

EEG-based depth of anaesthesia control

Kristjan Cuznar

August 7, 2023

Mentor: doc. dr. Gorazd Karer

1 Introduction

In addition to the industrial sector, we are witnessing a noticeable increase in the use of information technology combined with advanced process control methods in the field of medicine. In this context, numerous studies focus on computer-assisted drug dosing and anesthesia management during general anesthesia, falling under the realm of personalized medicine.

The depth of anesthesia, primarily quantified through the analysis of electroencephalogram (EEG) data, is commonly assessed by the Bispectral Index (BIS) and the Patient State Index (PSI). The objective of using these measurements is to better evaluate the level of hypnosis, with the possibility of incorporating other metrics such as the anesthetic flow rate within the patient's body. When combined with appropriate pharmacokinetic and pharmacodynamic models, this additional information offers valuable insights into the patient's condition.

Computer-assisted drug dosing aims to reduce the anesthesiologist's workload, enhance safety, and shorten postoperative recovery time by minimizing the quantity of administered medication.

Subsequently, a patient simulator is introduced, serving as the foundation for the development and testing of control algorithms. Progressing from basic pharmacokinetic (PK) and pharmacodynamic (PD) models, the simulator evolves into a sophisticated representation of the patient. The presentation then proceeds to outline the algorithm development procedures, encompassing multiple control methods that are thoroughly tested and implemented. Lastly, aggregated results of the control algorithm evaluations are presented.

2 Patient Simulator

For the purpose of developing a system to control the depth of anesthesia based on EEG index, an Open Source Patient Simulator [1] was utilized. In the following, the theoretical background pertaining to the functionalities of the patient simulator, which were employed in our work, is described.

2.1 Anesthesia

The possibilities [2] for maintaining the desired depth of anesthesia with propofol include:

- repetitive drug administration in a short period, known as bolus administration,
- manually controlled infusion (MCI), and
- target controlled infusion (TCI).

TCI systems [3] enable drug administration to achieve a target concentration at a specific site or tissue of interest. The dosing is automatically adjusted by the computer, employing simulations of PK models and implementing algorithms for drug administration. The anesthesiologist needs to input patient-specific parameters (age, body weight, height, gender, etc.) and the target concentration of the given drug at the effect site (concentration in the brain) or blood.

The increasing focus on automated anesthesia dosing research is driven by the capability to precisely and rapidly adapt to potential changes in surgical stimulation during the operation. Conventional methods exhibit notable shortcomings in maintaining a consistent drug concentration at the effect site.

2.2 Pharmacokinetic Model

The pharmacokinetic (PK) model describes the time course of drug concentration in the human body. Typically, models with three compartments are used, representing:

- the central, rapidly acting system (blood),
- the muscle mass system (slowly acting system), and
- the fat system (slowly acting system).

The utilized simulator and patient model are described using four compartments, with an additional, fourth, state representing the transport system to the effect site and denoting the concentration at the effect site. The model is presented in block diagram form in Figure 1.

The PK model can be represented in state-space form:

$$\dot{\mathbf{x}}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad \mathbf{y}(k) = \mathbf{Cx}(t) + \mathbf{Du}(t), \quad (1)$$

where the system matrices are defined as:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -(k_{10} + k_{12} + k_{13}) & k_{21} & k_{31} & 0 \\ k_{12} & -k_{21} & 0 & 0 \\ k_{13} & 0 & -k_{31} & 0 \\ k_{1e} & 0 & 0 & -k_{e0} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{C} &= [0 \quad 0 \quad 0 \quad 1] \\ \mathbf{D} &= [0] \end{aligned} \quad (2)$$

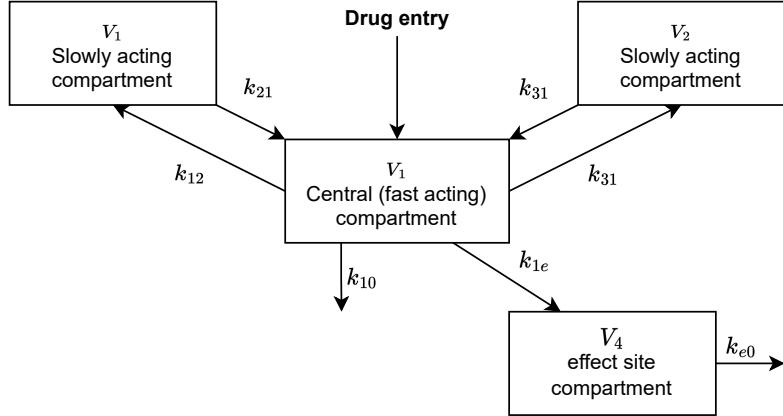


Figure 1: Block diagram of the PK model.

Parameters of the PK model vary among patients and types of drugs and are calculated based on empirically derived equations [1].

2.3 Pharmacodynamic Model

The pharmacodynamic (PD) model describes the effect of a drug on the human body in relation to its concentration. It is represented by a nonlinear Hill equation, which relates the drug concentration values to its effect.

In the case of modeling the effect of a single drug, the relationship is given by Equation (3):

$$E = E_{max} - \frac{E_{max} \cdot x_e^\gamma}{C_{50}^\gamma + x_e^\gamma}, \quad (3)$$

where:

- $E[\%]$ — predicted drug effect,
- x_e — current drug concentration,
- C_{50} — concentration required for 50% of the maximum effect,
- $\gamma[-]$ — Hill coefficient (slope factor),
- $E_{max}[\%]$ — maximum drug effect.

The PD model with selected parameters for the drug propofol is illustrated in Figure 2.

However, when modeling the effect of two or more drugs, the model becomes more complex [1]:

$$E = E_0 - E_{max} \frac{I^\gamma}{1 + I^\gamma}, \quad (4)$$

where

$$I = U_A + U_B + \sigma U_A U_B$$

and the following terms are defined:

- U_A — normalized effect of the first drug $U_A = \frac{x_A}{C_{50A}}$
- U_B — normalized effect of the second drug $U_B = \frac{x_B}{C_{50B}}$

$\sigma[-]$ — the degree of synergy in the range $[0, 1]$ between the effects of the drugs.

The PD model with selected parameters for drugs A (propofol) and B (remifentanil) is illustrated in Figure 3.

2.4 Complex Patient Model

Figure 4 presents the concept of the patient simulator. The simulator functions incorporate complex synergistic and antagonistic interaction aspects between general anesthesia and hemodynamic

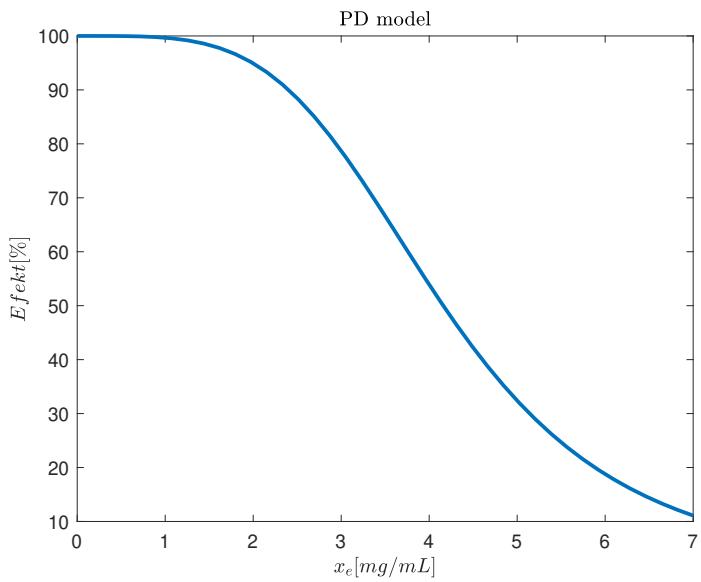


Figure 2: PD model with parameters $\gamma = 4$ and $C_{50} = 4.16$.

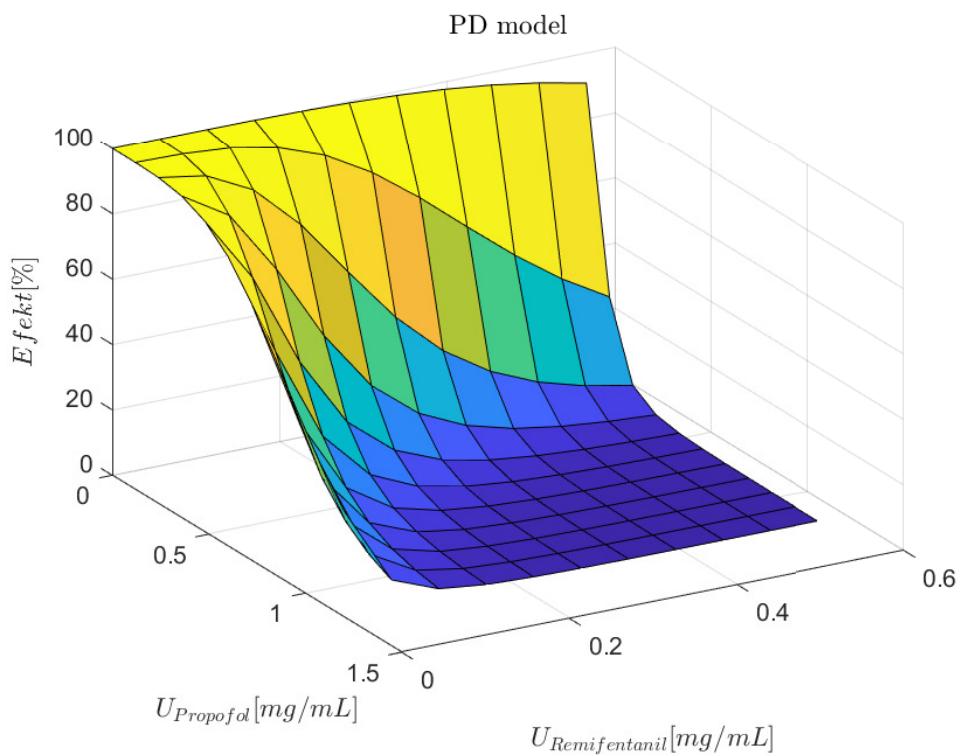


Figure 3: PD model with parameters $\gamma = 4$, $\sigma = 8, 20$, $C_{50P} = 4, 16$ ter $C_{50R} = 8, 84$.

