H1.

The motivation to follow this random stimulation protocol is the attempt to map both regulation and tracking situations in a single stimulation test, and to recognize the nonlinear and time-varying nature of muscles during a high time of electrical stimulation. The pulse width and angular position data are automatically recorded with a sampling period of 20 ms, i.e., $Ts = 0.02$ (s), resulting in datasets with approximately 2000 samples (40 s) for SCI individuals and 3000 samples (60 s) for healthy ones.

Furthermore, we instructed healthy individuals to relax during the experiments to not influence the leg motion voluntarily and allow the stimulation to control it. Under any displeasure situation, we warned individuals to deactivate the stimulation pulses using a stop button (as one can see in Fig. 4).

Data encoding

Each individual participated in only one stimulation session. So, each acquired dataset (labeled as P1, P2, H1–H7) contains

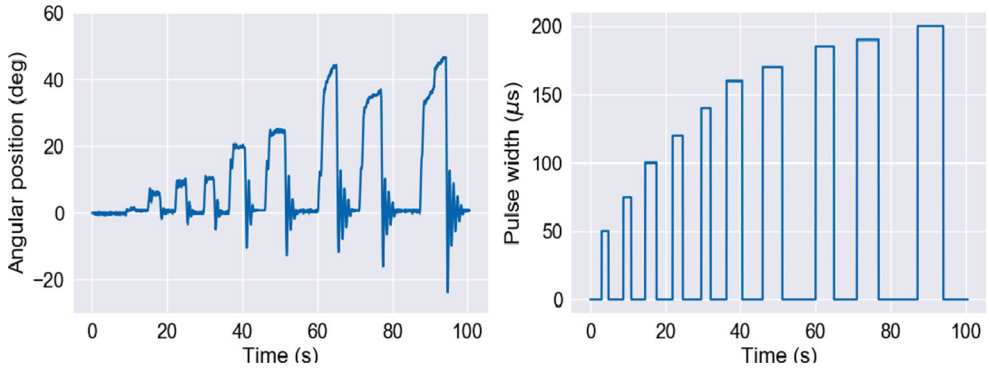


Fig. 5 Pulse width assessment test, where gradually different pulse widths ρ (right side image) are applied and the angular position θ (left side image) of the lower limb is evaluated

information about the delivered pulse width (μs) and knee angular position (radian). These data were suitably treated to feed as input to the models. First, due to the different measurement units (μs versus radian), the "StandardScaler" method from the scikit-learn library (Pedregosa et al. 2011) was applied to re-scale the distribution of values to zero mean and unit variance. To avoid that one feature with higher values has more contribution during training (misleading the models' training), most machine learning techniques (as NNs) can benefit from rescaling the attributes to the same distribution.

Second, both SimpleRNN and LSTM models require three-dimensional input data with the following configuration (*samples*, *timesteps*, *features*). One *sample* is one sequence, a *timestep* is a point of observation in the sample, a *feature* is an observation at a timestep, and the sequence is fed in a one timestep at a time. In this work, we used three timestep values as $m = n = [1, 5, 10]$ (the reader can refer to Eq. 3 for more details). For instance, when using timestep values as $m = n = 1$, each arranged dataset contains *samples* with two $(m + n)$ *features*: the last input value "$\text{Pulse}_{\text{Width}}(k-1)$" and the last output value "$\text{Angular}_{\text{Position}}(k-1)$." Hence, in this case, the *input data* is $[[\text{Pulse}_{\text{Width}}(k-1), \text{Angular}_{\text{Position}}(k-1)], ts, ft]$, where *ts* is the number of *timesteps* and *ft* is the number of *features*. And, the actual output value "$\text{Angular}_{\text{Position}}(k)$" is the *target*. Without loss of generality, the same concept is applied when using higher timestep values m and n.

In contrast, the MLP (NARX model) takes a two-dimensional input data configured as *samples* and *features*. That is, the whole sample is fed at once as this NN does not have a recurrent setting. So, for the case $m = n = 1$, the *input data* is $[[\text{Pulse}_{\text{Width}}(k-1), \text{Angular}_{\text{Position}}(k-1)], ft]$. Similarly, one can extend this concept for higher timestep values.

Finally, after scaling and encoding the data, we divided each dataset into a training set (the initial 70%), validation set (the next 15%), and testing set (the remaining 15%). The testing set comprises the final seconds of the stimulation test, which is a more stringent evaluation for the models. That is, due to the high time of electrical stimulation, there exist factors such as fatigue, tremors, and even voluntary movements (for healthy individuals), which will not be detected if a neural network is not properly trained for generalization.

