PATIENT TORQUE ESTIMATION FOR OPTIMAL CONTROL OF A ROBOTIC REHABILITATION DEVICE

Leonardo José Consoni, University of São Paulo, consoni_2519@hotmail.com
Guido Gomes Peña, University of São Paulo, doguigomez@hotmail.com
Adriano Almeida Gonçalves Siqueira, University of São Paulo, siqueira@sc.usp.br

Abstract. *Robotic rehabilitation might be a solution for the growing number of people with neuromotor disability. When programming robust control for therapy robots, the large variability in human body dynamics leads developers to leave it as unmodeled uncertainty. Here we propose and test a way to insert patient force estimates in a LQG compensator loop. Identification of a linearized leg model is combined with a Kalman observer to get a full LQG knee movement controller. A multilayer perceptron neural network is trained from reference trajectories and inverse dynamics data to calculate human-generate torque, subtracted from LQG state feedback, so that the controller is more compliant and incites orthosis user activity. Preliminary simulations of an impaired patient are used to validate the complete system, and to compare it to traditional approach. In the end, the more interesting results are discussed and considerations about shortcomings and future developments are made.*

Keywords: *rehabilitation robotics, optimal control, neural networks, kalman filter, system identification*

1. INTRODUCTION

Besides being one of world's major causes of mortality, leading to death about a third of its 15 million annual victims, the cerebrovascular accident (CVA or stroke) is also responsible for residual disability in half of survivors (World Health Organization, 2004), increasingly numerous, due to to improvements in first aid procedures. Spinal cord injuries, by its turn, have an incidence rate of 500 thousand cases each year, and in low- and middle-income countries only 15% of the impaired people have access to required assistive devices (World Health Organization, 2013).

Diseases like those, which compromise one's neuromotor capacity, are a ever growing problem, because of the global trend of population aging and proliferation of related risk factors, such as sedentary lifestyle, diabetes and hypertension. Overloading on health systems, increasingly required by patients in need of rehabilitation services, raises demand for innovative treatments that allow for quality recovery using fewer human and material resources (KREBS et al., 2008).

In view of this and recent technological advances, the last decades have seen an increasing interest in research on the application of robots (as orthoses or end-effectors) in physical rehabilitation. Using a robotic device as controllable interface between therapist and patient saves the first from manual work of guiding the exercised limb and enables finer adjustments in external aid to particular needs of the second (MARCHAL-CRESPO; REINKENSMEYER, 2009; CAO et al., 2014). In addition to the immediate advantages, having a computerized system integrated to the treatment opens the door for application of motivational virtual games, remote communication for home care, recording of data over several sessions to assess performance evolution, among others (GONÇALVES et al., 2014; CARIGNAN; KREBS, 2006).

The most successful approaches to robotic therapy so far are those based on learning: the synchronization of effort and precision during high intensity exercises, in order to stimulate the central nervous system to regain movement control. Although usage of robots, which can work uninterruptedly, facilitates performing a great number of repetitions, simply tracking predefined trajectories does not require any patient collaboration, resulting in only increased joint range of motion without improved coordination (*Slacking Hypothesis*). An appropriate control for the rehabilitation device should provide the minimum help necessary for users to complete motor tasks on their own (*assistance-as-needed* paradigm or AAN) (HOGAN, 2014).

In this article, an experiment with application of optimal torque control on an active knee orthosis is described. A real-time estimation of user activity is combined with the algorithm, in the expectation that it better follows an AAN criterion and promotes motor relearning.

The text is organized as follows: Section 2 presents the robotic platform; Section 3 shows how data collected from preliminary tests were processed to assist in system identification; in Section 4 that information is used to estimate a linearized plant model; Section 5 describes the optimal control architecture; Section 6 details the tools used in training an artificial neural network to calculate patient-generated torque; finally, in Sections 7 and 8, simulations of an impaired user are performed, to compare results obtained with and without network estimations, and some considerations about further improvements are made.

2. ROBOTIC REHABILITATION PLATFORM

Among the equipment available at the Rehabilitation Robotics Laboratory (ReRob) from São Carlos School of Engineering (EESC-USP) is the active knee orthosis, developed in Santos e Siqueira (2014). Based on a Rotational Series Elastic Actuator (rSEA), it allows to take measurements of interaction torque between its effector and the load (flexed leg). Such readings are necessary to implement compliant controllers, more suitable for robots acting as a human-machine interfaces (HMI) (HOGAN, 2014).

