Control of neuromuscular blockade with Gaussian process models

J.M. Lourenço\textsuperscript{a,b,*}, J.M. Lemos\textsuperscript{b,c}, J.S. Marques\textsuperscript{c,d}

\textsuperscript{a} EST Setúbal/IPS, Campus do IPS, Estefanilha, 2910-761 Setúbal, Portugal
\textsuperscript{b} INESC-ID, R. Alves Redol 9, 1000-029 Lisboa, Portugal
\textsuperscript{c} Instituto Superior Técnico/Universidade Técnica de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
\textsuperscript{d} ISR – Instituto de Sistemas e Robótica, Torre Norte Piso 7, Av. Rovisco Pais 1, EST Lisboa, Portugal

\textbf{A R T I C L E   I N F O}

Article history:
Received 12 January 2012
Received in revised form 26 September 2012
Accepted 29 October 2012
Available online 4 December 2012

Keywords:
Anesthesia
Gaussian process models
Nonlinear model predictive control
Nonlinear systems
Neuromuscular blockade

\textbf{A B S T R A C T}

This paper presents Nonlinear Model Predictive Control (NMPC) of neuromuscular blockade induced by atracurium on patients subject to general anesthesia. In order to tackle the high levels of uncertainty in the process behavior, probabilistic and non-parametric Gaussian process models are used in the NMPC approach. The proposed control structure was tested in a bank of models that represent patients subject to general anesthesia under elective surgery. All patients models were stabilized and yield a satisfactory performance.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Anesthesia comprises three main effects:

- Hypnosis, that aims at placing the patient in an adequate state of loss of self-consciousness;
- Analgesia, that aims at blocking the effect of nocuous stimuli (in plain words, to reduce or eliminate pain);
- Muscle relaxation, that aims at preventing involuntary patient movements.

Muscle relaxation is induced by drugs that block the neuromuscular function \cite{1} such as atracurium \cite{2}. The administration of muscle relaxant drugs may be formulated as a feedback control problem, in which the amount of drug to be continuously perfused in the patient is computed by a control algorithm depending on the neuromuscular blockade level, measured by a sensor, and such as to achieve a desired target for this variable \cite{3}.

This work presents a study on the use of Gaussian process models to implement a Nonlinear Model Predictive Control (NMPC) algorithm in automatic drug delivery for inducing muscular relaxation in human patients subject to general anesthesia in order to undergo elective surgery.

A high degree of uncertainty is present in the dynamics of biomedical systems due to its nonstationary, nonlinear behavior, external disturbances and inter and intra-individual variability. In particular, a large variability of muscle relaxation is observed in response to the infusion of atracurium, the drug considered in this article. Although the PID controller can be used to control muscle relaxation, the high level of uncertainty in patient dynamic behavior requires proper gain tuning based on patient data \cite{3,4}. When considering parametric methods, among other possibilities, the use of switched multiple model adaptive control \cite{5} has been successfully tested in clinical cases. An alternative to muscle relaxation control in the available literature consists of the use of controllers based on nonparametric models, in particular kernel based algorithms \cite{6}. Among these methods, Gaussian processes (GP) have already been successfully applied to control problems in contexts other than physiological variables \cite{7,8}. Gaussian process models provide a flexible, probabilistic, and nonparametric approach that allows to tackle the high uncertainty of these nonlinear systems. With this methodology we obtain a predictive distribution of the system output (instead of a point prediction). The predicted output is the mean of the distribution and the uncertainty of the prediction is given by its variance \cite{9,10}. The prediction value variance, that depends on the data density and noise, can be viewed as the prediction level of confidence and is a major advantage of this method when compared to neural network or fuzzy models. Another advantage consists in the fact that the GP model has a
structure that is determined only by the selection of the covariance function and the regressors. However, using the GP model suffers from high computational load due to the need to invert the covariance matrix at every iteration of the optimization algorithm, whose dimensions depend on the size of the data set and the number of regressors. Some applications of the GP model to nonlinear system identification are described in [11–17].

The GP model can be used with model predictive control since this control algorithm does not require a specific model form. Along with its predictive features, NMPC also allows to include constraints and feedforward from accessible disturbances, that are key aspects for the control of anesthesia. Some applications of NMPC based on Gaussian process models are described in [7,8,18,19].

The main contributions of this article is to present a viability study on the application of NMPC based on Gaussian process models to neuromuscular blockade, with atracurium as blocking agent.

