Multiple Strategies for Parameter Estimation via a Hybrid Method: a Comparative Study

Teresa Mendonça *, Hugo Alonso *, Hugo Magalhães * and Paula Rocha +
*Departamento de Matemática Aplicada
Faculdade de Ciências da Universidade do Porto
Rua do Campo Alegre, 687, 4169-007 Porto, Portugal
Phone: +351-220100871, Fax: +351-220100809
email:{tmendo,hugo.alonso,hfmagalh}@fc.up.pt
+Departamento de Matemática
Universidade de Aveiro
Campo de Santiago, 3810-193 Aveiro, Portugal
Phone: +351-234370359, Fax: +351-234382014
email: procha@mat.ua.pt

ABSTRACT: In this paper, two different strategies are considered for application of a previously proposed hybrid method designed to model parameter estimation. This method combines the neural networks ability to produce initial parameter estimates close to the real values with the fast convergence of the Levenberg-Marquardt method using such estimates. The first strategy is of general applicability, while the second one is intended for models having a structure defined by various blocks in series. The neuromuscular blockade of patients undergoing general anaesthesia is taken as the case study for comparing the performances of the two strategies.

KEYWORD: Levenberg-Marquardt Method, Neural Networks, Parameter Estimation, Pharmacokinetic-Pharmacodynamic Model, Atracurium.

INTRODUCTION

In practical applications, classical estimation techniques often diverge or do not converge to the real values of model parameters. This is mainly due to their iterative nature, high sensitivity to the choice of parameters involved in their formulation and to the initial guesses for the parameters to be estimated. In methods like the Levenberg-Marquardt optimization method (LM) [1], [2], a guaranteed and fast convergence to the real values of the parameters can however be accomplished if their initial guesses are sufficiently close. These guesses are usually taken from a finite set of known realizations for the parameters, so that some measure of distance between observed and computed model outputs is minimized. Even if such set is defined, knowing the statistical distribution for the parameters, one can never be sure of its “richness” to guarantee the convergence to the real values when curve fitting is carried out by application of the LM method. In order to overcome this problem and to improve the speed of convergence, it has been proposed in [3] that initial guesses should be given by an artificial neural network (ANN) able to produce estimates closer to real values than the ones in the set. This combination represents a Hybrid Method (HM) between a non-iterative – ANN – and an iterative – LM. It was shown to have better performance than a separate traditional application of LM and ANN, methods having comparable performances.

In this paper, two different strategies for applying the HM are compared. Neuromuscular blockade model parameter estimation is used as a case study.

In the sequel, notation and relevant concepts that will be used throughout the paper are introduced.

All input-output model data are supposed to be free from noise. Assume that the model structure and the input u as a real function of time are known. Therefore, the output y can be described as

$$y(\boldsymbol{\Theta},t) = f_{\boldsymbol{\Theta}}(t), \tag{1}$$

i.e., by means of a parametric real function of time, where $\boldsymbol{\Theta} \in \mathbb{R}^{p\times 1}$ is the vector of the p model parameters having a not necessarily known statistical distribution. Furthermore, suppose that $f_{\boldsymbol{\Theta}}$ is at least twice continuously differentiable with respect to $\boldsymbol{\Theta}$.

For some realization $\boldsymbol{\theta}$ of $\boldsymbol{\Theta}$, let

$$\mathbf{y}(\boldsymbol{\theta}) = (y_1,\ldots,y_n)^T(\boldsymbol{\theta}) = \left(y(\boldsymbol{\theta},t_1),\ldots,y(\boldsymbol{\theta},t_n)\right)^T \tag{2}$$

denote the corresponding vector of n observations of the model output at time instants $t_1,\ldots,t_n$ such that $0 \le t_1 < \ldots < t_n$. As is the general case, consider that $n \ge p \ge 1$. The sensitivity of the observations to the parameters is given by the matrix

$$\mathbf{S}(\boldsymbol{\theta}) = \left(\mathbf{S}(\boldsymbol{\theta})_{ij}\right)_{i=1,\ldots,n,\, j=1,\ldots,p}, \tag{3}$$

where, for $i=1,\ldots,n, j=1,\ldots,p$,

$$\mathbf{S}(\boldsymbol{\theta})_{ij} = \frac{\partial y_i}{\partial \theta_j}(\boldsymbol{\theta}). \tag{4}$$

Assume first that the probability density function $p_{\boldsymbol{\Theta}}$ of $\boldsymbol{\Theta}$ is known. Then, define the observation covariance matrix as

$$\boldsymbol{\Sigma} = E_{\boldsymbol{\Theta}}\left[\left(\mathbf{y}(\boldsymbol{\Theta}) - \boldsymbol{\mu}_{\mathbf{y}}\right)\left(\mathbf{y}(\boldsymbol{\Theta}) - \boldsymbol{\mu}_{\mathbf{y}}\right)^T\right], \tag{5}$$

where $E_{\boldsymbol{\Theta}}[\bullet] = \int_{D_{\boldsymbol{\Theta}}} \bullet\, p_{\boldsymbol{\Theta}}(\boldsymbol{\theta})d\boldsymbol{\theta}$ represents the mathematical expectation over the parameter space $D_{\boldsymbol{\Theta}}$, and $\boldsymbol{\mu}_{\mathbf{y}} = E_{\boldsymbol{\Theta}}\left[\mathbf{y}(\boldsymbol{\Theta})\right]$ represents the mean observation vector. Additionally, define the vector of the p first principal components of $\mathbf{y}(\boldsymbol{\theta})$ as

$$\mathbf{z}(\boldsymbol{\theta}) = \mathbf{V}^T\left(\mathbf{y}(\boldsymbol{\theta}) - \boldsymbol{\mu}_{\mathbf{y}}\right), \tag{6}$$

where $\mathbf{V} \in \mathbb{R}^{n\times p}$ is a matrix whose orthonormal columns are eigenvectors corresponding to the p largest eigenvalues of $\boldsymbol{\Sigma}$ [4].