The control hardware for this device is comprised of *Maxom Motor EPOS* DC motor drives and a *National Instruments PXIe-8115* embedded computer running the code on a real-time operating system (*Phar Lap ETS*). In addition, a desktop with network access can be added to run graphical user interfaces (GUIs) for monitoring or direct operation (Fig. (1)).

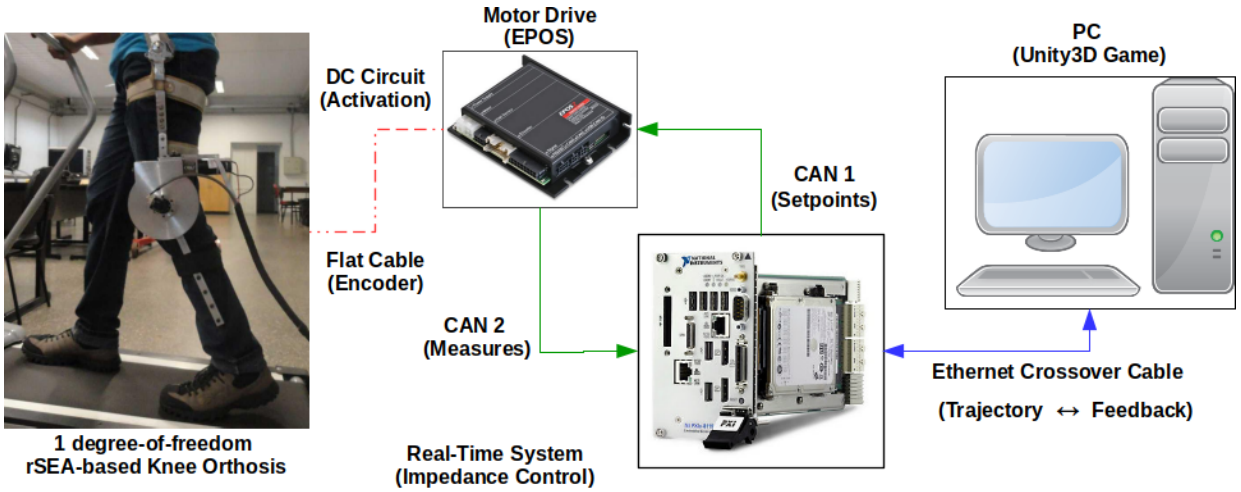


Figure 1: Diagram showing hardware connections used for controlling the active knee orthosis, on the left (adapted from Consoni et al. (2016))

Being performed via software, developed in Consoni et al. (2016), the orthosis control is discrete-time type (so are all its equations), with 5 milliseconds update step Δt. Inside that time frame, each processing iteration, including file recording of related variables (when set for any of them), must be completed to ensure real-time performance.

3. DATA ACQUISITION AND PREPROCESSING

In order to adapt therapy for each patient physical characteristics, the system operation has a preliminary phase, when it attempts to map limb dynamics. Equation (1), adapted from Mefoued, Mohammed e Amirat (2011), is the reference model for the case studied here: person in sitting down position, moving shank (of mass m_s and length l_s) around knee joint (with damping b_k and kinetic dry friction f_k^c) in the sagittal plane, from $0°$ (lowest point) to $90°$ (highest point).

$$\tau_r + \tau_p = j_s\ddot{\theta} + b_k\dot{\theta} + f_k^c sign(\dot{\theta}) + w_s sin(\theta)\ , \quad \text{where } j_s \approx \frac{m_s l_s^2}{3},\ w_s \approx \frac{m_s g l_s}{2},\ \tau_p = f(\theta^d, \theta, \dot{\theta}, \ddot{\theta}, \tau_r) \tag{1}$$

For that, values of desired position θ^d, effective position θ and interaction torque τ_r of the elastic actuator, indexed by time, are registered over 3 sampling periods, during which the user has to follow a reference sinusoidal movement, displayed on a GUI. Here, data was collected from a healthy subject performing the task.