This work is organized as follows: dynamic Gaussian processes models are described in Section 2. Section 3 introduces the model predictive control algorithm principle and in Section 4 the empirical model for neuromuscular blockade is presented. Section 5 illustrates the GP modeling of neuromuscular blockade and the simulations results obtained for model predictive control based on GP models. Conclusions are summarized at the end of the paper in Section 6.

2. Gaussian processes

A Gaussian process is a collection of random variables, that have a joint multivariate Gaussian distribution. Assuming a relationship of the form \( y = f(x) \) between an input \( x \in \mathbb{R}^D \) (where \( D \) denotes the set of real numbers and \( D \) is an integer) and an output \( y \in \mathbb{R} \), the output can be viewed as a collection of random variables \( y(1), \ldots, y(n) \sim \mathcal{N}(0, \Sigma) \) that have a joint multivariate Gaussian distribution. The covariance matrix \( \Sigma \) of this distribution can be parameterized and computed by means of a function \( \Sigma_{pq} = \text{Cov}(y(p), y(q)) = C(x(p), x(q)) \) that determines the covariance between the output points corresponding to the input points \( x(p) \) and \( x(q) \). The Gaussian process can be fully specified by its mean \( \mu(x) \) (uniformly assumed to be zero) and its covariance function \( C(x(p), x(q)) \). It is remarked that, although not all data can be modeled as a zero-mean process, this assumption is correct if the data is properly scaled and detrended [20].

A common choice of the covariance function, when the process is assumed to be stationary (the mean is constant and the covariance function only depends on the distance between the inputs \( x(i) \)), that has been proven to work well in practice [9] is

\[
C(x(p), x(q)) = \nu_1 \exp \left[ -\frac{1}{2} \sum_{d=1}^{D} w_d (x_d^p - x_d^q)^2 \right] + \nu_0 \delta(p, q),
\]

where \( x_d^p \) denotes the \( d \)-th component of the \( D \)-dimensional input vector \( x(p) \), \( \Theta = \{ \nu_1, w_1, \ldots, w_D, \nu_0 \} \) is the vector of hyperparameters and \( \delta(p, q) \) is the Kronecker operator defined as

\[
\delta(p, q) = \begin{cases} 1 & p = q \\ 0 & p \neq q \end{cases}
\]

The parameters \( w_1, \ldots, w_D \) control the scaling of the distances along each input dimension \( x_1^i, \ldots, x_D^i \). The parameter \( \nu_1 \) is the overall scale of correlations and \( \nu_0 \) expresses the process noise variance. The exponential term suggests that closer input vectors lead to highly correlated outputs while more distant input signals will be associated with less correlated outputs. The \( w_1, \ldots, w_D \) parameters can be used to evaluate the relative importance of the corresponding input components [dimensions], i.e., a high or low \( w_i \) value means that the inputs in dimension \( i \) contain high or low

information, respectively. Other forms of covariance functions may be chosen [21], the only restriction being that these covariance functions must generate non-negative definite covariance matrices for any set of input points.

2.1. Static nonlinear regression models

Assume that the data \( y(k) \) and \( x(k) \) are related by,

\[
y(k) = f(x(k)) + \epsilon(k),
\]

where \( k \) is an integer index, \( f \) is a function \( \mathbb{R}^D \rightarrow \mathbb{R} \) with an additive uncorrelated Gaussian white noise with variance \( \nu_0 \) and \( \epsilon \sim \mathcal{N}(0; \Sigma) \). Given a set of training data pairs of input data \( X = [x(1), x(2), \ldots, x(n)] \), the corresponding vector of output data \( y = [y(1), y(2), \ldots, y(n)] \) and a GP prior on \( f(x) \), with zero-mean and Gaussian covariance function given by (1), we wish to get the distribution of \( y(n+1) \) that corresponds to a new input \( x(n+1) \).

For the random variables \( y(1), \ldots, y(n), y(n+1) \) we can write

\[
y, y(n+1) \sim \mathcal{N}(0, K_{n+1}),
\]

where \( K_{n+1} \) is the covariance matrix made of submatrices as follows

\[
K_{n+1} = \begin{bmatrix}
K & k(x(n+1)) \\
[k(x(n+1))^T] & [k(x(n+1))] 
\end{bmatrix}.
\]

The matrix \( K \) is the \( n \times n \) covariance matrix for the training data such that

\[
K = \Sigma_{pp} = C(x(p), x(q)),
\]

and the vector

\[
k(x(n+1)) = [C(x(1), x(n+1)), \ldots, C(x(n), x(n+1))]^T
\]

is the \( n \times 1 \) vector of covariances between the training inputs and the new input. The expression

\[
k(x(n+1)) = C(x(n+1), x(n+1))
\]

represents the autocovariance of the new input.