If $p_{\boldsymbol{\Theta}}$ is not known, assume that a sample $\{\boldsymbol{\theta}_s\}_{s=1}^N$ of size N, exhaustively covering $D_{\boldsymbol{\Theta}}$, is available. Then, $\boldsymbol{\Sigma}$ is replaced by the sample observation covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{s=1}^{N}\left(\mathbf{y}(\boldsymbol{\theta}_s) - \overline{\mathbf{y}}\right)\left(\mathbf{y}(\boldsymbol{\theta}_s) - \overline{\mathbf{y}}\right)^T, \tag{7}$$

where $\overline{\mathbf{y}} = \frac{1}{N}\sum_{s=1}^{N}\mathbf{y}(\boldsymbol{\theta}_s)$ represents the sample mean observation vector. It follows that $\mathbf{z}$ gives place to

$$\hat{\mathbf{z}}(\boldsymbol{\theta}) = \widehat{\mathbf{V}}^T\left(\mathbf{y}(\boldsymbol{\theta}) - \overline{\mathbf{y}}\right), \tag{8}$$

as $\mathbf{V}$ is replaced by a $\widehat{\mathbf{V}}$ derived from (7), and $\boldsymbol{\mu}_{\mathbf{y}}$ is replaced by $\overline{\mathbf{y}}$.

The problem of parameter estimation is that of determining an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ from $\mathbf{y}(\boldsymbol{\theta})$ or $\mathbf{z}(\boldsymbol{\theta})$ ($\hat{\mathbf{z}}(\boldsymbol{\theta})$, if $\mathbf{z}(\boldsymbol{\theta})$ is not available).

METHODS

This section presents a brief overview of the Levenberg-Marquardt Method and Artificial Neural Networks, and a description of the two proposed strategies for applying the Hybrid Method.

THE LEVENBERG-MARQUARDT METHOD

The problem of parameter estimation can be viewed as an optimization one described by

$$\min_{\hat{\boldsymbol{\theta}}}\left\{E_{\mathrm{y}}(\hat{\boldsymbol{\theta}})=e_{\mathrm{y}}^{T}(\hat{\boldsymbol{\theta}})e_{\mathrm{y}}(\hat{\boldsymbol{\theta}})\right\}, \tag{9}$$

where

$$e_{\mathrm{y}}(\hat{\boldsymbol{\theta}})=(e_1,\ldots,e_n)_{\mathrm{y}}^{T}(\hat{\boldsymbol{\theta}})=\mathbf{y}(\boldsymbol{\theta})-\mathbf{y}(\hat{\boldsymbol{\theta}}), \tag{10}$$

and $\hat{\boldsymbol{\theta}}$ is an estimate of $\boldsymbol{\theta}$. The minimization of E_{y} in $\hat{\boldsymbol{\theta}}$ can be reduced to a linear problem in $\Delta\hat{\boldsymbol{\theta}}^k=\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}^k$ for the current k-th estimate $\hat{\boldsymbol{\theta}}^k$ by first expanding e_{y} in a Taylor series about $\hat{\boldsymbol{\theta}}^k$ and ignoring nonlinear terms as

$$e_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k)\cong e_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)+J(\hat{\boldsymbol{\theta}}^k)\Delta\hat{\boldsymbol{\theta}}^k, \tag{11}$$

where

$$J(\hat{\boldsymbol{\theta}}^k)=\left(\frac{\partial e_{i,\mathrm{y}}}{\partial\hat{\theta}_j}(\hat{\boldsymbol{\theta}}^k)\right)_{i=1,\ldots,n,\,j=1,\ldots,p} \tag{12}$$

is the Jacobian matrix of e_{y} evaluated at $\hat{\boldsymbol{\theta}}^k$. Then, replacing (11) in the definition of E_{y} in (9) to obtain a quadratic approximation of E_{y} around $\hat{\boldsymbol{\theta}}^k$ as

$$\begin{aligned}E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k)\ \cong\ \tilde{E}_{\mathrm{y}}(\Delta\hat{\boldsymbol{\theta}}^k)\ =\ & e_{\mathrm{y}}^{T}(\hat{\boldsymbol{\theta}}^k)e_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)+2e_{\mathrm{y}}^{T}(\hat{\boldsymbol{\theta}}^k)J(\hat{\boldsymbol{\theta}}^k)\Delta\hat{\boldsymbol{\theta}}^k+\\ & +\Delta\hat{\boldsymbol{\theta}}^{k^T}J^{T}(\hat{\boldsymbol{\theta}}^k)J(\hat{\boldsymbol{\theta}}^k)\Delta\hat{\boldsymbol{\theta}}^k,\end{aligned} \tag{13}$$

the linear problem arises from finding a minimizer $\Delta\hat{\boldsymbol{\theta}}^k$ for $\tilde{E}_{\mathrm{y}}$, which amounts to the solution of the normal equations

$$J^{T}(\hat{\boldsymbol{\theta}}^k)J(\hat{\boldsymbol{\theta}}^k)\Delta\hat{\boldsymbol{\theta}}^k=-J^{T}(\hat{\boldsymbol{\theta}}^k)e_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k), \tag{14}$$

derived from $\nabla\tilde{E}_{\mathrm{y}}(\Delta\hat{\boldsymbol{\theta}}^k)=0$. This establishes the basis for the Gauss method, resumed in the following algorithm:

1. Compute $J(\hat{\boldsymbol{\theta}}^k)$;
2. Compute $\Delta\hat{\boldsymbol{\theta}}^k$ from equation (14);
3. Set $\hat{\boldsymbol{\theta}}^{k+1}=\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k$, $k=k+1$;
4. Repeat from 1 until a stopping criterion is met.

The Gauss method is descendent, in the sense that there exists a step $\Delta\hat{\boldsymbol{\theta}}^k$ for which

$$E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k)=E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)+\Delta\hat{\boldsymbol{\theta}}^{k^T}\nabla E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)+\mathrm{O}(\Delta\hat{\boldsymbol{\theta}}^{k^T}\Delta\hat{\boldsymbol{\theta}}^k)<E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k), \tag{15}$$

since it can easily be seen that $\Delta\hat{\boldsymbol{\theta}}^{k^T}\nabla E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)<0$ for all $\Delta\hat{\boldsymbol{\theta}}^k$. Nevertheless, this method exhibits a poor performance and is potentially divergent whenever $J^{T}(\hat{\boldsymbol{\theta}}^k)J(\hat{\boldsymbol{\theta}}^k)$ is ill-conditioned, *i.e.*, when the ratio between maximum and minimum eigenvalues of this matrix is too large. Levenberg [1] and Marquardt [2] solved this problem by introducing an additional term in the normal equations (14), thus resulting in the augmented normal equations

$$(J^T(\hat{\boldsymbol{\theta}}^k)J(\hat{\boldsymbol{\theta}}^k)+\lambda_k I)\Delta\hat{\boldsymbol{\theta}}^k=-J^T(\hat{\boldsymbol{\theta}}^k)e_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k). \quad (16)$$

The choice of the value for λ can be controlled in each iteration by the gain ratio

$$\gamma=\frac{E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)-E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k)}{\tilde{E}_{\mathrm{y}}(0)-\tilde{E}_{\mathrm{y}}(\Delta\hat{\boldsymbol{\theta}}^k)}. \quad (17)$$

Note that $\tilde{E}_{\mathrm{y}}(0)=E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)$ and $\tilde{E}_{\mathrm{y}}(0)-\tilde{E}_{\mathrm{y}}(\Delta\hat{\boldsymbol{\theta}}^k)>0$. Therefore, a large value of γ indicates that $\tilde{E}_{\mathrm{y}}(\Delta\hat{\boldsymbol{\theta}}^k)$ is a good approximation to $E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k)$, and λ can be decreased so that the next Levenberg-Marquardt step is closer to the Gauss step. On the contrary, if γ is small (maybe even negative) then $\tilde{E}_{\mathrm{y}}(\Delta\hat{\boldsymbol{\theta}}^k)$ is a poor approximation to $E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k+\Delta\hat{\boldsymbol{\theta}}^k)$, and λ should be increased with the aim of getting a step closer to the steepest descent direction (determined by $-\nabla\tilde{E}_{\mathrm{y}}(0)=-\nabla E_{\mathrm{y}}(\hat{\boldsymbol{\theta}}^k)$) and of reduced length.

The stopping criterion for the described optimization methods can be the satisfaction of at least one of the following conditions:

- $\left\|\nabla\tilde{E}_{\mathrm{y}}(0)\right\|\le\varepsilon_1$;
- $\left\|\hat{\boldsymbol{\theta}}^k-\hat{\boldsymbol{\theta}}^{k-1}\right\|\le\varepsilon_2\left(\left\|\hat{\boldsymbol{\theta}}^{k-1}\right\|+\varepsilon_2\right)$;
- $k\ge k_{\max}$;

where $\varepsilon_1,\varepsilon_2\in\mathbb{R}^+$ and $k_{\max}\in\mathbb{Z}^+$ are chosen by the user. For further details refer to, *e.g.*, [5].

ARTIFICIAL NEURAL NETWORKS

The purpose here is to describe the use of neural networks for estimation of realizations $\boldsymbol{\theta}$ of the parameter vector either from $\mathbf{y}(\boldsymbol{\theta})$ or $\mathbf{z}(\boldsymbol{\theta})$ ($\hat{\mathbf{z}}(\boldsymbol{\theta})$, if $\mathbf{z}(\boldsymbol{\theta})$ is not available). Hence, a brief overview of the relevant concepts is given. For further details on neural networks refer to, *e.g.*, [4].