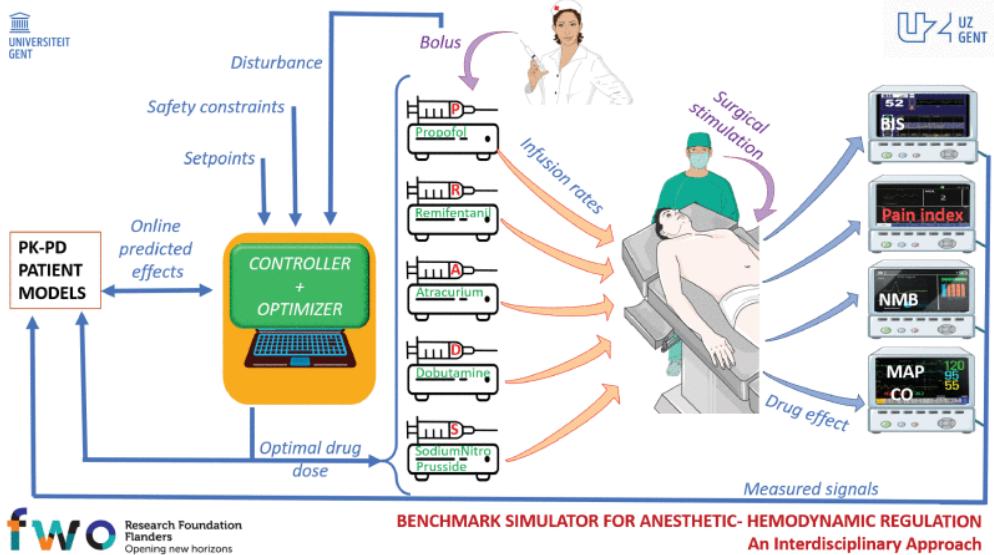


Figure 4: Patient Simulator [1].

stabilization variables.

In the simulator, which the authors assert to be the result of 15 years of experience and collaboration in the fields of medicine and control systems, the manipulative variables used for the control of hypnosis, analgesia, and neuromuscular blockade are Propofol, Remifentanil, and Atracurium respectively.

In our specific case, the primary manipulative variable employed is Propofol, which is intended for sedation (induction and maintenance of general anesthesia). However, the depth of anesthesia is also influenced by Remifentanil, and therefore its presence must be taken into account.

3 Open-Loop Control of Effect-Site Concentration

Computer-controlled infusion pumps (CCIPs), utilizing internal pharmacokinetic models of drugs, can rapidly and efficiently achieve and maintain a constant concentration of drugs in the patient [4]. Compared to traditional methods such as bolus or constant-rate infusions, these pumps offer a more precise approach to intravenous drug administration.

3.1 TCI - Stanpump Commercial Algorithm Variant

Schwilden [5] proposed a method for maintaining a constant plasma drug concentration. First, the target plasma concentration (C_T) is achieved with a bolus of C_TV_1 . Simultaneously, an infusion $I(t) = C_TV_1(k_{10} + k_{12}e^{-k_{21}t} + k_{13}e^{-k_{31}t})$ is initiated to maintain C_T (assuming a three-compartment model), where t represents the time since the initial bolus. As $I(t)$ continuously changes over time, computer-controlled target-controlled infusion (TCI) is necessary to maintain a constant plasma drug concentration.

However, the algorithms proposed by Schwilden are designed to control the drug concentration in the plasma, which may not be the desired target site for most drugs administered through intravenous infusion.

Therefore, Shafer et al. [4] suggested both a numerical method based on Euler's numerical approximation and an analytical closed-form solution to find the optimal drug administration over time ($I(t)$), which rapidly achieves and then maintains the desired effect-site concentration without overshooting. Here, we summarize the numerical solution, which will be used later for comparison among closed-loop methods.

The authors proposed modifying the pharmacokinetic model (illustrated in Figure 1) to improve symmetry by transforming the fourth compartment to return the drug to the central compartment. The effect-site compartment is small and has no influence on the drug's pharmacokinetics, so it is not critical whether the drug is eliminated from the effect site through k_{e0} or returns to the central compartment via k_{41} . The volume of the effect site V_E is defined as 1/10000 of the volume of the central compartment, $k_{41} = k_{e0}$, and $k_{14} = k_{e0}/10000$. The mathematical model is formulated in terms of quantities, not concentrations, where the concentration in each compartment is the quotient of the quantity and its corresponding volume.

The state-space model, which will be used in the control algorithm, is defined as:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -(k_{10} + k_{12} + k_{13} + k_{14}) & k_{21} & k_{31} & k_{41} \\ k_{12} & -k_{21} & 0 & 0 \\ k_{13} & 0 & -k_{31} & 0 \\ k_{14} & 0 & 0 & -k_{41} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{C} &= [0 \ 0 \ 0 \ 1] \\ \mathbf{D} &= [0] \end{aligned} \tag{5}$$

The computer adjusts the infusion rate every few seconds. The interval is typically set to 10 to 20 seconds, but it can be adjusted in real-time. Let's assume the computer adjusts the infusion rate every 10 seconds. We can utilize an internal pharmacokinetic algorithm to update the state variables in response to each 10-second infusion (Algorithm 1). For clarity, we define $k_1 = k_{10} + k_{12} + k_{13} + k_{14}$.

In the first step, it is necessary to calculate the amount of drug at the effect site until the peak concentration (maximum concentration at the effect site) occurs at time t_{peak} during and after a 10-second infusion $I(t) = 1$ unit/s, which is introduced into the body without any previously

Algorithm 1 Update of state variables using the pharmacokinetic algorithm.

```

for  $x = n$  to  $n + 10$  do
     $\Delta A_1 \leftarrow A_2 k_{21} + A_3 k_{31} + A_4 k_{41} - A_1 k_1 + I$ 
     $\Delta A_2 \leftarrow A_1 k_{12} - A_2 k_{21}$ 
     $\Delta A_3 \leftarrow A_1 k_{13} - A_3 k_{31}$ 
     $\Delta A_4 \leftarrow A_1 k_{14} - A_4 k_{41}$ 
     $A_1 \leftarrow A_1 + \Delta A_1$ 
     $A_2 \leftarrow A_2 + \Delta A_2$ 
     $A_3 \leftarrow A_3 + \Delta A_3$ 
     $A_4 \leftarrow A_4 + \Delta A_4$ 
end for
 $n \leftarrow n + 10$ 

```

administered drug (Algorithm 2). The concept is illustrated in Figure 5. This calculation needs to be performed only once and can be done before the actual start of the infusion. The variable a_i is used for temporary state variables to avoid confusion with the actual pharmacokinetic model state variables in real-time A_i . The response of the effect site is stored in an array $E[]$.

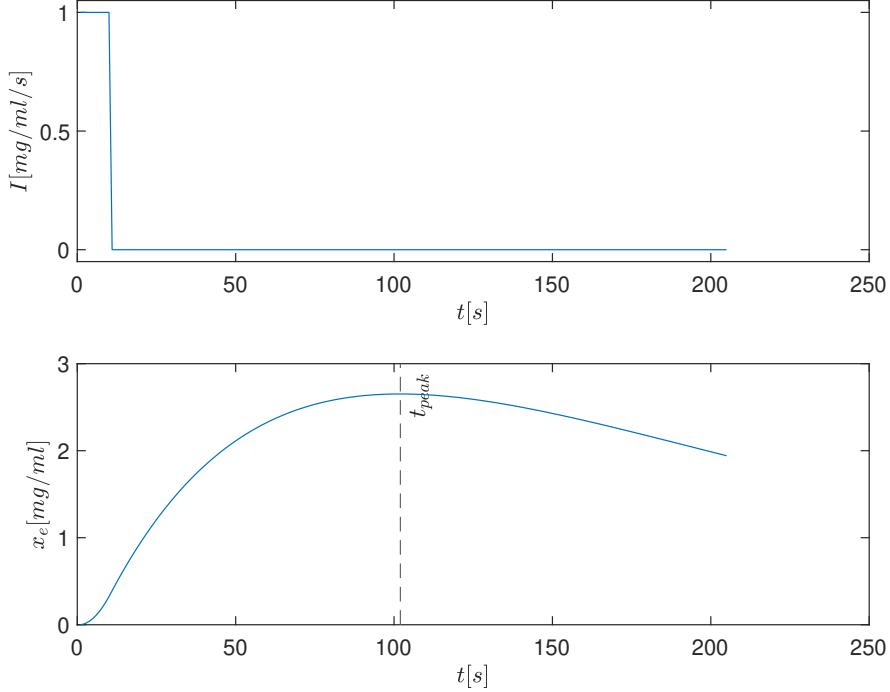


Figure 5: Effect site concentration (x_e) after a 10-second bolus ($tci.m$).

The algorithm calculates the infusion rate using linear pharmacokinetics to rapidly achieve the desired drug concentration at the effect site without overshooting. First, using temporary state variables, it determines the amount of drug at the effect site at the time of peak concentration if no drug is being infused (Algorithm 3). The amount of drug at the effect site at a specific time is stored in the array $B_4[]$.