The conditional distribution of (4) allows to obtain the predictive distribution of \( y(n+1) \), which is also Gaussian [22],

\[
p(y(n+1)|y, X, x(n+1)) \sim \mathcal{N}(\mu(x(n+1)), \sigma^2(x(n+1)),
\]

where \( \mu(x(n+1)) \) and \( \sigma^2(x(n+1)) \) are the mean and variance of the Gaussian predictive distribution, and are given by

\[
\mu(x(n+1)) = k(x(n+1))K^{-1}y
\]

\[
\sigma^2(x(n+1)) = k(x(n+1)) - k(x(n+1))^TK^{-1}k(x(n+1)).
\]

Given a new input vector \( x(n+1) \), we take as the predicted model output \( \hat{y}(n+1) \) the mean of the Gaussian distribution, i.e., \( \hat{y}(n+1) = \mu(x(n+1)) \) and the uncertainty of this prediction is given by the variance of the Gaussian distribution \( \sigma^2(x(n+1)) \). The predictive mean (10) can be interpreted as a weighted sum of the training outputs \( y \), to make a prediction at the test point \( x(n+1) \). For new inputs that are distant from the training inputs the value of \( k(x(n+1))^TK^{-1}k(x(n+1)) \) will be small and consequently the predicted variance (11) will be large.

2.2. Hyperparameter estimation

To be able to make predictions based on (10), the vector of hyperparameters \( \Theta \) has to be provided either as prior knowledge or estimated from the available data. An estimate is obtained by
performing the maximization of the log-likelihood of the hyperparameters given by

\[ \mathcal{L}(\Theta) = \log P(y | X) = -\frac{1}{2} \log \det(\mathbf{K}) - \frac{1}{2} \mathbf{y}^{T} \mathbf{K}^{-1} \mathbf{y} - \frac{n}{2} \log(2\pi). \]  

(12)

This optimization requires the computation of the derivative of \( \mathcal{L}(\Theta) \) with respect to each parameter \( \theta_i, i = 1, 2, \ldots, D+2 \), which is given [23] as

\[ \frac{\partial \mathcal{L}(\Theta)}{\partial \theta_i} = \frac{1}{2} \mathbf{tr} \left[ \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_i} \right] + \frac{1}{2} \mathbf{y}^{T} \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_i} \mathbf{y}, \]  

(13)

where \( \mathbf{tr} \) denotes the trace.

To solve the \( D+2 \) algebraic equations that result from equating (13) to zero in order to achieve the \( \theta_i, i = 2, \ldots, D+2 \) hyperparameters, a nonlinear optimization algorithm is necessary. To find the maximum of \( \mathcal{L}(\Theta) \) a conjugate gradient method with line searches was used with success [24].

2.3. Dynamic models and prediction

Gaussian processes can be used for the modeling of dynamic systems of the form defined by (3) if delayed input and output signal samples are used as regressor entries [26,12]. This procedure amounts to define

\[ \mathbf{x}(k) = [y(k-1), y(k-2), \ldots, y(k-L)], \]

\[ u(k-1), u(k-2), \ldots, u(k-L)]. \]  

(14)

In such cases a nonlinear autoregressive model is considered, such that the current output \( y(k) \) depends on previous output samples of \( y \), as well as on previous control input samples, up to a given lag \( L \).

The Gaussian process model not only describes the dynamic characteristics of the system but also provides information about the confidence in the predictions. This feature can be used to point out predictions of poor quality indicated by the high values of the corresponding variance.

The multiple-step-ahead predictions can be performed by repeating iteratively step-wise ahead predictions up to a desired horizon ahead (iterative method). This method can be performed by feedback to the input of the GP model only the mean values of the predicted output (naive approach), or the complete output distributions (exact approach), together with the future control inputs in both approaches. In the “naive" approach, a \( L \)-step-ahead prediction knowing the \( L \) past input and output signals as well as the future control input values, can be obtained if at time steps \( k, k+1 \) the following regressors are fed into the GP model

\[ \mathbf{x}(k) = [y(k-1), \ldots, y(k-L), u(k-1), \ldots, u(k-L)] \]

\[ \mathbf{x}(k+1) = [y(k), y(k-1), \ldots, y(k-L+1), u(k), \ldots, u(k-L+1)] \]

\[ \vdots \]

\[ \mathbf{x}(k+l) = [y(k+l-1), y(k+l-2), \ldots, y(k+l-L), u(k+l), \ldots, u(k+l-L)]. \]  

(15)

where \( \hat{y} \) is the predicted model output that is computed with (10).