Model selection

The programming language used in this research was Python, and all neural network models (MLP, SimpleRNN, and LSTM) were developed using the Keras library (Chollet et al. 2015). In the literature, the process of finding the best architecture and tuning its hyperparameters is known as *model selection*. In this investigation, we considered the following hyperparameters: number of layers, number of neurons per layer, mini-batch size, and learning rate.

Hence, each model was tuned via a random search procedure (Bergstra and Bengio 2012) during 300 iterations. This technique randomly selects a combination of hyperparameters to find the best solution for the built model. The proposal of randomly selecting attributes at each instance, where the

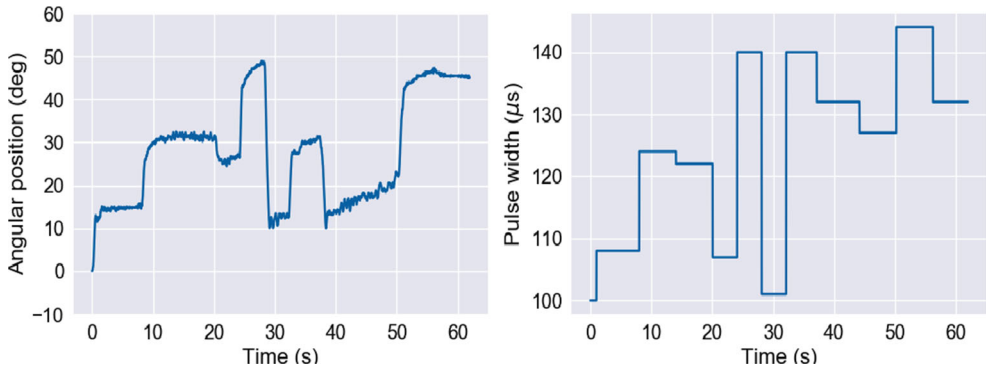


Fig. 6 Proposed stimulation test. On the left-side image, we apply a random PW during a random time. The achieved angular position is shown in the right-side image

whole set of solutions is likely to be reached due to randomness, might probably end up with a good solution with fewer computational costs.

Our random search optimization consisted of gradually stacking recurrent (SimpleRNN or LSTM) or dense (MLP) layers starting with 1 to at most 9 with a fully connected layer (dense) in the end to output the predicted value. Next, we randomly selected the rest of the hyperparameters for each iteration defining an *architecture*. Each architecture was trained during at most 2000 epochs minimizing the mean squared error loss function via the adaptive moment estimation (Adam) optimizer. Furthermore, to avoid overfitting, we used the "*EarlyStopping*" method. We configured it with 15 epochs to monitor the validation loss function decrement recuperating the best weight encountered during training. In order to select the best architecture per model, we averaged the results of each architecture for all individuals and the three timestep values. To do so, we used the root mean square error metric (explained in the next subsection).

Table 1 indicates the range of each hyperparameter we tuned and the best architecture we found per model. Notice that the hyperparameter "number of layers" indicates the total number of layers including the last dense one.

Analysis

The models were evaluated using the following metrics:

(i) the Pearson correlation coefficient (Corr) between the input (pulse width) and the output (angular position) data. This metric will be used as an indication of how correlated the input-output data is. This helps in understanding how "difficult" it is to identify the dynamics of the system. More specifically, Corr measures the linear correlation between two variables x and y. This value ranges from -1 to 1. The higher the value, the stronger the correlation. If negative, it is an inverse correlation. And, if positive, it is a regular correlation;
(ii) The well-known root mean square error (RMSE) metric. It measures the square root average of the squares of the errors, which benefits of penalizing large errors. Mathematically:

$$\mathrm{RMSE} = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)2} \tag{12}$$

where Y is the real output, $\widehat{Y}$ is the predicted output, and n is the total number of samples. The RMSE will be the most important metric to evaluate our models. The smaller the RMSE, the best the model fits the data. The RMSE will be averaged for all individuals and each timestep value. Last, these three averaged RMSE ($m = n = [1, 5, 10]$) will be averaged again as a total RMSE per model;
(iii) The training time (TT) in seconds (s) that each model takes for training. In many applications, the TT metric should be taken into account, and hence, it will be indicative of the time-utility trade-off for the developed models.

Results

In order to run the codes, we used two machines that have the following characteristics: one with a "Titan X", Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60 GHz, 64 Gb of RAM, and a GPU with 3072 cores and 12 Gb of RAM. The other with a "3 Titan V100", Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz with 128 Gb of RAM, and each "Titan V100" with 5120 cores and 16 Gb of RAM.

