UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Graduado en Ingeniería de la Salud

Diseño de un Sistema de Control de Anestesia Basado en Máquinas de Soporte Vectorial

Design of a Support Vector Machine-Based Anesthesia Control System

Realizado por
Gonzalo Mesas Aranda

Tutorizado por
Francisco Javier Fernández de Cañete Rodríguez

Departamento
Ingeniería de Sistemas y Automática
UNIVERSIDAD DE MÁLAGA

MÁLAGA, junio de 2025

UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADUADO EN INGENIERÍA DE LA SALUD

Diseño de un Sistema de Control de Anestesia Basado en Máquinas de Soporte Vectorial

Design of a Support Vector Machine-Based Anesthesia Control System

Realizado por
Gonzalo Mesas Aranda

Tutorizado por
Francisco Javier Fernández de Cañete Rodríguez

Departamento
Ingeniería de Sistemas y Automática

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2025

Fecha defensa: julio de 2025

Abstract

This work presents the design and implementation of a closed-loop anesthesia control system based on Generalized Predictive Control (GPC) using a Support Vector Machine (SVM) model. The objective is to regulate the depth of anesthesia, measured through the Bispectral Index (BIS), by automatically controlling the infusion rate of propofol. A three-compartment pharmacokinetic/pharmacodynamic (PK/PD) model (Schnider model) was used to simulate the patient dynamics. The SVM model was trained to predict BIS values based on historical infusion data and BIS trends, enabling the predictive controller to estimate future states and optimize the control signal accordingly.

The proposed architecture was compared to a traditional PID-based controller under identical simulation conditions in Simulink. Several performance metrics were evaluated, including overshoot, settling time, mean squared error (MSE), integral of absolute error (IAE), and integral of squared error (ISE). Additional experiments introduced perturbations such as reference changes, measurement noise, and infusion loss to assess robustness.

Results show that the SVM-based controller achieves smoother control actions and comparable or improved performance over the PID controller across all evaluated metrics. The system demonstrated robustness in the presence of disturbances, with better anticipation and recovery characteristics in some scenarios. These findings suggest that data-driven predictive control strategies are a viable alternative to classical approaches for automated anesthesia delivery and offer promising potential for patient-specific adaptation in clinical applications.

Keywords: Predictive Control, Support Vector Machine (SVM), Anesthesia automation, BIS monitoring, Pharmacokinetic/pharmacodynamic (PK/PD) modeling

Resumen

Este trabajo presenta el diseño e implementación de un sistema de control en lazo cerrado para la administración de anestesia, basado en Control Predictivo Generalizado (GPC) utilizando un modelo de Máquinas de Vectores de Soporte (SVM). El objetivo es regular la profundidad anestésica, medida a través del Índice Bispectral (BIS), controlando automáticamente la tasa de infusión de propofol. Para simular la dinámica del paciente, se utilizó un modelo farmacocinético/farmacodinámico (PK/PD) de tres compartimentos (modelo de Schnider). El modelo SVM fue entrenado para predecir los valores de BIS a partir de datos históricos de infusión y tendencias del BIS, permitiendo al controlador predecir estados futuros y optimizar la señal de control.

La arquitectura propuesta fue comparada con un controlador PID tradicional bajo condiciones de simulación idénticas en Simulink. Se evaluaron métricas de rendimiento como el sobreimpulso, el tiempo de establecimiento, el error cuadrático medio (MSE), el error absoluto integral (IAE) y el error cuadrático integral (ISE). Además, se realizaron experimentos con perturbaciones, incluyendo cambios de referencia, ruido en la medición y pérdida de infusión, para evaluar la robustez de ambos sistemas.

Los resultados muestran que el controlador basado en SVM proporciona acciones de control más suaves y un rendimiento comparable o superior al del controlador PID en todas las métricas evaluadas. El sistema demostró una buena capacidad de recuperación ante perturbaciones, con mejor anticipación y tiempo de respuesta en ciertos escenarios. Estos resultados sugieren que las estrategias de control predictivo basadas en datos son una alternativa viable a los métodos clásicos para la administración automatizada de anestesia y ofrecen un gran potencial de adaptación personalizada en aplicaciones clínicas.

Palabras Clave: **Control Predictivo, Máquinas de Vectores de Soporte (SVM), Automatización de anestesia, Monitoreo de BIS, Modelado farmacocinético/farmacodinámico (PK/PD)**

Contents

1 Introduction

1.1 Objectives

The main objective of this project is to design and implement a control architecture based on Support Vector Machine (SVM) regression for the automatic administration of anesthesia during surgery. This involves modeling the patient's pharmacokinetic and pharmacodynamic (PK/PD) response using a mathematical model, training an SVM model to predict the patient's response to anesthetic infusion, and developing the corresponding SVM-based control strategy.

To assess the effectiveness of the proposed approach, a classical Proportional-Integral-Derivative (PID) controller will also be implemented and evaluated under the same conditions. This comparison aims to highlight the potential advantages of the SVM-based predictive control.

1.2 State of the Art

Effective anesthesia management is critical for patient safety and surgical success. Anesthesia must maintain a balance of hypnosis, analgesia and physiological stability; failure can lead to awareness or postoperative complications [1], [2]. With millions of surgeries worldwide requiring intravenous anesthesia (especially propofol-based total intravenous anesthesia (TIVA)), optimizing drug delivery and monitoring is essential. For example, inadequate anesthesia under-dosing can cause intraoperative awareness, while over-dosing delays recovery [2]. Thus, scientific research aims to improve monitoring and control methods (e.g. EEG/BIS processing, feedback control algorithms) to enhance precision, reduce anesthetic complications, and ease clinician workload [1], [3].

Automatic anesthesia controllers have employed a range of control algorithms. The most

common are PID controllers, which continuously adjust drug infusion to drive a measured index (e.g. BIS or EEG-based depth) to a setpoint. In practice, multi-loop PID schemes are used — for example, one loop may regulate hypnosis (BIS), another analgesia (via blood pressure/heart rate) — sometimes in a multivariable (MIMO) design [2]. PIDs are popular for their simplicity and ease of implementation, and many studies (simulations and clinical) have validated their basic efficacy. However, PIDs assume linear, time-invariant behavior and must be tuned, so patient-specific pharmacokinetics can degrade performance. In real patients, nonlinearities and inter-patient variability often lead to steady-state errors or oscillations. For example, researchers have found that a standard PID may require gain scheduling or an additional feedforward term to handle Propofol's saturating dose–response [2].

To overcome the shortcomings of PID approaches, more advanced strategies have been explored. Model Predictive Control (MPC) uses an explicit PK–PD nonlinear model of the patient to predict future BIS and optimize dosing over a horizon. Clinical simulations report that MPC can better manage disturbances and multi-drug interactions than PID — for instance, MPC controllers outperformed PID in trials of propofol/remifentanil infusion under patient variability [1]. In MPC, machine learning models can be trained on past anesthesia records to predict BIS or patient response to propofol. These can capture complex dynamics and patient variability. For example, Support Vector Regression (SVR) is noted for handling nonlinear BIS–drug relationships [4]. As noted in [3], building predictive models of anesthesia depth is "of great value" for research and clinical practice, enabling new methods to monitor and maintain anesthetic levels during surgery. Overall, machine learning–based models offer a promising complement to traditional control; they may enable predictive control, where the controller anticipates patient response (via learned models) and adjusts dosing proactively rather than purely reactively.

TIVA is a common target for closed-loop control due to propofol's rapid kinetics and established PK–PD models [2]. Anesthesiologists often use Target-Controlled Infusion (TCI) systems based on models (e.g. Schnider/Minto) to set a desired plasma or effect-site concentration of propofol. In closed-loop schemes, the Bispectral Index (BIS) — a processed EEG score ranging from 0 (deep unconsciousness) to 100 (fully awake) — is the preferred controlled variable. Clinically, maintaining BIS in the range ~40–60 is considered optimal for surgical anesthesia [4]. Closed-loop controllers automatically adjust propofol infusion every few seconds to keep

BIS at the target. Studies show that such systems can keep the patient in the desired hypnotic range the vast majority of the time; for example, one closed-loop propofol/SEDLine system maintained the target BIS-equivalent (PSI) 92% of surgery time [5].

1.3 Justification of the Project

Effective control of anesthesia during surgery remains a critical challenge in biomedical engineering, with direct implications for patient safety, surgical outcomes, and clinical workflow. While conventional closed-loop systems—especially those based on PID (Proportional-Integral-Derivative) control—have demonstrated practical utility in regulating hypnotic states using indicators such as the Bispectral Index (BIS), their limitations are well-documented. These include reduced robustness to patient variability, limited capacity to adapt to nonlinear pharmacodynamics, and reliance on manual gain tuning [1], [2].

In response to these challenges, the research community has explored more advanced techniques, including Model Predictive Control (MPC), reinforcement learning, and data-driven modeling. Within this context, Support Vector Machines (SVM), and particularly Support Vector Regression (SVR), have emerged as a powerful method for learning nonlinear relationships between anesthetic dose and physiological response [3], [4]. However, the majority of studies using SVMs in anesthesia focus on prediction tasks, such as estimating BIS or forecasting infusion adjustments based on historical data [3]. To date, there is a notable absence of studies where SVMs are used directly within a closed-loop control architecture for real-time infusion rate optimization during surgery.

This gap in the literature motivates the present project. By designing and implementing an SVM-based control architecture that predicts the BIS response and adjusts the propofol infusion rate accordingly, this work seeks to evaluate whether machine learning–based predictive control can outperform traditional controllers in a simulated surgical context. To the best of the author's knowledge, this constitutes one of the first attempts to apply SVM-based models in a real-time closed-loop anesthesia control system. Moreover, the project benchmarks the proposed controller against a classic PID design, contributing not only an empirical comparison but also a reproducible methodology that may guide future developments in intelligent anesthesia systems.

1.4 Structure of the Document

This document is organized into eleven chapters, each contributing to the development and evaluation of an automatic control system for anesthesia administration based on Support Vector Machine (SVM) regression.

- **Chapter 1** introduces the topic, presents the objectives, motivation, state of the art, and the justification for the project.
- **Chapter 2** presents the theoretical background necessary to understand the pharmacological and control system principles involved, including concepts related to pharmacokinetics/pharmacodynamics and machine learning.
- **Chapter 3** describes the methodology followed to develop the system, including the simulation approach, model training, and evaluation procedure.
- **Chapter 4** details the materials and software tools used throughout the project.
- **Chapter 5** explains the mathematical modeling of the patient using the Schnider model, which simulates the effect of propofol in the body.
- **Chapter 6** discusses the process of training the SVM model to predict the Bispectral Index (BIS) as a function of the propofol infusion rate and prior state.
- **Chapter 7** presents the design and implementation of the SVM-based predictive control architecture.
- **Chapter 8** describes the PID-based control architecture, which is used as a baseline for comparison.
- **Chapter 9** contains the experimental setup and results, where both controllers are evaluated and compared in terms of accuracy, stability, and performance.
- **Chapter 10** discusses the results and performance of both controllers stated in Chapter 9.
- **Chapter 11** summarizes the main conclusions drawn from the project and proposes future lines of research.

- **Chapter 12** provides the same conclusions and future work proposals as Chapter 11, but in Spanish, as required by university regulations.

Finally, the **appendix** contains the source code developed for this work.

2

Theoretical Framework

This section presents a theoretical background necessary for the understanding of the methodology followed in this project.

2.1 Pharmacokinetic Modeling

Pharmacokinetic (PK) models describe how an administered drug is distributed and eliminated in the body over time. A common approach is a **three-compartment model**, shown in Figure 1, which partitions the body into a central compartment (plasma and highly perfused tissues) and two peripheral compartments (a shallow/rapidly equilibrating compartment and a deep/slow one). In this scheme, propofol (for example) is infused into the central compartment (volume V_1) at rate $U(t)$ and cleared from it with elimination rate k_{10}. Drug exchanges occur between the central and shallow compartments (volumes V_2, rate constants k_{12} and k_{21}) and between the central and deep compartments (volume V_3, rate constants k_{13} and k_{31}) [6]. The dynamics are governed by the differential equations (for concentrations C_1, C_2, C_3):

$$\frac{dC_1(t)}{dt} = -\left(k_{10} + k_{12} + k_{13}\right) C_1(t) + k_{21}C_2(t) + k_{31}C_3(t) + \frac{U(t)}{V_1}, \tag{1}$$

$$\frac{dC_2(t)}{dt} = k_{12}C_1(t) - k_{21}C_2(t), \tag{2}$$

$$\frac{dC_3(t)}{dt} = k_{13}C_1(t) - k_{31}C_3(t). \tag{3}$$

In practice, the widely used Schnider model for propofol PK is a three-compartment model whose parameters are adjusted for patient demographics [7]. For instance, Schnider et al. fixed V_1, V_3, k_{13}, k_{31} at constant values but scaled V_2, k_{12}, k_{21} by age, and scaled k_{10} (clearance) by

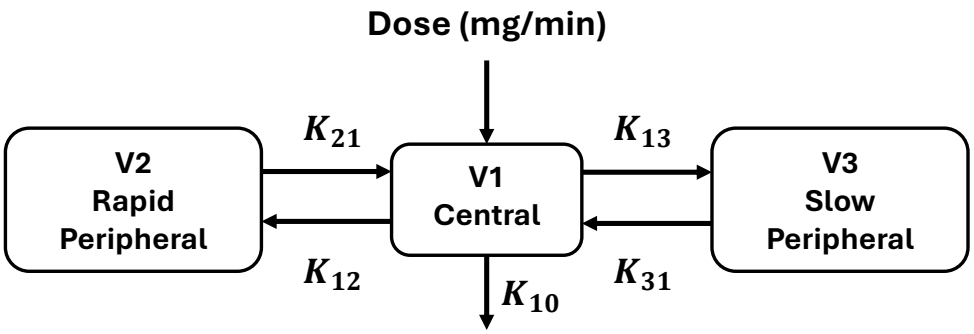


Figure 1: Representation of the three-compartment model.

weight, height, and lean body mass. This yields a personalized PK model for each patient based on age, weight, height and gender [8].

