Predictive Control of Depth of Anaesthesia
Contrôlé Prédictif de profondeur d’anesthésie

Nicolas Cardoso De Castro

**KEYWORDS (MOT CLÉS)**

Depth of Anaesthesia; Model Predictive Control; Switched Multiple Model Predictive Control; Control algorithm; MATLAB; SIMULINK; (Profondeur d’anesthésie; Contrôle prédictif de modèles multiples commutés; Algorithme de contrôle prédictif;)

**ABSTRACT**

A major difficulty for Depth of Anaesthesia controller design is parameters uncertainty. The aim of this internship is to derive a controller that overwhelm this difficulty. Firstly, a MATLAB/SIMULINK model for Depth of Anaesthesia is derivated and evaluated in open-loop. Secondly, three Model Predictive Control controller configurations are tested on the model. Finally, a MATLAB-based algorithm that couple Model Predictive Control and Switched Multiple Model Predictive Control is developed.

**RÉSUMÉ**

La conception d’un contrôleur de profondeur d’anesthésie présente une difficulté majeure : l’incertitude des paramètres. Le but de ce stage est de développer un contrôleur qui surmonte cette difficulté. La première partie consiste à développer et tester un modèle pour la profondeur d’anesthésie. Le rapport teste ensuite trois configurations de contrôleurs MPC (Model Predictive Control). La fin du rapport décrit le développement d’un algorithme qui associe les techniques de MPC et de contrôle prédictif de modèles multiples commutés (Switched Multiple Model).
# Table of Contents

1 Introduction  
   1.1 Context and objective ................................................. 4  
   1.2 Description of the work ............................................. 4

2 Depth of Anaesthesia model derivation  
   2.1 Clinical explanation .................................................. 5  
   2.2 Model derivation ...................................................... 6  
   2.3 Open-loop simulations ................................................ 8

3 Model Predictive Control structures study  
   3.1 Structures introduction .............................................. 9  
   3.2 Controllers configuration ........................................... 10  
   3.3 Closed-loop simulations ............................................. 13

4 Switched Multiple Model Predictive Control algorithm  
   4.1 Controller introduction ............................................. 15  
   4.2 Algorithm derivation ................................................. 19  
   4.3 Closed-loop simulations ............................................. 22

5 Conclusion ................................................................. 24

References ................................................................. 25

A Appendix  
   A.1 Coefficients in the DoA model ..................................... 26  
   A.2 Pseudo-code of the algorithm ..................................... 27  
   A.3 Extra content ......................................................... 27
I would like to thank all the employees from INESC-ID I have been in contact with. I have been very well welcomed, making my time spent abroad pleasant.

I particularly give thanks to my supervisor, Mr. Miranda Lemos, who has been a precious help in many ways.

I finally thank all the people I have met in the meetings in Porto.

This work has been done under the framework of project IDeA - Integrated Design for Automation of Anaesthesia, contract PTDC/EEA-ACR/69288/2006.
1 INTRODUCTION

1.1 Context and objective

This work is done under the framework of project IDeA (Integrated Design for Automation of Anaesthesia) \(^1\). The objective of the project IDeA is the development of an autonomous integrated system for the automation of anaesthesia that incorporates advanced control algorithms able to tackle the specific challenges of anaesthesia.

Anaesthesia comprises three actions \(^1\): keeping the patient still (called NeuroMuscular Blockade (NMB) achieved through atracurium drug), inducing a desired level of loss of consciousness (called Depth of Anaesthesia (DoA) achieved through propofol drug) and reducing the response to noxic stimuli (“alleviating the pain”, analgesia achieved through remifentanil drug). The internship focuses on the Depth of Anaesthesia, called DoA hereafter.

The control of anaesthesia has already been considered in the bio-medical framework \(^2, 3\) using various methods. Although the regulation of NMB level can be achieved through simple control laws, the control of the DoA level turned out to be more delicate. Indeed the DoA level is affected by both propofol and remifentanil drugs. Moreover actions of the surgeon on the patient may alter the DoA level. At least parameters of the patient change over the time of the surgery and from one patient to another. Thus a major difficulty for DoA controller design is parameters uncertainty.

Developing and experimenting controller design algorithms that overwhelm this difficulty is the major challenge of the work performed in the internship. The objective of the work is to couple Model Predictive Control (MPC) with supervised Switched Multiple Model Predictive Control (SMMPC) technics to design a controller suitable in most of the cases.

But before developing such algorithm, preliminary tasks are completed, consisting of derivating and evaluating a DoA model and studying a MPC controller on the DoA model.

1.2 Description of the work

The work is divided into three steps:

Designing and evaluating the DoA model: A non-linear DoA model is derivated using the documentation available (\(^4, 5, 6\)). It is used to simulate the patient and linearized to be used in the MPC and SMMPC controllers. The models are implemented using MATLAB and SIMULINK. Open-loop simulations are run and clinical data are available for comparison.

Evaluating MPC structures for DoA: In this step, several MPC controller configurations are designed to control the DoA model. The controllers are built and simulated in MATLAB and SIMULINK. The aim of this step is to give a feasibility study on the use of MPC to control DoA and to characterize the influence of the MPC design parameters on the performance of the regulation.

Developing an algorithm incorporating MPC and SMMPC technics to control DoA: The work previously done is extended to include a Switched Multiple Model block to overwhelm the patient uncertainty. In this step, several MPC controllers are used and a supervisor switches between the controllers to improve the performances of the regulation. The aim of this step is to develop a MATLAB based algorithm describing the SMMPC controller. The influence of the design parameters on the performance of the regulation are also characterized.

For each step, several MATLAB/SIMULINK files and a technical report are provided. The MATLAB/SIMULINK files can be used to run simulations, change parameters and observe performances. The technical report details the theoretical knowledge used to develop the MATLAB/SIMULINK files, explains how to use them and gives results, performances and parameters influence.

\(^1\)Contract PTDC/EEA-ACR/69288/2006.
2 DEPTH OF ANAESTHESIA MODEL DERIVATION

Section 2

Depth of Anaesthesia model derivation

2.1 Clinical explanation

Diffusion of a drug in the body (from the administration point to the actual effect) is a complex process. It is modeled using two stages [2] (as depicted on figure 1).

- The first stage, represented by the “Pharmaco-Kinetic” model (PK model), describes the spreading of the drug over the blood and tissue. The concentration of the drug on plasma is a function of time denoted \( c_p \).

- The second stage, represented by the “Pharmaco-Dynamic” model (PD model), describes the relationship between the plasma drug concentration and the drug effect. This stage comprises two steps: the transport of the drug from the plasma (with a certain plasma concentration, \( c_p \), given by the PK model) to the effect compartment (the brain, with an effect concentration denoted \( c_e \)), and the effect on the body of the drug present in the effect compartment.