The superposition dictates that the amount of drug at the effect site at time $n+j$ after the infusion I is equal to $B_4[j] + E[j] \cdot I$. We define j_{peak} as the index at the highest value of $B_4[] + E[] \cdot I$ for a given I . Reversing the equation, we get:

Algorithm 2 Calculation of t_{peak} and effect site response

```
a1, a2, a3, a4 ← 0
j ← 1
do
    if j <= 11 then
        Δa1 ← a2k21 + a3k31 + a4k41 - a1k1 + 1           ▷ Adding infusion
    else
        Δa1 ← a2k21 + a3k31 + a4k41 - a1k1           ▷ No infusion
    end if
    Δa2 ← a1k12 - a2k21
    Δa3 ← a1k13 - a3k31
    Δa4 ← a1k14 - a4k41
    a1 ← a1 + Δa1
    a2 ← a2 + Δa2
    a3 ← a3 + Δa3
    a4 ← a4 + Δa4
    E[j] ← a4
    j ← j + 1
while E[j] < E[j - 1]
tpeak ← j - 1
```

Algorithm 3 Update of auxiliary state variables using the pharmacokinetic model without infusion.

```
ΔB1 ← A1
ΔB2 ← A2
ΔB3 ← A3
ΔB4 ← A4
for j = 1 to tpeak do
    ΔB1 ← B2k21 + B3k31 + B4k41 - B1k1
    ΔB2 ← B1k12 - B2k21
    ΔB3 ← B1k13 - B3k31
    ΔB4 ← B1k14 - B4k41
    B1 ← B1 + ΔB1
    B2 ← B2 + ΔB2
    B3 ← B3 + ΔB3
    B4 ← B4 + ΔB4
    B4[j] ← B4
end for
```

$$I = \frac{A_{4_{ref}} - B_4[j_{peak}]}{E[j_{peak}]} \quad (6)$$

Since j_{peak} is not known in advance and can be anywhere between 0 and t_{peak} , we need to use a search algorithm to simultaneously solve for j_{peak} and I . We first select an initial value for j_{peak} , denoted as j_{peak_0} . When $A_{4_{ref}}$ increases, we set j_{peak_0} to 0; otherwise, it remains the previous value of j_{peak} . We determine I_0 as:

$$I_0 = \frac{A_{4_{ref}} - B_4[j_{peak_0}]}{E[j_{peak_0}]} \quad (7)$$

The peak of $B_4[] + E[] \cdot I$ occurs at j_{peak_1} . If j_{peak_1} is equal to j_{peak_0} , we obtain $I_0 = I$ and the problem is solved. Otherwise, we set j_{peak_0} to j_{peak_1} and repeat the calculation of I_0 (7). We repeat this process until j_{peak_1} is equal to j_{peak_0} .

If $B_4[10]$ exceeds $A_{4_{ref}}$, then I is set to 0 until B_4_{ref} becomes smaller than the target value $A_{4_{ref}}$.

The simulation results, where the internal PK model used in the TCI algorithm has the same parameter values as the patient model, are shown in Figure 6. The red plot ($x_{est_{TCI}}$) represents the estimated drug concentration at the effect site with the internal PK model, and the blue plot (x_{est}) represents the estimate with the patient PK model. The BIS estimate is also shown in the upper plot for completeness.

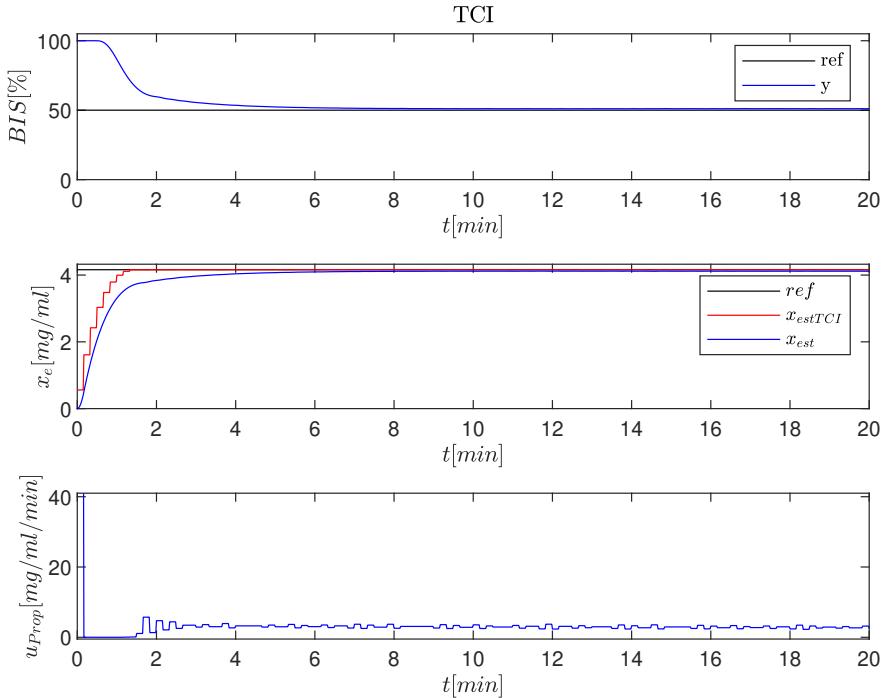


Figure 6: Simulation using TCI (*main_tci.m*).

In Figure 6, it can be observed that the results are good, and the desired drug concentration at the effect site is achieved. However, if the correct parameter values of the PK model are not estimated (which is generally not achievable with perfect accuracy), the actual drug concentration may not be the same as the estimated one. This is demonstrated in Figure 7, where the coefficients of the systemic matrix of the patient PK model for propofol in the state space are changed to random values between -10% and 10% . Therefore, it would be sensible to perform real-time estimation or correction of the parameters during the anesthesia process and implement feedback control. This

will be further explored in the following sections.

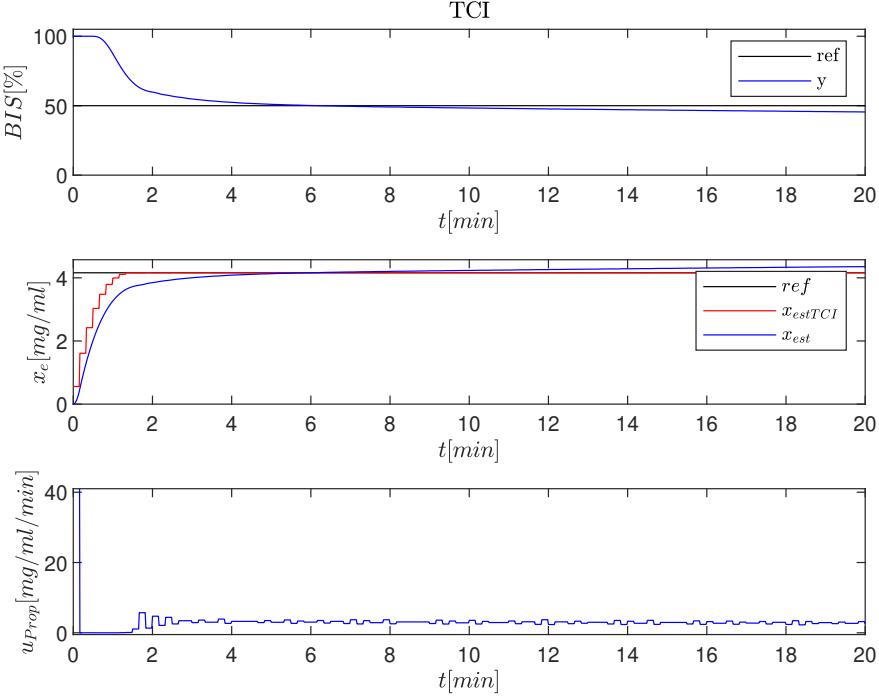


Figure 7: Simulation using TCI – incorrectly estimated PK model parameters (*main_tci.m*).

As depth of anesthesia is typically quantified using the bispectral index (BIS) based on EEG analysis, it is reasonable to use this information to calculate the target drug concentration at the effect site. By combining this with TCI, feedback control can be introduced, which will be further elaborated in the following sections. BIS values between 40 and 60 are considered to indicate appropriate general anesthesia for surgery [6]. This range prevents patient awareness and allows for reduced drug usage.

3.2 Model Reference Control with PSO

An alternative to the previously described TCI algorithm is the Model Reference Control with Particle Swarm Optimization (PSO), which is briefly presented below.

PSO is a method [7] that belongs to evolutionary computation, where particles move through the parameter space based on their inertia or velocity (\mathbf{v}), the value of the parameter vector with the highest fitness for that particle (\mathbf{p}^{best}), and the value of the parameter vector with the highest fitness in the entire population (\mathbf{g}^{best}).

One of the many algorithms, which is also implemented, adapts the velocity and position of the particles according to the equation:

$$\begin{aligned} \mathbf{v}_i &= \omega \cdot \mathbf{v}_i + c_1 \cdot rand \cdot (\mathbf{p}_i^{best} - \mathbf{x}_i) + c_2 \cdot rand \cdot (\mathbf{g}^{best} - \mathbf{x}_i) \\ \mathbf{x}_i &= \mathbf{x}_i + \mathbf{v}_i, \end{aligned} \quad (8)$$

where:

- \mathbf{v}_i — is the particle velocity,
- \mathbf{x}_i — is the particle position,

- ω — is the inertia factor (initialized to 1 in our case and then reduced after each iteration with the forgetting factor $\omega_{new} = \omega \cdot \mu$, where μ is chosen in the interval from 0.95 to 0.99),
- c_1 — is the cognitive constant, representing confidence in the particle's personal best, and
- c_2 — is the social constant, representing confidence in the swarm's best.

Certain constraints on the positions and velocities of the particles are taken into account. Typically, the limits of the parameter space (X_{min} and X_{max}) are defined to prevent the particle from escaping the feasible region. If a particle exceeds these boundaries, it is placed on the boundary and its velocity is set to 0. Another constraint is the velocity limit of the particle (V_{max}), which in our case is set to $V_{max} = (X_{max} - X_{min})/5$.

The number of particles in the swarm is smartly set to $PARTNUM = round(10 + 2 \cdot \sqrt{nr_parameters})$ based on one of the recommendations in the literature.

Algorithm 4 Initialization of PSO algorithm

- a. Randomly generate particle positions within the feasible range.
 - b. Set particle velocities to 0.
 - c. For each particle, calculate the fitness function value and store it as f^{best} .
 - d. Assign p_i^{best} to the particles.
 - e. Determine g^{best} among all particles.
-

In the loop, while the termination criterion for optimization is not satisfied, execute algorithm 5.

Algorithm 5 Main loop of the PSO algorithm

- a. Adapt particle positions based on equation (8).
 - b. Check if the particles are within the feasible range (if not, set them to the boundary and set their velocity to 0).
 - c. Check particle velocities (if they are out of bounds, set them to the boundary).
 - d. Calculate the fitness function value.
 - e. If the fitness function value is smaller than f^{best} , update it and replace p_i^{best} with the current particle position.
 - f. If the fitness function value is also smaller than f_{global}^{best} , update it and replace g^{best} with the current particle position.
 - g. Calculate the termination conditions for optimization (if they are satisfied, terminate the optimization).
-

Simulation results, where the same parameter values of the pharmacokinetic model as those in the patient model were used for the internal TCI algorithm, are shown in Figure 8. The upper graph displays the BIS estimation for completeness, the middle graph depicts the reference model value with a red curve (ref_{model}), and the blue curve (x_{est}) shows the estimation with the patient's PK model.