In this paper, we used the series-parallel structure mathematically described in Eq. (3) to identify the nonlinear and dynamic relationship of pulse width and angular position. Table 2 presents a summary of the identification results for our NN models. For each model, it presents the average RMSE for all individuals per timestep value, the total average RMSE for all individuals and timestep values, and the total training time. Additionally, one can find, in Appendix A, a more complete description of our results in Table 3 with all metrics described in the previous "Analysis" subsection. Finally, Figs. 7 and 8

Table 1 Hyperparameters grid setting and best architecture per model

Hyperparameter	Range	MLP	SimpleRNN	LSTM
Number of layers	[1, 10]	8	6	2
Number of neurons per layer	[1, 300]	101 → 45 → 172→ 242→248→31 → 187→1	29 → 198 → 153→46 →145→1	166 → 1
Mini-batch size	[5, 100]	29	5	9
Learning rate	[0.0001, 0.01]	0.0003	0.0002	0.0002

Table 2 Averaged RMSE per timestep value for all individuals and total training

Model	Average RMSE (°)				Total training time (s)
	$m = n = 1$	$m = n = 5$	$m = n = 10$	Total	
MLP	0.1795	0.1224	0.1375	0.1465	1386.84
SimpleRNN	0.1722	0.1093	0.1034	0.1283	8718.58
LSTM	0.1872	0.0603	0.0482	*0.0986*	38,600.55

illustrate results for the identified models of individuals P2 and H1 respectively for $m = n = [1, 5, 10]$. In these figures, we add a zoom to the last 50 samples to improve the visualization of these curves. For the sake of illustration, we select individual P2 due to a small Corr between data and H1 due to high tremors.

We highlight that we present our results in degrees instead of radians as all our procedures (the "Stimulation protocol for data acquisition" section) were described in this measurement unit. Furthermore, we present the RMSE on the real scale of degrees rather than the standardized data used for training the models.

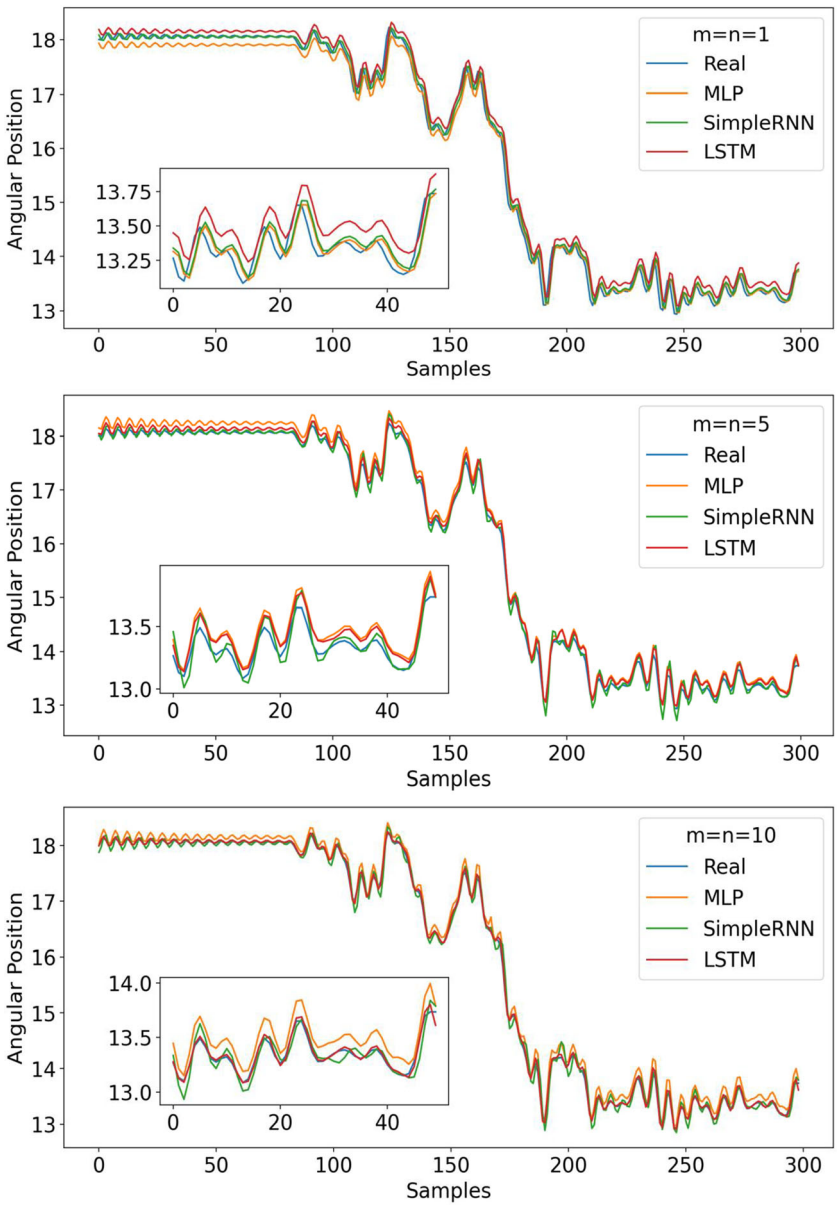


Fig. 7 NNs identified models for individual P2 with 1, 5, and 10 timesteps, respectively