A fourth **effect-site compartment** (volume V_e) is often added to represent the biophase (e.g. brain) for pharmacodynamics (PD). This virtual compartment does not exchange mass back to the plasma, but equilibrates with the central compartment with rate constant k_{e0}. Its concentration $C_e(t)$ follows:

$$\frac{dC_e(t)}{dt} = k_{e0}\left[C_1(t) - C_e(t)\right] \tag{4}$$

The effect-site compartment allows modeling the delay between plasma levels and drug effect without altering the plasma PK equations [6].

2.2 Pharmacodynamic Modeling

Pharmacodynamics (PD) describes the relationship between drug concentration and biological effect. In the context of propofol anesthesia, the Bispectral Index (BIS) is a widely used effect measure. BIS is a processed electroencephalogram (EEG) parameter scaled from 0 to 100, where a value near 100 corresponds to an awake state, 40–60 is considered optimal for surgical anesthesia, and values near 0 indicate deep unconsciousness. Clinically, maintaining BIS within the 40–60 range is crucial to ensure adequate anesthesia depth and minimize the risk of intraoperative awareness [4].

The BIS response to propofol concentration is commonly modeled using a sigmoid E_{max}

(Hill-type) function, capturing the nonlinear inhibitory relationship between effect-site concentration and hypnotic effect. The model takes the following form:

$$BIS(C_e) = BIS_{base} \cdot \frac{C_{50}^{\gamma}}{C_{50}^{\gamma} + C_e^{\gamma}} \tag{5}$$

Here, BIS_{base} represents the baseline BIS value in the absence of drug (typically around 100), C_e is the effect-site concentration of propofol, C_{50} is the concentration that produces 50% of the maximal effect, and γ is the Hill coefficient determining the steepness of the concentration–effect curve. This pharmacodynamic (PD) model reflects the decreasing sigmoidal nature of BIS with increasing C_e, capturing the saturation behavior of propofol's effect on cortical activity.

2.3 Support Vector Machines

A Support Vector Machine (SVM) is a supervised learning model that finds an optimal function (or decision hyperplane) fitting training data with maximum margin. In regression form (ε-Support Vector Regression, SVR), SVM seeks a function $f(x) = w^T x + b$ (possibly in a high-dimensional feature space via kernels) that deviates from the target values y_i by at most ε for most points, while minimizing model complexity. In this expression, w is the weight vector that determines the orientation of the regression hyperplane in the feature space, and b is the bias term (or offset), which shifts the hyperplane vertically. The standard ε-SVR optimization is:

$$\min_{w,b,\xi_i,\xi_i^*} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{6}$$

$$\text{subject to} \quad y_i - (w^T x_i + b) \leq \epsilon + \xi_i, \tag{7}$$

$$(w^T x_i + b) - y_i \leq \epsilon + \xi_i^*, \tag{8}$$

$$\xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, n. \tag{9}$$

where ξ_i, ξ_i^* are slack variables for points outside the "ε-tube". The solution is defined by support vectors, the training points with nonzero slack or lying on the tube boundary, which determine the regression function.

For nonlinear regression, a kernel function $K(x, x')$ (e.g. Gaussian RBF) is used. A common choice is the RBF kernel:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{10}$$

where σ is the kernel bandwidth [9], [10].

Key hyperparameters of SVR include:

- **BoxConstraint (C):** the regularization penalty on slack variables. Larger C enforces a smaller error margin (narrower tube) and can reduce bias at the risk of overfitting. A high C means errors are heavily penalized (complex model), while a low C allows more slack (simpler model).
- **Epsilon (ε):** the half-width of the ε-insensitive tube around the regression function. Errors (residuals) smaller than ε incur no loss. Increasing ε creates a wider tube (more tolerance).
- **KernelScale (σ):** the width of the Gaussian kernel. Smaller σ makes the RBF kernel narrower (allowing more complex, wiggly fits), whereas larger σ produces smoother functions. In practice, KernelScale can be set to 'auto' or tuned for best performance.

2.4 NARX Architecture

A **NARX (Nonlinear AutoRegressive with eXogenous inputs)** model is a dynamic system model in which the current output depends on past outputs and past (and possibly current) inputs. Formally, a NARX model can be written as:

$$y(t) = F(\, y(t-1), y(t-2), \ldots, y(t-N_y);\, u(t), u(t-1), u(t-2), \ldots, u(t-N_u)\,) \tag{11}$$

where y is the output (e.g. BIS) and u is the exogenous input (e.g. propofol infusion). In other words, the model regresses the current output on a vector of delayed outputs and inputs [9].

In practice, we use a NARX form to construct the SVM feature vector as shown in Figure 2. For example, if we use N_y past outputs and N_u past inputs, the input vector at time t is:

$$x(t) = [\, u(t), u(t-1), u(t-2), \ldots, u(t-N_u),\ y(t-1), y(t-2), \ldots, y(t-N_y)\,] \tag{12}$$

This captures the system's memory. By including delayed input (infusion) and output (BIS) terms, the model can represent the closed-loop dynamics of anesthesia delivery.

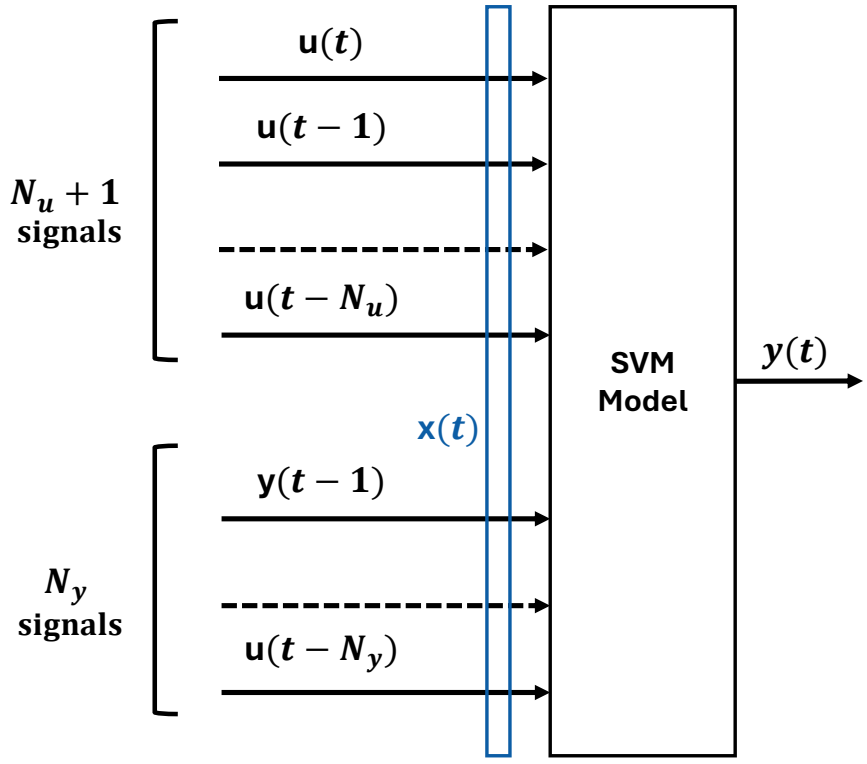


Figure 2: Representation of the NARX architecture used to construct the SVM feature vector.

2.5 PID Control

A Proportional-Integral-Derivative (PID) controller is one of the most widely used feedback control strategies in engineering systems due to its simplicity, robustness, and effectiveness in a wide range of applications [11]. It continuously calculates an error value as the difference between a desired reference signal and a measured process variable, and applies a correction based on proportional, integral, and derivative terms.

The control signal $u(t)$ generated by a PID controller is defined as:

$$u(t) = K_P\, e(t) + K_I \int_0^t e(t)\, dt + K_D \frac{de(t)}{dt} \tag{13}$$

where $e(t) = y_{ref}(t) - y(t)$ is the control error, K_P is the proportional gain, K_I is the integral gain, and K_D is the derivative gain. Each component has a specific function:

- The **proportional** term $K_P\, e(t)$ reacts to the current error.
- The **integral** term $K_I \int e(t)$ accounts for the accumulation of past errors, helping eliminate steady-state offsets.
- The **derivative** term $K_D \frac{de(t)}{dt}$ predicts future error based on its rate of change, improving stability and response.

In the context of anesthesia control, PID controllers are valued for their real-time simplicity, but may struggle with nonlinearity and inter-patient variability.

3

Methodology

This work follows a three-stage methodology, including patient modeling, control system design, and experimental evaluation, as outlined in the graphical abstract in Figure 3.

In the first stage, a patient model is constructed using the Schnider model, a PK/PD model commonly used in anesthesia simulation. This three-compartment model simulates the drug distribution and its effect on the BIS for a reference patient. The output of this model is used to generate training data for the control system, based on varying infusion profiles and the resulting BIS responses.

Next, two different control architectures are designed and implemented. The first is a predictive controller based on SVM regression. The SVM is trained to predict the BIS value at the next time step given previous BIS values and infusion rates. This predictive model is integrated into a control framework that applies gradient descent optimization to minimize a cost function. The function includes penalties for prediction error, sharp BIS fluctuations, and accumulated control error, allowing the controller to determine an optimal infusion rate dynamically. In parallel, a traditional PID controller is also implemented. This serves as a baseline for performance comparison.

In the final stage, both controllers are evaluated through closed-loop simulations of a 60-minute surgery scenario. In addition to the baseline simulation with a BIS reference of 50, robustness experiments are conducted, introducing reference changes, gaussian noise in the feedback, and temporary infusion loss. The performance of each controller is assessed using several metrics, including undershoot, settling time within clinical bounds, and accumulated error (MSE, IAE, and ISE). These results are analyzed to compare the effectiveness and robustness of both control strategies.

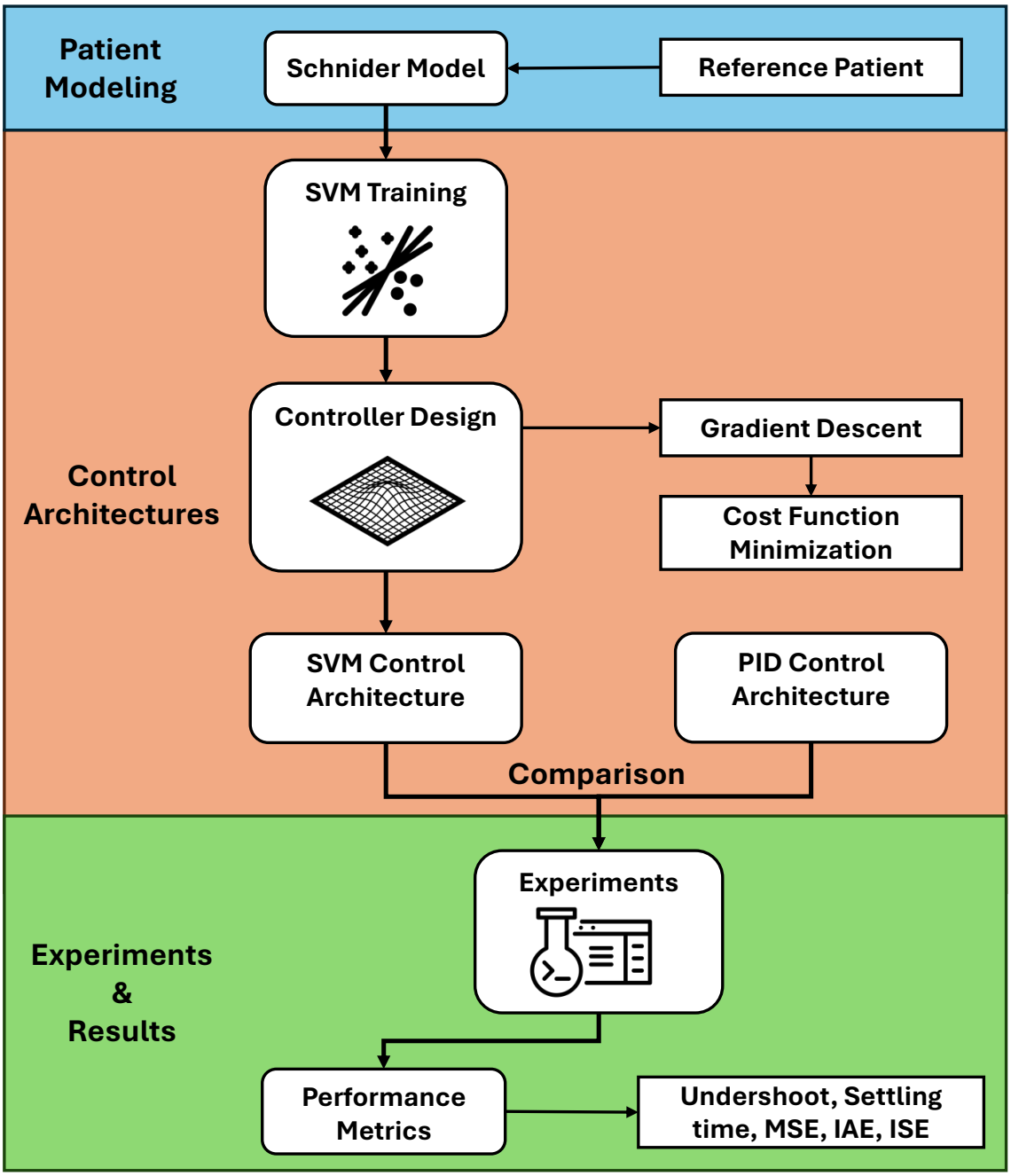


Figure 3: Graphical abstract of the methodology pipeline.