It can be observed that compared to the previous (commercial version) algorithm, a smoother control trajectory is achieved with the PSO algorithm, and it also allows for the possibility of

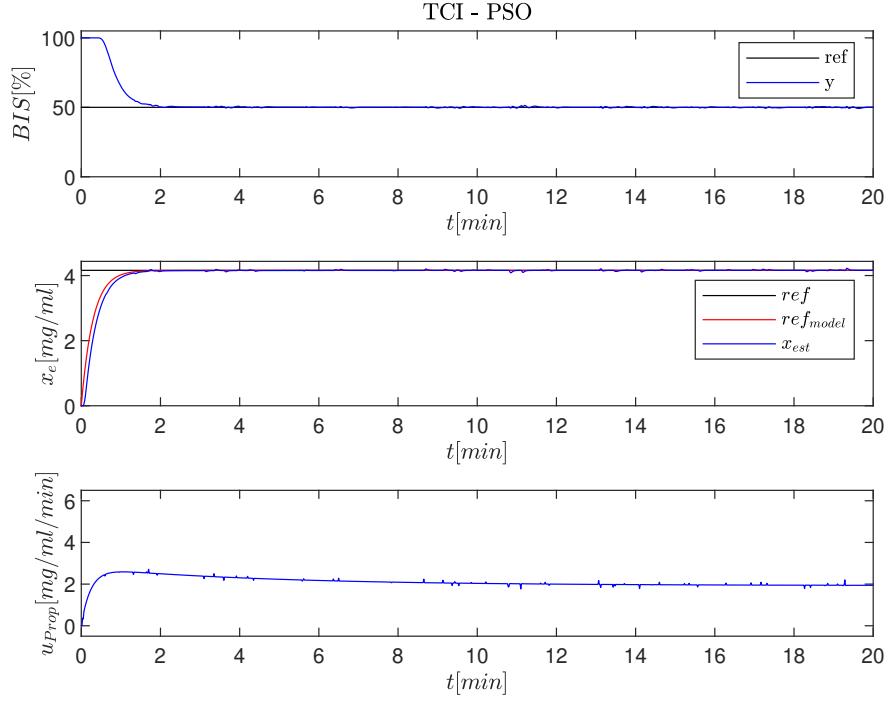


Figure 8: Simulation using TCI – PSO (*main_pso.m*).

rearrangement into closed-loop control. However, further investigation of this approach will not be addressed due to the desire for comparison with existing algorithms in clinical practice.

4 Development of Closed-Loop EEG-based depth of anaesthesia control

The effect of a drug on the human body is not linearly dependent on the drug input, which complicates the design of a control system. Assuming that the pharmacokinetic (PK) model is linear (which is usually the case), linearity occurs in the pharmacodynamic (PD) model (non-linear Hill equation).

4.1 Model Linearization

One attempt to solve this problem is to use soft models. With the Gustafson-Kessel method, we identified locally linear regions (Figure 9a), where locally linear PD models, or locally linear regions of the Hill equation, could apply. Based on the current normalized drug concentration at the site of action, we can calculate the membership (Figure 9b) to individual locally linear regions and, with the concept of a soft model, estimate the output value representing the drug effect, i.e., BIS or PSI.

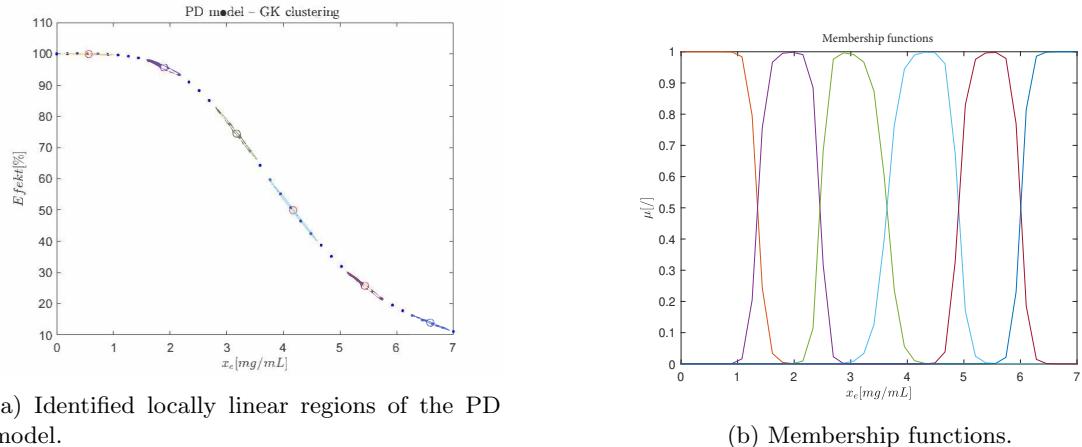


Figure 9: Model linearization.

Considering the additional concentration of remifentanil, this solution was not successful. Therefore, we used an alternative approach, which is presented below.

The PK/PD-model can be represented as a Wiener model [8] (Figure 10), composed of a series connection of a linear time-invariant (LTI) system and a static non-linear system. The linear system is modeled using a transfer function or a state-space representation, while the static non-linear system describes the non-linear Hill equation (4), denoted as f . If the non-linear part (f) is invertible, we can control the system with a linear controller using its inverse. The control structure is shown in Figure 11.

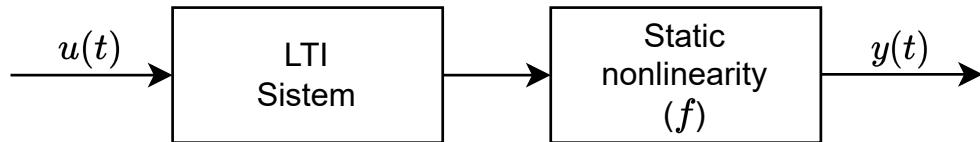


Figure 10: Wiener model.

By introducing the inverse Hill equation, we can set the desired depth of anesthesia index as a reference, which maps to the required drug concentration at the site of action. Thus, we transformed the non-linear problem into a linear one, for which there are many well-known control structures

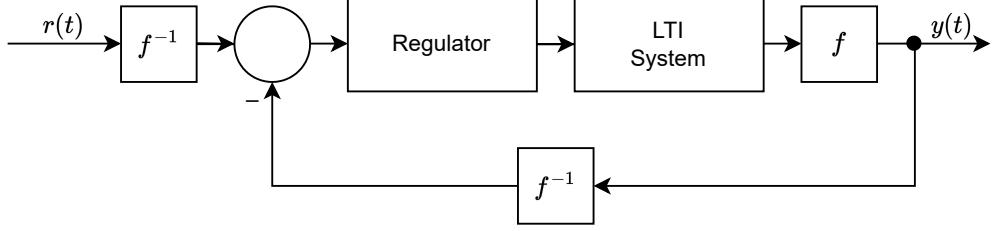


Figure 11: Control structure for the Wiener model.

available.

There is an analytical solution from which we can calculate the target propofol value if we know the concentration of remifentanil, the parameters γ and σ , and the desired depth of anesthesia index. This approach also allows for real-time adaptation to any changes in the parameters of the PD model (γ and σ). The derivation of the inverse Hill equation is shown in Equation (9):

$$\begin{aligned}
 E &= E_{max} - E_{max} \frac{I^\gamma}{1 + I^\gamma} \\
 J &= I^\gamma = (U_A + U_B + \sigma U_A U_B)^\gamma \\
 \frac{E - E_{max}}{-E_{max}} &= J \left(1 + \frac{E - E_{max}}{E_{max}} \right) \\
 J &= \frac{E - E_{max}}{-E_{max} \left(1 + \frac{E - E_{max}}{E_{max}} \right)} = \frac{Y - E_{max}}{-E_{max} - E + E_{max}} = \frac{E_{max} - E}{E} \\
 U_A (1 + \sigma U_B) + U_B &= J^{\frac{1}{\gamma}} \\
 U_A &= \frac{J^{\frac{1}{\gamma}} - U_B}{1 + \sigma U_B} = \frac{\left(\frac{E_{max} - E}{E}\right)^{\frac{1}{\gamma}} - U_B}{1 + \sigma U_B},
 \end{aligned} \tag{9}$$

where in our case, U_A is assumed to be the normalized effect of propofol, and U_B is the normalized effect of remifentanil.

4.2 PID Controller

The first method we implemented for anesthesia depth control is a simple PID controller. The simulation scheme is shown in Figure 12, and the initial results, with the controller parameters set using the PID Tuner tool, are shown in Figure 13.

In Figure 13, we observe unacceptable results, as the BIS value fluctuates between extreme values. Therefore, it is sensible to revisit the modeling process and examine the patient's response to a step excitation (constant propofol dosing), as shown in Figure 14a. By doing this, we can identify potential issues in the modeling procedure.

From the response, we can observe the presence of dead time, which occurs due to the delayed effect of the drug and the measurement of the EEG index. A zoomed-in response is shown in Figure 14b.

4.2.1 Smith Predictor

The Smith predictor [9] is a concept used for controlling systems with long dead times, as it can predict the future output of the process and adjust the control signal accordingly, allowing the system to compensate for time delays. This can significantly improve the dynamic response of the system and contribute to process stabilization.

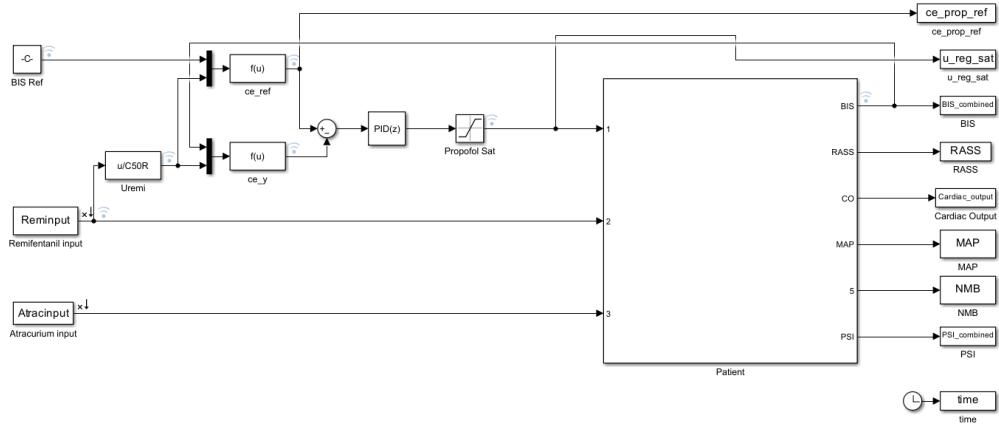


Figure 12: Simulation scheme for PID control.