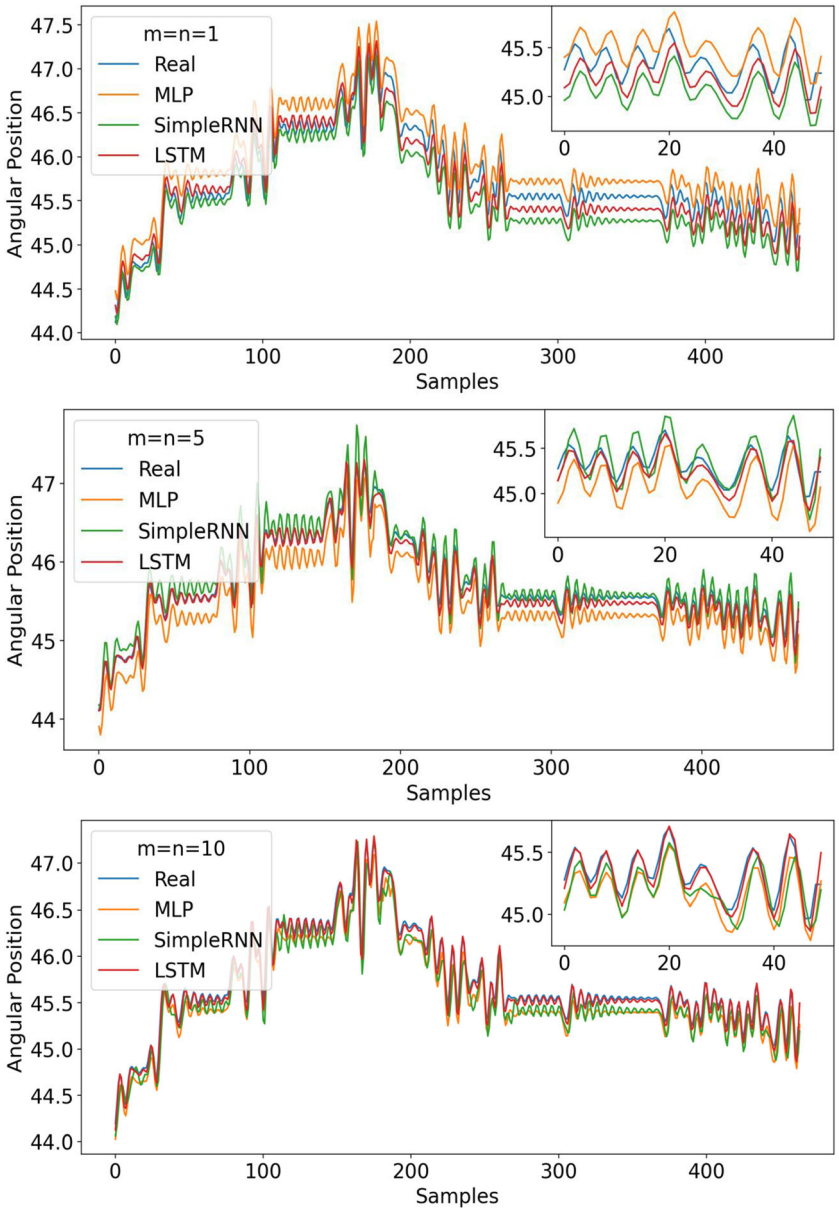


Fig. 8 NNs identified models for individual H1 with 1, 5, and 10 timesteps, respectively

Discussion

The results demonstrated very good identification for the designed NNs by following the procedures and models described in previous sections. The attempt to identify the nonlinear and time-varying relationship of the knee joint under NMES is very promising using LSTMs. Even with few data (only one stimulation session), this NN can accurately model this system. By few data, we mean that the identification data used for training/validation contains approximately 2550 samples while the testing set is composed of the remaining 450 samples (less but in the same proportion for SCI individuals).

Additionally, the test data consists of fuzzier behavior such as fatigue and tremors. For example, one can notice in Fig. 6 that in the final seconds, the pulse width is higher than any time before. On the other hand, the knee joint did not achieve a higher angular position. In our analysis, premature fatigue was noticed for SCI individuals due to long-time muscle paralysis. That is, the knee angular position for the individual P1 was between 0° and 5.5° in the test set and for P2 between 13° and 18° (as one can see in Fig. 7). In contrast, the knee angular position of healthy individuals continued in higher degrees, e.g., for H1 in Fig. 8, it is between 44° and 48°, but with several tremors.