4 Materials

This work was developed using a combination of MATLAB/Simulink and Python-based tools. Table 1 summarizes the software and tools used for simulation, data preprocessing, and model training in MATLAB.

Library / Tool	Description	Version
MATLAB	Core environment for algorithm development	R2024b
Simulink	Simulation environment for dynamic systems	24.2
Image Processing Toolbox	Used for signal preprocessing (e.g., noise filtering)	24.2
Statistics and Machine Learning Toolbox	Used to train the SVM regression model	24.2

Table 1: MATLAB and Simulink tools used in the project.

Table 2 summarizes the libraries used for data visualization in Python.

Library	Description	Version
Python	Programming language used for data handling and visualization	3.9.20
Pandas	Data manipulation and preprocessing	2.3.0
NumPy	Numerical computing and array operations	2.0.2
Matplotlib	Static plotting library for result visualization	3.9.4
Seaborn	Statistical data visualization	0.13.2
Cycler	Customization of plot styles and cycles	0.12.1
SciencePlots	Publication-quality plot styling	2.1.1

Table 2: Python environment and libraries used in the project.

5

System Modeling: Schnider model

The lack of publicly available datasets containing real patient data for anesthesia Target-Controlled Infusion (TCI), combined with the ethical and practical constraints of testing unvalidated closed-loop controllers on humans, necessitates the use of mathematical models to simulate patient responses. Over the years, several PK models have been developed for this purpose, including the Marsh, Schnider, and Eleveld models [12]. The choice of model significantly affects the accuracy and safety of both TCI systems and closed-loop anesthesia controllers [13]. Therefore, careful model selection is crucial to ensure validity and applicability in the context of this project.

The Marsh model was among the first PK models for propofol, known for its simplicity and widespread clinical use. However, its basic scaling mechanisms can result in suboptimal dosing, such as larger induction boluses, potentially causing hemodynamic instability [14]. On the other end of the complexity spectrum, the Eleveld model is the most recent and comprehensive, incorporating extensive population variability. Despite this, multiple studies suggest that it offers no substantial advantage for closed-loop control design and may introduce complications in elderly patients when compared to the Schnider model [13], [15]. Given this balance between personalization and simplicity, the Schnider model emerges as the most appropriate choice for this project.

As detailed in Section 2.1, the Schnider model estimates the parameters of a three-compartment PK/PD model based on patient-specific physiological data. The following equations summarize the calculation of each model parameter:

Lean Body Mass (LBM):

$$\text{LBM} = \begin{cases} 1.1 \cdot \text{Weight} - 128 \cdot \left(\frac{\text{Weight}}{\text{Height}}\right)^2, & \text{for males} \\ 1.07 \cdot \text{Weight} - 148 \cdot \left(\frac{\text{Weight}}{\text{Height}}\right)^2, & \text{for females} \end{cases} \tag{14}$$

Volume of Distribution:

$$V_1 = 4.27 \quad \text{(L)} \tag{15}$$

$$V_2 = 18.9 - 0.391 \cdot (\text{Age} - 53) \quad \text{(L)} \tag{16}$$

$$V_3 = 238 \quad \text{(L)} \tag{17}$$

Clearance Parameters:

$$\begin{aligned} CL_1 = {} & 1.89 + 0.0456 \cdot (\text{Weight} - 77) - 0.0681 \cdot (\text{LBM} - 59) \\ & + 0.0264 \cdot (\text{Height} - 177) \quad \text{(L/min)} \end{aligned} \tag{18}$$

$$CL_2 = 1.29 - 0.024 \cdot (\text{Age} - 53) \quad \text{(L/min)} \tag{19}$$

$$CL_3 = 0.836 \quad \text{(L/min)} \tag{20}$$

Transfer Rate Constants (Derived):

$$k_{10} = \frac{CL_1}{V_1} \tag{21}$$

$$k_{12} = \frac{CL_2}{V_1} \tag{22}$$

$$k_{13} = \frac{CL_3}{V_1} \tag{23}$$

$$k_{21} = \frac{CL_2}{V_2} \tag{24}$$

$$k_{31} = \frac{CL_3}{V_3} \tag{25}$$

Effect-Site Dynamics:

$$k_{e0} = 0.456 \quad \text{(1/min)} \tag{26}$$

$$BIS_{\text{base}} = 100 \tag{27}$$

$$C_{e50} = 4.93 \quad \text{(mg/L)} \tag{28}$$

$$\gamma = 2.46 \tag{29}$$

In this project, simulations are carried out using a standard patient profile for simplicity. Table 3 presents the computed PK and PD parameters for a standard patient (Age = 18 years; Weight = 60 kg; Height = 164 cm; Gender = Male).

Parameter	Description	Value
LBM (14)	Lean Body Mass (kg)	48.8673
C_{e50}	Effect-site concentration for 50% BIS (mg/L)	4.9300
γ	Hill coefficient	2.4600
V_1	Central compartment volume (L)	4.2700
V_2 (16)	Rapid peripheral compartment volume (L)	28.6750
V_3	Slow peripheral compartment volume (L)	238.0000
CL_1 (18)	Central compartment clearance (L/min)	1.4616
CL_2 (19)	Rapid peripheral compartment clearance (L/min)	1.8900
CL_3 (20)	Slow peripheral compartment clearance (L/min)	0.8360
k_{10} (21)	Elimination rate constant (1/min)	0.3423
k_{12} (22)	Rate constant to rapid peripheral (1/min)	0.4426
k_{13} (23)	Rate constant to slow peripheral (1/min)	0.1958
k_{21} (24)	Rate constant from rapid peripheral (1/min)	0.0659
k_{31} (25)	Rate constant from slow peripheral (1/min)	0.0035
k_{e0}	Effect-site equilibration rate constant (1/min)	0.4560

Table 3: Schnider model parameters for a standard patient (Age = 18 years; Weight = 60 kg; Height = 164 cm; Gender = Male).

Using these parameters, the differential equations defined in (1) and (4) can be solved numerically. For this purpose, MATLAB's ode45 function was used [16]. This solver is based on an explicit Runge–Kutta (4,5) formula, the Dormand–Prince pair [17], which is well-suited for solving non-stiff ordinary differential equations with moderate accuracy requirements. It offers a good balance between precision and computational efficiency, making it a suitable choice for PK simulations in real time.

6

SVM model Training

6.1 Data Generation

Training a data-driven model such as a SVR requires representative and diverse training data. Initially, a dataset was generated using random square-wave pulses of the infusion rate, simulating abrupt changes in drug delivery. While this approach provided some variability, the generated signals exhibited unrealistic behavior compared to the patterns expected under the closed-loop control system. As a result, the SVR model trained on this data showed good performance during validation but performed poorly when tested under more realistic conditions during closed-loop control simulation.

To address this limitation, a new data generator was implemented using a random walk-like process. This method produces smoother, more gradual variations in the infusion rate, closely mimicking the behavior of a real-world controller. Although no quantitative comparison is presented, preliminary experiments and qualitative observations confirmed that the random walk-like input led to more robust and generalizable model performance. This approach ensures that the SVR model is trained on input-output pairs that are representative of the actual application scenario.

This random walk-like generator produces a set of simulations by following the procedure below:

1. **Infusion profile generation:** Starting at 22 mg/min (a commonly used initial infusion rate and the starting point of the controller), the generator applies a random change to the infusion rate at each time step (every 1 second) within the interval $[-\Delta, \Delta]$. The updated value is then clamped to remain within the interval [INFUSION_MIN, INFUSION_MAX].

2. **Gaussian low-pass filtering:** A Gaussian low-pass filter is applied to the generated

infusion signal using a fixed window size and standard deviation (σ). This step smooths the profile to reduce high-frequency variations and avoid abrupt transitions, as can be seen in figure 4.

Figure 4: Example comparison between the raw and filtered infusion profile used during SVM model training. This corresponds to the first simulation in the dataset.

3. **PK/PD simulation:** Using the Schnider model parameters for a standard patient generated, the infusion profile is used as input to solve the differential equations of the three-compartment model (see Equations (1) and (4)) with MATLAB's ode45 solver. The resulting concentrations and BIS time series are recorded and stored along with the infusion input.

The random generation process is seeded to ensure reproducibility of results. Table 4 sums up the hyperparameters used for the random walk-like data generation.

Parameter	Description	Value
INFUSION_MIN	Minimum infusion rate (mg/min)	0
INFUSION_MAX	Maximum infusion rate (mg/min)	40
DELTA	Max change per step in random walk (mg/min)	0.9
SIM_DURATION	Duration of each simulation (minutes)	90
NUM_SIMULATIONS	Number of simulations generated	100
WINDOW_SIZE	Size of the Gaussian filter window	54
SIGMA	Standard deviation of Gaussian filter	40

Table 4: Hyperparameters used for the random walk-like data generation.

6.2 Model Training and Validation

Once the synthetic dataset is generated, it must be preprocessed to fit the NARX architecture used by the SVR model. As described in Section 2.4, a NARX model uses both past inputs and outputs as context. In this case, the inputs are infusion rates (u) and the outputs are BIS values.

The feature vector for each prediction is constructed as follows:

$$BIS(t) = \left[\, u(t-N_u), \ldots, u(t-1), u(t),\ BIS(t-1), BIS(t-2), \ldots, BIS(t-N_y) \,\right] \quad (30)$$

where $u(t)$ and $BIS(t)$ are the infusion rate and BIS value at time t, and N_u and N_y are the number of lags for input and output context, respectively.

The dataset is split into training, validation, and test sets. A min-max normalization is applied independently to inputs and outputs, ensuring all values fall within a common range. The SVR model is then trained using the normalized training data. The validation set is used to evaluate performance during hyperparameter tuning.

A manual hyperparameter tuning process was performed. This approach was considered sufficient due to the relatively small number of hyperparameters and the empirical observation that optimal validation metrics did not always correlate with best closed-loop controller performance. Hence, final model selection also considered its behavior in downstream control tasks.

To ensure realistic simulation and avoid data leakage, validation and test input matrices were constructed in a sequential (online) manner: each feature vector was built step-by-step while simulating, preserving the causal relationship between t and past values. The first prediction is performed at $t = \max(N_u, N_y)$, with earlier values used solely as context.

Beyond standard metrics, an additional *directional consistency* metric was employed. At each prediction step, a small perturbation ($\varepsilon = 0.001$) is added to the current input $u(t)$, and the model's output is recomputed. If the perturbation is positive (i.e., an increased drug dose), the predicted BIS value should decrease. The percentage of time steps in which this expected behavior fails is used as an additional qualitative metric.

Table 5 summarizes the hyperparameters used in this SVR model.

Hyperparameter	**Description**	**Value**
N_u	Number of infusion rate lags	70
N_y	Number of BIS lags	70
Simulation duration	Duration of each simulation (min)	90
Δ	Infusion variation step size	0.9
Window size	Gaussian filter window length (samples)	54
σ	Gaussian filter standard deviation	40
ε	Input perturbation magnitude	0.001
Kernel	SVR kernel function	RBF
BoxConstraint	Regularization parameter	1
Epsilon	Epsilon-insensitive loss margin	0.01

Table 5: Hyperparameters used for SVR training and validation.

7

SVM-based Control Architecture

In this section, the implementation of each component of the SVM-based controller is described in detail. The control architecture adopted in this work is based on the one proposed by İplikçi in [9], where a Support Vector Machine-based Generalized Predictive Control (SVM-GPC) scheme is introduced. The method, illustrated in Figure 5, operates as follows:

1. A candidate input vector following the NARX architecture described in Section 2.4 is generated:

$$u_{\text{candidate}} = \left[u_r(n+1),\, u_r(n+2),\, \ldots,\, u_r(n+N_{u_r}) \right] \tag{31}$$

   This vector is passed through the trained SVM model to obtain a prediction of the system response (i.e., the patient's BIS response to the proposed infusion rates).