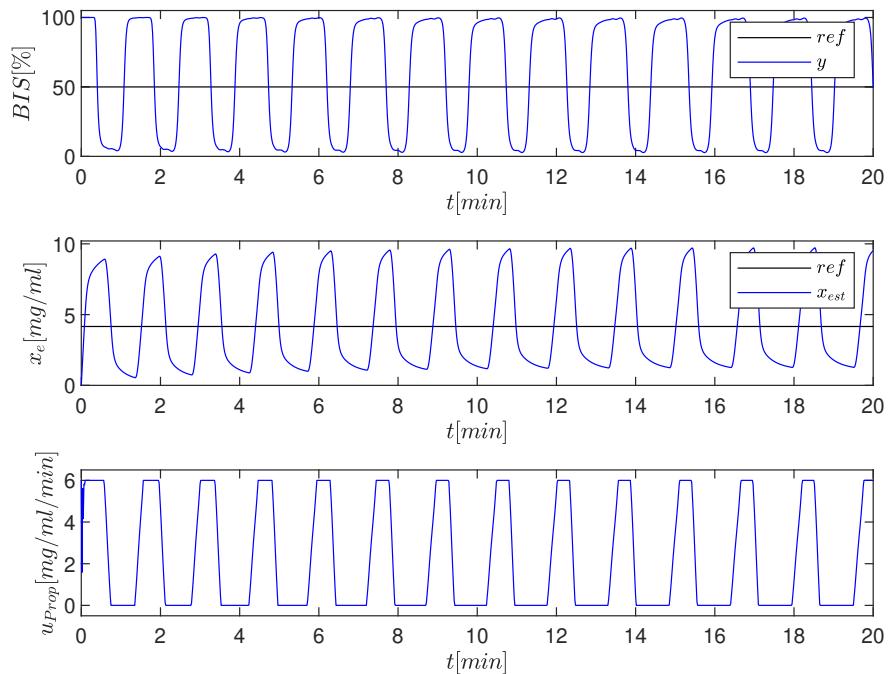
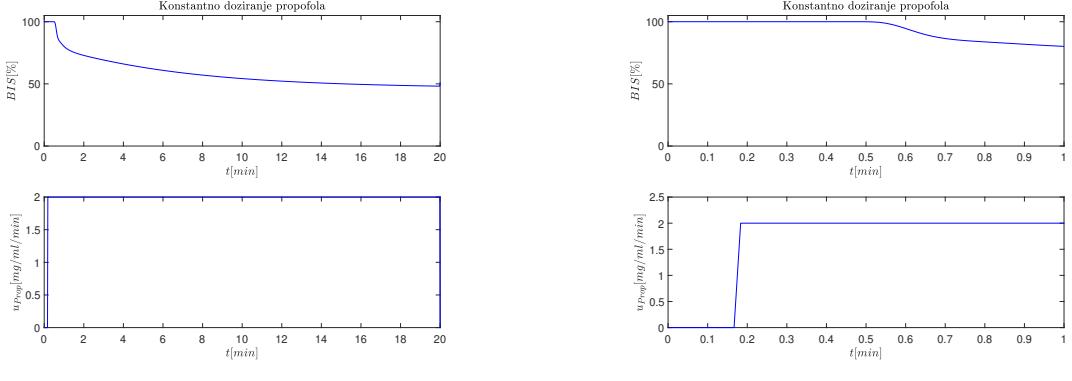


Figure 13: Simulation using a PID controller (*main_pid.m*).



(a) Complete response.

(b) Zoomed-in response.

Figure 14: Response to step excitation.

For the efficient use of the Smith predictor, it is crucial to have a good mathematical model of the process, including an accurate estimation of the dead time. The concept is illustrated in Figure 15.

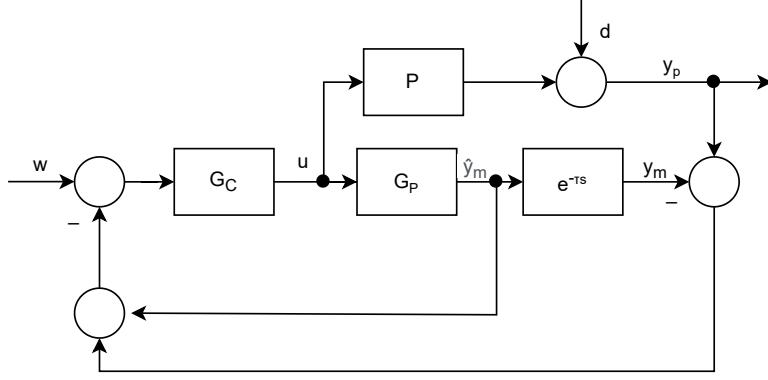


Figure 15: Smith Predictor concept.

When using the Smith predictor, we can calculate the estimate of the un-delayed output of the process through the equation:

$$y_p(k) - \hat{y}_p(k) = y_m(k) - \hat{y}_m(k)$$

which yields:

$$\hat{y}_p(k) = y_p(k) - y_m(k) + \hat{y}_m(k). \quad (10)$$

The simulation scheme with the implemented Smith predictor is shown in Figure 16, and the results with the controller parameters set as in the previous case are presented in Figure 17.

It can be observed that the system in this case is stable and reaches the reference value in a certain time. By adjusting the controller parameters differently, the control performance could be improved. However, for systems that are highly variable (such as patients), this type of control structure may not be as suitable as the methods described further below.

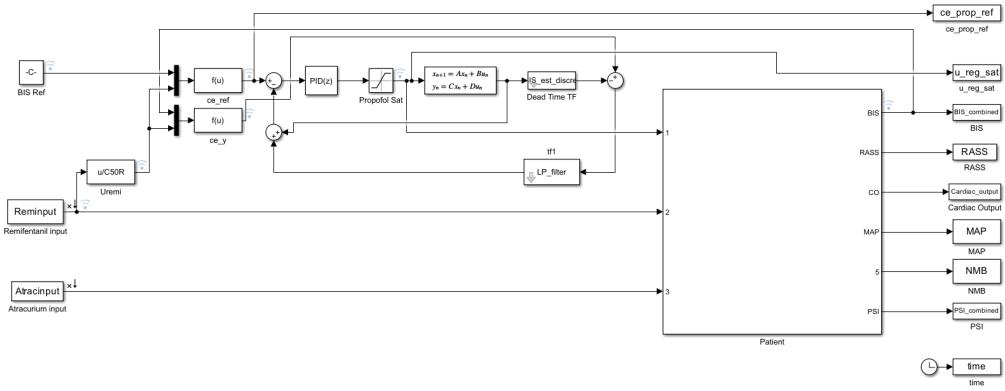


Figure 16: Simulation scheme for anesthesia control using PID controller with implemented Smith predictor.

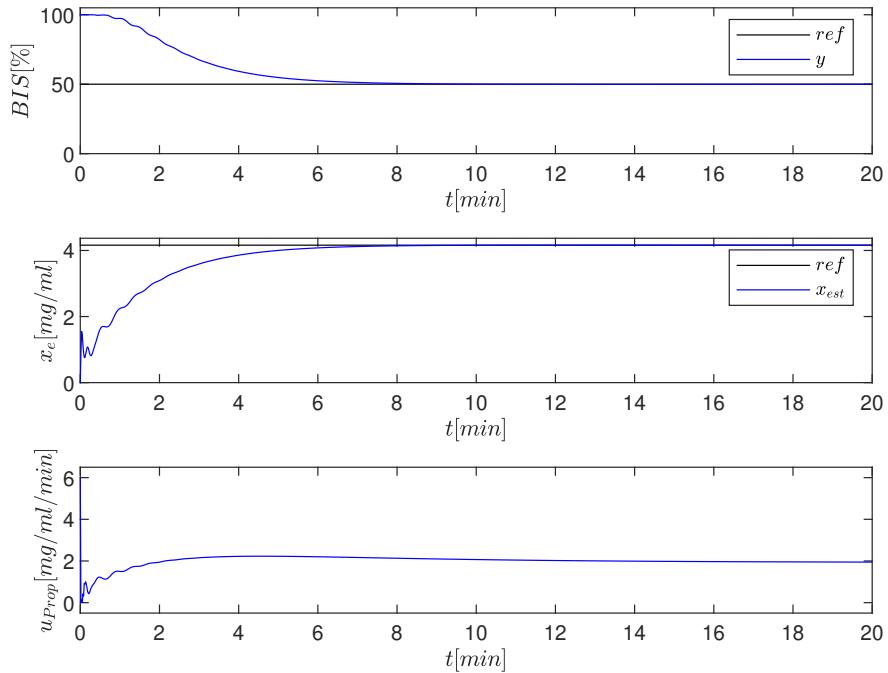


Figure 17: Simulation using PID controller with implemented Smith predictor (*main_pid.m*).

4.3 The Predictive Functional Controller

The Predictive Functional Controller (PFC) is a control strategy that utilizes a mathematical model of the process or system to predict its future behavior and then uses this prediction for control.

The PFC strategy [10] is based on using a process model to predict the process output at a certain horizon. The goal of the approach is to determine the control signal such that the difference between the predicted reference value $y_r(k + H)$ and the current process output value $y_p(k)$, and the difference between the predicted model output $y_m(k + H)$ and $y_m(k)$ are minimized. The concept of the PFC regulator is shown in Figure 18.