The main findings of comparing the LSTM architecture with the baselines MLP NARX and SimpleRNN models are that the LSTM has smaller RMSE for most of the individuals. On average, using $m = n = 1$, the LSTM model presented the highest RMSE but with a small difference (0.01°) compared with the other two simpler models as one can see in Table 2.

However, the LSTM outstands with a high difference when using more time step values. More precisely, for $m = n = 5$, the average RMSE for the LSTM model is approximately half the other two models. And, for $m = n = 10$, the average RMSE for the LSTM is one-third of the one from MLP NARX and half of the one from the SimpleRNN model. These results come with a high time-utility trade-off as the LSTM is a more complex architecture that takes advantage of using recurrent layers and nonlinear gating units to model the temporal dynamics of this system. The MLP NARX baseline model does not have any of these advantages and hence resulted in smaller utility. And, even though the SimpleRNN has recurrent layers to keep memories, its architecture is simpler than LSTMs, which resulted in intermediate accuracy.

In general, the baseline MLP NARX model showed inferior data fitting for this task, although its training is much faster than RNNs. For instance, one can see in Table 2 that when increasing the timestep value from $m = n = 1$ to $m = n = 5$, the performance increases by approximately 0.05° on average. However, when increasing it to $m = n = 10$, the performance was not better than $m = n = 5$. This might be due to the input data having more features ($m + n = 20$), which is fed at once and not as a sequence of timestep as in RNNs.

On the other hand, the SimpleRNN model has intermediate results. With the advantage of recurrent layers, this architecture takes advantage of contextual information using self-feedback connections. On average, this model achieved the best RMSE score when using $m = n = 1$. Additionally, it outstands the results of the baseline MLP NARX model by 0.02° in total with the cost of increasing the training time by a factor of 8 approximately.

One can see in Tables 2 and 3 and Figs. 7 and 8 that the identified models indicate good fitting to data and a low RMSE metric for all individuals and timestep values. Even though the testing data have nonideal conditions such as tremors, fatigue, spasms, and voluntary movements, which could be characterized as ill-defined behavior, the proposed NN models identified the dynamic systems very well. In general, the three models achieved the best performance for at least one individual in a given timestep value. For some individuals in which the Corr between input and output data was low (e.g., P1, P2, H1, and H7), the LSTM presented more robustness to identify these systems with lower RMSE. For instance, in Figs. 7 and 8, one can notice that the LSTM well-fits to data especially when using higher timesteps. Even though the LSTM can take more time for training due to a more complex architecture, time and computational costs could be increased if achieving better modeling of systems. For instance, in the methodology proposed in Arcolezi et al. (2020), the identified model is used in an off-line scheme for optimization, which allows having more time for training NN models.

Finally, the use of black-box models such as LSTMs for identifying this system is at first motivated by the advantages of this method itself, and second, by a high power for computation and storage of data encountered nowadays, which will allow continuously acquiring data through rehabilitation sessions. Additionally, NN structures are easier to train in comparison with mathematically modeling the knee joint dynamics and executing experimental tests to measure the parameters of each individual. Focusing more efforts on the model selection procedure and data encoding (adding more timestep values) could provide stronger models for input-output data modeling. Hence, using an appropriate choice of architecture and hyperparameters, the modeling of such dynamics would not be difficult for such recurrent neural networks.

Conclusion

In this paper, the identification of the dynamics between the knee joint position under NMES has been addressed for seven healthy individuals and two SCI ones using NNs. A proper description of the system's dynamics (lower limb response to electrical stimulus) can provide a more realistic simulation of its behavior. NN architectures, as the ones introduced in this paper, can be well-applied to model this system. This will help to develop powerful closed-loop control techniques for NMES systems.

We compared three NN models, namely LSTM, a SimpleRNN, and an MLP (NARX model) for this task. Using the series-parallel structure with three different timestep values ($m = n = [1, 5, 10]$), these NNs could accurately identify the knee angular position due to the electrical stimulus for all individuals. The LSTM model presented outstanding results in comparison with both SimpleRNN and MLP when using more regression terms (timesteps). This higher utility comes with a substantial increment in training time, which one should consider when using this model. The SimpleRNN was the intermediate between LSTM and MLP. With few differences in averaged RMSE, it presented the best results using $m = n = 1$ timestep with 8 times more training time than the MLP. The MLP, which has neither recurrent layers nor a complex architecture, presented worst fitting to data but with fast training.

Recurrent neural networks, in particular the LSTM architecture, are excellent for sequential and time-series data as reported in the literature, which makes them ideal for the identification and modeling of nonlinear systems as our problem. The LSTM is a prospective method for deeper investigation of nonlinear system identification and modeling, as shown in results and discussed in the "Related work" section. Rather than mathematically modeling and making identification tests to acquire parameters for each individual, training NNs is easier. As highlighted in the literature and shown in this paper,

the investigated technique is mature enough to be already used as a model for control purposes. Moreover, saving the data from each rehabilitation session can help these dynamic NNs to improve the system's identification for each patient.