Note on this figure that the two drugs only interact in the last block and there are independent PK and effect compartment models for both remifentanil and propofol.

Sensors are available to measure the DoA level but there is no way to measure the patient pain. The DoA level is measured with the Bispectral Index, called BIS, and given in [%]. The awake state corresponds to 97.7 % and a DoA of 50 % corresponds to a loss of the half of the consciousness. Levels lower than 15 or 10 % should never be reached since they correspond to coma.

The DoA model considers both propofol and remifentanil since this last one has a non-negligible effect on the DoA level.

Hereafter, \( c_{remi} \) (the remifentanil effect concentration) is assumed to be given and only the propofol chain is considered. This is acceptable in a first approximation if the TCI (Target Control Infusion [7]) method is used.

The DoA model\(^2\) takes the propofol infusion rate as input and provides the DoA level. The remifentanil effect concentration is taken into account as a measured disturbance. This is depicted on figure 2.

\(^{2}\)The data used in the model correspond to published results and/or typical values that lead to responses similar to published ones.
2.2 Model derivation

2.2.1 Pharmaco-kinetic model

The compartmental approach [6] is used to derive the PK model. According to this approach, the body is assumed to be divided into several compartments. In each compartment the drug concentration is homogeneous and there are exchanges between compartments. The propofol infusion rate \( u_{\text{prop}} \) is called \( u \) since it will be the manipulated variable later. The computation details are given in [8], and the values of the coefficients are given in table 2 in the appendix A.1. This yields the continuous linear state space model:

\[
\begin{align*}
\dot{x}_1 &= A_1 x_1 + B_1 u \\
c_{p}^{\text{prop}} &= C_1 x_1
\end{align*}
\]  

with

\[
A_1 = \begin{bmatrix}
-k_{10} & -k_{12} & -k_{13} \\
k_{12} & -k_{21} & k_{31} \\
k_{13} & 0 & -k_{31}
\end{bmatrix}, \\
B_1 = \begin{bmatrix}
10^4 \\
3600 \\
0
\end{bmatrix}
\]

and

\[
C_1 = \begin{bmatrix}
\frac{1}{0} & 0 & 0
\end{bmatrix}
\]

\( v_1 = \text{weight} \times v_c \) is computed from the weight of the patient and the coefficient \( v_c \) [L/kg] which represents the volume of compartment one per patient unit weight.

2.2.2 Pharmaco-dynamic model - Derivation of the effect compartment model

As in the comparative study between PK models by [6], a low-pass filter is used to relate the propofol plasma concentration \( c_{p}^{\text{prop}} \) and the propofol effect concentration \( c_{e}^{\text{prop}} \). This yields the following state space representation:

\[
\begin{align*}
\dot{x}_2 &= A_2 x_2 + B_2 c_{p}^{\text{prop}} \\
c_{e}^{\text{prop}} &= C_2 x_2
\end{align*}
\]  

where \( A_2 = -k_{e0}, B_2 = k_{e0} \) and \( C_2 = 1 \). The value of \( k_{e0} \) is given in table 2 in the appendix A.1.

2.2.3 Pharmaco-dynamic model - Derivation of the interaction model

The interaction model used here has been developed by [5]. It is called the Hill equation since it generalizes the Hill equation used for similar purposes in NMB modeling. It consists of a static model with two inputs (the remifentanil \( c_{e}^{\text{remi}} \) and propofol \( c_{e}^{\text{prop}} \) effect concentrations) and one output (the DoA level, called \( y \) since it is the measured output).

The Hill equation is given by:

\[
y = f(c_{e}^{\text{remi}}, c_{e}^{\text{prop}}) = E_0 \left( \frac{[U_{\text{prop}} + U_{\text{remi}}]/U_{50}(\theta)]^\gamma}{1 + [U_{\text{prop}} + U_{\text{remi}}]/U_{50}(\theta)]^\gamma} \right)
\]  

\[2.3\]
where:

- $E_0$ is the effect at zero concentrations,
- $U_{\text{prop}}$ and $U_{\text{remi}}$ are the normalized concentrations, given by $U_{\text{prop}} = \frac{C_{\text{prop}}}{C_{\text{50E}}} \text{ and } U_{\text{remi}} = \frac{C_{\text{remi}}}{C_{\text{50E}}}$ where $C_{50E}^{\text{prop}}$ and $C_{50E}^{\text{remi}}$ are the concentrations at half the maximum effect for each drug,
- $U_{50}(\theta) = 1 - \beta \theta + \beta \theta^2$ where $\theta = U_{\text{prop}}/(U_{\text{prop}} + U_{\text{remi}})$ and $\beta$ is a given constant,
- $\gamma$ is the steepness of the concentration-response relation.

The values of the coefficients are given in table 2 in the appendix A.1.

### 2.2.4 Non-linear global model

The global model, as described by figure 2, can be computed from equations (2.1), (2.2) and (2.3), yielding:

\[
\begin{align*}
\text{Linear part:} \quad & \quad \dot{x}(t) = A_{\text{nl}} x(t) + B_{\text{nl}} u(t) \\
& \quad c_{\text{prop}}(t) = C_{\text{nl}} x(t)
\end{align*}
\]

Non-linear part:

\[
y(t) = f(d(t), c_{\text{prop}}^{\text{nl}}(t))
\]

where:

- $t$ is the time variable.
- $u$ is the controlled variable, the propofol infusion rate ($u = r_{\text{prop}}$) given in [mL/h].
- $y$ is the output, the DoA level given in [%].
- $d$ is the disturbance ($d = c_{\text{remi}}^{\text{remi}}$), the remifentanil effect concentration given in [µg/mL].
- $c_{\text{prop}}^{\text{prop}}$ is the propofol effect concentration given in [µg/mL].

\[
x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad A_{\text{nl}} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad B_{\text{nl}} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \quad \text{and} \quad C_{\text{nl}} = \begin{bmatrix} 0 & 0 & C_2 \end{bmatrix}.
\]

### 2.2.5 Linearized global model

To linearize the model, equilibrium values are considered. The linearization point, denoted $\tilde{y}$, specifies the DoA level to consider the linearized model around, for instance $\tilde{y} = 50\%$. The linearized model only considers the variations around this equilibrium value, denoted $\Delta y$. The absolute value $y$ is thus computed by: $y = \Delta y + \tilde{y}$. The propofol infusion rate equilibrium is $\tilde{u}$ and the remifentanil effect concentration equilibrium is $\tilde{d}$.