The naive approach does not account for the uncertainty generated by each intermediate step prediction giving worse results especially in terms of predicted variance that is usually narrower than the obtained by the exact approach [13,25]. More on the GP model simulation and differences between both the above approaches can be found in [12,14,26].

2.4. GP modeling procedure summary

To sum up, assume that a training data set is known and consists of a \( n \times d \) matrix \( X \) of input measurements and a \( n \times 1 \) vector \( y \) of output or target values. Use the training data set to develop a model that can be used to make predictions with new data. Assume that the new data, called the testing data, is given by an \( 1 \times d \) input vector \( x^* \). The \( \mathbf{y}^* \) represents the target value corresponding to \( x^* \). The goal is to predict the value of \( y^* \) given \( X, y, \) and \( \mathbf{x^*} \).

In the Gaussian process approach the prediction of \( y^* \) involves the selection of a positive semidefinite covariance function \( C(x_i), x_j) \), where \( x_i \) and \( x_j \) are vectors with \( d \) components. The covariance function can be used to construct the \( n \times n \) covariance matrix \( \mathbf{K} \) with entries \( K_{ij} = C(x_i, x_j) \), where \( x_i \) and \( x_j \) are rows of \( X \), as well as the \( 1 \times n \) cross covariance vector \( \mathbf{k} \) with entries \( k_i = C(x_i, x_j) \). The prediction \( \mathbf{y}^* \) of \( y^* \) is obtained with the Gaussian processes equation (10). Besides the prediction \( \mathbf{y}^* \), the Gaussian process approach also leads to the prediction estimated variance (11), where \( \mathbf{K}(\mathbf{x}^*) = \mathbf{C}(\mathbf{x}^*, \mathbf{x}^*) \).

Note that the hyperparameters of the covariance function are not known in advance and must be estimated for instance, via the maximization of the log-likelihood (12) using the training data.

3. Model predictive control

The Model Predictive Control (MPC) algorithm used [27], aims at minimizing in discrete time and in a receding horizon sense the quadratic cost

\[ J(k, k + T_P) = \sum_{i=1}^{T_P} q_i (\hat{y}(k+i) - y_c(k+i))^2 + \sum_{i=1}^{T_u} u_i^2(k+i-1) \rho_i, \]  

(16)

where \( k \) is current time, \( \hat{y}(k+i) \) is the \( i \)-step ahead prediction, \( y_c \) is the reference to track, \( T_P \) is the prediction horizon, \( T_u \) is the control horizon, and \( q_i, i = 1, \ldots, T_P \) and \( \rho_i, i = 1, \ldots, T_u \) are error and control weighting factors, respectively. The quadratic cost function (16) is minimized subjected to the constraints imposed, among others, by actuator saturation, the maximum rate of change of the manipulated variables, and the bound of the output magnitude

\[ u_{\text{min}} \leq u(k) \leq u_{\text{max}}. \]  

(17)

\[ \Delta u_{\text{min}} \leq \Delta u(k) \leq \Delta u_{\text{max}}. \]  

(18)

\[ y_{\text{min}} \leq y(k) \leq y_{\text{max}}. \]  

(19)

The predictive control principle can be summarized as follows: at each current time \( k \), the process output \( y(k+i) \) is predicted over a time horizon \( k+1, \ldots, T_P \). The predicted values are indicated by \( \hat{y}(k+i) \) and the value \( T_P \) is called the prediction horizon. The prediction is done by a model of the process, in this case a GP model. The prediction depends on the past inputs and outputs, but also on the future control signals \( u(k+i), k = 1, \ldots, T_u \) (i.e. the control actions that are intended to be applied from a moment \( k \) onwards): At each time \( k \), among the admissible control scenarios, a set of future control samples is chosen \( u(k+i), k = 1, \ldots, T_u \) by optimizing a given criterion. The resulting control \( u(k) \) is then applied to the process, but only at instant time \( k \). According to a receding horizon strategy at the next sampling period a new measured output sample is available and the whole procedure is repeated.