For future work, the real implementation of the investigated dynamic neural networks on control-oriented models is planned. Moreover, we aim to evaluate these models in more rigorous tests such as multi-step prediction (e.g., identifying $[(k+1), \cdots, (k+10)]$) and using the parallel structure (NN model output rather than the real data output). The latter can be future-implemented in an online-learning scheme.

Acknowledgments Computations have been performed on the supercomputer facilities of Mésocentre de Calcul de Franche-Comté. The authors would also like to thank each volunteer who participated in this research, especially those with spinal cord injury.

Funding This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)—Brazil—Finance Code 001, by the Region of Bourgogne Franche-Comté CADRAN Project, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (research productivity scholarships 312170/2018-1 and 309872/2018-9) (CNPq)—Brazil.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Consent to participate All individuals signed written informed consent before participation in the study.

Appendix 1. Complete results

Table 3 presents the identification results for all individuals starting with P1–P2 and ending with H1–H7. This table contains the metrics we discussed in the "Analysis" subsection for each timestep value ($m = n = [1, 5, 10]$) and model (MLP NARX, SimpleRNN, and LSTM). The metrics are the Corr between pulse width and angular position data. The RMSE obtained from the testing set for each model highlighting the best one in italic. And the TT in seconds (s) of each final model.

Table 3 Identification results for all individuals (P1–P2, H1–H7)

Individual	Corr	Timestep	NN model	RMSE (°)	TT (s)
P1	0.4152	$m = n = 1$	MLP	0.1332	21.1
			SimpleRNN	*0.1318*	144.31
			LSTM	0.3051	282.22
		$m = n = 5$	MLP	0.1136	44.46
			SimpleRNN	0.0650	335.05
			LSTM	*0.0496*	1420.76
		$m = n = 10$	MLP	0.2063	54.37
			SimpleRNN	*0.0846*	694.28
			LSTM	0.0958	3218.08
P2	0.1034	$m = n = 1$	MLP	0.1632	19.03
			SimpleRNN	*0.1319*	67.61
			LSTM	0.1857	529.95
		$m = n = 5$	MLP	0.1551	43.06
			SimpleRNN	0.0909	320.41
			LSTM	*0.0893*	757.19
		$m = n = 10$	MLP	0.1288	46.58
			SimpleRNN	0.1007	332.29
			LSTM	*0.0311*	1369.57
H1	0.5907	$m = n = 1$	MLP	0.2411	20.1
			SimpleRNN	0.2527	87.6
			LSTM	*0.1664*	871.42
		$m = n = 5$	MLP	0.2827	48.05
			SimpleRNN	0.1527	267.31

Table 3 (continued)

Individual	Corr	Timestep	NN model	RMSE (°)	TT (s)
			LSTM	*0.0756*	1794.75
		$m = n = 10$	MLP	0.1583	65.34
			SimpleRNN	0.1623	512.60
			LSTM	*0.0438*	2818.26
H2	0.7788	$m = n = 1$	MLP	0.3368	24.30
			SimpleRNN	*0.2708*	94.50
			LSTM	0.2716	279.74
		$m = n = 5$	MLP	*0.0683*	72.47
			SimpleRNN	0.1655	306.41
			LSTM	0.0900	701.90
		$m = n = 10$	MLP	0.2388	67.81
			SimpleRNN	0.1183	580.90
			LSTM	*0.1035*	1135.61
H3	0.8332	$m = n = 1$	MLP	*0.1791*	40.36
			SimpleRNN	0.1837	84.07
			LSTM	0.1839	328.76
		$m = n = 5$	MLP	0.1100	74.11
			SimpleRNN	0.1312	343.45
			LSTM	*0.0562*	672.74
		$m = n = 10$	MLP	0.1327	94.95
			SimpleRNN	0.1690	377.77
			LSTM	*0.0439*	1321.90
H4	0.7181	$m = n = 1$	MLP	*0.1312*	17.26
			SimpleRNN	0.1353	111.17
			LSTM	0.1319	522.71
		$m = n = 5$	MLP	0.0410	92.23
			SimpleRNN	*0.0390*	437.73
			LSTM	0.0506	1356.40
		$m = n = 10$	MLP	0.0525	74.52
			SimpleRNN	0.0497	599.24
			LSTM	*0.0238*	2317.20
H5	0.6737	$m = n = 1$	MLP	*0.1842*	35
			SimpleRNN	0.1805	96.12
			LSTM	0.2008	298.08
		$m = n = 5$	MLP	0.1040	71.60
			SimpleRNN	0.0915	378.19
			LSTM	*0.0518*	1861.90
		$m = n = 10$	MLP	0.0584	57.97
			SimpleRNN	0.0458	740.42
			LSTM	*0.0350*	2420.25
H6	0.7338	$m = n = 1$	MLP	0.1198	22.64
			SimpleRNN	0.1150	199.64
			LSTM	*0.1149*	1244.25
		$m = n = 5$	MLP	0.0576	55.66
			SimpleRNN	0.0945	338.32
			LSTM	*0.0329*	3185.66
		$m = n = 10$	MLP	0.1424	64.60
			SimpleRNN	0.0743	609.58
			LSTM	*0.0301*	3283.61