Reversing the Hill equation leads to a Hill equation working point, denoted $[\tilde{d} \quad c_{\text{prop}}^{\text{remi}}]^{T}$. The system working point is then $[\tilde{d} \quad \tilde{u}]^{T}$.

The linearized global DoA model is given by:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + B\Delta u(t) \\
\Delta y(t) &= Cx(t) + D_3 \Delta d(t)
\end{align*}
\]

where:

- $\Delta u = u - \tilde{u}$, $\Delta y = y - \tilde{y}$, $\Delta d = d - \tilde{d}$,
- $A = A_{\text{nl}}$, $B = B_{\text{nl}}$, $C = \left( \frac{\partial f}{\partial c_{\text{prop}}^{\text{prop}}} \right) C_{\text{nl}}$ and $D_3 = \frac{\partial f}{\partial d} |_{d=c_{\text{prop}}^{\text{remi}}}$.

One of the MPC controller configurations designed in section 3 involves a patient with an extra integrator. A DoA model with an integral state is computed in [8].
2.3 Open-loop simulations

2.3.1 Non-linear model

Figure 3 plots the DoA level from the non-linear DoA model compared to data from clinical anaesthesia for two patients.

![Figure 3 - Open-loop simulation - Clinical DoA compared to modeled DoA (non-linear DoA model)](image)

It has to be noticed that the data from the anaesthesia are noisier than the ones from the model. A way to add noise to the model has been briefly described in [8, 9]. Open-loop results are in accordance with reality.

2.3.2 Linearized model

It makes no sense to compare the linearized model with the clinical data since the linearized model is only valid around its linearization point. Thus, the linearized model is compared to the non-linear model for the same scenarios in figure 4.

![Figure 4 - Open-loop simulation - Linearized model evaluation](image)

It can be seen that for moderate steps on the control signal, the response of the linearized model is close to the non-linear one.

The remifentanil effect concentration does not influence significantly the DoA level, and a very large step on this drug affect not much the output. This result has to be taken with care because, only the effect concentration is considered here, whereas it is the infusion rate that really enters the patient.
3.1 Structures introduction

3.1.1 Motivation

MPC is a model-based control. The inner-model used to compute the MPC control law should be linear. But using a linearized model in the controller leads to the problem of knowing the working point. A given linearization point \( \bar{y} \) corresponds to a given working point \([\bar{u} \quad \bar{d}]^T\). If the working point is not well identified, the patient model (which is not linear) is fed with biased values.

When the model is linearized (from a given linearization point), a Hill equation working point is identified \((c^\text{prop}_e \quad \bar{d})\). The difficulty resides in finding the propofol infusion rate equilibrium \(\bar{u}\), from the effect concentration value \(c^\text{prop}_e\).

Three methods to solve this problem are considered. They are called the “basic solution”, the “exact solution” and the “approximate solution” and are described hereafter.

3.1.2 Basic solution

The basic solution (figure 5) uses a guess for \(\bar{u}\). This solution suffers from the lack of accuracy on the guess of \(\bar{u}\), denoted \(\hat{\bar{u}}\). The two other solutions use structures that avoid an a priori value for \(\bar{u}\).

![Figure 5 – Closed-loop system - Basic solution](image)

For this solution and the others, the MPC controller works around zero, taking as input \(\Delta y\) (as a measured output), \(\Delta r^*\) (as the reference) and \(\Delta d\) (as a measured disturbance) and providing as output \(\Delta u\).

3.1.3 Exact solution

The so called “exact solution” (figure 6) introduces an integrator between the output of the MPC block and the input of the patient model.

![Figure 6 – Closed-loop system - Exact solution](image)
3 MODEL PREDICTIVE CONTROL STRUCTURES STUDY

The integrator compensates for the static error introduced by the implicit choice: \( \hat{u} = 0 \). Indeed, an integrator allows to get error-free tracking. On figure 6, the integrator is represented as an independent block. For simulation purposes, this integrator is embedded in the patient model. In this case, the prediction model in the MPC controller also takes into account the integrator (using the linearized augmented DoA). In a real application, the integrator should be added to the controller.

3.1.4 Approximate solution

The second way to avoid the use of \( \hat{u} \) is to add an integral action in a feed-back between the reference and the output (figure 7).

The idea is the same as in the previous solution, an integrator on the error signal is used to get error-free tracking. But this time, the integrator is only fed with the reference and the output, providing directly the equilibrium of the control signal. The integration time, \( T_i \), should be large since the integral action should only be visible on steady state.

The difference with respect to the exact solution consists in the fact that the design of the MPC controller does not take into account the integrator. This has the advantage that the on-line computational load is smaller, but the disadvantage of possible dynamic interactions between the MPC controller and the integrator.

3.1.5 Extra components

The technical report [9] introduces extra blocks to the solutions presented so far: a pre-filter on the reference, a filter on the output in case of measurement noise and two schemes to have a feed-forward action.

3.2 Controllers configuration

3.2.1 Introduction

MPC controllers include several features which need to be configured:

**Optimization:** MPC controllers are based on an optimal control problem. Therefore, the weights used in the cost function should be tuned. The cost function for the MPC block in MATLAB [10] can be simplified to the following equation:

\[
J(z, \varepsilon) = \|Y - R\|^2 + \rho \|\Delta U\|^2 + \rho \varepsilon^2
\]

(3.1)

where:

- \( Y \) is the vector of values of the output over the prediction horizon,
- \( R \) is the vector of values of the reference over the prediction horizon,
- \( \Delta U \) is the vector of values of the rate of the control signal over the control horizon,
• $\rho$ corresponds to the ratio between the weight associated to the error and the weight associated to the control,
• $\rho_\epsilon$ is the weight factor on the slack variable (used to penalize the violation of the constraints),
• $\epsilon$ is the slack variable, a variable to turn the inequality into an equation, it allows the constraints to be violated in a certain amount.

An alternative cost function (derived in section 3.2.5) consists in using exponential weights, introducing an additional parameter $\bar{R}$.

**State estimation**: the current state vector should be available to solve the cost function (3.1). Since it is often not the case, MPC controllers use estimation. It is assumed that a linear model on the form

$$x(k + 1) = Ax(k) + Bu(k)$$
$$y(k) = C_yx(k)$$
$$z(k) = C_zx(k) + D_zu(k)$$
$$z_c(k) = C_zc x(k) + D_zc u(k)$$

is available. Here $y(k)$ is the measured output, $z(k)$ the controlled output, $u(k)$ the control signal, $x(k)$ the state vector and $z_c(k)$ the constrained outputs. This model is called “prediction model” and given in discrete time since simulations are run in discrete time with a sampling time $T_s = 5$ s.