4. Neuromuscular blockade model

As shown in Fig. 1, the dynamic response of the Neuromuscular Blockade Model (NMB) induced by infusion of atracurium may be modeled by the Wiener structure [2,3] of Fig. 1, in which \( s \) denotes the Laplace transform variable. The linear compartmental pharmacokinetic model relates the drug infusion rate \( u(t) \)
\[ [\mu g\ kg^{-1}\ min^{-1}] \text{ with the plasma concentration } c_p(t) [\mu g\ ml^{-1}], \text{ and may be described by the state equations}
\]
\[
\dot{x}_1 = -\lambda_1 x_1(t) + a_1 u(t)
\]
\[
\dot{x}_2 = -\lambda_2 x_2(t) + a_2 u(t)
\]
\[
c_p(t) = \sum_{i=1}^{2} x_i(t),
\]

where \( \lambda_i [\text{min}^{-1}] \) and \( a_i [\text{kg mL}^{-1}] \) \((i = 1, 2)\) are patient-dependent parameters. The pharmacodynamic effect for \textit{atracurium} relates the plasma concentration \( c_p(t) \) to the level of NMB (normalized between 0 and 100, with 0 corresponding to full paralysis and 100 to normal muscular activity), \( r(t) \) (in percent). This involves a nonlinearity described by the Hill equation

\[
r(t) = \frac{100 c_p^{\beta}}{C_{50}^{\beta} + c_p^{\beta}},
\]

and the relation between the plasma concentration \( c_p(t) \) and the effect compartment concentration \( c_e(t) [\mu g\ ml^{-1}] \),

\[
\dot{c}_e(t) = -\lambda_c c_e(t) + \lambda_c c(t),
\]

where \( \lambda \) is an auxiliary signal and \( \tau [\text{min}], \lambda [\text{min}^{-1}], C_{50} [\mu g\ ml^{-1}] \), and \( \beta \) (dimensionless) are all patient-dependent parameters. Eq. (25) was included to allow a better clinical data replication [4]. To cover a wide range of neuromuscular blockade behaviors, a set of non-linear dynamic models, \( M_j, j = 1, \ldots, 100 \), has been generated (see Fig. 2) using the probabilistic model for \textit{atracurium} discussed previously [4].

The patient behaviors are characterized by a large dynamic variability, and motivates the need for a control strategy with a high degree of robustness.

In the simulations performed, a noise signal \( n(t) \) with a variance of \( \sigma^2 = 0.3 \) was superimposed to the output process. This noise is added in order to simulate the stochastic disturbances normally present in real situations.

5. Simulation results

5.1. GP model

As already mentioned in Section 3, a prediction model is necessary in order to use a MPC strategy.

Using a trial and error approach, model \( M_{94} \) was chosen to generate the identification data that is needed for the learning of the Gaussian process model. Fig. 3 represents the extremes (corresponding to \( M_{12} \) and \( M_{90} \)) and the average (corresponding to \( M_{44} \) and \( M_{44} \)) neuromuscular blockade dynamic behaviors induced by a bolus of 500 \( \mu g\ kg^{-1} \). We can see that model \( M_{94} \) has an almost central dynamic behavior.

For the modeling of a system, it is desirable to be able to predict the response to inputs that cover most or all of the operating range of the system.

To obtain the identification data, the NMB model \( M_{94} \) was initially excited by a bolus of 500 \( \mu g\ kg^{-1} \) (i.e., very large amplitude signal with a short duration, that mimics the drug administration by a syringe injection) and after that by a pseudo random binary signal with amplitude values between 0 and 11, multiplied by a random number issued from a normal distribution with mean 0 and variance 0.64. Fig. 4 shows the simulated neuromuscular blockade response and the input signal \( u(t) \) used that induced it. The output signal is contaminated by additive measurement noise with a normal distribution. The maximum limit of the infusion rate \( u(t) \)
was set as 20 µg kg⁻¹ min⁻¹ considering the limit of the infusion pumps used for the model data collection and in use in the operating room. As the number of samples used in the identification determines the dimension of the covariance matrix the identification data is was limited to 250 samples. This value was found to be a good compromise between computational load and performance. The regressor structure that together with the covariance function are the only choices that have to be made for the GP model, and the hyperparameters of the covariance function were selected by trial and error such as to maximize the likelihood of the training data, being
\[
x(k) = [y(k − 1), y(k − 2), y(k − 3), u(k − 1), u(k − 2)].
\]

As covariance function was chosen (1). The corresponding vector of hyperparameters, is given by:
\[
\Theta = [\omega_1, \ldots, \omega_D, v_1, v_0]^T
\]