Table 3 (continued)

Individual	Corr	Timestep	NN model	RMSE (°)	TT (s)
H7	0.5200	$m = n = 1$	MLP	0.1272	18.02
			SimpleRNN	0.1480	103.19
			LSTM	*0.1242*	456.92
		$m = n = 5$	MLP	0.1690	71.82
			SimpleRNN	0.1531	217.90
			LSTM	*0.0466*	2147.72
		$m = n = 10$	MLP	0.1190	56.55
			SimpleRNN	0.1259	315.76
			LSTM	*0.0271*	2013.57

References

Aguirre LA. Introdução à identificação de sistemas: técnicas lineares e não-lineares-teoria e aplicação. 4th ed. Belo Horizonte: Editora UFMG; 2015.

Arcolezi HH, Nunes WRBM, Araujo RA de, Cerna S, Sanches MAA, Teixeira MCM, Carvalho AA de. A robust and intelligent RISE-based control for human lower limb tracking via neuromuscular electrical stimulation. In: XIV Conferência Brasileira de Dinâmica, Controle e Aplicações. 2019. Online. http://soac.eesc.usp.br/index.php/dincon/xivdincon/paper/view/1683/1152. Accessed 11 Aug 2020.

Arcolezi HH, Nunes WRBM, Araujo RA de Cerna S, Sanches MAA, Teixeira MCM, de Carvalho AA. A novel robust and intelligent control based approach for human lower limb rehabilitation via neuromuscular electrical stimulation. arXiv preprint arXiv:2006.15605, 2020.

Benoussaad M, Hayashibe M, Fattal C, Poignet P, Guiraud D. Identification and validation of FES physiological musculoskeletal model in paraplegic subjects. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE. 2009; https://doi.org/10.1109/iembs.2009.5334507.

Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13:281–305.

Cattagni T, Lepers R, Maffiuletti NA. Effects of neuromuscular electrical stimulation on contralateral quadriceps function. J Electromyogr Kinesiol. 2018;38:111–8. https://doi.org/10.1016/j.jelekin.2017.11.013.

Chen, S, Billings, S A, Grant P M. Non-linear system identification using neural networks. International Journal of Control, 51, 1191–1214. 1990.

Chollet F, et al. Keras. 2015. Web. https://keras.io.Online. Accessed 25 Nov 2019

Doucet BM, Lam A, Griffin L. Neuromuscular electrical stimulation for skeletal muscle function. Yale J Biol Med. 2012;85:201–15.

Ferrarin M, Pedotti A. The relationship between electrical stimulus and joint torque: a dynamic model. IEEE Trans Rehabil Eng. 2000;8(3):342–52. https://doi.org/10.1109/86.867876.

Gaino R, Covacic MR, Cardim R, Sanches MAA, Carvalho AA, Biazeto AR, et al. Discrete Takagi-Sugeno fuzzy models applied to control the knee joint movement of paraplegic patients. IEEE Access. 2020;8:32714–26. https://doi.org/10.1109/ACCESS.2020.2971908.

Géron A Hands-on machine learning with scikit-learn and tensor-flow. volume 1. 1 ed., O'Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472. 2017.

Ghani NAM, Kamaruddin SBA, Ramli NM, Nasir NBM, Kader BSBK, Huq MS. The quadriceps muscle of knee joint modelling using neural network approach: Part 1. In: 2016 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), IEEE. 2016; https://doi.org/10.1109/ic3e.2016.8009039.

Gregory CM, Dixon W, Bickel CS. Impact of varying pulse frequency and duration on muscle torque production and fatigue. Muscle Nerve. 2007;35:504–9. https://doi.org/10.1002/mus.20710.

Gobbo M, Maffiuletti NA, Orizio C, Minetto MA. Muscle motor point identification is essential for optimizing neuromuscular electrical stimulation use. J Neuroeng Rehabil. 2014. https://doi.org/10.1186/1743-0003-11-17.

Gonzalez J, Yu W. Non-linear system modeling using LSTM neural networks. IFAC-PapersOnLine. 2018;51:485–9. https://doi.org/10.1016/j.ifacol.2018.07.326.