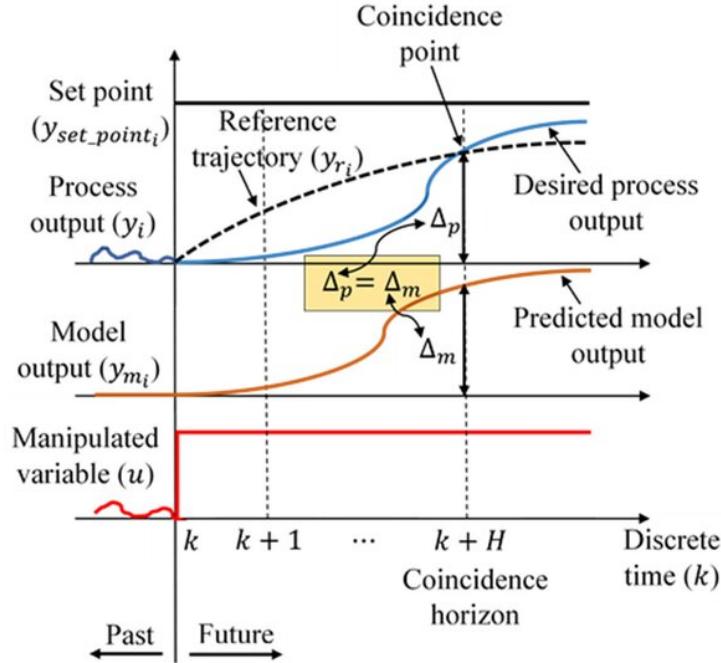


Figure 18: PFC concept [11].

For the implementation of PFC, a reference model of the trajectory is also considered, which describes how the reference value should change (usually this model is a first-order model).

$$\begin{aligned} \mathbf{x}_r(k+1) &= \mathbf{A}_r \mathbf{x}_m(k) + \mathbf{B}_r \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_r \mathbf{x}_m(k), \end{aligned} \quad (11)$$

The reference model must have a gain equal to one, which means that the matrices \mathbf{A}_r , \mathbf{B}_r , and \mathbf{C}_r must satisfy the condition:

$$\mathbf{C}_r (\mathbf{I} - \mathbf{A}_r)^{-1} \mathbf{B}_r = \mathbf{I}. \quad (12)$$

The control law reads as follows:

$$\mathbf{u}(k) = \mathbf{G} (\mathbf{w}(k) - \mathbf{y}_p(k)) + G_0^{-1} \mathbf{y}_m(k) - G_0^{-1} \mathbf{C}_m \mathbf{A}_m^H \mathbf{x}_m(k), \quad (13)$$

where:

$$\mathbf{G} = G_0^{-1} (\mathbf{I} - \mathbf{A}_r^H) \quad (14)$$

and:

$$G_0 = \mathbf{C}_m (\mathbf{A}_m^H - \mathbf{I}) (\mathbf{A}_m - \mathbf{I})^{-1} \mathbf{B}_m. \quad (15)$$

In this control law, \mathbf{w} represents the reference and H is the prediction horizon (within this horizon, the output of the process $\mathbf{y}_p(k)$ should match the reference response $\mathbf{y}_r(k)$).

In the case of controlling a system with dead time, the Smith predictor concept is introduced for the PFC regulator. The control law is then reformulated as:

$$\mathbf{u}(k) = \mathbf{G}(\mathbf{w}(k) - (\mathbf{y}_p(k) + \hat{\mathbf{y}}_m(k) - \mathbf{y}_m(k))) + G_0^{-1}\mathbf{y}_m(k) - G_0^{-1}\mathbf{C}_m\mathbf{A}_m^H\mathbf{x}_m(k), \quad (16)$$

where $\hat{\mathbf{y}}_m(k)$ represents the un-delayed model output.

The simulation diagram with the implemented PFC is shown in Figure 19, and the simulation results are displayed in Figure 20. The reference model was designed to achieve the desired dynamics of the reference trajectory:

$$\mathbf{A} = [0.945], \mathbf{B} = [0.055], \mathbf{C} = [1], \mathbf{D} = [0], \quad (17)$$

and the matching horizon is set to $H = 10$.

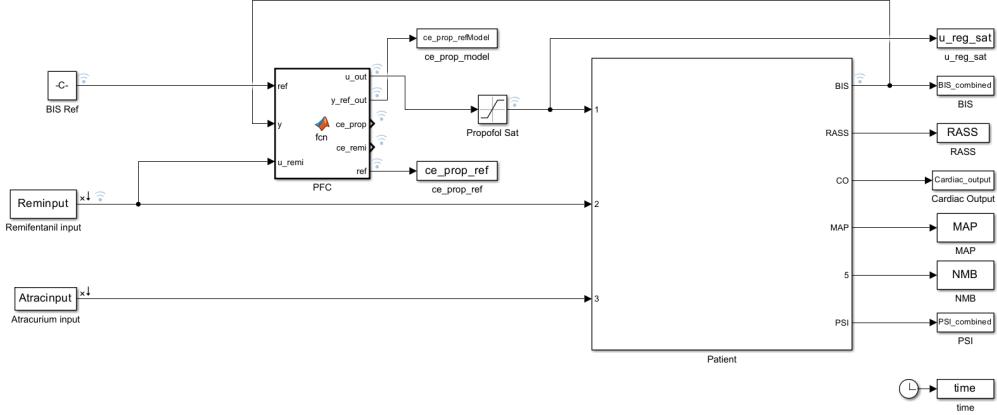


Figure 19: Simulation scheme for PFC.

In Figure 20, we can observe that the desired BIS value is achieved. However, in the case of not estimating the correct parameters of the PK model (which is generally never achieved perfectly accurately), the actual drug concentration would not match the estimated value in the internal model of the controller. To test the robustness of the control system, we randomly changed the PK model parameters for propofol in the state-space representation by -10% to 10% for each parameter. In comparison with the open-loop control, we can observe a significant difference - the desired BIS value is still achieved and maintained, as shown in Figure 21.

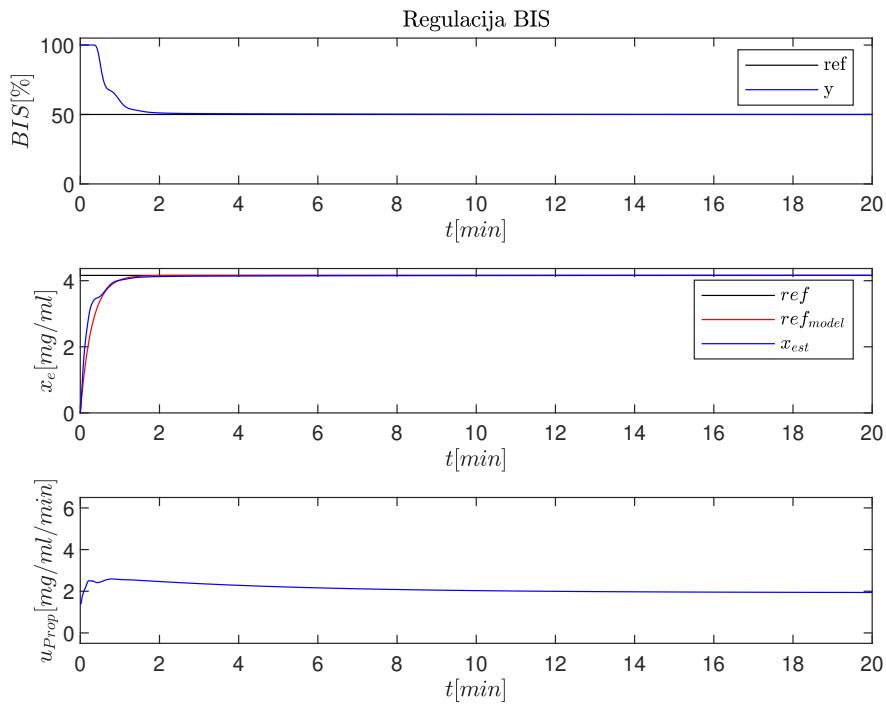


Figure 20: Simulation using PFC (*main-pfc.m*).

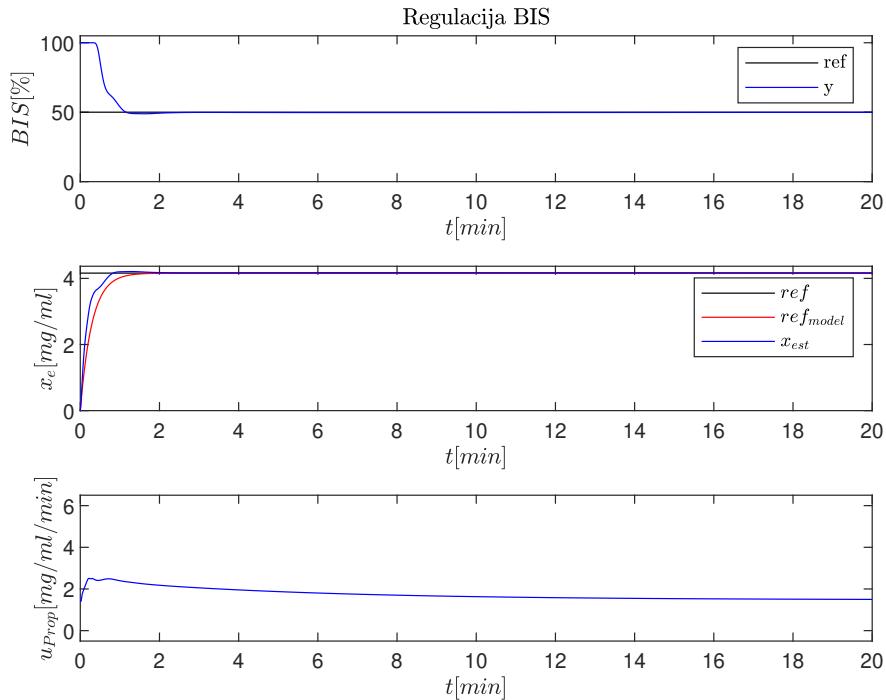


Figure 21: Simulation using PFC – incorrectly estimated PK model parameters (*main-pfc.m* – line 247: add distortion matrix).

5 Results

The key parameters of the patient under consideration are presented in tables 1, 2 and 3.

Table 1: Patient General Information

Index [/]	Age [leta]	Height [cm]	Weight [kg]	Gender [/]	BMI [/]	LBM [LBM]
1	74	164	88	1	32,7	60

Table 2: PK-model parameters for Patinet 1

k_{10} [l/min]	k_{12} [l/min]	k_{13} [l/min]	k_{21} [l/min]	k_{31} [l/min]	k_{e0} [l/min]	k_{e1} [l/min]
0,4638	0,1841	0,1958	0,0735	0,0035	0,4560	0,4560

Table 3: PD-model parameters for Pacepoint 1

C_{50P} [mg/ml]	C_{50R} [mg/ml]	γ [/]	σ [/]
4,1600	8,8400	4,000	8.2000

Assuming that the PK model parameters, dead time (19.7 seconds), and PD model parameters are precisely determined for the PFC algorithm, the results of the comparison are shown in Figure 22.