The maximum likelihood approach was used to estimate the hyperparameters. An optimization algorithm based on [24] was used to search for their optimal values yielding
\[
\Theta = 10^3 [0.0237, 0.3456, 0.1047, 1.2014, 0.5897, 6.6691, 0.0013]
\]

The response of the Gaussian process model to the validation signal is shown in Fig. 5(a). In the Fig. 5(b) the validation input signal that is different from the data used in the learning is shown. The shaded band represents a 95% confidence level of the GP model predicted mean. Regions with wider confidence bars indicate that the model is less certain about its prediction. In the range of the operating space there is a very good adequation between the prediction and the actual values, with most of the actual values lying within the prediction confidence region.

The GP model was also validated for a predictive horizon of length 6, since the model is intended for predictive control purposes. Fig. 6 shows the Gaussian process (obtained with M₉₄) six-step-ahead prediction, and the M₉₄ and M₉₉ responses for comparison. The results show a good adequation between the prediction and the actual values for both simulations. Also in these simulations most of the actual values lie within the prediction confidence region (more than 85% of the values). The six-step-ahead predictions were obtained using the naive approach that produces narrower predicted variance regions than the obtained with the exact approach. Using the exact approach would imply a significantly more complex algorithm. It is still an open issue to show its advantages in this particular situation.

5.2. Control

In this section we apply a NMPC algorithm based on a GP model to the problem of controlling the neuromuscular blockade by means of administration of atracurium.

It is assumed that an initial bolus of 500 µg kg⁻¹ is applied manually to the patient, driving the r% to a level less than 10% for a given period of time. After this initial period, the control objective consists in following a specific reference. The value of the reference is initially zero during approximately 30 min. It is then gradually increased to the final value of 10%. The MPC algorithm is turned on when the reference begins to increase.

The NMPC algorithm minimizes the cost function (16) subject to the GP model (9) and the constraint (17), where \(u_{\text{min}} = 0\) and \(\mu_{\text{max}} = 20\).
Several simulations, with different patient models (including patient M_{69} that is the most difficult patient model to control) have been conducted in order to find the best configuration for the NMPC algorithmic parameters $T_u$, $T_p$, $q_i$, $p_i$, the choice of these parameters was done by trial and error. A sampling interval of 20 s (the one used for real data collection) has been used. The parameters were selected as $T_u = 1$, $T_p = 20$, $q_1 = 1.14^2$, $q_2 = 1.14^4$, ..., $q_{20} = 1.14^{20}$ and $p = 10$. A low value of $T_u$ was chosen since the cost does not depend much on the control horizon and this choice corresponds to a lower computational time. The GP model used by the controller in all the simulations is fixed and was taught off-line by the model M_{69}, as explained in the last section. By following the procedure described above, one single controller is obtained which is able to stabilize all the patient models in the set considered while yielding an adequate performance.

The GP prediction of the level of NMB was done by iterative multi-step ahead prediction, feeding back at each time step the predictive mean only (naive approach), an approach that has shown similar results, in what concerns the mean predictions, when compared for other systems with the exact approach [12,14].

Figs. 7 and 8 illustrate the output and the control signal, respectively, corresponding to the application of the NMPC algorithm based on GP model to the models M_{12}, M_{61}, M_{44} and, M_{69}. It can be seen from these figures that the quality of the responses is better for model M_{44} and worse for model M_{69}. These results were expected, due to the differences between the model that was used in the learning of the GP model and the models used in the simulations (see Fig. 3). The oscillatory behavior in model M_{69} and to lesser extent in model M_{44} indicates that the model upon which the GP model was based do not provide a full coverage of the range of patient dynamics.

To overcome the output oscillations, mainly in patient M_{69}, the controller was modified by adding to the NMPC control signal $u(t)$ a filtered output derivative signal $u_d(t)$ (hereafter called NMPC+D),

$$u(t) = u_p(t) + u_d(t),$$

where $u(t)$ is the NMPC+D control signal. The transfer function relating the derivative control signal and the output signal is given by,

$$y(z) = \frac{K(1-a)(z-1)}{z-a},$$

where $z$ is the Z-transform variable and $K$ and $a$ are parameters. After several simulations, with different patient models (mainly patient M_{69}) in order to find the best configuration for the derivative term, the prediction horizon $T_p$ was settled to 14, $K$ was set to 5, and $a = 0.3$, and all the remaining parameters were not changed. As seen by comparing Figs. 7 and 9 with the derivative term introduction the initial transient almost disappeared in models M_{12}, M_{44}, M_{64}, and the output oscillations of model M_{69} was significantly reduced.