Grin L, Decker M, Hwang J, Wang B, Kitchen K, Ding Z, et al. Functional electrical stimulation cycling improves body composition, metabolic and neural factors in persons with spinal cord injury. J Electromyogr Kinesiol. 2009;19:614–22. https://doi.org/10.1016/j.jelekin.2008.03.002.

Gulli A, Pal S. Deep learning with Keras. Birmingham: Packt Publishing; 2017.

Laubacher M, Aksöz AE, Riener R, Binder-Macleod S, Hunt KJ. Power output and fatigue properties using spatially distributed sequential stimulation in a dynamic knee extension task. Eur J Appl Physiol. 2017;117:1787–98. https://doi.org/10.1007/s00421-017-3675-0.

Haykin SS. Neural networks and learning machines. 3rd ed. Upper Saddle River: Pearson Education; 2009.

Hirose N, Tajima R. Modeling of rolling friction by recurrent neural network using LSTM, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE. 2017; https://doi.org/10.1109/icra.2017.7989764.

Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Kamaruddin SBA, Ghani NAM, Ramli NM, Nasir NBM, Kader BSBK, Huq MS. The quadriceps muscle of knee joint modelling using neural network approach: part 2. In: 2016 IEEE Conference on Open Systems (ICOS), IEEE. 2016; https://doi.org/10.1109/icos.2016.7881988.

Law LF, Shields R. Mathematical models of human paralyzed muscle after long-term training. J Biomech. 2007;40:2587–95. https://doi.org/10.1016/j.jbiomech.2006.12.015.

Lynch CL, Graham GM, Popovic MR. Including nonideal behavior in simulations of functional electrical stimulation applications. Artif Organs. 2011;35:267–9. https://doi.org/10.1111/j.1525-1594.2011.01218.x.

Lynch CL, Popovic MR. Functional electrical stimulation. IEEE Control Syst. 2008;28:40–50. https://doi.org/10.1109/mcs.2007.914689.

Maffiuletti NA. Physiological and methodological considerations for the use of neuromuscular electrical stimulation. Eur J Appl Physiol. 2010;110:223–34. https://doi.org/10.1007/s00421-010-1502-y.

Narendra K, Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Trans Neural Netw. 1990;1: 4–27. https://doi.org/10.1109/72.80202.

Narendra KS, Mukhopadhyay S. Neural networks for system identification. IFAC Proc. 1997;30:735–42. https://doi.org/10.1016/s1474-6670(17)42933-8.

Nunes WRBM, Teodoro RG, Sanches MAA, de Araujo RA, Teixeira MCM, Carvalho AA. Switched controller applied to functional electrical stimulation of lower limbs under fatigue conditions: a linear analysis. In: Costa-Felix R, Machado J, Alvarenga A, editors. XXVI Brazilian Congress on Biomedical Engineering. IFMBE Proceedings, vol 70/1. Singapore: Springer; 2019. pp. 383–390; https://doi.org/10.1007/978-981-13-2119-1_59.

Ogunmolu OP, Gu X, Jiang SB, Gans NR. Nonlinear systems identification using deep dynamic neural networks. 2016; abs/1610.01439.arXiv:1610.01439.

Ordóñez F J, Roggen D. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors. 16(1). 2016; https://doi.org/10.3390/s16010115

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.

Previdi F. Identification of black-box nonlinear models for lower limb movement control using functional electrical stimulation. Control Eng Pract. 2002;10:91–9. https://doi.org/10.1016/s0967-0661(01)00128-9.

Riener R, Quintern J, Schmidt G. Biomechanical model of the human knee evaluated by neuromuscular stimulation. J Biomech. 1996;29: 1157–67. https://doi.org/10.1016/0021-9290(96)00012-7.

Sanches MAA. Sistema eletrônico para geração e avaliação de movimentos em paraplégicos (D.Eng. dissertation in portuguese). São Paulo: Universidade Estadual Paulista (UNESP); 2013.

Teodoro RG, Nunes WRBM, de Araujo RA, Sanches MAA, Teixeira MCM, de Carvalho AA. Robust switched control design for electrically stimulated lower limbs: a linear model analysis in healthy and spinal cord injured subjects. Control Eng Pract. 2020. https://doi.org/10.1016/j.conengprac.2020.104530.

Wang C, Wu X, Ma Y, Wu G, Luo Y. A flexible lower extremity exoskeleton robot with deep locomotion mode identification. Complexity.1–9. 2018; https://doi.org/10.1155/2018/5712108

Wang Y. A new concept using LSTM neural networks for dynamic system identification. In: 2017 American Control Conference (ACC), IEEE. 2017; https://doi.org/10.23919/acc.2017.7963782