The neuron, the neural network elementary unit, is capable of receiving, processing and transmitting data. The common representation of the neuron function is

$$\varphi_{\mathbf{w}}(\mathbf{x})=\sigma(\mathbf{w}^T\tilde{\mathbf{x}}), \quad (18)$$

where $\mathbf{x}=(x_1,\dots,x_n)^T\in\mathbb{R}^n$ is the vector of neuron inputs, and $\mathbf{w}=(w_0,w_1,\dots,w_n)^T\in\mathbb{R}^{n+1}$ is the vector of neuron parameters weighting $\tilde{\mathbf{x}}=(1,\mathbf{x}^T)^T$ to produce the neuron output $\varphi_{\mathbf{w}}(\mathbf{x})$ through the activation function σ. The standard logistic sigmoidal function is the usual choice for σ, *i.e.*,

$$\sigma(\xi)=\frac{1}{1+e^{-\xi}},\xi\in\mathbb{R}. \quad (19)$$

Note that this is a C^∞ function and hence so is $\varphi_{\mathbf{w}}$.

A neural network is an intricate set of neurons connected in such a way that the output of a neuron is generally an input to other ones. The number of neurons and the way how they are interconnected determine the neural network architecture. Multilayer feed-forward neural networks, a class of neural networks having an architecture defined by a directed acyclic graph, are the only ones here considered. In such networks, the neurons are disjointly split into ordered layers, the connections among neurons lead only from lower layers to upper ones, and each neuron in one layer is connected to all neurons in the next one. The input, hidden or intermediate, and output neurons belong to the first, intermediate, and last layers, respectively. The function of the input neurons is the identity function. Thus, for p output neurons, the neural network function is

$$\phi_{\mathbf{w}}(\mathbf{x}) = \left(\phi_l, \ldots, \phi_p\right)_{\mathbf{w}}(\mathbf{x}), \tag{20}$$

where, for $l = 1, \ldots, p$,

$$\phi_{l,\mathbf{w}}(\mathbf{x}) = \sigma_l\left(w_{l0} + \sum_k w_{lk}\sigma_k\left(\ldots\left(\sum_i w_{ji}\tilde{x}_i\right)\ldots\right)\right), \tag{21}$$

being w_{ji} the weight of the connection from the *i*-th to the *j*-th neurons, σ_k the activation function of the *k*-th neuron, etc. Assume that the activation functions of all the hidden neurons are defined as in (19), and that σ_l is the identity, $\forall l$. Therefore, $\phi_{\mathbf{w}}$ is a C^∞ function.

Let Ψ denote the set of all C^m functions from $\mathbb{R}^n$ to $\mathbb{R}^p$ and Φ the subset of Ψ of all single hidden layer networks functions. The universal approximation property of this class of networks is presented in the following theorem.

Theorem 1: [6] Given $\psi \in \Psi$, a compact $X \subset \mathbb{R}^n$, and $\varepsilon > 0$, there exists $\phi_{\mathbf{w}} \in \Phi$ such that, for $l = 1, \ldots, p$,

$$\max_{\alpha \le m} \sup_{\mathbf{x} \in X} \left| D^\alpha\left(\psi_l - \phi_{l,\mathbf{w}}\right)(\mathbf{x})\right| < \varepsilon,$$

where $D^\alpha = \dfrac{\partial^\alpha}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}}$ for $\alpha = \alpha_1 + \ldots \alpha_n$.

The theorem states that no matter how a function $\psi \in \Psi$, a compact $X \subset \mathbb{R}^n$, or the accuracy of approximation $\varepsilon > 0$ are chosen, one can always find a single hidden layer feed-forward neural network having in *X* a function and all derivatives up to order *m* lying within ε of ψ and all its derivatives. As ε tends to zero, the number of hidden neurons in the network tends to infinity.

Once the neural network architecture is set, the weights are determined to minimize a performance function relative to a set $T = \left\{\left(\mathbf{x}_s, \psi\left(\mathbf{x}_s\right)\right)\right\}_{s=1}^N$ as

$$E_T(\mathbf{w}) = \frac{1}{N}\sum_{s=1}^N \left\|\psi\left(\mathbf{x}_s\right) - \phi_{\mathbf{w}}\left(\mathbf{x}_s\right)\right\|^2. \tag{22}$$

This is known as training the network. Several optimization algorithms, such as the Levenberg-Marquardt [7] or the Improved Rprop [8], are used in practice to obtain a near-optimal $\mathbf{w}$ for E_T. It is important to note that the success and the duration of the training is intimately related to the complexity of the network architecture, in particular with the number of inputs. That is why dimensionality reduction by principal components analysis is often considered. Finally, note that an already trained network implements a ready-to-use function.

The problem of parameter estimation using neural networks is formalized as follows. Consider the functions

$$\psi_1 : \boldsymbol{\theta} \to \mathbf{y}(\boldsymbol{\theta}), \tag{23}$$

$$\psi_2 : \boldsymbol{\theta} \to \mathbf{z}(\boldsymbol{\theta}), \tag{24}$$

where $\mathbf{y}(\boldsymbol{\theta})$ and $\mathbf{z}(\boldsymbol{\theta})$ are defined as in (2) and (6), respectively. When do there exist neural networks implementing left inverse functions $\phi_{1,\mathbf{w}_1}$ and $\phi_{2,\mathbf{w}_2}$ of ψ_1 and ψ_2, respectively? Furthermore, does the existence of $\phi_{1,\mathbf{w}_1}$ is equivalent to the existence of $\phi_{2,\mathbf{w}_2}$? The following theorems give the answers to these questions.