Note that in the MATLAB MPC block, the model is a simple LTI system. Controlled output and constraints are specified in the controller configuration.

**Constraints**: MPC controllers allows to define constraints on the control signal and its rate and on the output, such that:

$$\Delta u_{\text{min}} \leq \Delta u(k) \leq \Delta u_{\text{max}}$$
$$u_{\text{min}} \leq u(k) \leq u_{\text{max}}$$
$$z_{\text{min}} \leq z_c(k) \leq z_{\text{max}}$$

**Receding horizon control**: in MPC, the optimal problem is considered over the prediction horizon ($T_p$), and the number of unknowns of the control signal to solve depends on the control horizon ($T_c$).

The parameter choices are introduced in the following sections. Comparative studies on the following parameter choices and an extra sub-section on the feed-forward action are given in [9].

### 3.2.2 Prediction models

The basic prediction model is given by equation (2.5). But, for the exact solution, since an extra integrator is added to the controller, the augmented Doa model should be used instead.

Moreover, since the real DoA level is very noisy, a filter can be used on the measured output. This filter can be added to the prediction model (normal and augmented). These cases are detailed in [9].

### 3.2.3 Constraints

The DoA signal range is between 0 and 100% (initial signal is at 97.7%) and the propofol infusion rate cannot be negative (which would have the physical meaning to take out propofol from the patient). These constraints are summed up in table (1).
### Table 1 – Constraints of the MPC controller

<table>
<thead>
<tr>
<th>Variables</th>
<th>Units</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u$</td>
<td>[mL/h]</td>
<td>0</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>$\frac{du}{dt}$</td>
<td>[mL/hs]</td>
<td>$-\infty$</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>DoA</td>
<td>[%]</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

These are the basic constraints. In reality, the maximal infusion rate and the changes in the infusion rate are constrained by the equipment, but these bounds are very high and are never reached in practice.

After the derivation of the MPC models, which is covered in this section, the next stage of the work is to couple MPC controllers with Supervised Switched Multiple Model Predictive Control (SMMPC). In this perspective, it would be preferable to have the MPC controllers without constraint so that they can easily be turned into Linear Quadratic designs. It seems reasonable to not constrain the controllers since the bounds defined above should not be reach with normal conditions.

Therefore, no constraints are set to the controllers in the default cases.

#### 3.2.4 Time horizons

The influence of the time horizons on the performances of the controllers has been tested in [9]. Simulations have been run with fixed parameters, except the prediction horizon ($T_p$) and the control horizon ($T_c$). Simulations are compared by computing the so called “standard deviation of the tracking error” which is the mean of the power of the difference between the reference and the output of the system (equation (3.4)). It is called “error standard deviation” hereafter.

$$\text{error standard deviation} = \frac{\sum_{k=0}^{N}(y(k) - r^*(k))^2}{N}$$  \hspace{1cm} (3.4)

where $N$ is the number of samples in the simulation of duration $t_{\text{max}}$ such that $t_{\text{max}} = N \times T_s$ [s].

#### 3.2.5 Weights in the cost function

The weight factor on the slack variable ($\rho_{\varepsilon}$), used to penalize the violation of the constraints will not be discussed here, since no constraint has been set up.

Two cases can be considered: a fixed $\rho$ parameter (as appearing in cost function (3.1)) or variable weights.

The aim of having variable weights is to force the latest compilation of $\Delta u$ and $y$ samples to be closer to zero. Therefore this lowers the influence of predicted data that might be expired.

It is shown [11] that by taking the cost function as equation (3.5), discrete poles in closed-loop are ensured to be contained in a circle or radius $\bar{R}$.

$$J(z, \varepsilon) = \sum_{k=0}^{p-1} \left( \frac{1}{\bar{R}} \right)^{2k} \left( [y(k+1) - r(k+1)]^2 + \rho \Delta u(k)^2 \right)$$  \hspace{1cm} (3.5)

The influence of the weights on the controller performances have been tested in [9]. Two indicators are computed to compare the simulations. The first is the error standard deviation given by equation (3.4). The second, called “standard deviation of the manipulated variable” and shortened “control standard deviation” hereafter, is the mean of the power of the manipulated variable (equation (3.6)).

$$\text{control standard deviation} = \frac{\sum_{k=0}^{N} u(k)^2}{N}$$  \hspace{1cm} (3.6)
Two main observations have to be done. Firstly, the choice of $\rho$ is a trade-off between the tracking error and the power of control allowed. Secondly, in the case of variable weights, the value given to $\bar{R}$ cannot be too low otherwise the optimal problem is unsolvable.

### 3.3 Closed-loop simulations

The aim of this section is to compare the three controller configurations and to observe the basic solution in realistic conditions.

The MPC controller configuration used for the different solutions is the same. The linearization point is taken at 50%, the filter (used to filter the output in case of measurement noise) is not included in the prediction model, the controller is not constrained, the prediction horizon is set to 50 s, the control horizon to 10 s, fixed weight are used and $\rho$ is set to 1.5.

The patient model simulated uses the non-linear DoA model.

#### 3.3.1 Tests around the linearization point

Figures 8 and 9 gives the results with the default configuration, for each solution.

![Figure 8](image1.png)

**Figure 8 – Closed-loop simulation - DoA and reference**

![Figure 9](image2.png)

**Figure 9 – Closed-loop simulation - DoA and reference**

As it can be seen on figure 9(b), dynamics are difficult to consider in presence of noise. Thus, only simulation without noise are considered hereafter.

A bump appears before $t = 500$ s on figure 8(b). This is due to the fact that the integrator in the non-linear model has to be initialized with correct initial conditions to be at the equilibrium point.
The problem is that the initial condition of the integrator (to keep the system at the linearization point in the beginning of the simulation) should be $\bar{u}$, which is not available in practice.

From these results, the exact solution seems to be the best solution. However, as it is detailed in [9], this solution is less robust. The two other solutions give also good results.

The response from the basic solution is a bit better than the approximate solution. But it has to be noticed that the prediction model embedded in the MPC controller is exactly the linearization of the patient model simulated. Thus the guessed value $\hat{u}$ in the basic solution is very closed to the real value. This could not be the case with a real patient and the approximate solution could give better results than the basic solution.

### 3.3.2 Realistic conditions

In this section, the basic solution is tested in realistic conditions. This means that the DoA model is not considered only around the linearization point. The DoA level starts at 97.7% as in a real surgery.

The model is led to its linearization point by a bolus of propofol. When the model is near from its linearization point, the MPC controller is plugged on. So, at a given time $t_{control}$, the propofol infusion rate is provided by the MPC controller.