2. The following components are then provided to the cost function minimization (CFM) block:

$$\text{Predicted output:} \quad \hat{y} = \left[\hat{y}_q(n+1),\, \hat{y}_q(n+2),\, \ldots,\, \hat{y}_q(n+N_{y_q}) \right] \tag{32}$$

$$\text{Reference signal:} \quad \tilde{y} = \left[\tilde{y}_1,\, \tilde{y}_2,\, \ldots,\, \tilde{y}_Q \right] \tag{33}$$

$$\text{Current system state:} \quad y(n) = \left[y_1(n),\, y_2(n),\, \ldots,\, y_Q(n) \right] \tag{34}$$

   The CFM block —the controller— minimizes the prediction error by optimizing the candidate input vector, yielding an optimized input sequence:

$$u_{\text{optimal}} = \left[u_r^*(n+1),\, u_r^*(n+2),\, \ldots,\, u_r^*(n+N_{u_r}) \right] \tag{35}$$

3. The optimized vector's current input value $u_r^*(n+k)$ is applied to the system.

4. This process repeats at each sampling instant n, adapting the control based on updated measurements.

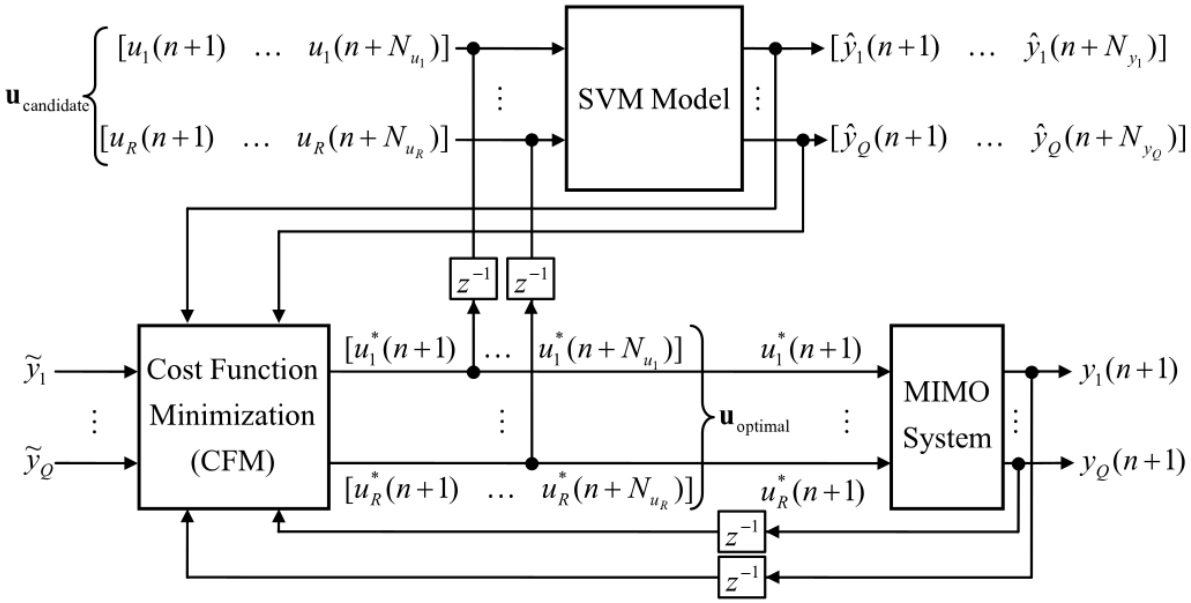


Figure 5: MIMO SVM-based GPC scheme proposed by İplikçi in [9].

Figure 6 shows the Simulink implementation of the proposed control architecture. It is composed of three main subsystems: (A) the MIMO plant, which models the patient and provides the current BIS value; (B) the $U_{\text{candidate}}$ generator, which proposes a candidate input vector for the optimization step in each iteration; (C) the controller, which predicts the BIS response using the trained SVM model, evaluates the cost, and optimizes the candidate input via a gradient descent algorithm.

This controller subsystem integrates the SVM Model and CFM blocks shown in Figure 5, implementing the SVM-GPC logic within Simulink.

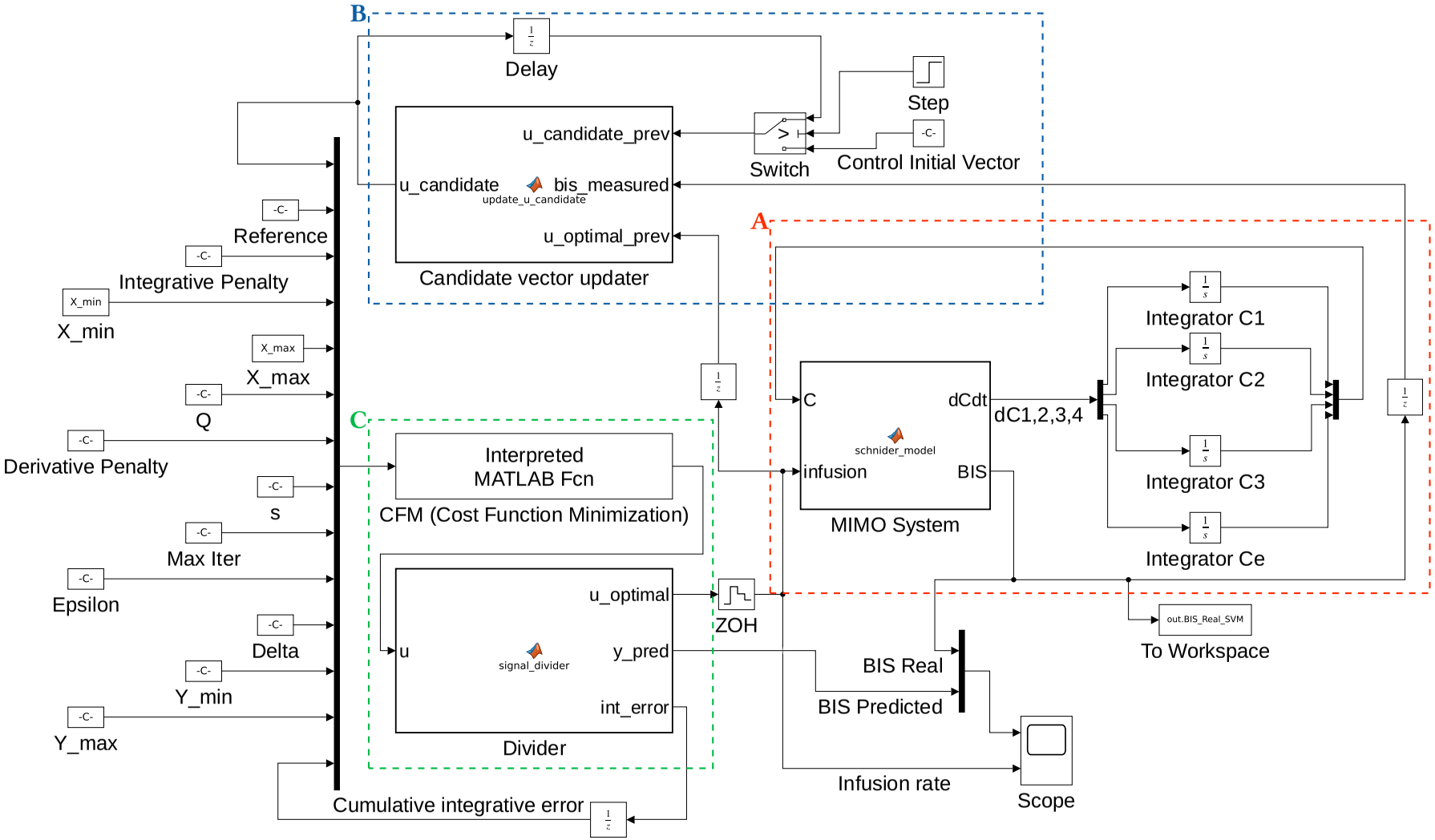


Figure 6: Simulink implementation of the SVM-based GPC architecture. Subsystems A, B, and C correspond to the MIMO plant (patient), the the $U_{\text{candidate}}$ generator, and the controller, respectively.

7.1 MIMO plant

The MIMO plant (see Figure 6, region A) is a mathematical representation of the patient, implemented using a three-compartment PK/PD model. It operates in continuous time and is used to compute the patient's current BIS value based on the administered infusion rate. This subsystem functions as follows:

1. A MATLAB Function block computes, at each time step, the BIS value and the derivatives of the drug concentrations in the central (C_1), rapid peripheral (C_2), slow peripheral (C_3), and effect-site (C_e) compartments. This computation is based on the current infusion rate and the previous concentration values. The function solves the model's system of differential equations using the Schnider parameter set.

2. Four integrator blocks ($\frac{1}{s}$) integrate the concentration derivatives, producing the updated compartment concentrations. These values are then fed back to the MATLAB Function block in the next iteration.

3. The calculated BIS value is passed to the next subsystem, the $U_{\text{candidate}}$ generator.

7.2 Candidate Vector Generator

The candidate vector generator (see Figure 6, region B) updates $U_{\text{candidate}}$ at each iteration, operating with a sampling time of $T_s = \frac{1}{60}$ minutes. It operates in two distinct modes depending on the current simulation time:

- **At simulation time $T = 0$:** Since no prior candidate vector exists, a predefined initial vector is used to start the simulation. Specifically, this vector assumes constant infusion rates of 22 mg/min (past and present), and a BIS context beginning at 100 (representing the awake state), with its subsequent evolution computed according to the infusion. This initial vector is obtained by simulating the three-compartment PK/PD model under these conditions.

- **At simulation time $T > 0$:** The candidate vector's context (past infusion rates and BIS values) is updated through a shift operation. The real BIS value from the MIMO plant and the previously applied infusion rate are appended to the context. The candidate infusion

rate proposed for the next step is initialized using the actual infusion rate applied in the previous iteration. This ensures that the optimization process always begins from the last known optimal input.

7.3 Cost Function Minimization (CFM)

The CFM (see Figure 6, region C) has the role of controller and is the subpart that optimizes the candidate vector and generates the optimal infusion rate. For the optimization, parameters and data shown in table 6 are passed to the CFM block.

Parameter	**Description**	**Value**
Reference	Target BIS value	50
Integrative Penalty	Penalty term for integrative action in cost function	3×10^{-4}
X_{min}	Minimum values used for SVM input normalization	Vector
X_{max}	Maximum values used for SVM input normalization	Vector
Q	Scalar weighting for prediction error in cost function	1.6
Derivative Penalty	Penalty for steep changes in BIS	2.5
s	Step size in steepest gradient descent	22×10^{-4}
Max_iter	Maximum number of optimization iterations	15
ϵ	Convergence threshold (cost improvement)	5×10^{-5}
Δ	Perturbation size to compute gradient direction	1×10^{-3}
Y_{min}	Minimum values for output SVM denormalization	Vector
Y_{max}	Maximum values for output SVM denormalization	Vector

Table 6: Default hyperparameters and data passed to the controller (CFM block).

The CFM uses a cost function to estimate a scalar value representing the total control error. This cost is minimized in the optimization loop. The proposed cost function is:

$$Cost = Q \cdot (y_{\text{ref}} - \tilde{y})^2 + \text{deriv\_penalty} \cdot \left(\frac{\tilde{y} - y_{\text{past}}}{T_s} \right)^2 + \text{int\_penalty} \cdot (e_{\text{int}})^2 \tag{36}$$

where $\tilde{y}$ is the predicted BIS value by the SVM model; y_{ref} is the reference BIS value; y_{past} is the previous BIS value; e_{int} is the accumulated integral error; T_s is the sampling time; and Q, deriv_penalty and int_penalty are scalar weights.

The optimization procedure follows these steps:

1. **Initialize cost:** Compute the initial cost using the proposed candidate vector.
2. **Optimization loop:**
   (a) **Estimate the gradient:** Slightly perturb the candidate vector by increasing the current infusion rate by a small amount Δ. Recalculate the cost to estimate the gradient direction (opposite of the cost increase).
   (b) **Suggest optimal infusion rate:** Adjust the candidate infusion rate by taking a small step s in the direction of the negative gradient. Recalculate the cost.
   (c) **Exit conditions:** If the maximum number of iterations is reached or the relative improvement in cost falls below the threshold ϵ, the loop terminates.
3. **Propagate signal:** The optimal infusion rate and the updated integral accumulated error are propagated through the control architecture.

8

PID-based Control Architecture

As discussed in previous sections, a PID-based closed-loop control architecture was implemented to compare the performance of the SVM-based GPC with a traditional controller. Figure 7 shows the Simulink implementation of the PID-based system. It consists of the same MIMO plant used in the SVM-based GPC, a continuous-time PID controller, a rate limiter block (to constrain the rate of change of the control signal), and the reference signal input.