Theorem 2: [9] There exists a neural network implementing a left inverse function $\phi_{1,\mathbf{w}_1}$ of ψ_1 in (23) if and only if **S** in (3) has full column rank.

Theorem 3: [9] There exists a neural network implementing a left inverse function $\phi_{2,\mathbf{w}_2}$ of ψ_2 in (24) if and only if $\mathbf{V}^{\mathrm{T}}\mathbf{S}$, for **V** from (6) and **S** in (3), is invertible.

Theorem 4: [9] If there exists a neural network implementing a left inverse function $\phi_{2,\mathbf{w}_2}$ of ψ_2 then there exists another one implementing a left inverse function $\phi_{1,\mathbf{w}_1}$ of ψ_1. The converse is false.

HYBRID METHOD

The implementation of the Hybrid Method can be divided into two phases: the first one consists in training a neural network to approximate well the map between the space of model input-output observations and the space of model parameters; the second one, carried out after a predetermined time-interval where data collection occurs, consists in applying the trained network to obtain a parameter estimate that will be further refined by application of the Levenberg-Marquardt method. The following algorithm fully describes the method.

First phase

Assume that a set $T=\left\{\left(\mathbf{z}\left(\boldsymbol{\theta}_{\mathbf{s}}\right),\boldsymbol{\theta}_{\mathbf{s}}\right)\right\}_{s=1}^{N}$, with $\left\{\boldsymbol{\theta}_{\mathbf{s}}\right\}_{s=1}^{N}$ exhaustively covering the parameter space $D_{\boldsymbol{\Theta}}$, is given. Choose to take *T* as: i) the set for selecting and training the best possible architecture, using cross-validation to estimate the generalization error of each of the candidate architectures; ii): the randomly split set into two disjoint sets: T_1 for training the best possible architecture chosen from a set of candidate architectures whose generalization error is estimated on T_2, being $\#T_1 \geq \#T_2$.

1. Using Theorem 3 verify if there exists a neural network capable of estimating $\boldsymbol{\theta}$ from $\mathbf{z}(\boldsymbol{\theta})$. If so, go to step 2, else stop the method application.
2. Find a neural network using *T*. Let $\phi_{\mathbf{w}}$ denote the neural network function.

Second phase

Assume that $\mathbf{y}(\boldsymbol{\theta})$ is observed and that $\mathbf{z}(\boldsymbol{\theta})$ is calculated from (6).

1. Obtain $\hat{\boldsymbol{\theta}}^i=\phi_{\mathbf{w}}(\mathbf{z}(\boldsymbol{\theta}))$;
2. Take as inputs for the Levenberg-Marquardt method the initial parameter estimate $\hat{\boldsymbol{\theta}}^0=\hat{\boldsymbol{\theta}}^i$ and $\mathbf{y}(\boldsymbol{\theta})$.
3. Take for final estimate $\hat{\boldsymbol{\theta}}^f$ the solution $\hat{\boldsymbol{\theta}}$ determined by the Levenberg-Marquardt method to the minimization problem (9).

Remark: an analogous description of the method can be obtained for $T=\left\{\left(\mathbf{y}\left(\boldsymbol{\theta}_{\mathbf{s}}\right),\boldsymbol{\theta}_{\mathbf{s}}\right)\right\}_{s=1}^{N}$. By Theorem 4, this description has to be considered if the condition of Theorem 3 fails.

Obviously, the final parameter estimate is more accurate than the initial one produced by the neural network. On the other hand, it is also expected that the Levenberg-Marquardt method converges to the real value and in fewer iterations when taking as first estimate the one produced by the neural network. Note that the estimate produced by the neural network is in principle closer to the real value than one minimizing, in a set of parameter realizations, some measure of distance between the model outputs calculated from such a set and the observed output $\mathbf{y}(\boldsymbol{\theta})$ or correspondent $\mathbf{z}(\boldsymbol{\theta})$.

This original implementation strategy of the HM will be referred from now on as HM1. A second implementation strategy, HM2, is specifically considered for the case where the model structure is defined by various blocks in series as in Figure 1:. It is based on classical approaches to the problem of estimating the parameters of generally overparameterized models of this kind. The key idea behind HM2 is to estimate all parameters from $u(t)$ to $y(t)$ and then, in a stepwise fashion, recover each of the intermediate signals while further parameter refinement is made. An algorithmic description of HM2 is given below Figure 1:.

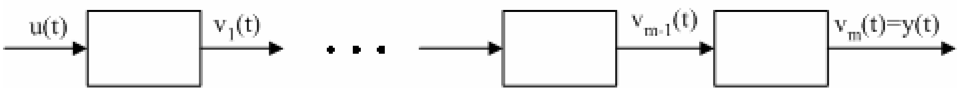

Figure 1: Block diagram of a model whose structure is defined by various blocks in series, where $u(t)$ is the input, $v_1(t),\ldots,v_{m-1}(t)$ are the intermediate signals, and $v_m(t)=y(t)$ is the output.

Assume a model structure like the one illustrated in Figure 1:.

1. Apply HM1 to the estimation of parameters from $u(t)$ to $y(t)$.
2. For $c=m-1$ to 1 do
   i. Recover $v_c(t)$ from $v_{c+1}(t)$ through the known relation between both and the estimated parameterization;
   ii. Refine the estimates for the parameters applying HM1 to the submodel from $u(t)$ to $v_c(t)$.

   end do

This second strategy of the HM application should be implemented if one verifies that the resulting estimation of the parameters is improved and pays for the increase of the computational burden.