These conditions yield the figures 10(a) (patient 1) and 10(b) (patient 2). To test the controller, steps are performed in the negative and positive sides. Upper plots do not include noise whereas downer plots.

![Figure 10](image)

**Figure 10** – Closed-loop simulation in realistic conditions - DoA and reference

Note that the controller is not at the correct state when plugged in. This results to a bump in the output after the switching time ($t_{control} = 1500$ s for patient 1 and $t_{control} = 3000$ s for patient 2).

It has to be noticed that even with a linearized prediction model, the MPC controller is able to follow the decreasing references with good performances. The controller is also able to follow small positive steps without important overshoot. But oscillations appear on the output for large positive steps. These points are detailed in [9]. It is also shown in [9] that the total amount of drug provided by the MPC controller is in the same range as the amount injected by the anaesthetist in a real surgery.
4 SWITCHED MULTIPLE MODEL PREDICTIVE CONTROL ALGORITHM

4.1 Controller introduction

4.1.1 Switched Multiple Model principle

A Switched Multiple Model (SMM) controller includes two major parts:

- a supervisor, that observes the state of the plant and chooses which of the controllers should be plugged to the plant,
- a bank of controllers, of which one is plugged to the plant.

The general structure of a typical SMM controller is depicted on figure 11.

![Figure 11 – Typical SMM controller global structure](image)

The plant input is the control signal $u$, corresponding to the output $u_i$ of the current controller $C_i$ chosen by the supervisor ($i \in \{1, 2, \ldots, N\}$). The output is $y$ and the plant is subject to the disturbance $d$.

The supervisor is fed with the input and the output of the plant. The bank of controllers is also fed by these two signals and the reference $r$.

However, this general structure differs in the case considered here since the models embedded in the supervisor and in the bank of controllers (respectively estimation and prediction models) are linearized around different linearization points. In this case the supervisor and the controllers should be fed with the incremental values ($\Delta u$ and $\Delta y$) in order to run the models around their linearization points.

By definition, the incremental values ($\Delta u$, $\Delta y$, $\Delta d$) describe the difference between the equilibrium values ($\bar{u}$, $\bar{y}$, $\bar{d}$) and the absolute values ($u$, $y$, $d$) such that:

$$
\begin{align*}
  u &= \bar{u} + \Delta u \\
  y &= \bar{y} + \Delta y \\
  d &= \bar{d} + \Delta d
\end{align*}
$$

(4.1)

However, different approaches can be considered to compute the incremental values. The main problem is that the models are not linearized around the same point, therefore no equilibrium value is common for all the models\(^3\). Sub-section 4.1.2 explains four approaches and chooses the suitable one for the current situation.

\(^3\)Moreover, section 3.1.1 explains that the propofol infusion rate equilibrium value is not well known and presents three ways to handle the $u$ signal.
4.1.2 Controller structure definition

An unseen side of the developed algorithm is the coupling of Switched Multiple Model with models linearized at different points. Having several models linearized around different points may cause problems in the supervisor (there may be a difficulty to compare the estimated outputs and the measured output and thus to choose the best controller) and in the controller (a suitable trajectory should be computed whatever the linearization point is).

Globally, two approaches can be considered. The first one consists in using incremental values ($\Delta y$ and $\Delta u$) common to all the models (i.e. only one incremental value is computed outside the Supervisor and Bank of controllers blocks). The second uses only the absolute values ($y$ and $u$) and lets each model compute the incremental value corresponding to its linearization point (i.e. as many incremental values as the number of estimation and prediction models are computed inside the Supervisor and Bank of controllers blocks).

The difficulty in the first approach, used in attempts one and two hereafter, is to compute the output incremental value $\Delta y$. Indeed, by definition, $\Delta y$ is given by $\Delta y = y - \bar{y}$, but since different linearization points $\bar{y}$ are considered, the definition cannot be used for all the models. Attempts one and two and the selected solution present three different ways to compute $\Delta \hat{y}$, the approximation of $\Delta y$.

The difficulty in the second approach is to define the equilibrium value of the propofol infusion rate $\bar{u}$. Indeed, as explained earlier, this value is not well known. But this difficulty inheres in the DoA problem considered here. The algorithm is a bit more complex in this approach since each model must hold its equilibrium values and compute the incremental signals.

Each solution presents other difficulties which are explained in the following paragraphs.

**First attempt** The first attempt (Figure 12) uses the following definition to compute the approximation of $\Delta y$ in the first approach:

$$\Delta \hat{y}_1(k) = y(k) - r(k)$$

The advantage of choosing a variable value ($r(k)$) to compute $\Delta \hat{y}_1(k)$ is that the incremental value is adapted according to the current state of the system. The drawback is that the linearization point is implicitly taken not fixed.

The drawback is however too much incapacitating since it makes the supervisor fail to guess the best model because of an offset introduced by the reference when it is not at the equilibrium point, as it is proved in [12].

This solution is not usable since the supervisor always fails when the reference is not at the linearization point. Moreover it is incapacitating to consider an equilibrium point on the reference since the non-linear patient model does not present such equilibrium value (this point is detailed in the second attempt).
**Second attempt** The second attempt tries to solve the problem appeared in the first attempt by using a fixed value for \( \tilde{\bar{y}} \) (the approximation of \( \bar{y} \)):

\[
\Delta \tilde{y}_2(k) = y(k) - \bar{r}
\]

This definition corresponds to the structure given in figure 13.

![Figure 13 – SMM controller global structure (attempt two)](image)

As explained in the comment of the first attempt, this choice has the advantage to use a fix linearization point, preventing the supervisor to fail due to a variable offset in the estimations. A drawback is that a value must be defined as the equilibrium point for the reference (\( \bar{r} \)).

This drawback is not as restrictive as the one from the first attempt. But it imposes to add an equilibrium point to the overall system, which is not in accordance with reality.

Moreover, using a common linearization point (\( \bar{r} = \tilde{\bar{y}} \)) for all the models imposes the models linearized far from this point to work far from their linearization point. This thus gives worst results when the effective reference is far from its equilibrium. Indeed, this means that the control is only optimal around the common linearization point, and the adaptive part of the control design is simply lost.

This solution has not been retained since it imposes unjustified limitations that another structure could avoid. However, it could have been used, whereas the solution introduced in the first attempt.

**Third attempt** The third attempt uses a mix of the two approaches. The models are fed with the absolute value of the output (\( y \)) and the incremental value of the control signal (\( \Delta u \)), as depicted on figure 14.