3.1 Provisional Control for Data Acquisition

As the knee orthosis is not backdriveable, some sort of actuation is required to enable its movement. So, initially, a basic virtual impedance control (JARDIM; SIQUEIRA, 2013) is applied. For the first period, the knee orthosis is set to a fully compliant (zero impedance) behavior, that lets users track the reference as much as they are able to. Over the last two time windows, the robot provides variant assistance and random disturbances are inserted at some points.

3.2 Kalman Filtering

For kinematic data sampling in that setup, it is only possible to reliably measure angular position θ, although velocity and acceleration are also needed. Instead of simply performing a double discrete derivative to get those values, a Kalman filter is used to estimate smoother signals $\hat{\theta}$, $\hat{\dot{\theta}}$ and $\hat{\ddot{\theta}}$. The general formulas (for state $\hat{x}$ calculation and Kalman gain $\mathbf{K}$ adjustment) presented in Faragher (2012) were adapted to this use case as in Eq. (2).

$$\hat{x}_{i+1} = F\hat{x}_i + K_{i+1}(y_i - HF\hat{x}_i)\,, \quad \text{where } F = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix},\ H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},\ \hat{x} = \begin{bmatrix} \hat{\theta} \\ \hat{\dot{\theta}} \\ \hat{\ddot{\theta}} \end{bmatrix},\ y = \theta \tag{2}$$

3.3 Inverse Dynamics

The last measurement necessary for motion analysis is patient-generated torque τ_p, which could only be obtained directly using electromyographic (EMG) readings and a muscle contraction equation, as in Hassani, Mohammed e Amirat (2013). As an alternative, here inverse dynamics (ID) is used to get it indirectly on *OpenSim* (SETH et al., 2011), a set of software libraries (available for *C++*, *Python*, *Java* and *MATLAB*) for human mechanics simulation.

With that framework, a base musculoskeletal model, provided as XML file, can be scaled to fit a given person anthropometric values. Then, taking the adjusted parameters, *OpenSim* ID solver calculates internal joint torques for a given kinematic state and list of external loads.

4. PLANT IDENTIFICATION

Considering that a MIMO (multiple input, multiple output) solution, applicable to multi-joint control, is being developed, a state-space (matrix) representation of the system is necessary. Equation (3) shows a standard linear model for plant observation, where matrices **A**, **B** and **C** determine the relationships between the state **x** variables and the input **u** and output **y**, accounting for unstructured (stochastic) uncertainties **v** (processing error) and **w** (measurement error).

$$x_{i+1} = Ax_i + Bu_i + v_i\,, \quad y_i = Cx_i + w_i \quad \left(\text{here, } x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix},\ y = \begin{bmatrix} \theta \end{bmatrix},\ u = \begin{bmatrix} \tau_r + \tau_p \end{bmatrix} \right) \tag{3}$$

However, the nonlinearities of Eq. 1 would not fit this formulation. A linearized variant of it, as in Eq. (4), must be used to approximate the original one, at the cost of greater state prediction errors.

$$\tau_r + \tau_p = \alpha\ddot{\theta} + \beta\dot{\theta} + \gamma\theta \implies A = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ -\gamma/\alpha & -\beta/\alpha & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 0 \\ 1/\alpha \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4}$$

Having done that, the preprocessed list of values from the first sampling period, described in Section 3., is used in a linear least squares solver (like *Python*'s *numpy.linalg.lstsq*) to estimate parameters α, β and γ (Tab. (1)).

Table 1: Unconstrained least squares coefficients and average residual for nonlinear and linear formulations

	$\ddot{\theta}$ coeff.	$\dot{\theta}$ coeff.	sign($\dot{\theta}$) coeff.	sin(θ) coeff.	θ coeff.	avg. residual
nonlinear equation	$j_s = 6.217$	$b_k = -2.637$	$f_k^c = 0.154$	$w_s = 13.485$	0	0.178
linear equation	$\alpha = 13.432$	$\beta = -5.824$	0	0	$\gamma = 10.363$	1.510

5. CONTROL ARCHITECTURE

As the control of the orthosis must simultaneously keep reference position **r** tracking (small error) by the patient and reduced input torque **u**, it involves an optimization problem: the minimization of a cost function with two inputs. In Linear–quadratic–Gaussian (LQG) "cheap" control, this cost **J** is defined as in Eq. (5), where **Q** and **R** are symmetrical weighting matrices for manual adjustment (one could be made constant, as only the ratio between them matters).