CASE STUDY

NEUROMUSCULAR BLOCKADE MODEL

The dynamic response of the neuromuscular blockade to intra-venously administered *atracurium* may be modelled as shown in Figure 2: [10], [11].

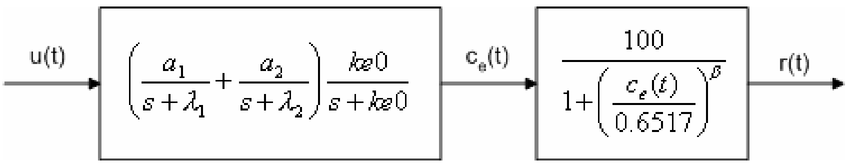

Figure 2: Block diagram of the neuromuscular blockade model.

The drug infusion rate $u(t)$ $\left[\mu g\ kg^{-1}\ min^{-1}\right]$ is related with the effect concentration $c_e(t)$ $\left[\mu g\ ml^{-1}\right]$ by a means of a linear pharmacokinetic-pharmacodynamic model, and the latter with the relaxation level $r(t)$ $[\%]$ by a non-linear static Hill equation. Hence, the relaxation level can be expressed as a parametric real function of time,

$$r(\Theta,t) = f_{\Theta}(t) = \frac{100}{1+\left(\frac{L^{-1}\left(\left(\frac{a_1}{s+\lambda_1}+\frac{a_2}{s+\lambda_2}\right)\frac{ke0}{s+ke0}u(s)\right)(t)}{0.6517}\right)^{\beta}}, \tag{25}$$

being $\Theta = \left(a_1, \lambda_1, a_2, \lambda_2, ke0, \beta\right)^T$ the vector of patient dependent parameters. The neuromuscular blockade model has therefore 6 parameters $\left(a_1, a_2\ \left[kg\ ml^{-1}\right], \lambda_1, \lambda_2, ke0\ \left[min^{-1}\right], \beta\right)$, which follow a multidimensional log-normal distribution [12]. Note that the $c_e(t)$ normalization factor, known as C_{50}, has been fixed in its mean value $0.6517\ \left[\mu g\ ml^{-1}\right]$, since it has been shown in [9] that this parameter can't be estimated from a *bolus* response $\left(u(t) = B\delta(t-t_B)\right)$.

RESULTS

A random sample of size $N = 9000$ from the multidimensional log-normal probability distribution for the pharmacokinetic-pharmacodynamic parameters was generated. The relaxation level to a bolus of $500\ \left[\mu g\ kg^{-1}\right]$ administered at $t = 0$ minutes was calculated. Data record was confined to the first 10 minutes as this is the mean time to recovery and the instant after which control action taking into account the estimates for the parameters is initiated. Since the relaxation level was calculated every 20 seconds, let

$$\mathbf{r}(\boldsymbol{\theta}) = \left(r(\boldsymbol{\theta}, t_1), \ldots, r(\boldsymbol{\theta}, t_{31})\right)^T \tag{26}$$

denote the resulting vector of 31 observations at time instants $t_i = (i-1)/3$ minutes, $i = 1, \ldots, n = 31$. The resulting set $\left\{\left(\mathbf{r}(\boldsymbol{\theta}_s), \boldsymbol{\theta}_s\right)\right\}_{s=1}^{9000}$ was randomly split into two disjoint sets: $T\ (\#T = 5000)$, for ANN selection and training, and $T'\ (\#T' = 4000)$ for HM1 and HM2 testing, in order to comparatively evaluate the performance of the two implementation strategies of the Hybrid Method.

On Table I: one can find some statistics of the absolute relative error

$$100 \times \left|\frac{\boldsymbol{\theta}_{ij} - \hat{\boldsymbol{\theta}}_{ij}}{\boldsymbol{\theta}_{ij}}\right|\ [\%], i = 1, \ldots, 6, j = 1, \ldots, \#T' = 4000, \tag{27}$$

for each of the 6 parameters $\theta_1 = a_1, \ldots, \theta_6 = \beta$ on the test set, when estimation is carried out by application of the ANN, HM1 and HM2. The results concerning the ANN should be seen as the starting point for the estimation process carried out by the HM.

Parameter	Method	Absolute relative estimation error [%]				
		Minimum	1st Quartile	Median	3rd Quartile	Maximum
a_1	ANN	1.1E-3	3.2	6.9	1.2E+1	1.4E+2
	HM1	8.1E-6	3.4E-3	9.3E-2	1.2E+1	5.2E+3
	HM2	2.4E-7	1.3E-3	2.6E-2	2.4	5.0E+2
λ_1	ANN	1.0E-4	1.0	2.1	3.9	1.4E+2
	HM1	6.1E-8	2.1E-5	4.0E-4	2.1E-2	3.0E+2
	HM2	9.5E-9	1.3E-5	2.1E-4	9.2E-3	3.7E+3
a_2	ANN	3.5E-3	1.6	3.4	6.5	7.4E+1
	HM1	7.3E-8	7.2E-4	3.1E-2	3.9	1.8E+4
	HM2	1.1E-9	2.4E-4	8.8E-3	7.8E-1	1.8E+2
λ_2	ANN	4.2E-3	1.9	4.1	7.3	7.3E+1
	HM1	4.8E-6	1.2E-3	1.5E-2	8.3E-1	1.1E+3
	HM2	3.0E-7	5.1E-4	5.1E-3	2.0E-1	1.0E+3
$ke0$	ANN	6.6E-3	2.3	4.7	8.1	3.6E+1
	HM1	6.7E-6	2.5E-3	5.9E-2	8.1	1.0E+3
	HM2	1.8E-7	9.5E-4	1.7E-2	1.6	1.6E+3
β	ANN	3.7E-4	2.4E-1	5.1E-1	9.3E-1	4.7E+1
	HM1	2.0E-12	1.6E-7	8.3E-7	4.3E-5	4.2
	HM2	2.0E-12	1.6E-7	8.3E-7	4.3E-5	4.2