![Figure 14 – SMM controller global structure (attempt three)](image)

This way, the difficulties from both approaches are avoided, nor common linearization point \( \tilde{\bar{y}} \) neither propofol infusion rate equilibrium \( \bar{u} \) have to be defined. Each model holds its linearization point (different from one model to another) and the comparison in the supervisor is made using the
absolute value, directly measured from the plant.

This solution gives good results to guess the suitable model in open-loop, but as soon as the loop is closed, the supervisor fails to guess the closest model to the plant. This seems to be due to an offset in the estimation caused by the common $\Delta u$ incremental signal whereas the output incremental value $\Delta y_i$ is particular to each model. This makes this solution unusable.

This attempt is detailed in the appendix of [12].

**Selected solution** The selected solution breaks with the attempts and does not use nor any equilibrium value to compute $\Delta \tilde{y}$ nor the absolute value $y$. It considers that the incremental value $\Delta \tilde{y}$ is the derivation of the absolute value instead of the difference with respect to the equilibrium:

$$\Delta \tilde{y}(k) = y(k) - y(k-1)$$

whereas $\Delta y(k) = y(k) - \bar{y}$

It has been seen during simulations that this solution imposes the $\Delta u(k)$ value entering the supervisor and the bank of controllers to be computed in the same way. Therefore, the $\Delta u(k)$ value designating the optimal control computed by the controller chosen by the supervisor is denoted $\Delta u^*(k)$ to prevent confusion.

$$\Delta u(k) = u(k) - u(k-1)$$

$$\Delta u^*(k) = \Delta u_i(k)$$

where $i$ is the index of the best controller at time $k$.

This is depicted on figure 15.

Hereafter, and since the notation $\Delta y$ defining the exact output incremental value is not used anymore, the notation $\Delta \tilde{y}$ is used as a shortcut for the approximation $\Delta \tilde{y}$. The expression “incremental value” is kept to designate the new $\Delta y$ with abuse of language.

Identically for the control, the expression “incremental value” is kept to designate $\Delta u$ with abuse of language.

This solution avoids drawbacks from the previous attempts. It gives generally good results as it is explained hereafter but some restrictions have to be noticed.

- This solution suppresses the information about the steady state and only the dynamics are represented. This makes the supervisor a bit less precise since the plant moves slowly.
- In presence of output additive noise, it appears that one of the model in the bank is always prefered among the others.
- Since the patient model is slow, the difference between two models is tiny at each iteration. That may imply a lot of switchings in the supervisor because a suitable model is hard to guess.
This problem is however not particular to the selected solution but to the type of the plant. Possible improvements can be considered like letting each model use their own data for several iterations or increasing the sampling time.

4.2 Algorithm derivation

This section gives the theoretical background computed to develop the algorithm. First, the DoA model is turned to a temporal form and a bank of models is derived, then the supervisor is designed and finally the MPC control law is developed. The final sub-section introduces the pseudo-code corresponding to the algorithm.

4.2.1 Bank of models

The algorithm is designed to simulate the patient. The DoA model previously derived needs to be turned to a temporal form to be easily simulated. This form is also needed for the computation of the estimation and prediction models. And finally, a bank of model should be available in the algorithm to make the comparison with the plant.

**DoA model in temporal form** The state space representation of the DoA model (equation (2.4)) can be turned to a discrete transfer function and then, to the following temporal form:

\[ c_{\text{prop}}^p(k) = -a_1' c_{\text{prop}}^p(k-1) - a_2' c_{\text{prop}}^p(k-2) - \ldots - a_n' c_{\text{prop}}^p(k-n) + b_0' u(k-1) + b_1' u(k-2) + \ldots + b_m' u(k-(m+1)) \]

(4.2)

The details of the coefficients and the computations can be found in [12].

The same operation is done for the linearized model:

\[ y(k) = -a_1 \Delta y(k-1) - a_2 \Delta y(k-2) - \ldots - a_n \Delta y(k-n) + b_0 \Delta u(k-1) + b_1 \Delta u(k-2) + \ldots + b_m \Delta u(k-m-1) + \bar{D} \Delta d(k) + \bar{y} \]

(4.3)

Derivation of the bank of models The models in the bank are generated from the DoA model where some parameters are taken different from their default value. Five parameters vary: \( v_c, k_{12}, k_{e0}, c_{\text{SOE}}^p \) and \( \gamma \) (see table 2 in appendix A.1 for parameters description). Two values are used for each parameter leading to ten models, linearized around four points (97.7, 80, 60 and 40 %), totaling forty models.

If a relevant number of clinical data from real anaesthesia were available, the bank of models could have been build from these data. The models would have been then clustered according to some of their characteristics. The resulting bank of models would have cover a range of patient in accordance with reality.

This approach has not been considered through lack of time and clinical data. Indeed it implies a deep study of the clinical data and needs methods to compare the models and to build a relevant clusterization. The purpose of the work was firstly to develop the algorithm and validate it on suitable models.

4.2.2 Switched Multiple Model implementation

The supervisor embeds several estimators and compares the estimated outputs to the measured output and chooses the best estimated. In the final version of the algorithm, a controller is associated to each estimator as it is shown on figure 16.
The details of the computation are not given here (and they are available in [12]). The estimation of the incremental value $\Delta y(k)$ for the model $i$ from the bank is given by:

$$\Delta \hat{y}_i(k) = \Theta_i^T x_E(k)$$  \hspace{1cm} (4.4)

where the parameters vector $\Theta_i$ and the shared state for estimation $x_E$ are defined by:

$$\Theta_i \triangleq \begin{bmatrix} y_1 - a_{i,1} \\ \vdots \\ y_n - a_{i,n} \\ b_{i,0} \\ \vdots \\ b_{i,m} \end{bmatrix}; \quad x_E(k) \triangleq \begin{bmatrix} \Delta y_f(k-1) \\ \vdots \\ \Delta y_f(k-n) \\ \Delta u_f(k-1) \\ \vdots \\ \Delta u_f(k-(m+1)) \end{bmatrix}$$

where:

- $n$ is the dimension of the system, the order of the denominator of the transfer function describing the linearized DoA model, and $m$ is the order of the numerator. By definition, the nominal delay is 1, i.e. $n - m = 1$.
- $\gamma_j$ (with $j \in \{1, \ldots, n\}$) are the coefficients of the characteristic polynomial $A_0^*$ such that $A_0^* (q^{-1}) \triangleq 1 + \sum_{i=1}^{n} \gamma_i q^{-i}$ ($q^{-1}$ is the backward shift operator).
- $a_{i,j}$ (with $j \in \{1, \ldots, n\}$) and $b_{i,k}$ (with $k \in \{0, \ldots, m\}$) are the coefficients of the $i$th model from the bank defined in equation (4.3).
- $\Delta y_f$ and $\Delta u_f$ are the signals $\Delta y$ and $\Delta u$ filtered by $A_0^*$ as follows:

$$\Delta y_f(k) \triangleq \frac{1}{A_0^*} \Delta y(k) = - \sum_{j=1}^{n} \gamma_j \Delta y_f(k-j) + \Delta y(k)$$  \hspace{1cm} (4.5)

$$\Delta u_f(k) \triangleq \frac{1}{A_0^*} \Delta u(k) = - \sum_{j=1}^{n} \gamma_j \Delta u_f(k-j) + \Delta u(k)$$  \hspace{1cm} (4.6)

Once the estimations for each model from the bank have been computed, the supervisor computes the error from these estimations to the measured output (equation (4.7)) and the performance evaluator for each model (equation (4.8)) to choose the best model at the current time.

$$\epsilon_i(k) \triangleq \Delta y(k) - \Delta \hat{y}_i(k)$$  \hspace{1cm} (4.7)

$$\pi_i(k) = \lambda \pi_i(k-1) + (1-\lambda) \epsilon_i^2(k)$$  \hspace{1cm} (4.8)
The parameter $\lambda$ (tuned between 0 and 1) is used to set a weight on the previous model used. The larger $\lambda$ is, the less the model tends to be changed. Case $\lambda = 1$ should be avoided as it means that the model should never be changed.

The model ($\Sigma$) used at time $k$ is the one presenting the lowest performance evaluator.

$$\Sigma(k) = \Sigma_i \text{ such that } i \Rightarrow \pi_i(k) = \min_{j \in \{1, \ldots, N\}} \pi_j(k)$$

To ensure stability in a switching context, it has been shown [13] that a controller, once plugged to the plant, must be kept a minimum amount of time (called Dwell time, denoted $T_D$). However the proof given in [13] cannot be simply applied to the algorithm developed here.

Anyway, the Dwell time is used here as it seems obvious that even if the current controller is not the best one, the overall behavior is better with this controller than changing too often the controller. The Dwell time is taken at least as large as the order of the system.

### 4.2.3 Model Predictive Control implementation

This sub-section only gives the results, details of the computations are available in [12].

The figure 17 depicts the main aspects of MPC principle: the prediction, the optimization, the receding horizons and the tracking trajectory.

![MPC Principles in a Temporal View](image)

**Figure 17 – MPC principles in a temporal view**

The best model at time $k$ choosen by the supervisor is denoted $\Sigma_i = \frac{B_i^*}{A_i^*}$. From this model, a number of $T_p$ of $K$ steps ahead predictors are computed. $T_p$ is the prediction horizon. The prediction of $\Delta y$ at time $k + K$ knowing $k$ is given by:

$$\Delta \hat{y}_i(k + K|k) = \sum_{j=1}^{K} w_{i,j} \Delta u_f(k + K - j) + \Gamma_x T x_C(k) \tag{4.9}$$

where $w_{i,j}$ and $\Gamma_x$ are coefficients and matrices computed from the model $\Sigma_i$ and the characteristic polynomial $A_0$, $\Delta u_f$ is the filtered control signal and $x_C$ is a compilation of past incremental values of the output and the control signal (both filtered).

Then, the predictors are ordered in the following matrix form:

$$\Delta \hat{Y}_i|_{k+1}^{k+T_p} = W_i^r \Delta U_f|_{k}^{k+T_p-1} + \Gamma_i^T x_C(k) \tag{4.10}$$

where $\Delta \hat{Y}_i$, $\Delta U_f$, $W_i^r$, and $\Gamma_i^T$ are the compilations of $\Delta \hat{y}_i$, $\Delta u_f$, $w_{i,j}$ and $\Gamma_{i,K}$ respectively, from equation (4.9).
The cost function of the optimal problem is given by:

$$J_i(k, k + T_p) = \left\| \Delta \hat{Y}_i^{[k+T_p]} - \Delta Y_i^{[k+T_p]} \right\|_Q^2 + \left\| \Delta U_f^{[k+T_p - 1]} \right\|_R^2$$

(4.11)

where $\Delta Y_i^*$ is the tracking trajectory to be tracked over the prediction horizon, as it can be seen on figure 17 and $Q$ and $R$ are the weighting matrices associated to the tracking and the control.

The optimal control which minimized the cost function is given by:

$$\left\{ \begin{array}{l}
\Delta U_f^{[k+T_c - 1]}_i \\
M = W_i^{T_c} Q W_i^{T_c} + R
\end{array} \right. = -M^{-1} W_i^{T_c} Q (\Gamma_i^T x_C(k) - \Delta Y_i^{[k+T_p]} )$$

(4.12)

The first element of the optimal solution $\Delta U_f^*$ can be unfiltered and used as the incremental control signal at time $k$.

**Pseudo-code of the algorithm** All the computations given in the previous sub-sections are used to derive the algorithm. A pseudo-code of the algorithm is given in the appendix A.2 (code 1). This pseudo-code is designed as an algorithm to be implanted into a microcontroller, which the structure is given in figure 18.

![Figure 18 – Algorithm in a temporal view](image)

But since the algorithm is meant to be run in MATLAB, the use of interruptions has been replaced by a loop. This implies that the `Wait` statement is not used. Moreover, since the patient is simulated in the algorithm itself, the `Get signals` step is replaced with the simulation of the patient.

### 4.3 Closed-loop simulations

Figure 19 plots a simulation where the models in the bank are linearized around different points ($\bar{y}_1 = 40 \%$, $\bar{y}_2 = 60 \%$ and $\bar{y}_3 = 80 \%$). The patient model simulated is exactly one of these models. The reference goes from 40 \% to 80 \% and the patient model simulated is changed according to the reference.
The results are very good in this situation. The reference is perfectly tracked and the supervisor correctly guesses the right model. Two negligible errors appear in figure 19(b).

[12] shows that by taking the reference as a series of steps instead of a ramp makes the output oscillate, but the supervisor still guesses perfectly the correct model.

Last figure (20) simulates the algorithm in realistic conditions. Only one patient model is simulated, but it is non-linear (it has the index 1). The bank is composed of four models, linearized around different points ($\bar{y}_2 = 97.7\%$, $\bar{y}_3 = 80\%$, $\bar{y}_4 = 60\%$ and $\bar{y}_5 = 40\%$). The patient starts at a conscious state (DoA = 97.7%) and is led to the unconscious state (DoA = 40%) with a bolus of propofol, then the controller is plugged to the patient. The patient is waken up at the end of the simulation with a going up ramp on the reference.