$$J = \sum_{i=0}^{N} \left(z_i^T Q z_i + u_i^T R u_i \right), \quad \text{where } Q = C'C,\ R = \rho I\ (\rho > 0) \tag{5}$$

The full discrete optimal controller equations are a combination of state-space plant model, internal state **z** observer (Kalman filter) and compensator (feedback gain) (Eq. (6)). Although not intrinsically robust, the LQG method leaves room for stability optimization through the LTR process (Loop Transfer Recovery), that for "cheap" control consists of lowering ρ to near 0 (SKOGESTAD; POSTLETHWAITE, 2005).

$$z_{i+1} = Az_i + Bu_i + K_{i+1}(y_i - r_i - C(Az_i + Bu_i)) \,, \quad u_i = -(B^T X B + R)^{-1} B^T X A z_i \tag{6}$$

Matrix **X** is the solution to discrete-time algebraic Ricatti equation (Eq. (7)), also available in *Python* (function *scipy.linalg.solve_discrete_are*).

$$A^T X A - X - (A^T X B)(R + B^T X B)^{-1}(B^T X A) + Q = 0 \tag{7}$$

6. NEURAL NETWORK TORQUE CALCULATION

Up to this point, the approach taken follows what is commonly done for robust control of rehabilitation devices, which treat human interaction as unknown input disturbance (SANTOS; SIQUEIRA, 2014; JUTINICO et al., 2017). The proposed improvement consists of subtracting user torque estimates $\hat{\tau}_p$ from state feedback control input **u**, in order to lower actuator work when the patient is able to partially track a given trajectory.

OpenSim ID solver, however, can not be used to simulate the person behavior, as it does not take reference positions into account. Besides, its algorithm is slow for required time constraints and its libraries are not easily integrated in a real-time system.

As a solution to model that relationship and speed up processing, a multilayer perceptron (MLP) neural network with improved learning rate (SUTSKEVER et al., 2013) is employed. Data from second and third sampling routines are used for MLP training and validation, respectively. As the appropriate amount of samples and hidden neurons is unknown, a discrete optimization process searches for a suitable combination to minimize a fitness function, as in Alg. (1).

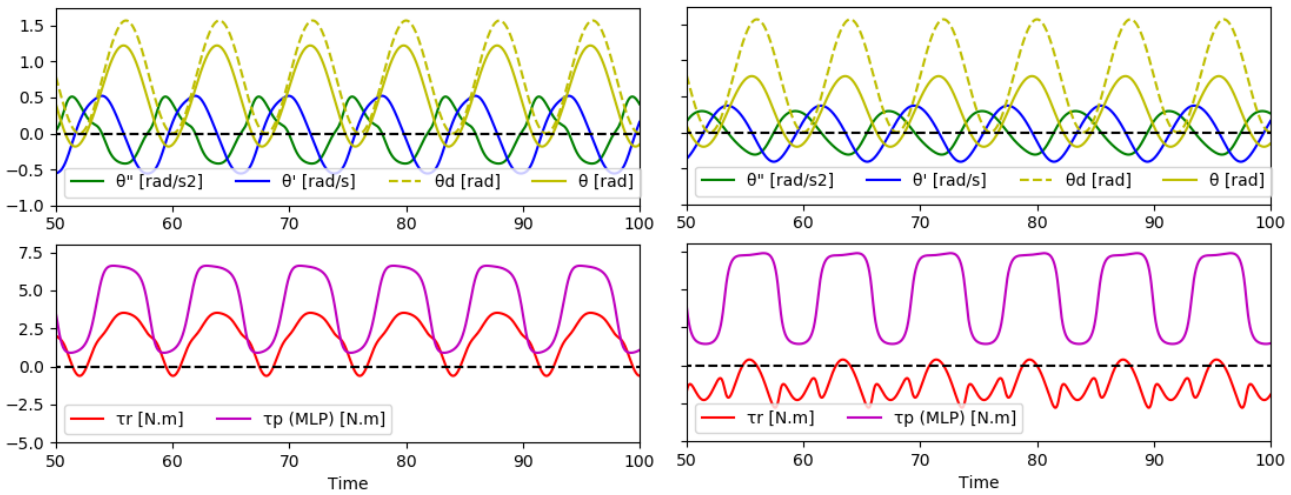
Algorithm 1: Code for perceptron optimization (Python)

```
bestNeurons=10; nStep=5; bestSamples=10000; sStep=5000
bestFitness = 1.0
# iterative optimization loop
isRunning = True
while isRunning:
  # generate 8 new points around last best fit one
  neuronsList = [ bestNeurons+nStep, bestNeurons, bestNeurons-nStep ]
  samplesList = [ bestSamples+sStep, bestSamples, bestSamples-sStep ]

  oldBestFitness = bestFitness
  for neuronsNumber in neuronsList:
    for samplesNumber in samplesList:
      # call MLP training and validation routines
      trainingError = TrainMLP( neuronsNumber, samplesNumber )
      validationError = ValidateMLP( neuronsNumber, samplesNumber )
      fitness = trainingError + 0.5 * validationError
      # update fittest MLP training properties
      if fitness < bestFitness:
        bestFitness = fitness
        bestNeurons = neuronsNumber
        bestSamples = samplesNumber
  # optimization loop stop condition
  if abs( bestFitness - oldBestFitness ) < 10e-3: isRunning = False
  # update steps to their half greater nearest integer
  nStep = ceil( nStep / 2 )
  sStep = ceil( sStep / 2 )
```