The infusion rate of the NMPC+D controller, as expected, shows a more random and spiky behavior than the NMPC infusion rate (compare Figs. 8 and 10), however, that one has never reached his maximum limit of $u_{\text{max}} = 20 \mu g kg^{-1} min^{-1}$. Table 1 shows the

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean square tracking error</th>
<th>Mean % steady state</th>
<th>Mean % steady state standard deviation</th>
<th>Mean infusion rate</th>
<th>Infusion rate standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.927</td>
<td>10.20</td>
<td>0.550</td>
<td>7.197</td>
<td>3.521</td>
</tr>
<tr>
<td>41</td>
<td>0.465</td>
<td>9.55</td>
<td>0.592</td>
<td>5.271</td>
<td>3.512</td>
</tr>
<tr>
<td>44</td>
<td>0.466</td>
<td>9.67</td>
<td>0.592</td>
<td>5.543</td>
<td>3.554</td>
</tr>
<tr>
<td>69</td>
<td>2.181</td>
<td>9.28</td>
<td>0.558</td>
<td>4.562</td>
<td>3.788</td>
</tr>
</tbody>
</table>

Table 1: The four models parameters for NMPC with filtered derivative and a random output noise of $\sigma^2 = 0.3$. |
Fig. 8. Predictive control ($T_p = 20$, $T_u = 1$). Infusion rate (controller output) obtained with patient models 12, 41, 44, and 69.

Fig. 9. Predictive control ($T_p = 14$, $T_u = 1$) plus output derivative control. Simulation results obtained with patient models 12, 41, 44, and 69 showing the reference (black) and the neuromuscular blockade level $r(t)$ (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)
controller NMPC+D performance analysis for the patients M_{12}, M_{41}, M_{44} and, M_{69}. The Mean Square Tracking Error (MSTE),

$$\text{MSTE} = \frac{1}{n} \sum_{i=1}^{n} [r(t) - y_c(t)]^2,$$

was calculated for each simulation after the initial 30 min (i.e., 90). The variable \( r(t) \) is near the reference and the mean infusion rate values are far from maximum limit. Due to his slow recovery after the initial bolus infusion and the chosen reference (that begins to increase after 30 min), patient M_{69} takes about 60 min to get to \( r(t) = 10\% \) after the bolus infusion. For this patient model, the MSTE is always higher than for the other patient models, under similar conditions.

To test the effect of the prediction horizon on the performance of the NMPC+D controller, simulations were run keeping all parameters constant, except the prediction horizon \( T_p \). Fig. 11 shows the MSTE as a function of the prediction horizon \( T_p \). For low values of \( T_p \), a big error is obtained. Due to decreasing prediction precision of the GP model high values of \( T_p \) also yield a decreasing performance. The best \( T_p \) values are 11, 11, 11 and 16 for models M_{41}, M_{44}, M_{12} and M_{69}, respectively. For the simulations made with controller NMPC+D the prediction horizon was always chosen as \( T_p = 14 \), a value that is a compromise for all the tested models.

The effect of the control effort weighting matrix on the performance of the NMPC+D controller was also tested. The MSTE as a function of the control effort weight \( \rho \) is depicted in Fig. 12. This figure shows that there is a slight change in the tracking error, whenever the \( \rho \) is changed between 0.01 and 62, with values in the intervals of [0.91, 2.00], [0.34, 0.48], [0.43, 0.47] and [1.91, 2.20] for models M_{12}, M_{41}, M_{44} and, M_{69}, respectively. The MSTE values for models M_{41}, M_{44} are very similar and show almost no changes except for \( \rho = 62 \) where M_{41} takes its minimum value. The model M_{12} has an ascending tendency with a MSTE minimum value of 0.91 for \( \rho = 1 \), whereas the model M_{69} has a tendency for descending with a minimum value of 1.91 for \( \rho = 62 \). These different behaviors are justified by a faster muscle relaxation recovery for M_{12} than for M_{69}. Thus, the former patient model requires a higher infusion rate (lower control effort weighting values) to keep the neuromuscular relaxation level near the reference, than the latter. Again and because of the different models behaviors it is necessary to reach a

---

**Fig. 10.** Predictive control (\( T_p = 14, T_o = 1 \)) plus output derivative control. Simulation infusion rate (controller output) obtained with patient models 12, 41, 44, and 69.