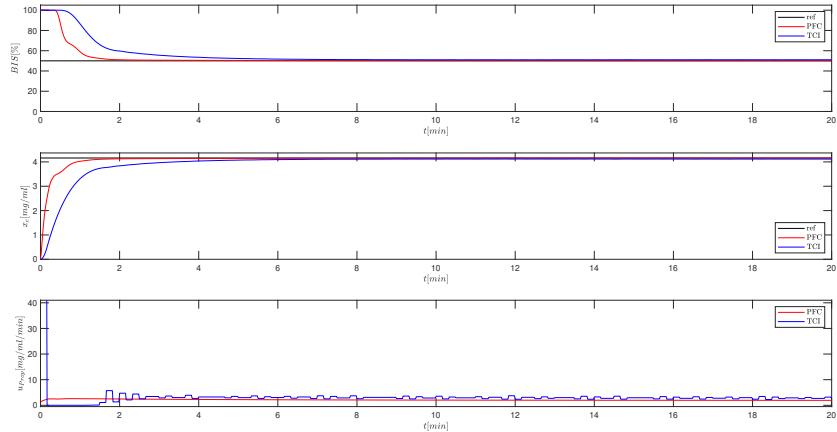


Figure 22: Comparison between PFC and TCI (*main_pfc_vs_tci.m*).

We can observe that the use of feedback control algorithm for anesthesia management (PFC) compared to the open-loop algorithm (TCI) resulted in faster attainment of the desired BIS value and reduced drug administration to the body by 39.05%.

However, if the PK model parameters are not accurately estimated, there may be a difference between the actual state and the estimated state in the internal model of the controller. To test the robustness of the control system, we intentionally changed the PK model parameters for propofol. With closed-loop control using the PFC algorithm, we can observe that the desired BIS value is achieved and maintained. When comparing it with open-loop control (TCI algorithm), we can see

a deviation of the actual values from the desired values. The results of both approaches are shown in Figure 23.

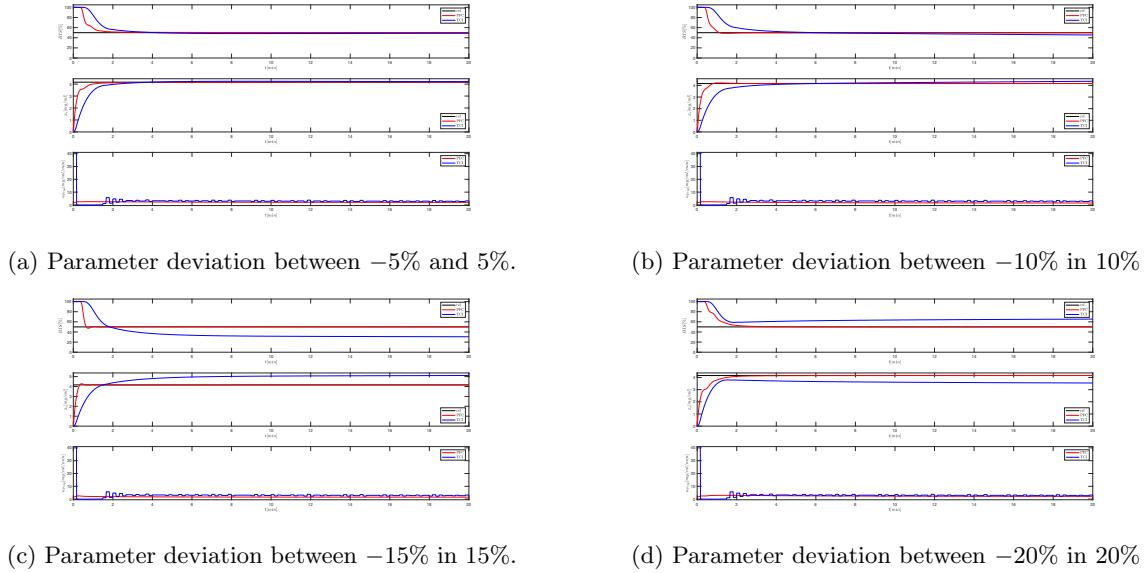


Figure 23: The comparison between PFC and TCI – non-exact PK model parameters (*main_pfc_vs_tci.m* – 6 row: add distortion matrix).

Results of our testing show the advantage of closed-loop control over open-loop control. However, in closed-loop control, we are exposed to the condition of knowing accurate parameters of the PD model and dead time. Therefore, it would be sensible to examine the robustness of the closed-loop control system when there are incorrect estimates of these parameters.

For clarity, we will compare the responses depending on the incorrect estimates of parameters only for the closed-loop control system, as they do not affect the open-loop control.

Figure 24 presents the results of a simulation in which we intentionally varied the parameter γ . We can observe that incorrect estimates of the parameter do not have a significant impact in the vicinity of the desired value $BIS = 50$, as the Hillova equation is more linear in this region compared to others. This demonstrates that the closed-loop control approach using the PFC algorithm performs well even with incorrect estimates of the parameter.

Figure 25 shows the results of the simulation, where we intentionally varied the parameter σ . We can observe that incorrect estimates of the parameter do not have a significant impact on the simulation results. The parameter σ only affects the amount of synergy between the drugs (in this case, we assumed $u_{Remi} = 0$, meaning that we did not administer remifentanil). This is reflected in equation (4).

Figure 26 shows the results of the simulation, where we intentionally varied the parameter σ , but in this case, the remifentanil input was not set to 0 (it was chosen such that with the correct estimates of PK model parameters for remifentanil, the target concentration $x_{e_{Remi}} = 0.8mg/ml$ is achieved). We can observe that incorrect estimates of the parameter have a significant impact on the simulation results, especially during the transient response. However, in the steady-state, the control algorithm compensates well for the incorrectly estimated parameter.

Figure 27 illustrates the results of the simulation where we manipulated the parameter T_d , representing the dead time of the system. We can observe that incorrect estimates of the dead time can impact the overall dynamics, as an overestimated dead time compared to the actual value can lead to more oscillations in the BIS values. On the other hand, if the estimated dead time

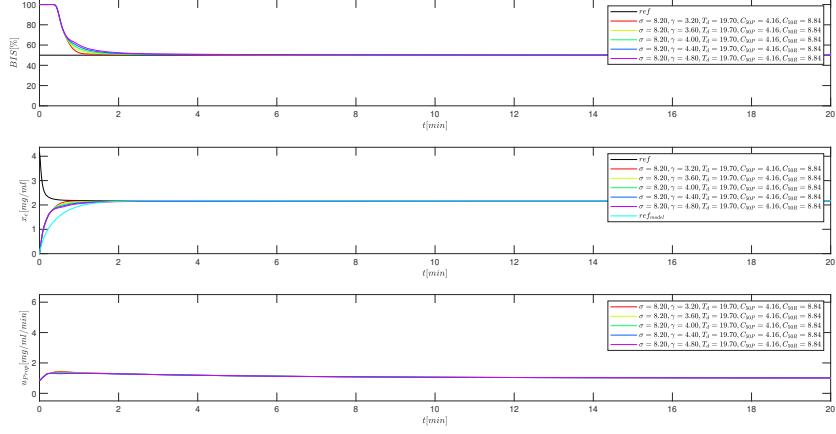


Figure 24: Analysis of the impact of incorrect estimates of parameter γ (*main_pfc_pd_analysis.m* – 201 row: assembly experiment).

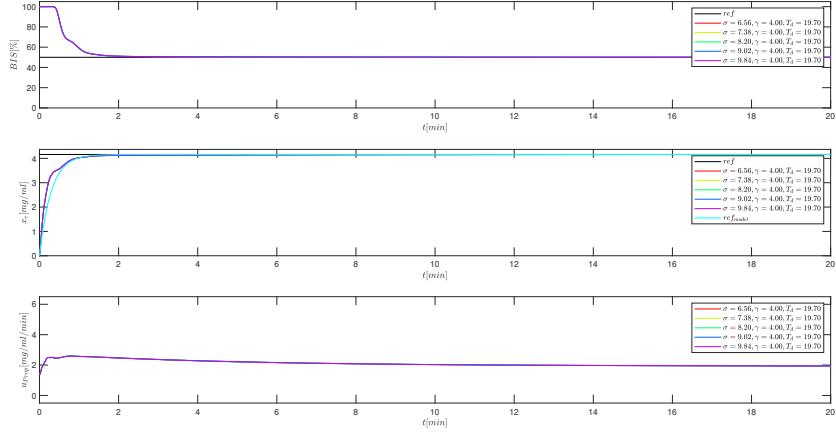


Figure 25: Analysis of the impact of incorrect estimates of parameter σ , without the input of remifentanil (*main_pfc_pd_analysis.m* – 201 row: assembly experiment).

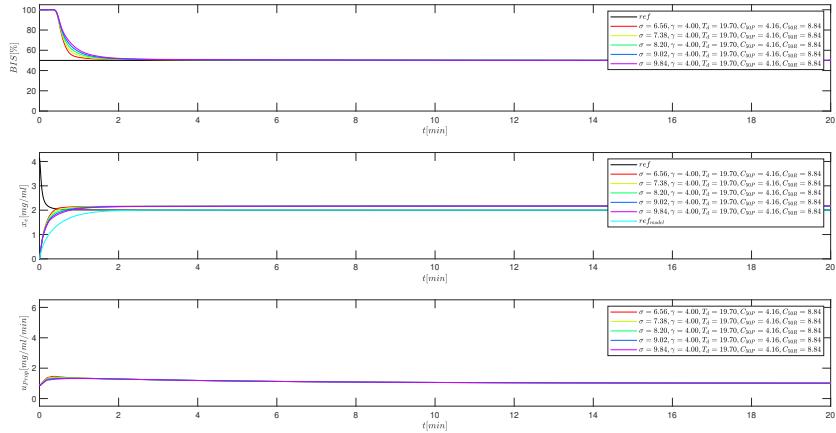


Figure 26: Analysis of the impact of incorrect estimates of parameter σ , with remifentanil input (*main_pfc_pd_analysis.m* – 201 row: assembly experiment).

is lower than the actual value, we do not observe such pronounced changes in the system's response.