Since the bolus cannot led the patient to the exact DoA level needed for the surgery, some oscillations appear once the controller is plugged (between time $k = 1000$ s and $k = 5000$ s) on figure 20(a). The upward slope is also tracked with some oscillations. Globally, the response of the controller is good.

The problem in this simulation is the guesses of the supervisor. As it can be seen on figure 20(b), the supervisor is always switching between the models in the bank. Between time $k = 2500$ s and $k = 8000$ s, the 5th model (linearized around 40%) seems to be preferred by the supervisor (and it is the right guess), but the supervisor still experiments other models.

More tests would have been necessary to really characterize the situation, and explaining why the supervisor (working well so far) is totally lost in this test. The models considered here only differ from their linearization points, they are maybe too close ones to each others. Also, the supervisor seems to have problem to compare the non-linear plant and the linearized models.

Note that the tracking would have been better with fewer switchings.
The work done during this internship was really interesting, from a personal point of view, but it was also useful to the IDeA project. Although the two first steps (designing the DoA model and evaluating the MPC structures) have been completed in time, unexpected difficulties in the choice of the structure of the algorithm (which have been detailed with the three attempts and the selected solution) prevented me from running deep tests on the algorithm. However, the IDeA project is still continuing for the next two years and I will follow with great interest the results to come. Objectives in the framework of the internship have been reached.

This internship has been a rewarding experience in many ways. Firstly, it allowed me to discover the daily work in a laboratory. I could attend with great pleasure many meetings or workshop.

Moreover, M. Lemos summed up my work into a paper proposed and accepted for the 2008 International Conference of the IEEE Engineering in Medicine and Biology Society in Vancouver.

This internship has been useful to experiment directly the knowledge learnt from my lectures in Lund during the first semester. The subject covers perfectly my course option and it is also in accordance with the areas I am proud to contribute in and to bring my personal energy.

In a personal point of view, this internship widened the opportunities in my young engineer career. Even if I would like to develop myself in an industrial context now, I would like to keep the contact with the research area and I have thus decided to search for a PhD thesis in a company. I could enlarge my knowledge and improve my work habits. I am now better in controlling my time and organizing my tasks list.

I draw a very positive conclusion from this internship.
## References


### A.1 Coefficients in the DoA model

This section gives a table that sum up the values of the coefficients used in the DoA level.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_c$</td>
<td>volume coefficient for the compartment model of the PK model</td>
<td>0.228</td>
<td>[L/kg]</td>
</tr>
<tr>
<td>$k_{10}$</td>
<td>transfer coefficient for the compartment model of the PK model</td>
<td>0.119/60</td>
<td>[s$^{-1}$]</td>
</tr>
<tr>
<td>$k_{12}$</td>
<td>transfer coefficient for the compartment model of the PK model</td>
<td>0.112/60</td>
<td>[s$^{-1}$]</td>
</tr>
<tr>
<td>$k_{13}$</td>
<td>transfer coefficient for the compartment model of the PK model</td>
<td>0.0419/60</td>
<td>[s$^{-1}$]</td>
</tr>
<tr>
<td>$k_{21}$</td>
<td>transfer coefficient for the compartment model of the PK model</td>
<td>0.055/60</td>
<td>[s$^{-1}$]</td>
</tr>
<tr>
<td>$k_{31}$</td>
<td>transfer coefficient for the compartment model of the PK model</td>
<td>0.0033/60</td>
<td>[s$^{-1}$]</td>
</tr>
<tr>
<td>$k_e$</td>
<td>pharmacodynamic coefficient for linear the PD model</td>
<td>0.25/60</td>
<td>[s$^{-1}$]</td>
</tr>
<tr>
<td>$C_{remi}^{50E}$</td>
<td>concentration at half the maximal effect (for remifentanil)</td>
<td>49.9998</td>
<td>[µg/mL]</td>
</tr>
<tr>
<td>$C_{prop}^{50E}$</td>
<td>concentration at half the maximal effect (for propofol)</td>
<td>2.3708</td>
<td>[µg/mL]</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>the steepness of the concentration-response relation</td>
<td>1.2578</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>a constant</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$E_0$</td>
<td>effect at zero concentrations</td>
<td>97.7</td>
<td>[%]</td>
</tr>
</tbody>
</table>

*Table 2 – Coefficients in the DoA model*
A.2 Pseudo-code of the algorithm

This section gives a the pseudo-code of the SMMPC algorithm.

**Code 1 – Pseudo-code of the algorithm**

```plaintext
for time = 0 to final time do
    /* Get signals at time k */
    get plant output  // compute y(k) from past values of y and u
    get reference  // return r(k)
    /* Compute control signal */
    // Supervisor
    compute the estimated outputs  // return delta_y_hat(k) for each models
    knowing xE(k-1)
    compute the performance evaluators  // return p_i(k) for each model
    choose the best model  // return the index of the best model
    /* Compute the control law */
    load the predictors parameters  // return W_Tc and GAMMA
    // Predictors
    update xC  // return xC(k) with current value of delta_yf(k)
    compute trajectory  // return delta_Y_star
    compute M
    compute Uf_star
    extract and unfilter current control signal  // return delta_u(k)
    compute absolute value  // return u(k)
    /* Send data */
    send control signal
    /* Update the state of the estimators and controllers */
    compute the filtered signals  // return delta_yf(k) and delta_uf(k)
    compute the state of the estimators  // return xE(k+1) = [delta_yf(k) ...
    delta_yf(k-n+1) delta_uf(k) ... delta_uf(k-m)] but the state will
    only be used at the next iteration, yielding: xE(k) =
    [delta_yf(k-1) ... delta_yf(k-n) delta_uf(k-1) ...
    delta_uf(k-(m+1))]
    compute the state of the controllers  // return xC(k+1) = [delta_yf(k+1) ...
    delta_yf(k-(n-2)) delta_uf(k) ... delta_uf(k-(m-1))] but the state
    will only be used at the next iteration, yielding: xC(k) =
    [delta_yf(k) ... delta_yf(k-(n-1)) delta_uf(k-1) ...
    delta_uf(k-m)] (Note that delta_yf(k+1) is not available at time k,
    and xC will need to be updated before been used in the next
    iteration)
    /* Wait until next sampling time */
end
```

A.3 Extra content

During the internship, three technical reports have been written: [8, 9, 12]. These reports detail the work completed and they are available on INESC-ID website: [http://www.inesc-id.pt/publications.php?showAll=true](http://www.inesc-id.pt/publications.php?showAll=true) (select the year 2008, and search for “Nicolas Castro”). They can be consulted in case of need of any further information.