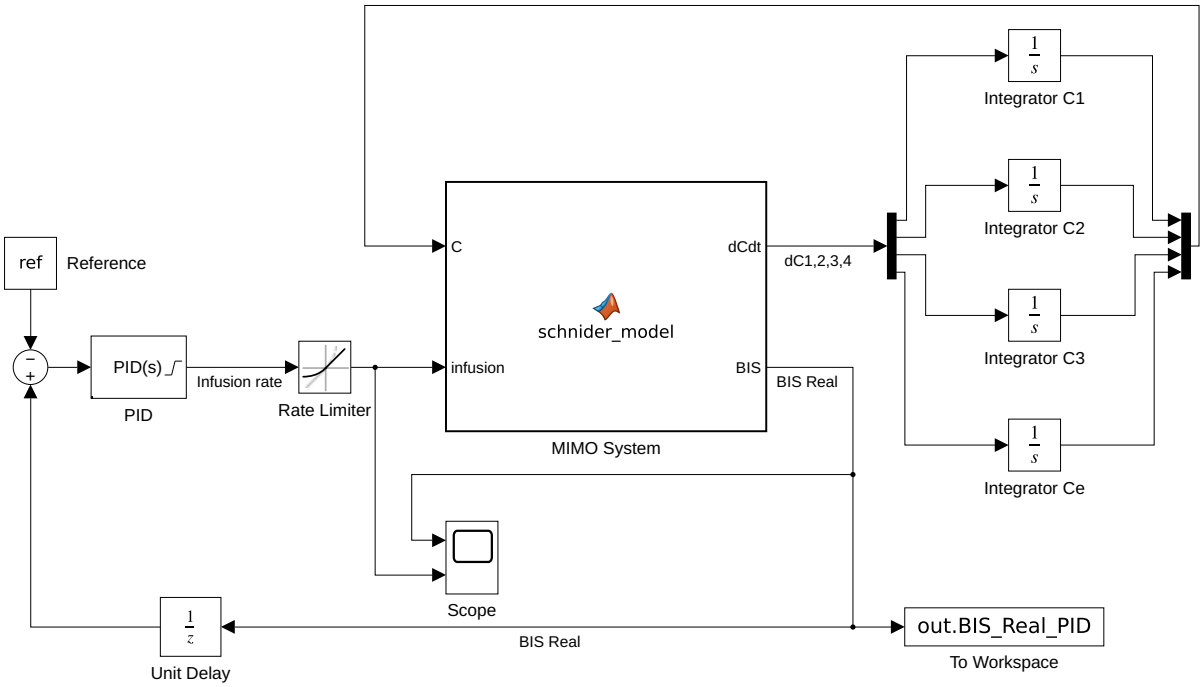


Figure 7: Simulink implementation of the PID-based architecture.

The tuning of the PID parameters was performed manually using an iterative trial-and-error process, following the standard sequential approach:

1. **Proportional gain (K_p) tuning:** The integral and derivative gains (K_i and K_d) were initially set to zero. The proportional gain was increased gradually until the BIS value approached the reference within a reasonable settling time, without inducing excessive oscillation.

2. **Integral gain (K_i) tuning:** To eliminate steady-state error, the integral gain was adjusted. A balance was sought between eliminating the offset quickly and avoiding excessive oscillations or instability.

3. **Derivative gain (K_d) tuning:** Finally, the derivative gain was introduced to reduce undershoot and to dampen the response, improving the overall stability and smoothness of the BIS signal.

The hyperparameters used in this implementation are shown in Table 7.

Parameter	**Description**	**Value**
K_p	Proportional gain	2.5
K_i	Integral gain	0.1
K_d	Derivative gain	0.1
Reference	Target BIS value	50
Rising slew rate	Maximum increase rate of infusion (mg/min)	2
Falling slew rate	Maximum decrease rate of infusion (mg/min)	-2

Table 7: Hyperparameters for the PID-based control architecture.

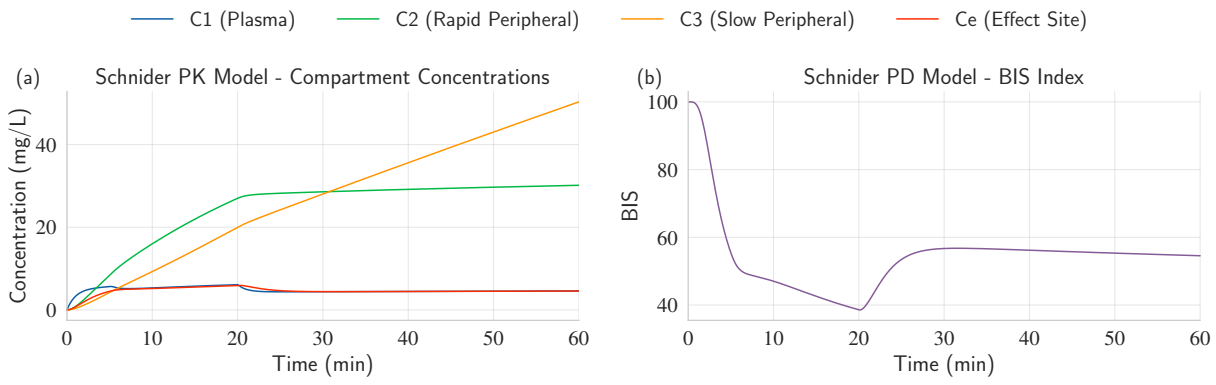
9

Experiments and Results

The purpose of this section is to present the experimental results obtained from different components of the system, in order to evaluate both performance and robustness.

9.1 Open-loop Response of the Three-Compartment PK/PD Model

Figure 8 shows the open-loop response of the three-compartment PK/PD model to a standard infusion profile (see subfigure 8(b)). Subfigure 8(a) plots the concentration trajectories of the four compartments over time: C1 (plasma), C2 (rapid peripheral), C3 (slow peripheral), and Ce (effect site). Subfigure 8(c) focuses on C1 and Ce only, allowing a clearer comparison between plasma and effect-site concentrations. Finally, subfigure 8(b) displays the BIS response, which drops from 100 (awake state) to within the target range of 40–60 in approximately 5 minutes and remains stable in this range for the majority of the simulation.



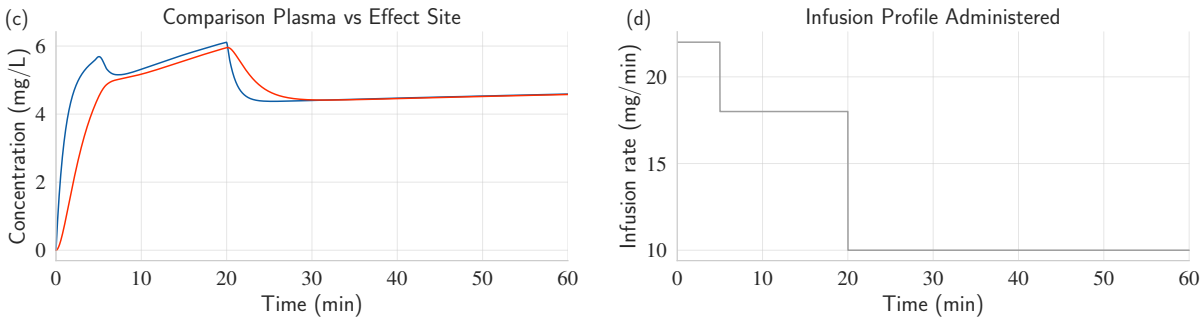


Figure 8: Open-loop simulation of the Three-Compartment PK/PD model using a standard infusion profile.

9.2 SVM Training Validation

Figure 9 shows the performance of the SVM model on the test simulation. Subfigure 9(a) displays the real BIS time series along with the SVM predicted and perturbed BIS profiles. The predicted and perturbed curves are nearly identical, since only a small perturbation was applied. The SVM predictions generally follow the real BIS dynamics.

It is important to note that the contexts for the predicted and perturbed BIS profiles are constructed using the SVM predictions themselves, whereas in the control architecture, the context is constructed using actual BIS values. This can introduce compounding error in open-loop evaluation.

Subfigure 9(b) illustrates the difference between the perturbed and predicted profiles. The relatively low directional consistency metric highlights the importance of robust error correction mechanisms within the controller.

Table 8 summarizes the performance metrics computed for this test simulation.

Metric	**Description**	**Value**
MSE	Average squared difference in BIS prediction	54.70
MAE	Average absolute difference in BIS prediction	6.69
R^2	Proportion of variance in BIS explained by the model	0.80
Directional Consistency	Percentage of correct behavior in (ΔBIS)	54.26%

Table 8: SVM Test Validation Metrics.

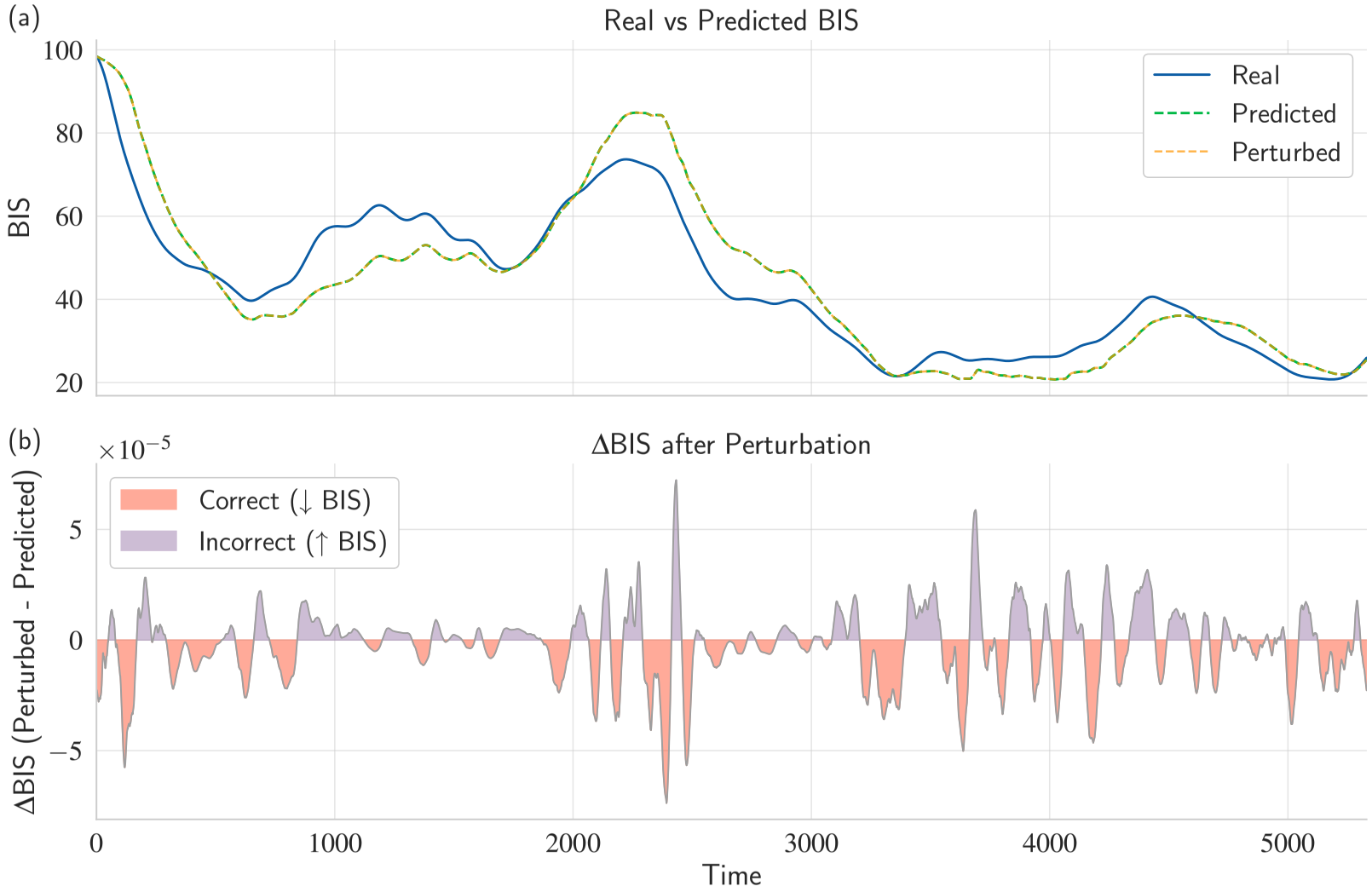


Figure 9: Test simulation results. (a) Real BIS profile vs. predicted and perturbed profiles from the SVM. (b) Difference between perturbed and predicted BIS profiles.

9.3 Closed-Loop Simulations

Figure 10 shows the Simulink simulation of a 60-minute surgery using the SVM-based controller with the reference point set to 50, the clinical standard. In the top subplot, the BIS profile shows a rapid and smooth response. In the bottom subplot, the anesthesia infusion profile is as well smooth, without abrupt changes—an indication of controlled and stable behavior.