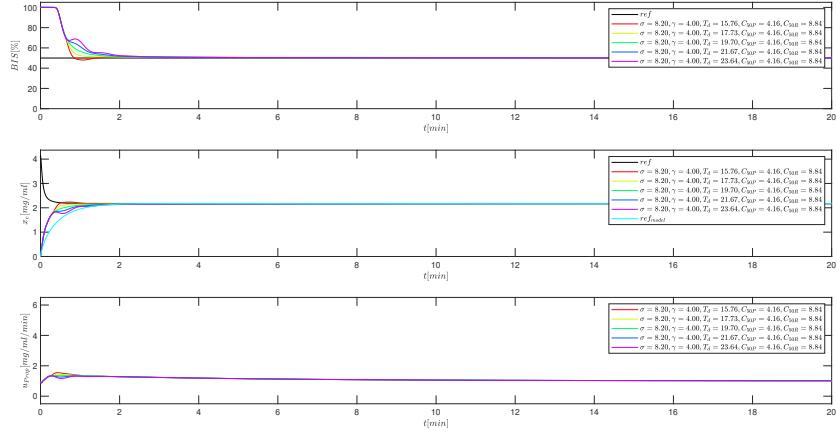


Figure 27: Analysis of the impact of incorrect estimates of parameter T_d (*main_pfc_pd_analysis.m* – 201 row: assembly experiment).

Figure 28 shows the impact of the parameter C_{50P} , which represents the concentration of propofol required to achieve 50% of the maximum effect. We can observe that incorrect estimates of this parameter can influence the overall dynamics, as an overestimated parameter represents a higher concentration needed to achieve the same effect.

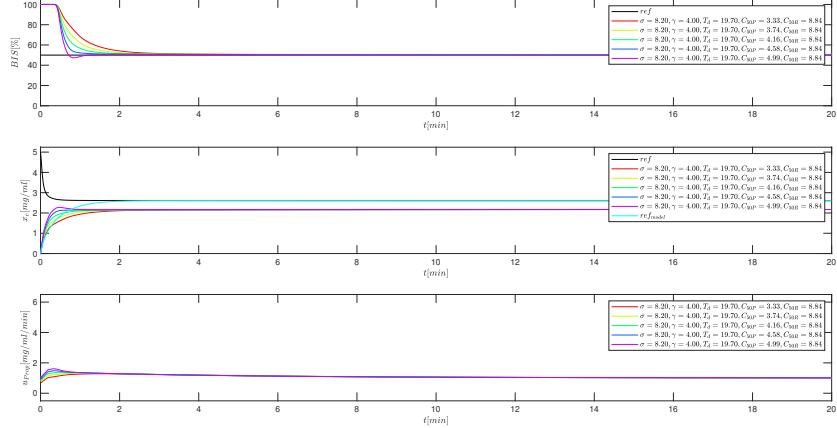


Figure 28: Analysis of the impact of incorrect estimates of parameter C_{50P} (*main_pfc_pd_analysis.m* – 201 row: assembly experiment).

Similarly, in Figure 29, we can observe the impact of the parameter C_{50R} , which represents the concentration of remifentanil required to achieve 50% of the maximum effect. The influence is smaller, partly because the concentration does not directly affect the feedback loop to the same extent as the concentration of the main manipulative variable (propofol).

In the simulations, which yielded the above-presented results of the analysis of the impact of incorrectly estimated parameters, we assumed that we knew the exact parameters of the PK models. However, in the case where we do not know these parameters precisely, the final results can, of course, differ. For instance, when we incorrectly estimate the PK model parameters within a range of 20% and also misestimate the PD model parameters, the simulation results are shown in Fig-

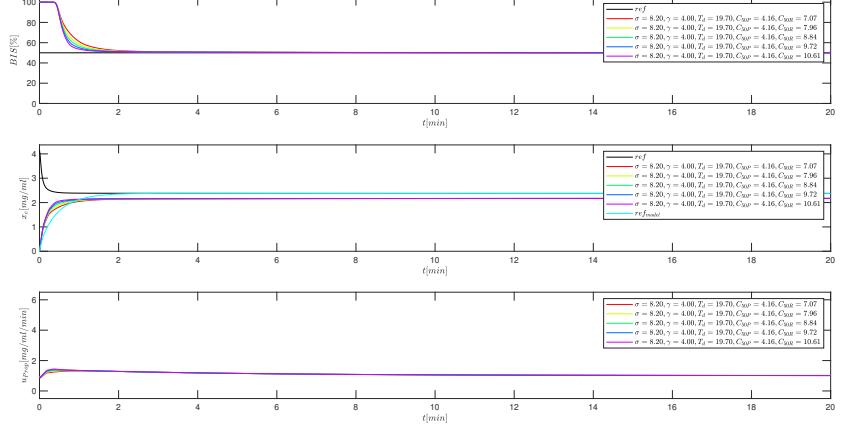


Figure 29: Analysis of the impact of incorrect estimates of parameter C_{50R} (*main_pfc_pd_analysis.m* – 201 row: assembly experiment).

ure 30.

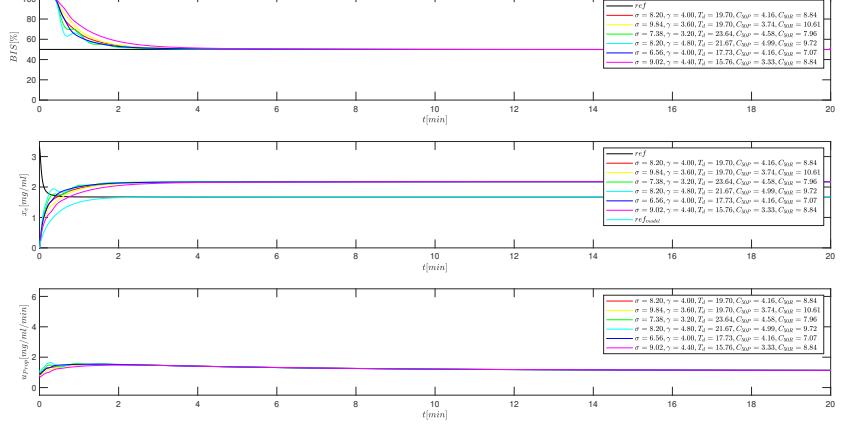


Figure 30: Analysis of the impact of several incorrectly estimated parameters (*main_pfc_detail_analysis.m* – 202 row: assembly experiment).

We can observe that even with incorrectly estimated multiple parameters, we still obtain stable results. However, these results would need to be quantitatively evaluated by specialists in the field to draw meaningful conclusions.

6 Conclusions

In this work, we have described the necessary theoretical background of the anesthesia depth control problem and connected it to the trend of personalized medicine. We have presented a comparison between the existing TCI algorithm and the developed algorithm for closed-loop anesthesia depth control based on EEG index and assessed the control quality under various levels of incorrectly estimated PK model parameters.

We can observe opportunities for further research in the development of a robust algorithm for real-time estimation of PK/PD model parameters. Such an algorithm would allow for adapting the models to the actual patient's state during anesthesia, resulting in even better control quality. To achieve this, the developed algorithms will need to be validated using real-world data to ensure their effectiveness and safety. This opens up additional possibilities for developing methods to estimate actual measurements in case of instrument failure in any form.

In conclusion, the described research and its results indicate that the use of EEG index is a step towards personalized medicine, and further exploration in this direction is essential to provide patients with the best possible healthcare services.

References

- [1] C. M. Ionescu, M. Neckebroek, M. Ghita in D. Copot, "An Open Source Patient Simulator for Design and Evaluation of Computer Based Multiple Drug Dosing Control for Anesthetic and Hemodynamic Variables," *IEEE Access*, vol. 9, str. 8680–8694, 2021. Conference Name: IEEE Access.
- [2] T. Gale, K. Leslie in M. Kluger, "Propofol anaesthesia via target controlled infusion or manually controlled infusion: effects on the bispectral index as a measure of anaesthetic depth," *Anaesthesia and Intensive Care*, vol. 29, str. 579–584, dec. 2001.
- [3] L. F. Laso, A. López-Picado, E. O. de La Fuente, A. M. Murua, C. Sánchez-Castro, L. P. Ruilope in C. Valero-Martínez, "Manual vs. target-controlled infusion induction with propofol: An observational study," *Colombian Journal of Anesthesiology*, vol. 44, str. 272–277, okt. 2016. Publisher: Elsevier.
- [4] S. L. Shafer in K. M. Gregg, "Algorithms to rapidly achieve and maintain stable drug concentrations at the site of drug effect with a computer-controlled infusion pump," *Journal of Pharmacokinetics and Biopharmaceutics*, vol. 20, str. 147–169, apr. 1992.
- [5] H. Schwilden, "A general method for calculating the dosage scheme in linear pharmacokinetics," *European Journal of Clinical Pharmacology*, vol. 20, str. 379–386, sept. 1981.
- [6] Y. Punjasawadwong, A. Phongchiewboon in N. Bunchungmongkol, "Bispectral index for improving anaesthetic delivery and postoperative recovery," *The Cochrane Database of Systematic Reviews*, vol. 2014, str. CD003843, jun. 2014.
- [7] I. Škrjanc, "Inteligentni sistemi za podporo odločanju." , 2016.
- [8] M. Boroujerdi Alavi in M. Tabatabaei, "Control of depth of anaesthesia using fractional-order adaptive high-gain controller," *IET Systems Biology*, vol. 13, str. 36–42, feb. 2019.
- [9] "Control of Processes with Long Dead Time: The Smith Predictor - MATLAB & Simulink Example - MathWorks United Kingdom." Dosegljivo: <https://uk.mathworks.com/help/control/ug/control-of-processes-with-long-dead-time-the-smith-predictor.html#d124e35030>. [Dostopano: 30. 12. 2022].
- [10] E. F. Camacho in C. Bordons, *Model Predictive control*. Advanced Textbooks in Control and Signal Processing, London: Springer, 2007.
- [11] P. Vallejo in P. Vega, "Practical Computational Approach for the Stability Analysis of Fuzzy Model-Based Predictive Control of Substrate and Biomass in Activated Sludge Processes," *Processes*, vol. 9, str. 531, mar. 2021.