Table I:. Some statistics of the absolute relative error [%] on the test set when estimation is carried out by application of the ANN and the two HM implementation strategies.

In what concerns error central tendency, one reads from the table that the estimates for all parameters are, in general, and as expected, much better when produced by the HM than by the ANN, in the sense that they are less biased, if biased at all. Note that the worst HM case is that of a_1 estimation by HM1, where the median error is only about 0.1%. One concludes that the two HM implementation strategies are comparable in this context of a generally very well succeeded estimation. With respect to the error dispersion, measured by the inter-quartile range IQR, it can be seen that, for all estimated parameters, IQR for HM2 is less than IQR for HM1. Furthermore, in 75% of the cases, the estimation error by HM2 is no more than 2.4%. Note that in the 25% of the cases considered between the median and the 3rd quartile, the quality of estimation decreased much more by HM1 application than by HM2 application, namely in parameters a_1, a_2 and $ke0$. Finally, worst estimation cases, corresponding to the maximum value of the error, are worse in the HM application, with the exception of parameter β. This is due to the high sensitivity of the Levenberg-Marquardt method to the initial estimates.

In order to evaluate the overall impact of estimating all parameters on the relaxation level *r*, one has first investigated the contribution of each parameter to the shape of the relaxation curve. To that end, it was computed a time-varying parameter weight for each of the 6, raging from 0 (no contributing parameter) to 1 (only contributing parameter), and given by

$$PW_i(t) = \frac{SAD(\theta_i, t)}{\sum_{i=1}^{6} SAD(\theta_i, t)}, i = 1, \ldots, 6, \quad (28)$$

being the sum of absolute deviations *SAD* defined as

$$SAD(\theta_i, t) = \eta_\Theta \left[\sum_{\tilde{t}=0:1/3:t} \left| r(\Theta, \tilde{t}) - r\left(\Theta\left(\theta_i = \overline{\theta}_i\right), \tilde{t}\right) \right| \right] \tag{29}$$

where η_Θ denotes the median with respect to the vector of parameters. Figure 3: illustrates $PW_i(t), i = 1, \ldots, 6$, for $t = 0:5:120$ minutes. It can be seen that the shape of *r* is dominated in the beginning by a_1, and by a_2 beyond 35 minutes.

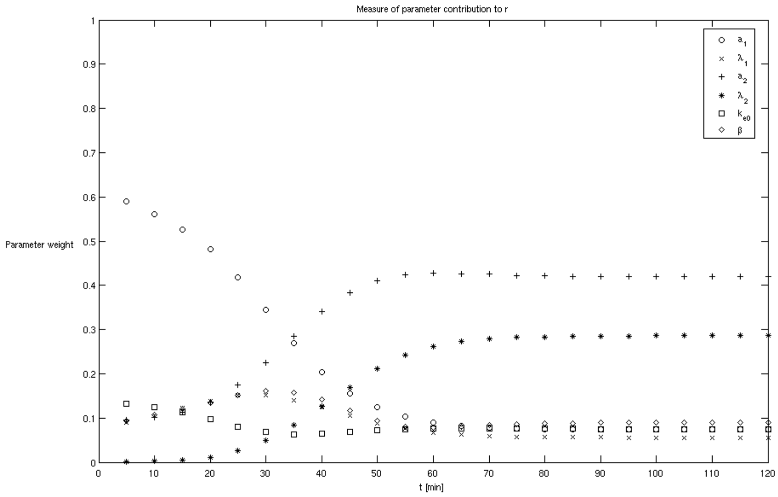

Figure 3: Measure of parameter contribution through time to the relaxation level r.

The global estimation error [%], representing the overall impact on the relaxation level of estimating all 6 parameters, was then defined as

$$\sum_{i=1}^{6} PW_i(10) \left| \frac{\theta_i - \hat{\theta}_i}{\theta_i} \right|. \tag{30}$$

The reason why this measure of goodness-of-fit was considered instead of the common sum of absolute (or squared) deviations between the observed and the estimated relaxation level curves has to do with the fact the Levenberg Marquardt method does a curve fitting and, therefore, a small error produced by the latter measure may not reflect a bad parameter estimation. Table II: shows some statistics of the global estimation error [%] on the test set when estimation is carried out by application of the ANN, HM1 and HM2.

Method	**Global estimation error [%]**				
	Minimum	1st Quartile	Median	3rd Quartile	Maximum
ANN	1.4E-1	2.9	5.3	8.8	9.4E+1
HM1	5.4E-6	2.3E-3	6.3E-2	8.4	4.8E+3
HM2	2.0E-7	8.7E-4	1.8E-2	1.7	4.7E+2

Table II:. Some statistics of the global estimation error [%] on the test set when estimation is carried out by application of the ANN and the two HM implementation strategies.