**Fig. 11.** Predictive control (\( T_p = 1 \)) plus output derivative control. Mean Square Error obtained with different values of the prediction horizon for the models 12, 41, 44, and 69. The circles indicate the model minimum error.
compromise for the $R$ value. For $\rho$ values higher than 62 the system becomes unstable.

The effect of the error weighting factor values $q_1 = f^1$, $q_2 = f^2$, ..., $q_6 = f^6$ on the MSTE was also analyzed for values of $f$ in the interval $[1.36, 20]$. The changes in the MSTE were not significant, for the model $M_{12}$, $0.93 \leq \text{MSTE} \leq 0.99$ and for the model $M_{69}$, $2.10 \leq \text{MSTE} \leq 2.18$. For $f$ values lower than 1.36 the system becomes unstable.

Because of the inclusion of the derivative term in the control signal and to further test the control algorithm robustness, it was decided to analyze the effect of the output noise in the performance of the controller. Fig. 13 illustrates the MSTE as a function of the output noise variance. Although, this figure shows a decreasing performance for higher values of the output noise variance, Fig. 14 shows that even for high values of the noise ($\sigma^2 = 0.8$) the response is still acceptable and similar to the one obtained with $\sigma^2 = 0.3$.

Fig. 12. Predictive control ($T_p = 14$, $T_s = 1$) plus output derivative control. Mean Square Error obtained with different values of the control effort weight $\rho$ for the models 12, 41, 44, and 69. The circles indicate the model minimum error.

Fig. 13. Predictive control ($T_p = 14$, $T_s = 1$) plus output derivative control. Mean Square Error obtained with different output noise values for the models 12, 41, 44, and 69.

Fig. 14. Predictive control ($T_p = 14$, $T_s = 1$) plus output derivative control. Simulation results obtained with patient models 12, 41, 44, and 69 for an output noise variance of 0.8. This figure presents reference in black and the neuromuscular blockade level $r(t)$ in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)
Fig. 15. Predictive control \(T_r = 14, T_n = 1\) plus output derivative control. Simulation infusion rate (controller output) obtained with patient models 12, 41, 44, and 69 with a output noise variance of 0.8.

![Graph showing infusion rate over time for different models](image)

Table 2

<table>
<thead>
<tr>
<th>Model 12</th>
<th>Model 41</th>
<th>Model 44</th>
<th>Model 69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean square tracking error</td>
<td>1.397</td>
<td>1.125</td>
<td>1.021</td>
</tr>
<tr>
<td>Mean % steady state</td>
<td>10.13</td>
<td>9.38</td>
<td>9.56</td>
</tr>
<tr>
<td>% steady state standard deviation</td>
<td>0.904</td>
<td>0.948</td>
<td>0.879</td>
</tr>
<tr>
<td>Mean infusion rate</td>
<td>7.225</td>
<td>5.318</td>
<td>5.585</td>
</tr>
<tr>
<td>Infusion rate standard deviation</td>
<td>5.199</td>
<td>4.963</td>
<td>5.037</td>
</tr>
</tbody>
</table>

(Fig. 9). Table 2 shows the controller NMPC+D performance analysis for the patient models \(M_{12}, M_{41}, M_{44}\) and, \(M_{69}\) with \(\sigma^2 = 0.8\) output noise variance. The results show that the control performance is reduced essentially by a higher output noise that is shown in the output standard deviation, rather than in an inferior control performance. The mean infusion rate is almost equal to the one obtained with \(\sigma^2 = 0.3\), however, is more dispersed and for all the models the infusion rate maximum limit was reached several times (Fig. 15).

6. Conclusions

This paper presents a viability study on the application of Non-linear Model Predictive Control based on Gaussian process models to neuromuscular blockade. The results obtained suggest that the application of the method described to real patients is viable, yielding results that are comparable with the ones obtained with other approaches such as [3, 5]. From a modeling point of view the NMPC based on GP models has the advantage of avoiding the need to make assumptions on the nonlinearity associated to the pharmacodynamic of the drug used. A major disadvantage of the Gaussian process model is the computational load associated with the need to invert the covariance matrix, at every iteration of the optimization algorithm. The computational load increases with higher prediction horizon values. For muscle relaxation this is not however an issue due to the value of the sampling period used (20 s).

Acknowledgements

This work was supported by national funds through FCT – Fundação para a Ciência e a Tecnologia under project GALENO PTDC/SAU-BEB/103667/2008 and PEst–OE–EEI/LA0021/2011.

References