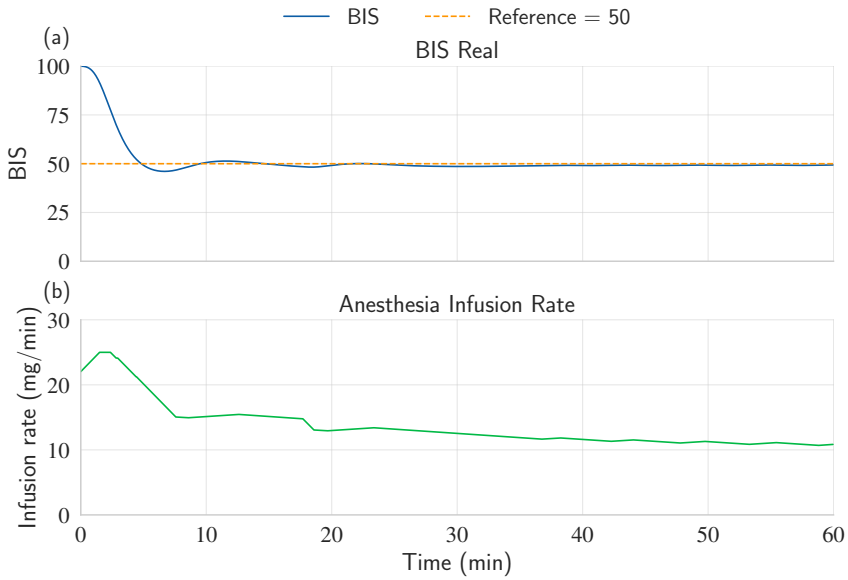


Figure 10: Closed-loop simulation of the SVM-based controller over a 60-minute surgery. Top: Real vs. predicted BIS. Bottom: Infusion rate profile.

Figure 11 presents the same surgery duration simulated with the PID controller. In the top subplot, the BIS profile shows a response comparable to that of the SVM simulation. However, the bottom subplot reveals a more abrupt anesthesia infusion profile, which may be less clinically desirable.

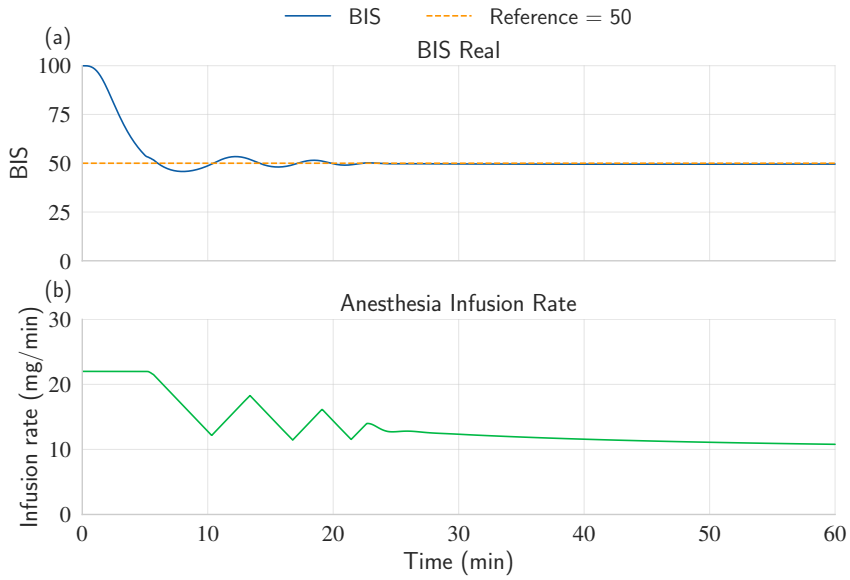


Figure 11: Closed-loop simulation of the PID-based controller over a 60-minute surgery. Top: BIS response. Bottom: Infusion rate profile.

9.3.1 Perturbation experiments

In order to evaluate and compare the robustness of both control architectures, a set of perturbation experiments was conducted.

Reference change

Three experiments were performed in which the reference BIS value was altered from the baseline of 50 to: (i) 40, (ii) 60, and (iii) a triangular waveform oscillating between 45 and 55 with a 20-minute period.

Figure 12 shows the closed-loop simulation of the SVM-based controller for the experiment in which the reference signal was set to 40. The BIS response is smooth and stable; however, a noticeable deviation from the reference can be observed over an extended portion of the simulation, even though the values remain within the clinically acceptable range.

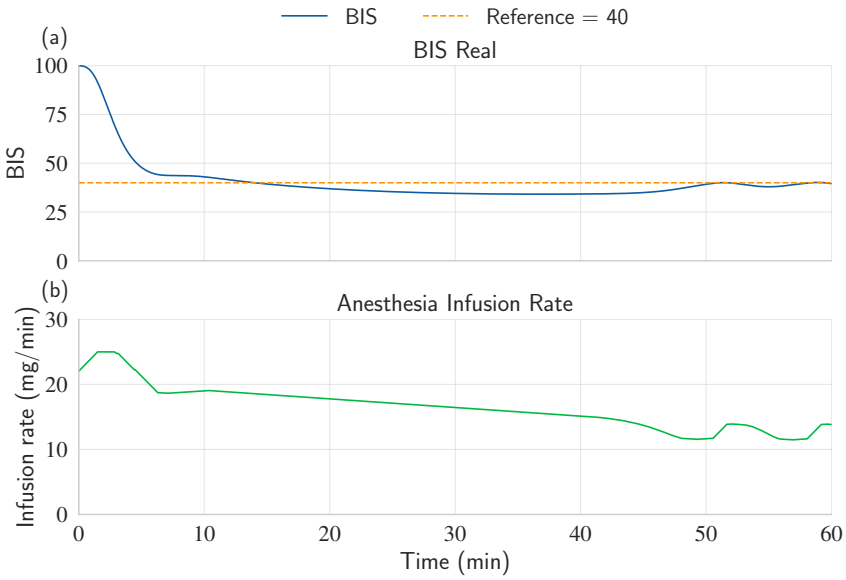


Figure 12: Closed-loop simulation of the SVM-based controller for the reference change to 40 experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 13 shows the corresponding closed-loop simulation for the PID-based controller under the same reference condition. Compared to the SVM controller, the PID controller reaches and maintains the reference value more closely, with minimal steady-state error and a similarly smooth response.

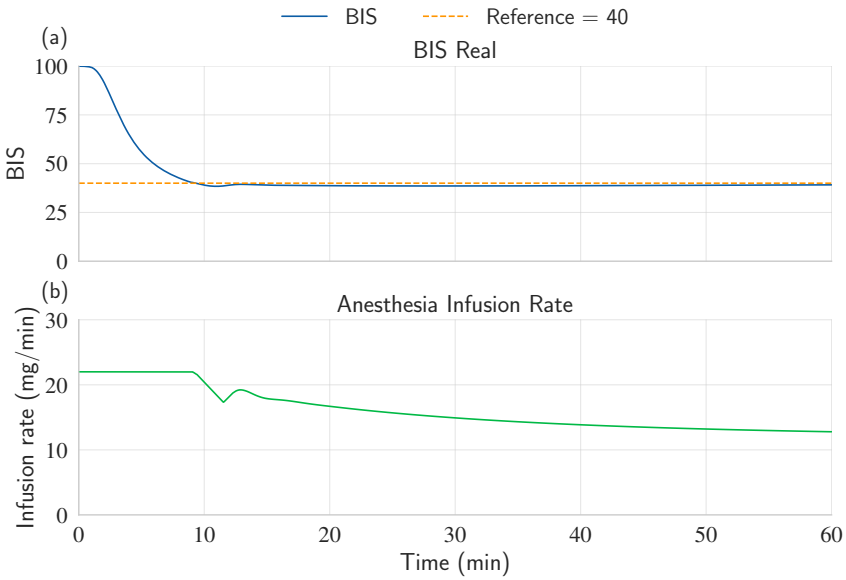


Figure 13: Closed-loop simulation of the PID-based controller for the reference change to 40 experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 14 shows the closed-loop simulation of the SVM-based controller for the experiment in which the reference signal was set to 60. The BIS response exhibits a brief initial undershoot but then stabilizes smoothly around the reference value.

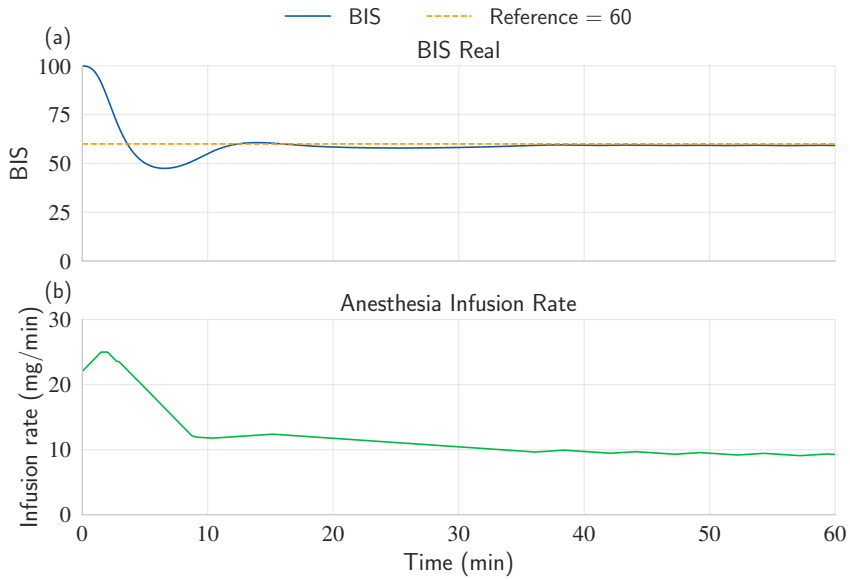


Figure 14: Closed-loop simulation of the SVM-based controller for the reference change to 60 experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 15 shows the closed-loop simulation of the PID-based controller under the same conditions. The BIS response exhibits more pronounced oscillations compared to the SVM controller. These fluctuations are driven by instability in the infusion rate, which is also less desirable from a clinical standpoint.

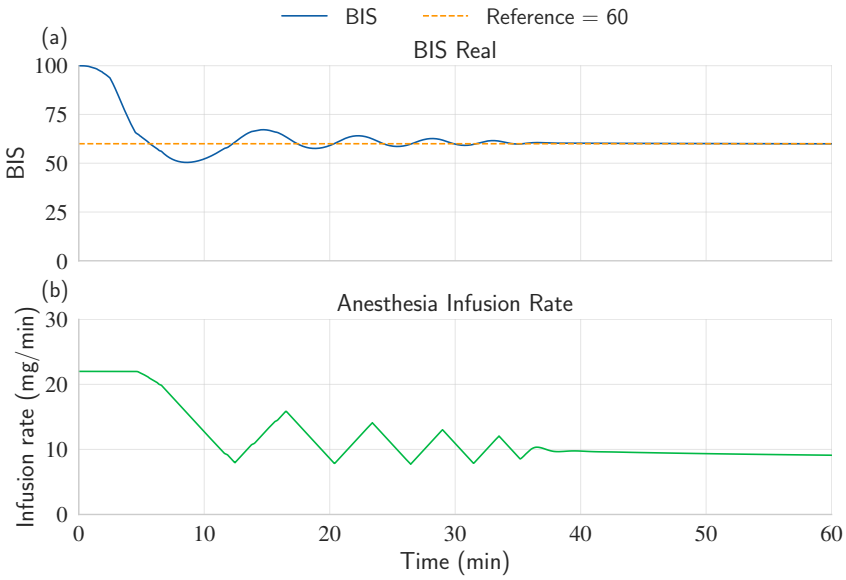


Figure 15: Closed-loop simulation of the PID-based controller for the reference change to 60 experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 16 shows the closed-loop simulation of the SVM-based controller for the experiment in which the reference signal follows a triangular waveform with a 20-minute period. The BIS response partially tracks the reference, but notable deviations occur in certain intervals, indicating limited adaptability to dynamic reference changes.

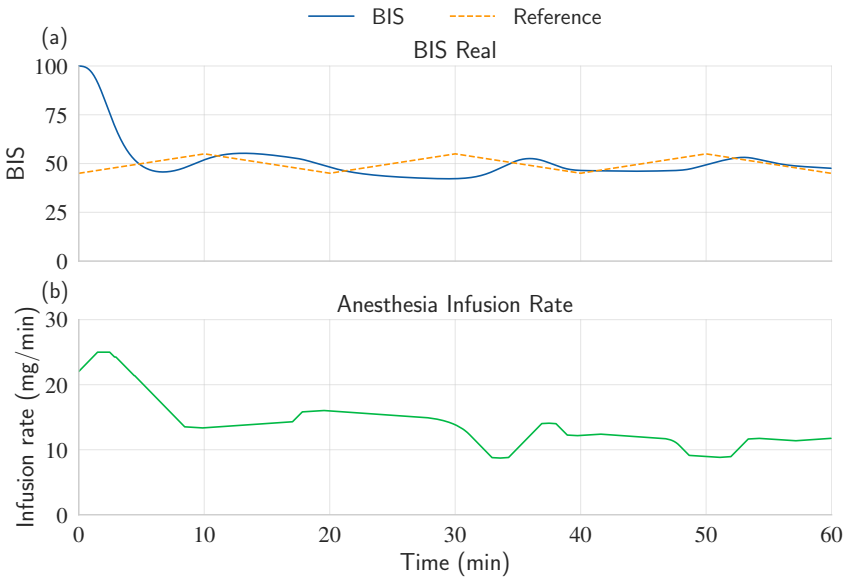


Figure 16: Closed-loop simulation of the SVM-based controller for the reference change to a triangular wave form experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 17 shows the closed-loop simulation of the PID-based controller under the same triangular reference condition. The BIS response aligns more closely with the reference compared to the SVM controller. However, this comes at the cost of a more abrupt and oscillatory infusion profile, which is clinically undesirable.