One concludes once again that there is a clear superiority of HM2 over HM1 beyond the 50% of the cases associated with the median value for the global error. This is a consequence of the dominating parameter a_1 on the first 10 minutes of the relaxation level curve (Figure 3:) being better estimated by HM2 than HM1 (Table I:).
Figure 4: illustrates the worst estimation case (maximum global error) by the ANN and the two HM implementation strategies.

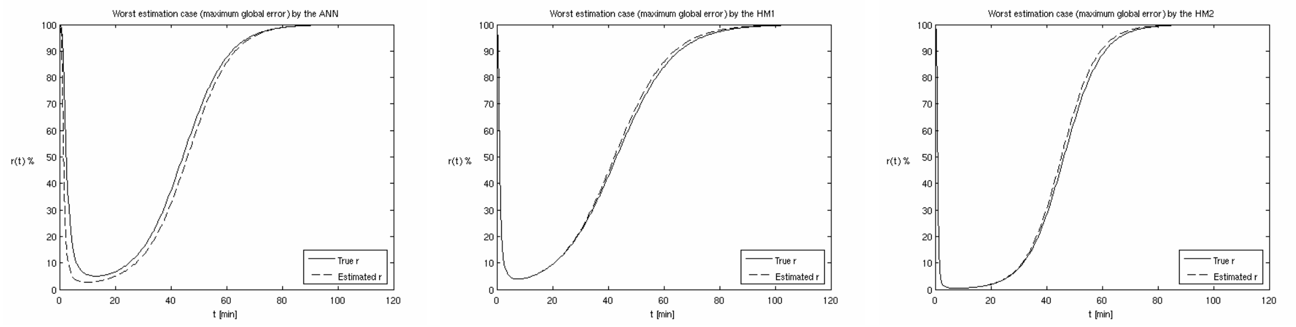

Figure 4: Illustration of the worst estimation case (maximum global error) by the ANN and the two HM implementation strategies.

CONCLUSIONS

In this paper, two different strategies were considered for application of a previously proposed hybrid method designed to model parameter estimation. The first strategy is of general applicability, while the second one is intended for models having a structure defined by various blocks in series, as is the case of the model representing the neuromuscular blockade of patients undergoing general anaesthesia, used as a case study for comparing the performances of the two strategies. One could conclude that both are comparable in terms of the estimates biases, but not of the variances, in which case the second strategy is better at the expense of a higher computational burden. Future work will consider the presence of noise at the model output.

ACKNOWLEDGEMENTS

The second and the third authors would like to thank the Fundação para a Ciência e a Tecnologia (FCT) for the financial support (PhD grant) during the course of this project. All the authors have been partially supported by the Unidade de Investigação Matemática e Aplicações (UIMA), Universidade de Aveiro, Portugal, through the Programa Operacional “Ciência e Tecnologia e Inovação” (POCTI) of the FCT, co-financed by the European Union fund FEDER.

REFERENCES

[1] Levenberg, K., 1944, “A method for the solution of certain nonlinear problems in least squares”, Quarterly of Applied Mathematics, vol. 2, pp. 164 - 168.

[2] Marquardt, D., 1963, “An algorithm for least-squares estimation of nonlinear parameters”, SIAM Journal Applied Mathematics, vol. 11, pp. 431 - 441.

[3] Alonso, H.; Magalhães, H.; Mendonça, T.; Rocha, P, 2005, “A hybrid method for parameter estimation”, WISP 2005 – IEEE International Symposium on Intelligent Signal Processing, Faro, Portugal.

[4] Haykin, S., 1999, “Neural Networks, A Comprehensive Foundation (Second Edition)”, Prentice-Hall.

[5] Gill, P.; Murray, W.; Wright, M., 1981, “Practical Optimization”, Academic Press.

[6] Hornik, K.; Stinchcombe, M.; White, H., 1990, “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks”, Neural Networks, vol. 3, pp. 551-560.

[7] Hagan, M.T.; Menhaj, M., 1994, “Training feedforward networks with the Marquardt algorithm”, IEEE Transactions on Neural Networks, vol. 5(6), pp. 989-993.

[8] Igel, C.; Husken, M., 2003, "Empirical evaluation of the improved rprop learning algorithms", Neurocomputing, vol. 50, pp. 105-123.

[9] Alonso, H.; Mendonça, T.; Rocha, P., 2005, "Contributions to parameter estimation using neural networks", Cadernos de Matemática – Série de Investigação, Universidade de Aveiro, http://pam.pisharp.org/handle/2052/86.

[10] Mendonça, T.; Lago, P., 1998, "PID control strategies for the automatic control of neuromuscular blockade", Control Engineering Practice, vol. 6(10), pp.1225-1231.

[11] Weatherley, B.; Williams, S.; Neill, E, 1983, "Pharmacokinetics, pharmacodynamics and dose-response relationships of *atracurium* administered i.v.", British Journal of Anaesthesia, vol. 55, pp. 39s-45s.

[12] Lago, P.; Mendonça, T.; Gonçalves, L., 1998, "On-line autocalibration of a PID controller of neuromuscular blockade", IEEE International Conference on Control Applications, Trieste, Italy, pp. 363-367.