7. SIMULATION RESULTS

To try getting a better understanding of how the control system would behave with an actual impaired patient, 2 simulations were run reducing MLP-estimated $\hat{\tau}_p$ (equal to the effective one τ_p for this virtual subject), by 40% (KELLER; ENGELHARDT, 2013). Reference trajectory was a generated sinusoidal curve with 8s period. During the first simulated activity, $\hat{\tau}_p$ is only monitored, and not subtracted from control input **u**. In the second, calculated values enter the control loop. The actual plant was replaced by the nonlinear model (Eq. (1)) in order to reproduce more reliable results (Fig. (2)).



(a) (b)

Figure 2: Motion outputs (on top) and robot and patient inputs (below) for simulated assisted activity when (a) user torque is taken as disturbance and (b) accounted as control action. The stable LQG compensator used $\rho = 0.00001$

Despite lower actuator torques, human ones remained mostly the same, causing larger trajectory deviation. However, as the main concern is stimulating synchrony between effort and desired (not actual) motion, another evaluation metric is necessary. Summation of the pointwise product of $\hat{\tau}_p$ and position error e_y $(= \theta^d - \theta)$ sign (always pointing the correct direction) was chosen as a preliminary performance assessment. Table (2) present that and other measurements.

Table 2: Patient performance measurements for N=50000 samples of both simulation runs

	$\sqrt{\frac{\sum e_y^2}{N}}$ [rad]	$\sqrt{\frac{\sum \tau_p^2}{N}}$ [N.m]	Work: $\sum \tau_p \cdot \dot{\theta} \Delta t$ [J]	Performance index: $\sum \tau_p \cdot sign(e_y)$
Simulation 1	0.2834	4.7352	4112.98	207009 (-325084 at 100% τ_p)
Simulation 2	0.5663	5.4934	-4312.07	244021 (494985 at 100% τ_p)

As depicted, in simulation 2 the virtual patient produced slightly higher torques, but great part was spent on negative work, resisting actual movement (probably due to unsustainable leg's weight). On the other hand, higher performance index possibly indicates an extra amount of ability of reacting to the task which is not used during simulation 1.

8. CONCLUSIONS

This work experimented with aspects of optimal and robust control (LQG, H_∞, etc.) for human-robot interaction where they seem to be lacking. Making no considerations about external loads makes sense to keep the algorithm as generic and adaptable as possible, but in the case of rehabilitation, in controlled environments, at least a black-box (no defined structure) approach could be employed to preadapt models for more performance.

The enhancement proposed here shows an impact worth looking at, and it is far from fully developed, as room for improvement is clear. The usage of time-invariant matrices limits range of operation, as state prediction error raises considerably for larger joint angles, making online recalculation of **A** and **B** matrices an interesting next step. Also, EMG measurements could be compared or even combined with current torque estimates, in order to fine tune it. Finally, real tests on healthy and mainly impaired subjects will be essential to confirm the whole strategy viability.