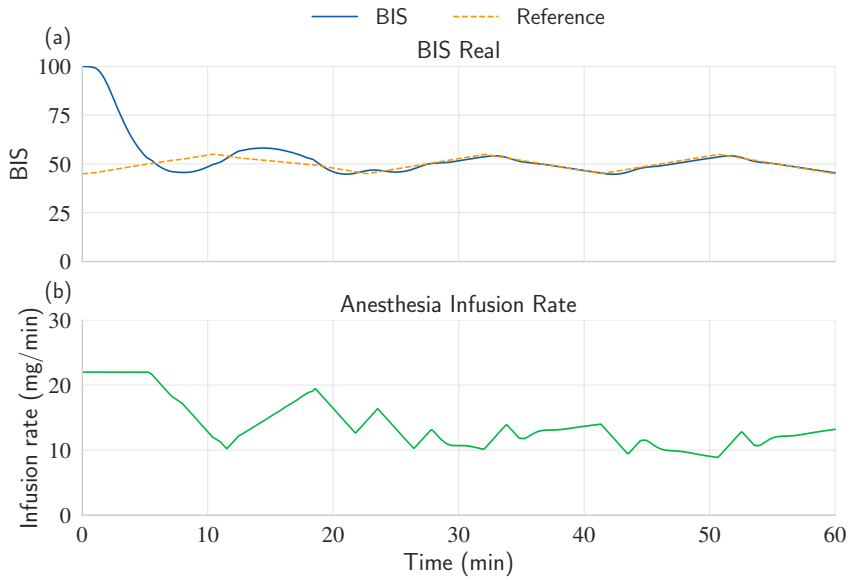


Figure 17: Closed-loop simulation of the PID-based controller for the reference change to a triangular wave form experiment. Top: BIS response. Bottom: Propofol infusion rate.

Gaussian noise

Additive Gaussian noise with zero mean and unit variance was introduced into the feedback loop to imitate a real-life scenario of a noisy feedback from the patient.

Figure 18 shows the closed-loop simulation of the SVM-based controller for the experiment in which Gaussian noise was introduced into the feedback loop. The BIS response remains very smooth and rapidly converges to the reference in a steady manner. The anesthesia infusion profile is also smooth and appears unaffected by the noise, demonstrating strong robustness to disturbances in the plant.

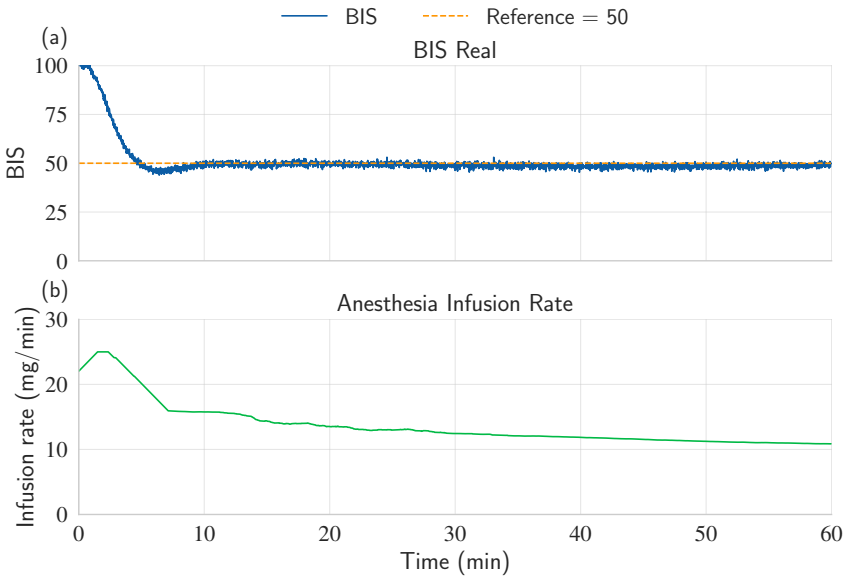


Figure 18: Closed-loop simulation of the SVM-based controller for the gaussian noise in the feedback loop experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 19 shows the closed-loop simulation of the PID-based controller under the same noise conditions. The BIS response is noticeably slower and exhibits more oscillations compared to the SVM controller. The infusion rate profile also shows significant fluctuations, contributing to a longer settling time in the BIS response.

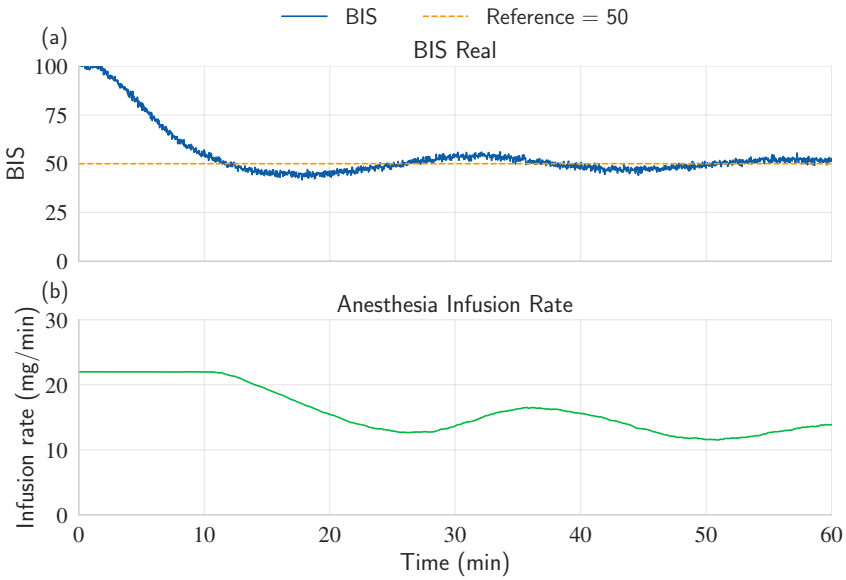


Figure 19: Closed-loop simulation of the PID-based controller for the gaussian noise in the feedback loop experiment. Top: BIS response. Bottom: Propofol infusion rate.

Infusion loss

The infusion dose was set to zero for 20 seconds at the 30-minute mark—once both controllers had reached steady state—simulating a temporary failure in drug delivery.

Figure 20 shows the closed-loop simulation of the SVM-based controller for the experiment in which an infusion loss was introduced. The BIS response exhibits minimal oscillation in reaction to the disturbance and quickly returns to the reference. The infusion profile adjusts smoothly, indicating robust performance in the presence of unexpected input losses.

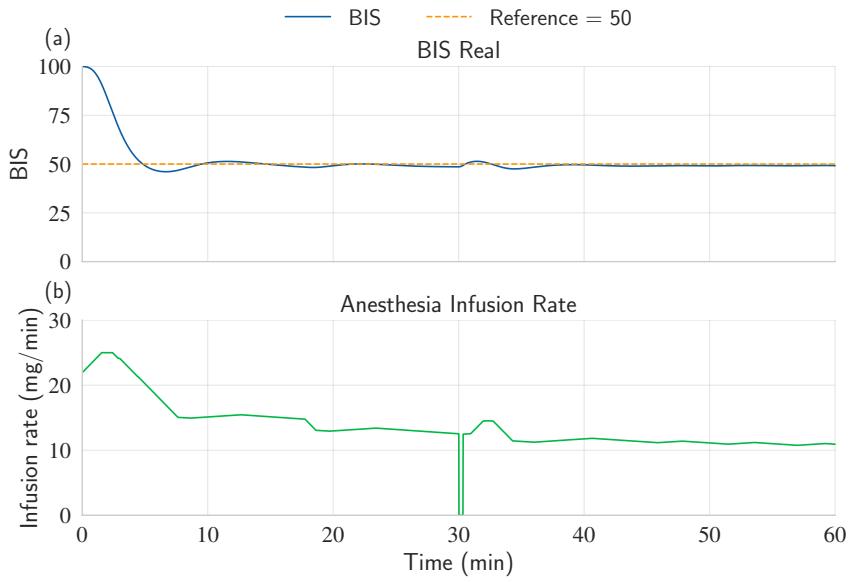


Figure 20: Closed-loop simulation of the SVM-based controller for the infusion loss experiment. Top: BIS response. Bottom: Propofol infusion rate.

Figure 21 shows the closed-loop simulation of the PID-based controller under the same infusion loss conditions. The BIS response displays more noticeable oscillations compared to the SVM controller. The infusion rate reacts abruptly to the disturbance, suggesting a less stable and less desirable response to this type of scenario.

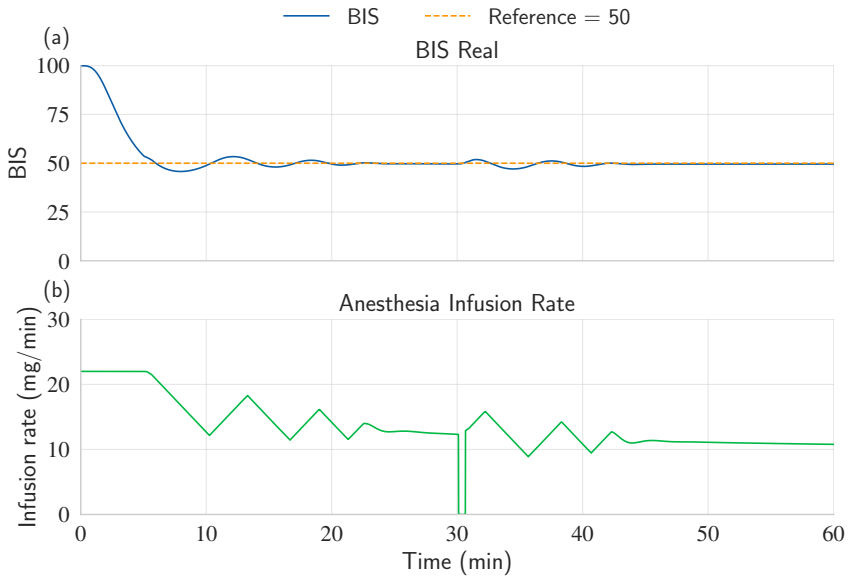


Figure 21: Closed-loop simulation of the PID-based controller for the infusion loss experiment. Top: BIS response. Bottom: Propofol infusion rate.

In order to quantify the performance of these control simulations and experiments, the following metrics were used for comparison:

- **Undershoot:** The percentage of initial overshoot or undershoot relative to the reference value. A lower undershoot indicates a smoother and more stable convergence to the target BIS level.
- **Settling time:** The time required for the BIS to enter and remain within a predefined margin around the reference value. In addition to the conventional $\pm 5\%$ band, a $\pm 20\%$ margin was also evaluated—corresponding to a BIS range of 40–60, which is considered clinically desirable.
- **Error metrics:** Mean Squared Error (MSE), Integral Absolute Error (IAE), and Integral Squared Error (ISE), which collectively reflect the overall deviation of the BIS from the reference over time.

The results of these experiments, compared to the default simulation (reference set to 50), are summarized in Figure 22. In the standard scenario, the SVM-based architecture slightly outperforms the PID controller across all evaluated metrics.

In the Gaussian noise experiment, both controllers exhibit similar behavior, although the SVM-based controller achieves a slightly lower undershoot. In the infusion loss experiment, the SVM controller recovers more quickly, reaching the $\pm 5\%$ steady state faster than the PID controller.

In the reference change experiments:

- For the reference set to 40, the SVM controller reaches the clinically acceptable range more quickly, though it takes longer to converge to the $\pm 5\%$ band.
- For the reference set to 60, it takes the SVM controller slightly longer (over 2 minutes more than the PID) to reach the $\pm 20\%$ margin. However, it converges to the $\pm 5\%$ band nearly at the same time, indicating a smoother stabilization.
- In the triangular reference experiment, both controllers perform comparably, but the SVM-based controller shows lower error values (MSE and ISE).