9. REFERENCES

CAO, J. et al. Control strategies for effective robot assisted gait rehabilitation: the state of art and future prospects. *Medical Engineering & Physics*, v. 36, n. 12, p. 1555–66, 2014. ISSN 1873-4030.

CARIGNAN, C. R.; KREBS, H. I. Telerehabilitation robotics: Bright lights, big future? *The Journal of Rehabilitation Research and Development*, v. 43, n. 5, p. 695, 2006. ISSN 0748-7711.

CONSONI, L. J. et al. A robotic telerehabilitation game system for multiplayer activities. In: *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*. Singapore: IEEE, 2016. v. 2016-July, p. 798–803. ISBN 9781509032877. ISSN 21551774.

FARAGHER, R. *Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]*. 2012. 128–132 p.

GONÇALVES, A. C. B. F. et al. Serious games for assessment and rehabilitation of ankle movements. In: *IEEE 3rd International Conference on Serious Games and Applications for Health*. Rio de Janeiro: IEEE, 2014. p. 1–6. ISBN 9781479948239.

HASSANI, W.; MOHAMMED, S.; AMIRAT, Y. Real-Time EMG Driven Lower Limb Actuated Orthosis for Assistance As Needed Movement Strategy. *Robotics: Science and Systems*, 2013.

HOGAN, N. Robot-Aided Neuro-Recovery. *Mechanical Engineering*, v. 136, n. 9, p. S3–S5, 2014.

JARDIM, B.; SIQUEIRA, A. a. G. Development and analysis of series elastic actuators for impedance control of an active ankle–foot orthosis. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 36, n. 3, p. 501–510, 2013. ISSN 1678-5878.

JUTINICO, A. L. et al. Impedance control for robotic rehabilitation: A robust markovian approach. *Frontiers in Neurorobotics*, v. 11, n. AUG, p. 1–16, 2017. ISSN 16625218.

KELLER, K.; ENGELHARDT, M. Strength and muscle mass loss with aging process. Age and strength loss. *Muscles, Ligaments and Tendons Journal*, v. 3, n. 4, p. 346–350, 2013. ISSN 22404554.

KREBS, H. I. et al. A paradigm shift for rehabilitation robotics. *IEEE Engineering in Medicine and Biology Magazine*, v. 27, n. 4, p. 61–70, 2008. ISSN 07395175.

MARCHAL-CRESPO, L.; REINKENSMEYER, D. J. Review of control strategies for robotic movement training after neurologic injury. *Journal of Neuroengineering and Rehabilitation*, v. 6, n. 1, p. 20, 2009. ISSN 1743-0003.

MEFOUED, S.; MOHAMMED, S.; AMIRAT, Y. Knee joint movement assistance through robust control of an actuated orthosis. *IEEE International Conference on Intelligent Robots and Systems*, p. 1749–1754, 2011. ISSN 2153-0858.

SANTOS, W. M. dos; SIQUEIRA, A. a. G. Robust Torque Control Based on H-infinity criterion of an Active Knee Orthosis. In: *IEEE International Conference on Biomedical Robotics and Biomechatronics*. [S.l.: s.n.], 2014. p. 644–649. ISBN 9781479931279.

SETH, A. et al. OpenSim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange. *Procedia IUTAM*, v. 2, p. 212–232, 1 2011. ISSN 2210-9838.

SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable feedback control: analysis and design*. [S.l.: s.n.], 2005. 575 p. ISSN 1049-8923. ISBN 978-0470011683.

SUTSKEVER, I. et al. On the importance of initialization and momentum in deep learning. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, n. 2010, p. 8609–8613, 2013. ISSN 15206149.

World Health Organization. *Global burden of stroke*. 2004. 50 - 51 p.

World Health Organization. International Perspectives on Spinal Cord Injury. *The Lancet*, p. 4, 2013.

10. ACKNOWLEDGMENTS

This research is supported by National Council for Scientific and Technological Development (CNPq) under grant 141058/2017-0.

11. RESPONSIBILITY FOR INFORMATION

The authors are solely responsible for the information included in this work.