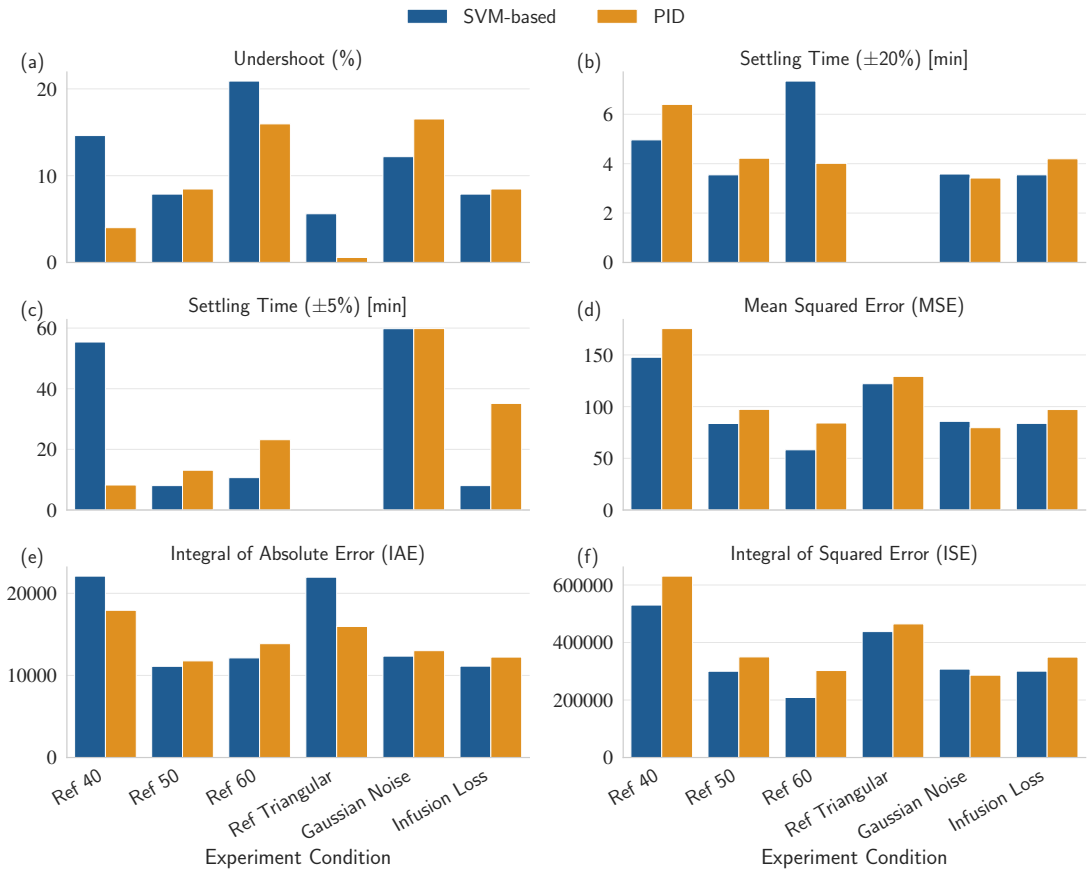


Figure 22: Comparison of performance metrics for all control experiments across both architectures.

10 Discussion

The results of this study demonstrate that both control architectures—SVM-based gradient predictive control and traditional PID—are capable of regulating BIS levels effectively within a simulated anesthesia environment. However, several performance differences emerged that are worth discussing in depth.

Overall, the SVM-based controller exhibited smoother control behavior, particularly visible in the infusion rate profile (see Figure 10), which lacked abrupt transitions and may offer advantages in clinical scenarios where gradual dosing is preferred. In contrast, the PID controller (Figure 11) achieved comparable BIS tracking performance but produced more abrupt control signals, which could translate to less predictable pharmacodynamic responses in a real patient.

Quantitatively, the SVM controller outperformed the PID controller across all performance metrics in the default simulation scenario (Figure 22). These metrics include lower error values (MSE, IAE, ISE), shorter settling times, and reduced undershoot. This suggests that the machine-learning-based architecture can offer improved tracking accuracy and stability under standard conditions.

The perturbation experiments further highlight the robustness of the SVM controller. In the infusion loss experiment, the SVM controller regained steady-state control more rapidly than the PID controller, suggesting enhanced resilience to sudden disturbances. Similarly, in the presence of Gaussian noise, both controllers showed comparable BIS tracking, but the SVM exhibited a slightly lower undershoot, indicating more stable behavior in noisy conditions.

Interestingly, under dynamic reference changes, the controllers exhibited complementary strengths. For instance, the PID controller adapted faster when the reference increased to 60, but the SVM controller converged more smoothly, as evidenced by the minimal difference between its $\pm 20\%$ and $\pm 5\%$ settling times. This implies that the SVM-based method may be

better suited to scenarios that require smooth transitions rather than speed alone.

Additionally, it should be noted that the PID controller is already operating near the limits of its simplicity and tuning potential, whereas the SVM-based control architecture offers significant room for further improvement. With enhanced training, expanded datasets, and other architecture improvements, the SVM controller could potentially outperform the PID controller by a larger margin. Furthermore, the SVM model can be trained to adapt to inter-patient variability by incorporating physiological parameters, enabling the predictive control system to generalize more effectively across different patient groups. This adaptability is crucial for the development of a robust, real-world automated anesthesia infusion system.

Finally, although both controllers were tested under several realistic perturbations, further validation under more complex multi-drug scenarios or with physiological feedback loops would strengthen the case for clinical translation.

11

Conclusions and Futures Lines of Research

11.1 Conclusions

This work has presented the design, implementation, and evaluation of a closed-loop control architecture for propofol-based anesthesia using a SVM integrated into a GPC framework. The proposed approach was compared against a traditional PID-based control system under identical simulation conditions using a three-compartment PK/PD model (Schnider model).

Simulation results indicate that the SVM-based GPC controller achieves comparable or better performance than the PID controller across several metrics, including undershoot, settling time, and total control error. The SVM controller produced smoother control actions with reduced abrupt changes, which are more desirable from a clinical perspective. Moreover, robustness tests under various perturbations (e.g., reference variation, measurement noise, and infusion loss) showed that the SVM-based controller maintains stability and good performance under challenging conditions.

Overall, the results support the feasibility and potential advantages of using data-driven models such as SVMs in predictive control for automated anesthesia delivery systems.

11.2 Future lines of Research

While the current implementation demonstrates the promise of SVM-based predictive control, several directions can be explored to enhance and extend this work:

- **Multi-patient generalization:** The current controller is trained and validated on a

single-patient model. Future work should incorporate inter-patient variability, training the model with datasets representing different physiological conditions (e.g., age, weight, height) to improve robustness and generalizability.

- **Multi-step prediction:** Extending the prediction horizon could improve the controller's performance by enhancing its ability to anticipate changes in patient dynamics. This would more fully exploit the advantages of predictive control, particularly in complex and nonlinear systems such as automated anesthesia regulation.

- **Model optimization and architecture scaling:** More complex machine learning models, such as deep neural networks or ensemble methods, could be integrated into the control architecture to improve predictive accuracy and control performance.

- **Real-time implementation:** The control system could be adapted for deployment in embedded hardware or real-time simulation platforms to test its feasibility in clinical environments.

- **Adaptive or online learning:** Incorporating adaptive control techniques or online retraining mechanisms could allow the controller to adjust in real time to unanticipated changes in patient dynamics.

- **Clinical validation:** A critical next step would be validating the system with real patient data or in hardware-in-the-loop simulations to assess performance and safety under realistic conditions.

By addressing these aspects, the proposed architecture could move closer to practical application in automated anesthesia delivery systems.

12

Conclusiones y Líneas Futuras

12.1 Conclusiones

Este trabajo ha presentado el diseño, implementación y evaluación de una arquitectura de control en lazo cerrado para la administración de anestesia basada en propofol, utilizando una Máquina de Vectores de Soporte (SVM) integrada en un esquema de Control Predictivo Generalizado (GPC). El enfoque propuesto se comparó con un sistema de control tradicional basado en PID bajo condiciones de simulación idénticas, empleando un modelo farmacocinético/farmacodinámico (PK/PD) de tres compartimentos (modelo de Schnider).

Los resultados de las simulaciones indican que el controlador GPC basado en SVM alcanza un rendimiento comparable o superior al del controlador PID en varias métricas, incluyendo sobreimpulso, tiempo de establecimiento y error total de control. El controlador basado en SVM generó señales de control más suaves, con menos cambios abruptos, lo cual es más deseable desde una perspectiva clínica. Además, las pruebas de robustez frente a distintas perturbaciones (como variaciones en la referencia, ruido de medición y pérdida de infusión) mostraron que el controlador basado en SVM mantiene la estabilidad y un buen rendimiento en condiciones desafiantes.

En general, los resultados respaldan la viabilidad y las ventajas potenciales del uso de modelos basados en datos, como las SVM, dentro del control predictivo para sistemas automatizados de administración de anestesia.

12.2 Líneas Futuras

Si bien la implementación actual demuestra el potencial del control predictivo basado en SVM, existen varias direcciones que podrían explorarse para mejorar y ampliar este trabajo:

- **Generalización multi-paciente:** El controlador actual ha sido entrenado y validado con un modelo de un solo paciente. Trabajos futuros deberían incorporar variabilidad entre pacientes, entrenando el modelo con conjuntos de datos que representen distintas condiciones fisiológicas (por ejemplo, edad, peso, altura) para mejorar la robustez y la capacidad de generalización.
- **Predicción a múltiples pasos:** Ampliar el horizonte de predicción podría mejorar el rendimiento del controlador al aumentar su capacidad para anticipar cambios en la dinámica del paciente. Esto permitiría explotar mejor las ventajas del control predictivo, especialmente en sistemas complejos y no lineales como la regulación automatizada de la anestesia.
- **Optimización del modelo y escalado de la arquitectura:** Se podrían integrar modelos de aprendizaje automático más complejos, como redes neuronales profundas o métodos de conjunto (ensemble), para mejorar la precisión de predicción y el rendimiento del control.
- **Implementación en tiempo real:** El sistema de control podría adaptarse para su implementación en hardware embebido o plataformas de simulación en tiempo real, con el fin de evaluar su viabilidad en entornos clínicos.
- **Aprendizaje adaptativo o en línea:** La incorporación de técnicas de control adaptativo o mecanismos de reentrenamiento en línea permitiría que el controlador se ajuste en tiempo real ante cambios imprevistos en la dinámica del paciente.
- **Validación clínica:** Un paso crítico sería validar el sistema con datos reales de pacientes o mediante simulaciones hardware-in-the-loop, para evaluar su rendimiento y seguridad en condiciones realistas.

Abordando estos aspectos, la arquitectura propuesta podría avanzar hacia una aplicación práctica en sistemas automatizados de administración de anestesia.

References

[1] X. Cai, X. Wang, Y. Zhu, Y. Yao, and J. Chen, "Advances in automated anesthesia: A comprehensive review," *Anesthesiology and Perioperative Science*, vol. 3, no. 1, pp. 1–20, 2025.

[2] M. Ghita, M. Neckebroek, C. Muresan, and D. Copot, "Closed-loop control of anesthesia: Survey on actual trends, challenges and perspectives," *Ieee Access*, vol. 8, pp. 206 264–206 279, 2020.

[3] L. Shi, X. Li, and H. Wan, "A predictive model of anesthesia depth based on svm in the primary visual cortex," *The open biomedical engineering journal*, vol. 7, p. 71, 2013.

[4] A. T. Namel and M. A. Sahib, "Data-driven based bispectral index prediction during general anesthesia.," *Journal Européen des Systèmes Automatisés*, vol. 57, no. 4, 2024.

[5] C. Castellanos Peñaranda, F. D. Casas Arroyave, F. J. Gómez, *et al.*, "Technical and clinical evaluation of a closed loop tiva system with sedline tm spectral density monitoring: Multicentric prospective cohort study," *Perioperative Medicine*, vol. 9, pp. 1–11, 2020.

[6] A. Simalatsar, M. Guidi, and T. Buclin, "Cascaded pid controller for anaesthesia delivery," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2016, pp. 533–536.

[7] T. W. Schnider, C. F. Minto, P. Cambus, *et al.*, "The influence of method of administration and covariates on the pharmacokinetics of propofol in adult volunteers," *ANESTHESIOLOGY-PHILADELPHIA THEN HAGERSTOWN-*, vol. 88, pp. 1170–1182, 1998.

[8] A. Absalom, V. Mani, T. De Smet, and M. Struys, "Pharmacokinetic models for propofol—defining and illuminating the devil in the detail," *British journal of anaesthesia*, vol. 103, no. 1, pp. 26–37, 2009.

[9] S. Iplikci, "A support vector machine based control application to the experimental three-tank system," *ISA transactions*, vol. 49, no. 3, pp. 376–386, 2010.

[10] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, pp. 199–222, 2004.

[11] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. ISA—The Instrumentation, Systems, and Automation Society, 1995.

[12] A. Absalom and M. M. Struys, "An overview of tci & tiva," 2019.

[13] S. Hosseinirad, K. van Heusden, and G. A. Dumont, "Evaluating inter-individual variability captured by the eleveld pharmacokinetics model," *Journal of Clinical Monitoring and Computing*, vol. 38, no. 2, pp. 505–518, 2024.

[14] M. M. Sahinovic, M. M. Struys, and A. R. Absalom, "Clinical pharmacokinetics and pharmacodynamics of propofol," *Clinical pharmacokinetics*, vol. 57, no. 12, pp. 1539–1558, 2018.

[15] F. Linassi, P. Zanatta, L. Spano, P. Burelli, A. Farnia, and M. Carron, "Schnider and eleveld models for propofol target-controlled infusion anesthesia: A clinical comparison," *Life*, vol. 13, no. 10, p. 2065, 2023.

[16] I. The MathWorks, *Matlab version r2024b*, https://www.mathworks.com/products/matlab.html, The MathWorks, Inc., Natick, Massachusetts, 2024.

[17] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

Appendix A
Code Repository

The source code for this project, including system modeling scripts, SVM training, controller design, Simulink models, all experiments, and data visualization scripts is available on GitHub at the following URL:

https://github.com/GonzaloM786/SVM-based-closed-loop-anesthesia-control-system

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA