Anesthesia Delivery Control

EEME 6602: Modern Control Theory

Patrick Munar

Final Report

Electrical Engineering
Columbia University
New York, USA
May 8 2023

Contents

1 Introduction **3**

2 Literature Review **4**
- 2.1 Bispectral Index . . . 4
- 2.2 Compartment Models . . . 4
- 2.3 Parsimonious Wiener Model . . . 5
- 2.4 Generalized Hill Equation . . . 5

3 Problem Statement **6**
- 3.1 Anesthesia System Development . . . 6
  - 3.1.1 Pharmacokinetics and Pharmacodynamics . . . 6
  - 3.1.2 State-Space Model . . . 7
- 3.2 Proposed Control Strategy . . . 9

4 Results **10**
- 4.1 Model Predictive Control . . . 10
  - 4.1.1 Static BIS Progression . . . 10
  - 4.1.2 Dynamic BIS Progression . . . 12

5 Future Work **15**

6 Conclusion **15**

7 References **16**

A Model Predictive Control **17**
- A.1 Overview . . . 17
- A.2 Derivation . . . 18
  - A.2.1 Continuous-Time to Discrete-Time . . . 18
  - A.2.2 Augmentation . . . 19
- A.3 Optimization . . . 23
  - A.3.1 Cost Function . . . 23
- A.4 Constraints . . . 23
  - A.4.1 Input Amplitude . . . 24
  - A.4.2 Input Rate of Change . . . 26
  - A.4.3 Combining Input Constraints . . . 28
  - A.4.4 Input Change Equality . . . 28
- A.5 Realization . . . 29
  - A.5.1 Modified Cost Function . . . 29
  - A.5.2 Gradient Descent Algorithm . . . 30
  - A.5.3 MATLAB . . . 31

1 Introduction

Advancements in the fields of engineering and mathematics have significantly contributed to the medical industry. Integrating modern mathematical and analytical techniques opens opportunities for new surgical, diagnostic, and pharmaceutical solutions in healthcare. The proposed project addresses the latter, specifically anesthesia distribution and control, and is an interdisciplinary work of biochemistry and control systems engineering.

Many, if not all, surgical procedures are going to be invasive and will put a tremendous amount of strain on the body of the patient. It goes without saying that providing the proper amount of general anesthesia is paramount to the success of any surgical procedure. Administering anesthetics to a surgical patient can be done either through inhalation or intraveneous fluids directly entering the bloodstream. Both approaches will lead to similar outcomes where the combination of the specific anesthetic drugs will block the pain receptors in the peripheral nervous system from sending electrical signals to the central nervous system. If these electrical signals are not registered by the central nervous system, this keeps the patient from experiencing any discomfort and pain during the surgery. Providing general anesthesia ensures that the patient's vitals are kept stable throughout the surgery and that the amount of trauma and stress the body endures is limited. It is for the previously mentioned reasons that general anesthesia can be seen as the foundation to any surgical procedure.

While general anesthesia can be administered as an inhaled drug, the proposed project primarily considers the use of intraveneous fluids as the means for delivering the proposed anesthetics of propofol and remifentanil. Much of the existing literature examines and models anesthesia delivery with these drugs, and this project aims to build upon that work. However, even when narrowing the scope to one type of general anesthesia technique, the task of maintaining the proper level of sedation throughout a given surgical procedure is still quite complex. A variety of factors need to be considered, most of which revolve around the patient. These would include age, gender, weight, height, as well as their accompanying medical history. However, as mentioned earlier, there has been much research in the area of biochemistry and systems engineering that makes it possible to model the human body and its reaction to anesthetic drugs. Being able to model a system for anesthesia distribution does not mean the current results match the performance of being able to properly match a desired sedation level during a surgical procedure.

Depending on the individual, as well as the type of procedure and amount of anesthetic used, the required sedation level of a patient is very hard to match with what the actual reading may be. There is also the concern for administering too much or too little of either anesthetic. Being on either side of these extremes may result in further health complications for the patient as well as a poorly managed surgical procedure. The proposed project discusses and elaborates on the aforementioned goal and how the use of optimization and modern control techniques can synthesize a solution that improves the ability of anesthesiologists to safely control and maintain the sedation of level of a patient.

2 Literature Review

Anesthesia control, and more broadly pharmacokinetics and phramacodynamics, is a vast interdiscplinary field of study. With advances in the areas of mathematics, medicine, biochemistry, and pharmacology, we have been able to model the reaction that the human body has to drugs used for general anesthesia. Much of the recent literature used to inform my understanding of this type of system comes from studies as recent as the past decade. This section looks to summarize those findings and how they contribute to informing the system model that I would like to control.

2.1 Bispectral Index

The bispsectral index, or BIS, is a metric used in anesthesiology to quantify the consciousness of a patient. During a medical procedure, electrodes are placed on the forehead of a patient and return the brain activity through electroencephalogram, or EEG, readings. These are used to calculate and monitor the BIS of a patient in real time. The bispectral index is a dimensionless metric and ranges between 0-100. Readings that are between 90-100 are where people are at full consciousness. Readings that are between 70-90 denote a state of moderate sedation and even some levels of deep sleep. For both of these ranges, the central and peripheral nervous systems of the brain are still able to detect and report pain or discomfort. However, anything between 30-60 is defined as the appropriate BIS range for general anesthesia. This region indicates a deep hypnotic state where pain receptors in the peripheral nervous systems do not relay these electrical signals back to the central nervous system. Anything below a BIS of 30 is extremely low electrical activity and close to flatlining.

2.2 Compartment Models

Pharmacokinetic/Pharmacodynamics, or PK/PD, compartment models are the basis for describing and modelling how drugs are absorbed, metabolized, and distributed throughout the body. These types of models, as the name implies, breaks up the body into a series of compartments. These compartments can represent almost anything in the human body from tissues, muscle, and organs. These individual compartments will interact dynamically with one another to create a relation for the drugs behavior in the body. The interaction between the compartments is usually related through drug flow rates, concentration, and dissipation. However, some compartments have a stronger affect for certain drugs, but that is obviously dependent on the drug being analyzed. Adding more compartments to a PK model will increase the complexity and accuracy of the drug behavior.

With respect to modelling anesthesia control, the primary compartments that need to be considered are parts of the body rich and deficient in blood vessels. Areas of the body that are rich in blood vessels include the brain and muscles. Consequently, areas of the body that are deficient in blood vessels are fatty tissues and smaller organs. A three compartment model that incorporates the previously mentioned categories is displayed in Figure 1.

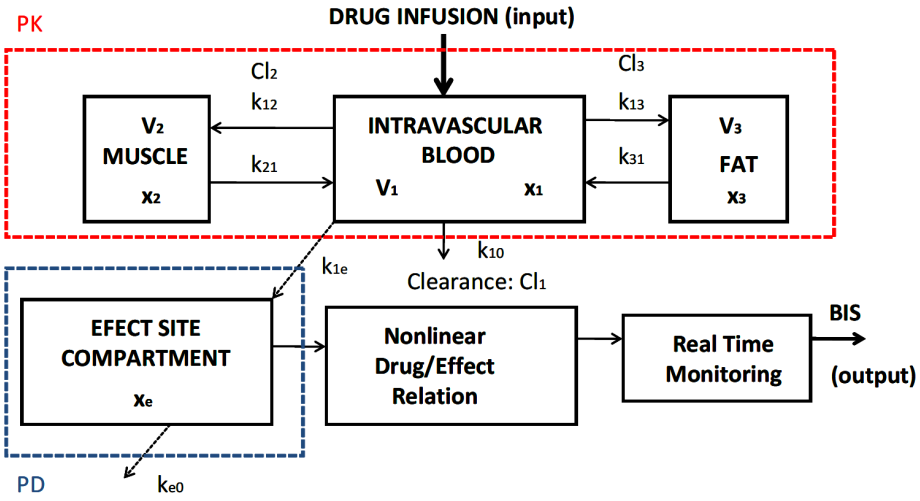

Figure 1: PK/PD Compartment Model

2.3 Parsimonious Wiener Model

A parsimonious Wiener model is used in conjunction with the previously compartment model to fully outline the relationship between the infusion of propofol and remifentanil, and the bispectral index of the patient. The parsimonious Wiener model is a a type of mathematical model used to describe input and output relationships of signals. The model contains linear system dynamics which feed into a nonlinear operator that produces the entire system output. This is mainly different from the classical Wiener model which contains a nonlinear function as a part of the plant. Additionally, the traditional Wiener model is MIMO whereas the parsimonious Wiener model is SISO. Finally, the parsimonious Wiener model has specific applications to pharmacokinetics/pharmacodynamics, thus making it a favorable tool for modelling these types of systems.

For anesthesia control, the parsimonious Wiener model incorporates the linear system dynamics of the PK/PD compartment models as well as the combined nonlinear effect that propofol and remifentanil have on the bispectral index of the patient. A general block diagram of the parsimonious Wiener model used for anesthesia delivery is displayed in Figure 2.

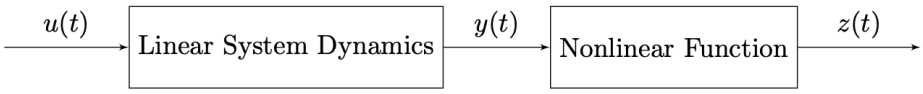

Figure 2: Parsimonious Wiener Model

2.4 Generalized Hill Equation

The final concept acquired from my literature review of anesthesia control is the use of the generalized Hill equation to model nonlinear effects that intravenous anesthetics have on the body.

The generalized Hill equation for anesthesia relates the combined potency of a given drug to the BIS of a patient. It requires the base BIS that a patient would normally be at with no sedation, as well as the current potency of the anesthetics present in the bloodstream. The form of the function makes it nonlinear since it does not abide by superposition or homogeneity. This will be further elaborated on when discussing the system model.

3 Problem Statement

Now that we have discussed the related literature to anesthesia control, we are able to formalize the problem that we would like to address in this project. Being able to properly manipulate the sedation level and hypnotic state of a patient would help medical procedures and surgeries operate more smoothly. Therefore, we would like to create and test a control strategy that would improve the current anesthesia system's ability to track a desired BIS for different patients and procedures. The control strategy should be robust to noise and disturbances as either of these present in the system would be disruptive to any medical procedure, and be detrimental to patient health. As such, the proposed control strategy is to implement model predictive control as its architecture is tailored for tracking desired outputs, even in the presence of disturbances and noise.

3.1 Anesthesia System Development

As seen in Section 2, and its associated subsections, the overall process of relating the infusion of propofol and remifentanil to the patient BIS is complex. The following derivations aim to show the synthesis of this anesthesia delivery system using the previously outlined concepts

3.1.1 Pharmacokinetics and Pharmacodynamics

A linear system model can be developed from transfer functions that relate the infusion rates of propofol and remifentanil to their relative concentrations. These transfer functions provide the linear system dynamics to the system and show the individual effect of each anesthetic where α and η are patient specific parameters determined through clinical trials.

$$\begin{aligned} c_p(s) &= \frac{90\alpha^3}{(10\alpha + s)(9\alpha + s)(\alpha + s)} u_p(s) \\ c_r(s) &= \frac{6\eta^3}{(3\eta + s)(2\eta + s)(\eta + s)} u_r(s) \end{aligned} \tag{1}$$

Knowing the concentration of each drug still does not provide enough information to determine the effect on the BIS. As such, the concentrations of propofol and remifentanil need to be normalized. This converts the concentrations of these drugs into their relative potencies.

$$\begin{aligned} y_p(t) &= 0.1 c_p(t) \\ y_r(t) &= 100 c_r(t) \end{aligned} \tag{2}$$

The individual potencies of propofol and remifentanil can be superimposed to determine the overall potency of the two anesthetic drugs. Essentially the above expressions will be summed together where the potency for propofol is scaled by μ, another patient specific parameter.

$$\begin{aligned} y(t) &= \mu y_p(t) + y_r(t) \\ &= 0.1\mu c_p(t) + 100 c_r(t) \end{aligned} \tag{3}$$

The combined potency is then finally inputted into the nonlinear, generalized Hill equation. This nonlinear function relates the combined potency to the BIS of the patient. Again, a patient specific parameter γ is used to improve simulation accuracy.

$$z(t) = \frac{z_0}{1 + y(t)^\gamma} \tag{4}$$

It is important to note that these PK/PD expressions will have varying effects on the overall system. This is primarily due to the inclusion of the set of patient specific parameters $\alpha, \eta, \gamma, \mu$. While not fully apparent until the final result, we see that these PK/PD relationships begin to create the basis for our state-space model.

3.1.2 State-Space Model

Now that the pharmacokinetics and pharmacodynamics have been properly discussed, we are able to incorporate them in our synthesis of an anesthesia control system that uses the two drugs propofol and remifentanil. Our main objective is to convert the previously derived expressions into a system of linear equations that can be modelled by (A, B, C, D).

To start with the state matrix A, we primarily consider the transfer functions relating the changes individual changes of the concentrations of propofol and remifentanil. These transfer functions each have 3 poles, and thus it follows that state matrices for each drug $A_p, A_r \in \mathbb{R}^{3\times 3}$.

$$A_p = \begin{bmatrix} -10\alpha & 0 & 0 \\ \alpha & -9\alpha & 0 \\ 0 & \alpha & -\alpha \end{bmatrix} \quad , \quad A_r = \begin{bmatrix} -3\eta & 0 & 0 \\ 2\eta & -2\eta & 0 \\ 0 & \eta & -\eta \end{bmatrix} \tag{5}$$

The input matrix B also stems from the transfer functions for propofol and remifentanil and takes into account the associated infusion rate of each drug. It follows that the input matrices for each drug $B_p, B_r \in \mathbb{R}^3$.

$$B_p = \begin{bmatrix} 10\alpha \\ 0 \\ 0 \end{bmatrix} \quad , \quad B_r = \begin{bmatrix} 3\eta \\ 0 \\ 0 \end{bmatrix} \tag{6}$$

The output matrix C takes into account the relationship between the concentrations of propofol and remifentanil with respect to their individual and combined potencies. As expected, the output matrices for each drug $C_p^T, C_r^T \in \mathbb{R}^3$.

$$C_p = \begin{bmatrix} 0 & 0 & 0.1\mu \end{bmatrix} \quad , \quad C_r = \begin{bmatrix} 0 & 0 & 100 \end{bmatrix} \tag{7}$$

This system does not include a D matrix, and therefore we are unable to properly model disturbances that directly affect the output of the system.

Each anesthetic also had associated with it state vectors $x_p(t), x_r(t) \in \mathbb{R}^3$ and inputs $u_p(t), u_r(t) \in \mathbb{R}$. The states for each anesthetic corresponded to the changes in the concentration of each anesthetic and the inputs corresponded to the infusion rates at which each drug was administered to the patient.

$$x_p(t) = \begin{bmatrix} x_{p1}(t) \\ x_{p2}(t) \\ x_{p3}(t) \end{bmatrix} \quad , \quad x_r(t) = \begin{bmatrix} x_{r1}(t) \\ x_{r2}(t) \\ x_{r3}(t) \end{bmatrix} \tag{8}$$

The previously defined matrices, states, and inputs only model the individual effects of propofol and remifentanil. In other words, so far we have only defined two smaller systems (A_p, B_p, C_p) and (A_r, B_r, C_r) which are independent of one another. Concatenating these systems requires the inclusion of the appropriate zero matrices O to create compatible system sizing. As such, the new sizing for the entire anesthesia delivery system has $A \in \mathbb{R}^{6\times6}$,$B \in \mathbb{R}^{6\times2}$, and $C^T \in \mathbb{R}^6$.

$$A = \begin{bmatrix} A_p & O_{3\times3} \\ O_{3\times3} & A_r \end{bmatrix} \quad , \quad B = \begin{bmatrix} B_p & O_{3\times1} \\ O_{3\times1} & B_r \end{bmatrix} \quad , \quad C = \begin{bmatrix} C_p & C_r \end{bmatrix} \tag{9}$$

We are able to concatenate these smaller systems with one another to form a larger system (A, B, C) which incorporates both effects of propofol and remifentanil. This provides a comprehensive system model of the effects that these drugs have on the BIS of a patient.

Combining the state, input, output matrices of propofol and remifentanil also results in combined state, input, and output vectors. Based on the structure of concatenating the state and input matrices, the effects of these drugs in the state remain independent of one another. However, the structure of the output matrix imposes the effects of both anesthetics together.

$$x(t) = \begin{bmatrix} x_p(t) \\ x_r(t) \end{bmatrix} \quad , \quad u(t) = \begin{bmatrix} u_p(t) \\ u_r(t) \end{bmatrix} \tag{10}$$

These new state and input vectors help fully relate the concentrations and flow rates of propofol and remifentanil. Based on the sizing of these new matrices, it is not difficult to derive that $x(t) \in \mathbb{R}^6$ and $u(t) \in \mathbb{R}^2$.

Finally, after all the appropriate derivations and review of PK/PD expressions of the two desired anesthetics, we are able to fully express this anesthesia delivery system in the form of a linear dynamical system.

$$
\begin{aligned}
\dot{x(t)} &= \begin{bmatrix} \dot{x_p}(t) \\ \dot{x_r}(t) \end{bmatrix} = \begin{bmatrix} A_p & O_{3\times 3} \\ O_{3\times 3} & A_r \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_r(t) \end{bmatrix} + \begin{bmatrix} B_p & O_{3\times 1} \\ O_{3\times 1} & B_r \end{bmatrix} \begin{bmatrix} u_p(t) \\ u_r(t) \end{bmatrix} \\
y(t) &= \begin{bmatrix} C_p & C_r \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_r(t) \end{bmatrix} \\
z(t) &= \frac{z_0}{1 + y(t)^\gamma}
\end{aligned} \tag{11}
$$

Where the system matrices (A, B, C) and states take the form of a concatenated version of (A_p, B_p, C_p) and (A_r, B_r, C_r) with patient specific parameters $\alpha, \eta, \gamma, \mu$ included for further specificity.

After having defined the anesthesia delivery system model, we need to examine what are the useful control techniques and strategies that we can apply to improve the base system performance. This will be elaborated in Section 4 which discusses the numerical results when testing this system.

3.2 Proposed Control Strategy

As stated in the problem statement, the primary goal of this project is to design a controller that will assist the current anesthesia delivery system in tracking a reference BIS level. Therefore, the control strategy should be able to determine the optimal input flow rates of propofol and remifentanil that can achieve the desired effect. When stated as an algorithm, a candidate control strategy should perform the following:

1. Take as input a reference BIS level $z_{\text{des}}(t)$.
2. Determine the combined potency $y(t)$ and subsequently the individual potencies of propofol and remifentanil $y_p(t)$ and $y_r(t)$.
3. Determine the assoiciated concentrations of propofol and remifentanil $c_p(t)$ and $c_r(t)$.
4. Calculate from the concentrations the required flow rates of propofol and remifentanil $u_p(t)$ and $u_r(t)$.
5. Optimize the input flow rates to minimize the error at the output $z(t)$.
6. Repeat for the entire duration of the procedure.

Even though in many ways this algorithm is vague and potentially naive, it provides a framework for the expected functionality of the control strategy that we would like to implement for this system. Some bonus features to the control strategy may include having robustness to disturbances or the inclusion of some sort of filter to diminish noise contributions.

Based on the problem statement requires and overall goals of the project, I have determined that model predictive control is an excellent control strategy that can realize the desired closed-loop performance. For a comprehensive derivation of this control strategy, refer to Appendix A.

4 Results

After implementing the previously discussed control strategy in MATLAB, I was able to evaluate the performance of the controller and its ability to track a desired BIS. Before actually applying the proposed control strategy, it was important to first characterize the uncontrolled system. This would mainly inform what needs to be improved and what the control strategy should help accomplish. Evaluating the controlled system was also done in parallel to make direct comparisons and analysis of the closed-loop system performance.

4.1 Model Predictive Control

The primary objective is to create a control law that is capable of effectively tracking a desired BIS while minimizing the amount of anesthetic injected into the patient. Prioritizing these results will ensure that our prescribed control law is safe for the patient and maintains proper sedation throughout a surgical procedure. Model predictive control is one such strategy that can help achieve these objectives.

4.1.1 Static BIS Progression

Before examining the various BIS progression cases, we first applied the model predictive controller to the different step responses for a static BIS. Again, these were iterated for each patient and for the appropriate BIS levels during surgery. This refers to bringing down the patient BIS within the range of 40-60 which has been specified as the desired sedation level for invasive medical procedures. Figures 3-5 display the results of applying model predictive control to each of the patients. Each column of plots shows the static progression of bringing a patient to the desired BIS, as well as the amount of propofol and remifentanil administered to the patient measured in flow rates. The red line in the larger BIS plot denotes the reference signal that we want the patient to track. The solid lines represent the uncontrolled system response and the dotted lines represent the system response using model predictive control. Finally, each column shows the results for tracking a BIS of 60, 50, and 40.

These results provide some insight about how the model predictive controller affects the uncontrolled system. While only a static BIS progression was used, these observations can serve as a baseline for the overall behavior of the implemented control strategy:

- **Undershoot** - A primary observation about the controlled output is the presence of undershoot. This is indiscriminate of the varying patient models and parameters since there is undershoot in every response. The severity of the undershoot could be examined more closely however. It is hard to determine if the magnitude of the undershoot is due to the model for each patient or the configuration of the model predictive controller.
- **Fall Time** - While specific fall time metrics were not recorded, it is clear that the output using model predictive control has less lag than the uncontrolled response. This improvement in fall time not only indicates good system performance, but for a physical interpretation this means that the patient better reaches the desired BIS in an appropriate amount of time. Specifically,

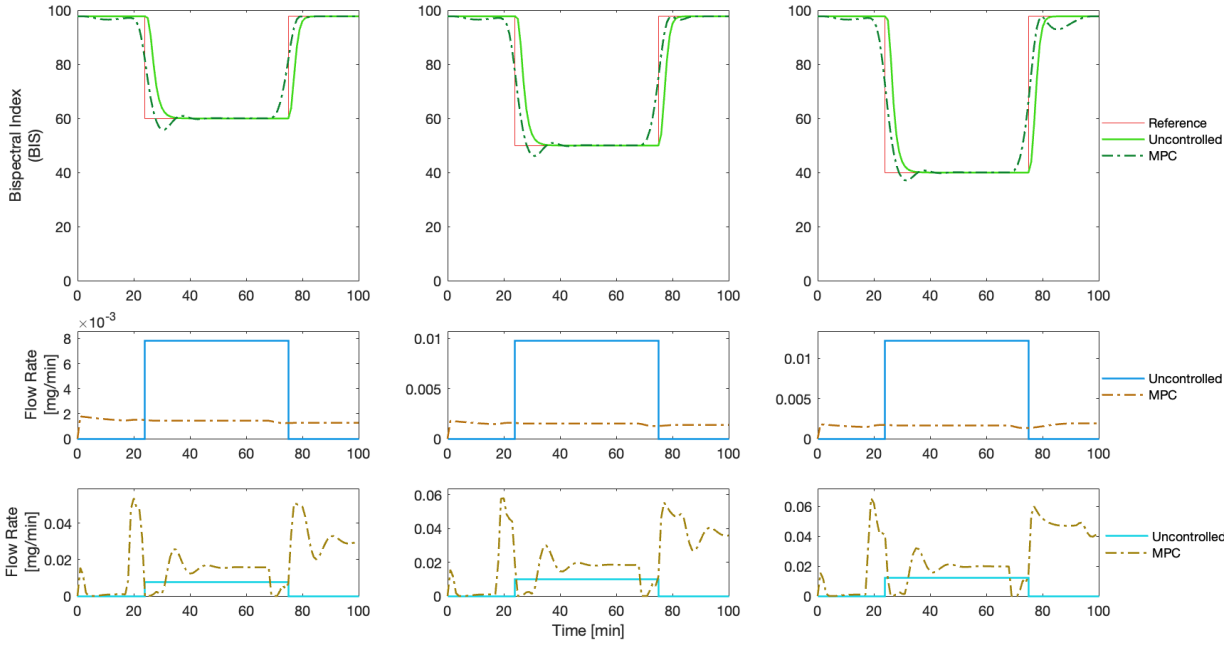

Figure 3: Static BIS Progression - Patient 1

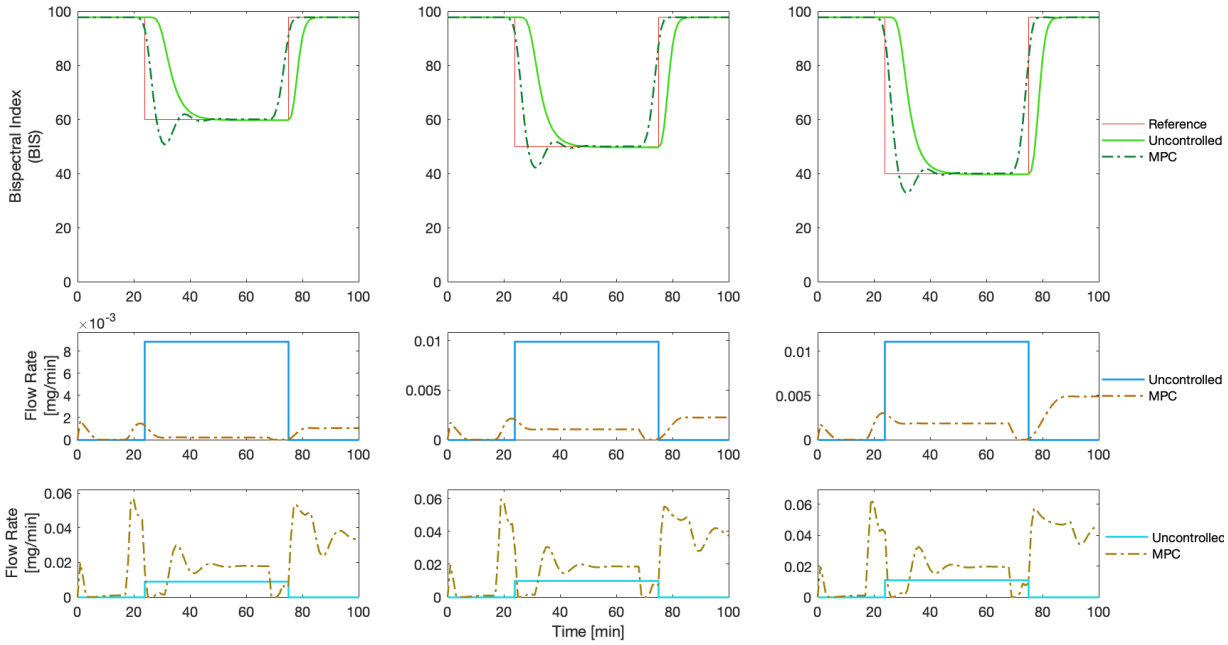

Figure 4: Static BIS Progression - Patient 2

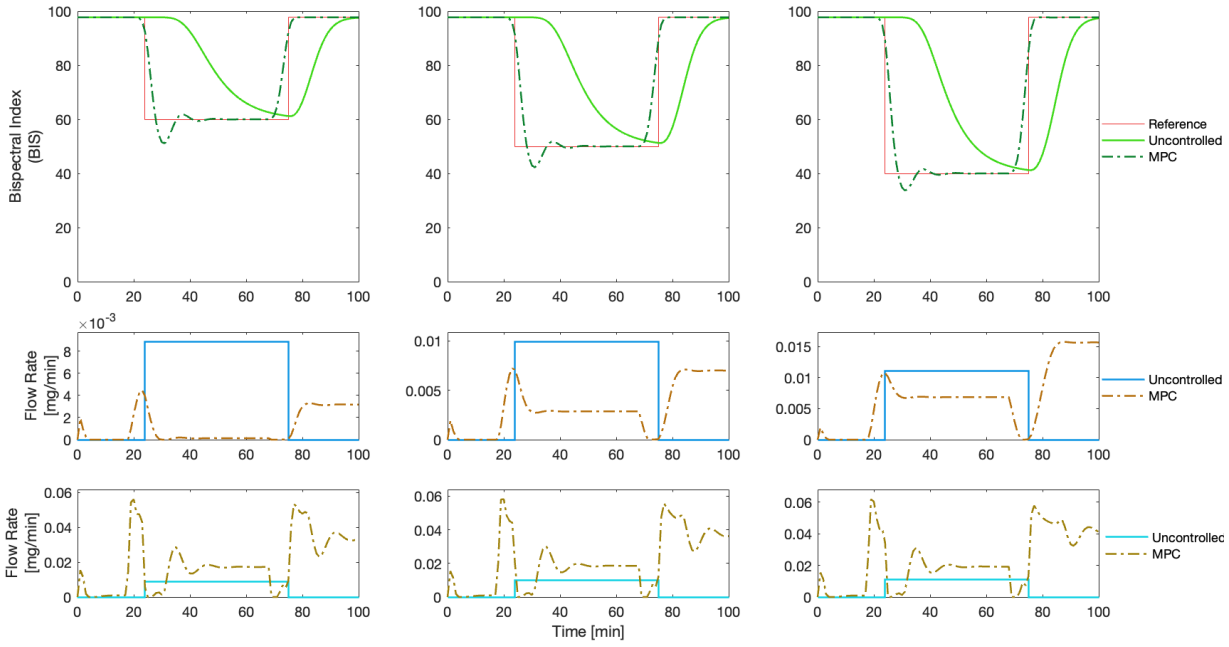

Figure 5: Static BIS Progression - Patient 3

the plots for patient 3 show how the model predictive controller is able to significantly reduce the lag of the output response.

- **Drug Usage** - Unlike the steady-state approach used for the uncontrolled system, applying the model predictive controller produces a nonlinear input sequence of flow rates. A key result is that the overall use of propofol is diminished significantly for every patient and every BIS level. Even though the use of remifentanil is increased, the reduced infusion of propofol decreases the total amount of anesthetic used for each patient.

Despite gaining a lot of insights about the general effects of the implemented model predictive controller on this system, there needs to be further testing on dynamic BIS progressions as they better reflect what the actual sedation level of a patient behaves like during a surgical procedure.

4.1.2 Dynamic BIS Progression

Now after getting conducting some preliminary analysis on general effects of this control law on static BIS progressions, we can more further explore the effects on dynamic BIS progressions. Following the same format as before, each set of plots shows the uncontrolled and patient response to four different dynamic BIS progression cases. These reference BIS signals are no longer just dropping down to a single level as that does not accurately represent what occurs in an actual procedure. The reference signals for each case follow the pattern of first bringing down the patient BIS to operating level as quickly as possible, then adding some fluctuations in the patient BIS while

they are in a deeply sedated state, and finally gradually bringing the patient back to consciousness. Again, each plot is accompanied by the propofol and remifentanil infusion rates for each case. Figures 6-8 show the simulated results.

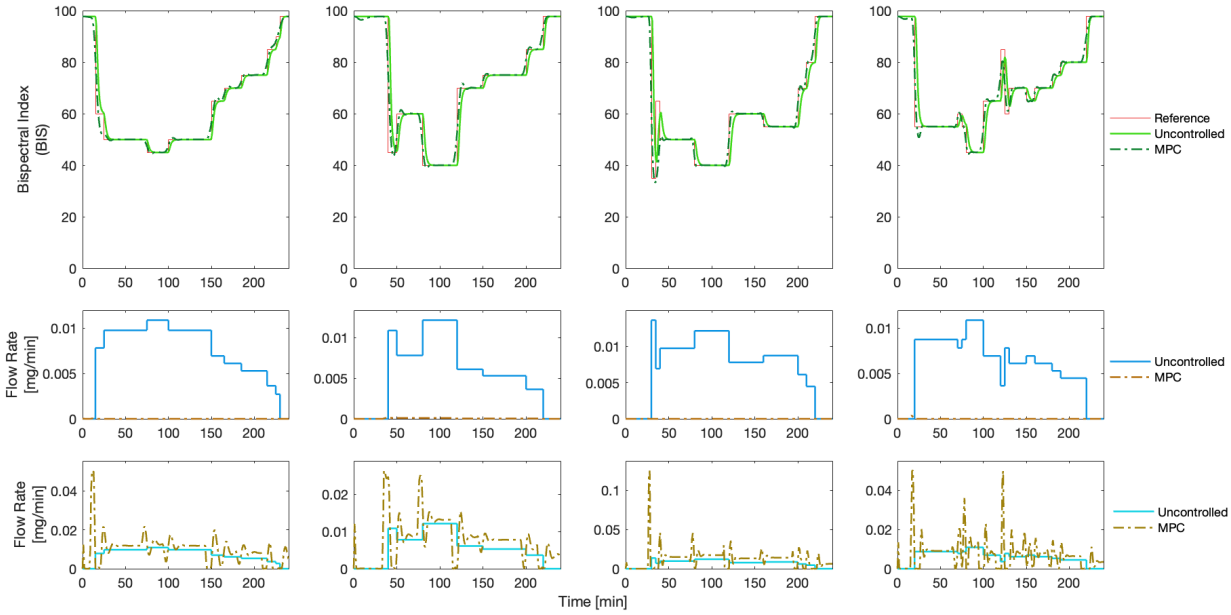

Figure 6: Dynamic BIS Progression - Patient 1

Many of the observation that were made for the static BIS progression cases can be generalized to the dynamic BIS progression cases as well. There is the presence of undershoot in the different cases for each patient especially in the initial portion of the outlined procedure. Overshoot is also present in the sequence of gradually bringing the patient back to full consciousness. This is mostly minimal since the difference between increasing BIS levels is not as abrupt. While mostly visible with patient 3, the model predictive controller still provides good tracking at the output since the lag on the actual patient BIS compared to the desired BIS is reduced as well. Finally, we see a diminished use of propofol and intermittent use of remifentanil which contributes to less anesthetic used for each procedure and each patient.

After analyzing the simulated results, we see that model predictive control is a suitable strategy for improving the reference tracking of the uncontrolled system while minimizing the amount of anesthetic administered to the patient. The main limitations to the strategy would have to be the available hardware. This strategy revolves around optimization which can be iterated for several epochs to determine the minimal flow rates for the desired BIS. Even though this control strategy is computationally expensive, the results show that this price is worth paying especially for systems that perform poorly in the open-loop configuration.

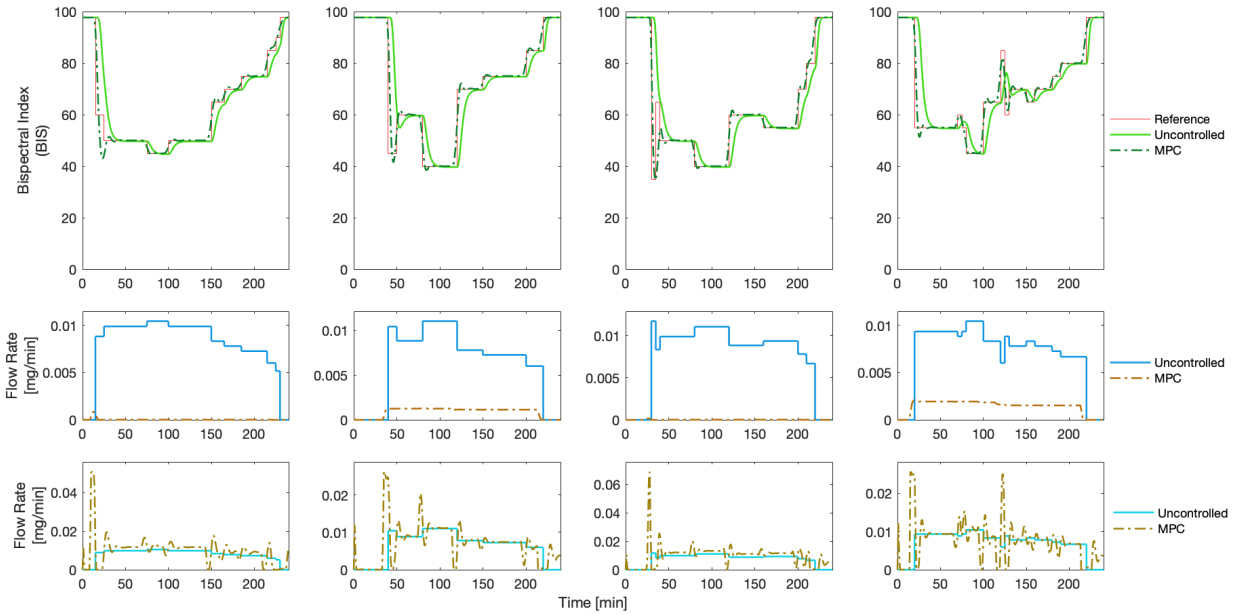

Figure 7: Dynamic BIS Progression - Patient 2

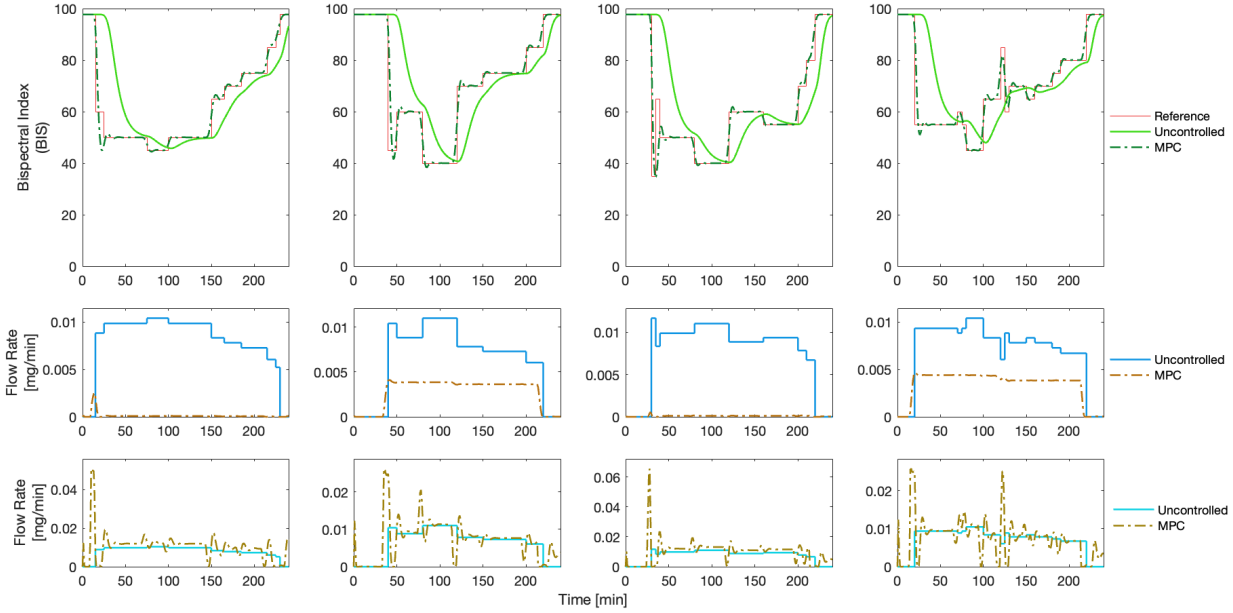

Figure 8: Dynamic BIS Progression - Patient 3

5 Future Work

As of now, what was currently accomplished is excellent in its own right, but is strictly theoretical. If there was more time in the semester, a next step would be to further complexify the system model. More research would need to be conducted on pharmacokinetics, pharmacodynamics, and compartment models to potentially reflect any higher order effects that the current anesthesia delivery system did not consider.

This primarily alludes to unforeseen side effects and limitations of propofol and remifentanil. Additionally, another direction to further take the project in is to explore how the model works with inhalable anesthetics. Anesthetics administered as a gas changes how the model and system dynamics would need to be constructed. The system would not only need to consider the gas flow rates of the anesthetics, but the breathing rate of the patient. Pursuing this direction would certainly be quite challenging, but it would be interesting to compare and contrast how the current control strategy implemented in this project would fare with a new system model.

Other areas of future work would potentially include a transition between theory and physical implementation. Anesthetics, whether they are administered through intraveneous fluids or inhalation, can be delivered through automation of the actual hardware. Since this solution produces the optimal flow rates of propofol and remifentanil, the next step would be to see how these flow rates can be actuated. Exploring fields of ASIC design or microcontroller programming could be areas to consider if taking the project into industry.

6 Conclusion

Through the use of modern mathematical concepts, optimization techniques, and control strategies, this project was able to synthesize an advanced solution for safely manipulating the sedation level of a patient undergoing an invasive surgical procedure. A culmination of biochemical and pharmaceutical models made it possible to accurately evaluate the anesthetic effects of propofol and remifentanil on the sedation level of a patient. The tested control strategy also provides great flexibility in tailoring what is the appropriate amount of anesthetic required for a certain patient and their desired sedation levels. This improved the safety and utility of the analyzed control strategy since it determines what is the minimum amount of anesthetic required to get the desired results. This control method provides anesthesiologists an effective and robust option for determining the optimal combination of anesthetic drugs to administer to a patient. This project represents the combined work of varying scientific fields. The resulting controller is an interdisciplinary accomplishment of biochemistry, pharmacology, mathematics, and engineering all coming together to further advance the medical industry.

7 References

S. Mandal, N. Mandal, B. Pal. *Study of the Performance of an Automatic Electrical Control System for Anesthesia of a Patient in a Clinic.* 2015.

F.N. Nogueira, T. Mendonc¸ P. Rocha. *Controlling the Depth of Anesthesia by a Novel Positive Control Strategy.* Elsevier Ireland 2013.

F.N. Nogueira, T. Mendonc¸ P. Rocha. *A New Retuning Approach for DoA Reference Tracking Improvement.* Proceedings of the 19th World Congress: The International Federation of Automatic Control, 2014, pp. 9890–9894.

O. Rosen, M.M. Silva. A. Medvedev. *Nonlinear Estimation of a Parsimonious Wiener Model for the Neuromuscular Blockade in Closed-Loop Anesthesia.* Proceedings of the 19th World Congress: The International Federation of Automatic Control, 2014, pp. 9258–9264.

Zak, S 2017, *An Introduction to Model-based Predictive Control (MPC)*, lecture notes, ECE 680, Purdue University, delivered Fall 2017.

A Model Predictive Control

A.1 Overview

Model predictive control, otherwise known as "MPC" or the moving horizon or receding horizon control, is a control strategy for assisting linear dynamical systems track a desired output. Model predictive control can be used with stable/unstable and linear/nonlinear system dynamics. This technique performs its computation online for a given reference signal, and thus is a computationally expensive control method. However, the computational price to pay is justified in that MPC, when configured with the proper settings and constraints, yields exceptional performance in controlling the plant output.

The high level description of model predictive control is as follows. The controller will take as input the current state, input, and output of the system and then compute over a finite prediction horizon how the state and output will change based on the next input in the predicted control sequence. While only the first element in the predicted control sequence is used as the actual plant input, having a longer prediction horizon, and subsequently more predicted inputs and outputs, will better inform the controller about what is the optimal plant input that will track the reference signal. This strategy operates in discrete-time and uses both controllability and observability properties to perform the online inference of the optimal input sequence for tracking the desired output at the plant.

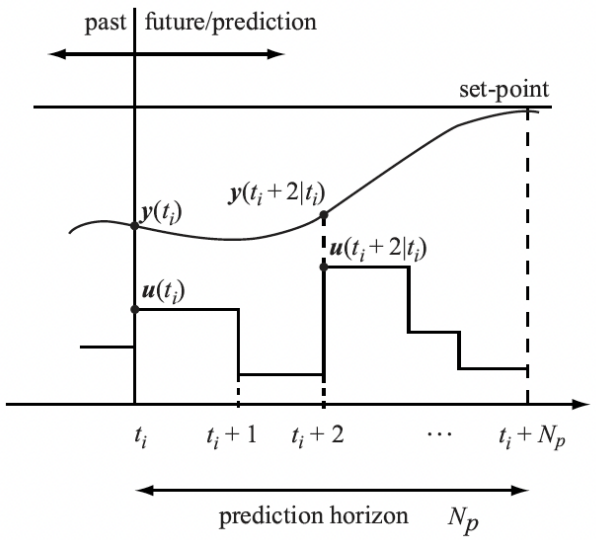

Figure 9: Model Predictive Control Illustration

As such continuous-time systems need to first be discretized, typically by a zero order hold (ZOH), before any computation can be performed. This will be discussed in later subsections, but the discrete-time system is further augmented into a system that becomes a function of predicted input changes, current states, and predicted outputs. This augmented system, sometimes referred to as the "plant model", is essential for the model predictive controller in performing inference.

In addition to creating an augmented, discrete-time plant model, model predictive control utilizes an optimizer to determine the optimal input sequence for the plant. The optimizer itself takes as input the predicted error between the actual output and the predicted output, a quadratic cost function and any accompanying equality or inequality constraints, and outputs a predicted input at the given time step.

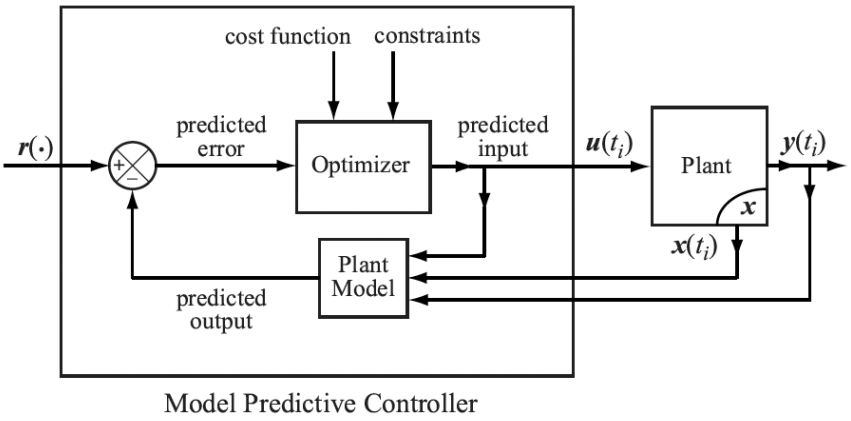

Figure 10: Model Predictive Control Architecture

More will be discussed about how the optimizer performs the inference of the optimal input sequence in later subsections, but this overview primarily illustrates the main control methodology being applied to this anesthesia delivery system.

A.2 Derivation

Before implementing model predictive control in MATLAB, an essential preliminary step was to derive the expected dimensions and sizes of the matrices used in this technique. The following derivations will cover the system conversion from continuous-time to discrete-time and the augmentation of the discrete-time model for MPC.

A.2.1 Continuous-Time to Discrete-Time

Suppose we have the continuous-time system (A, B, C) with the following state-space model:

$$\begin{aligned} \dot{x(t)} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \\ z(t) &= f(y(t)) \end{aligned} \tag{12}$$

This system is then passed through a ZOH to be discretized. Applying the MATLAB command `c2dm()` to (A, B, C) for some sampling time T_s will automatically perform this conversion with the below calculations.

$$A_d = e^{AT_s} \qquad B_d = \int_0^{T_s} e^{A\tau} B d\tau \qquad C_d = C \tag{13}$$

This results in the discrete-time system (A_d, B_d, C_d) with the following state-space model:

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \\ z(k) &= f(y(k)) \end{aligned} \tag{14}$$

Depending on the continuous-time system, specifically the patient specific parameters, different sampling times T_s will affect the performance of the model predictive control strategy. Further discussion on sampling time selection is done in Section –.

A.2.2 Augmentation

From Section –, which defines the anesthesia delivery system model, we know that $A \in \mathbb{R}^{6\times6}$, $B \in \mathbb{R}^{6\times1}$, and $C \in \mathbb{R}^{1\times6}$. Consequently, converting from continuous- to discrete-time does change the dimensions of the system which accordingly makes $A_d \in \mathbb{R}^{6\times6}$, $B_d \in \mathbb{R}^{6\times1}$, and $C_d \in \mathbb{R}^{1\times6}$. Reviewing the current dimensions of the base system model is important in ensuring that we build the proper augmented system model.

Currently, our discrete-time system operates in terms of the input and output at time step k and the state at the current time step k and the next time step $k+1$. To construct the augmented system model we need to consider the changes in our discrete-time system by applying the backwards difference operator, denoted by $\Delta f(k+1) = f(k+1) - f(k)$, to the entire system.

First, we apply the backwards difference operator to the state equation:

$$\begin{aligned} \Delta x(k+1) &= x(k+1) - x(k) \\ &= A_d x(k) + B_d u(k) - A_d x(k-1) - B_d u(k-1) \\ &= A_d(x(k) - x(k-1)) + B_d(u(k) - u(k-1)) \end{aligned} \tag{15}$$

This allows us to get an expression for the state change at the next time step $k+1$ in terms of the current change in the state and input.

$$\Delta x(k+1) = A_d \Delta x(k) + B_d \Delta u(k) \tag{16}$$

Second, we apply the backwards difference operator to the output equation:

$$\begin{aligned}\Delta y(k+1) &= y(k+1) - y(k) \\ &= C_d x(k+1) - C_d x(k) \\ &= C_d(x(k+1) - x(k)) \\ &= C_d \Delta x(k+1) \\ &= C_d A_d \Delta x(k) + C_d B_d \Delta u(k)\end{aligned} \tag{17}$$

It follows that the output at the next time step $k+1$ is now a function of the current change in the state and input.

$$y(k+1) = y(k) + C_d A_d \Delta x(k) + C_d B_d \Delta u(k) \tag{18}$$

Finally, we combine the two results into a new system of linear equations where the future output and state change are a function of the current output, state and input change:

$$\begin{aligned}\begin{bmatrix}\Delta x(k+1) \\ y(k+1)\end{bmatrix} &= \begin{bmatrix}A_d & O_{n\times p} \\ C_d A_d & I_p\end{bmatrix}\begin{bmatrix}\Delta x(k) \\ y(k)\end{bmatrix} + \begin{bmatrix}B_d \\ C_d B_d\end{bmatrix}\Delta u(k) \\ y(k) &= \begin{bmatrix}O_{p\times n} & I_p\end{bmatrix}\begin{bmatrix}\Delta x(k) \\ y(k)\end{bmatrix}\end{aligned} \tag{19}$$

This augmented system states can be formalized with the following definitions that combine the current input change and output into a single state variable.

$$x_a(k) = \begin{bmatrix}\Delta x(k) \\ y(k)\end{bmatrix} \tag{20}$$

The augmented system matrices can also be assigned with the following definitions to further simplify our analysis of constructing a model predictive controller.

$$\Phi = \begin{bmatrix}A_d & O_{n\times p} \\ C_d A_d & I_p\end{bmatrix} \qquad \Psi = \begin{bmatrix}B_d \\ C_d B_d\end{bmatrix} \qquad \Omega = \begin{bmatrix}O_{p\times n} & I_p\end{bmatrix} \tag{21}$$

With the previous definitions, the new system can be written in terms of the augmented state variable and augmented system matrices:

$$\begin{aligned}x_a(k+1) &= \Phi x_a(k) + \Psi \Delta u(k) \\ y(k) &= \Omega x_a(k)\end{aligned} \tag{22}$$

If we examine closely the augmented system (Φ, Ψ, Ω), as well as the associated matrix multiplications that go into constructing these augmented matrices, we are able to determine the expected dimensions of the augmented system model. In particular, we know that there are $n = 6$ states, $m = 2$ inputs, and $p = 1$ outputs. This would indicate that $\Phi \in \mathbb{R}^{7\times 7}$, $\Psi \in \mathbb{R}^{7\times 2}$, and $\Omega \in \mathbb{R}^{1\times 7}$.

From the Cayley-Hamilton theorem we know that for some square matrix $S \in \mathbb{R}^{N \times N}$:

$$S^N \in \text{span}\{S^{N-1}, S^{N-2}, \ldots, S, I\} \tag{23}$$

By using the Cayley-Hamilton theorem with the augmented state matrix Φ we can determine the minimum prediction horizon N_p associated with the predictive anesthesia system model. Since there are $n' = 7$ augmented states, the predictive plant model requires at least a prediction horizon of $N_p = 7$.

This system is only a function of the current time step k and the next time step $k+1$. As stated earlier, model predictive control is able to compute more accurately the optimal control sequence over a prediction horizon N_p. This would result in a running computation of predicted input changes,

$$\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+N_p-1) \tag{24}$$

The augmented states x_a which contain the state change $\Delta x(k)$ and output $y(k)$ of the plant,

$$x_a(k+1), x_a(k+2), \ldots, x_a(k+N_p) \tag{25}$$

And the predicted outputs of the plant based on the predicted input changes.

$$y(k+1), y(k+2), \ldots, y(k+N_p) \tag{26}$$

Taking into account the previous observations and properly understanding the premise of model predictive control, we can further modify the augmented system (Φ, Ψ, Ω) to take into account the prediction horizon N_p.

First, we can rewrite the predicted augmented states as a recursion of one another and the predicted input changes:

$$\begin{aligned}
x_a(k+1) &= \Phi x_a(k) + \Psi \Delta u(k) \\
x_a(k+2) &= \Phi x_a(k+1) + \Psi \Delta u(k+1) = \Phi^2 x_a(k) + \Phi\Psi\Delta u(k) + \Psi\Delta u(k+1) \\
&\vdots \\
x_a(k+N_p) &= \Phi^{N_p} x_a(k) + \Phi^{N_p-1}\Psi\Delta u(k) + \Phi^{N_p-2}\Psi\Delta u(k+1) + \cdots + \Psi\Delta u(k+N_p)
\end{aligned} \tag{27}$$

A similar recursion method can be applied to the predicted outputs since they are directly a function of the predicted augmented states:

$$\begin{aligned}
y(k+1) &= \Omega x_a(k+1) = \Omega\Phi x_a(k) + \Omega\Psi\Delta u(k) \\
y(k+2) &= \Omega x_a(k+2) = \Omega\Phi^2 x_a(k) + \Omega\Phi\Psi\Delta u(k) + \Omega\Psi\Delta u(k+1) \\
&\vdots \\
y(k+N_p) &= \Omega x_a(k+N_p) = \Omega\Phi^{N_p} x_a(k) + \Omega\Phi^{N_p-1}\Psi\Delta u(k) + \cdots + \Omega\Phi\Delta u(k+N_p-1)
\end{aligned} \tag{28}$$

Second, we convert the derived recursions into a system of linear equations that will encapsulate the augmented state, input change, and output over the specified prediction horizon N_p. Starting with the predicted augmented states we are able to write:

$$\begin{bmatrix} x_a(k+1) \\ x_a(k+2) \\ \vdots \\ x_a(k+N_p) \end{bmatrix} = \begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \Phi^{N_p} \end{bmatrix} x_a(k) + \begin{bmatrix} \Psi & & \\ \Phi\Psi & \Psi & \\ \vdots & & \ddots \\ \Phi^{N_p-1}\Psi & \cdots & \Psi \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_p-1) \end{bmatrix} \tag{29}$$

Again, we are able to follow a similar procedure for converting the predicted outputs into a system of linear equations represented as:

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_p) \end{bmatrix} = \begin{bmatrix} \Omega\Phi \\ \Omega\Phi^2 \\ \vdots \\ \Omega\Phi^{N_p} \end{bmatrix} x_a(k) + \begin{bmatrix} \Omega\Psi & & \\ \Omega\Phi\Psi & \Omega\Psi & \\ \vdots & & \ddots \\ \Omega\Phi^{N_p-1}\Psi & \cdots & \Omega\Psi \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_p-1) \end{bmatrix} \tag{30}$$

We formalize this modified augmented system of linear equations by defining new variables over the prediction horizon for the predicted input changes and the predicted output.

$$Y = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_p) \end{bmatrix} \qquad \Delta U = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_p-1) \end{bmatrix} \tag{31}$$

Based on the previous derivations of the prediction horizon and the sizing of the augmented system matrices, we know that $Y \in \mathbb{R}^{7\times 1}$ and $\Delta U \in \mathbb{R}^{14\times 1}$. This is because there are two inputs to the anesthesia delivery system and thus the prediction horizon needs to cover both inputs for proper inference.

The modified augmented system matrices can also be assigned with the following definitions to formalize the model predictive controller.

$$W = \begin{bmatrix} \Omega\Phi \\ \Omega\Phi^2 \\ \vdots \\ \Omega\Phi^{N_p} \end{bmatrix} \qquad Z = \begin{bmatrix} \Omega\Psi & & \\ \Omega\Phi\Psi & \Omega\Psi & \\ \vdots & & \ddots \\ \Omega\Phi^{N_p-1}\Psi & \cdots & \Omega\Psi \end{bmatrix} \tag{32}$$

Again, to determine the sizing of the modified augmented system matrices we can examine the previous sizing derivations. Specifically, $Y(k) \in \mathbb{R}^{7\times 1}$ and $x_a(k) \in \mathbb{R}^{7\times 1}$. W needs to be compatible with $x_a(k)$ to get the proper output dimensions, therefore $W \in \mathbb{R}^{7\times 7}$. Similarly, Z needs to map ΔU to $\mathbb{R}^{7\times 1}$. Since $\Delta U \in \mathbb{R}^{14\times 1}$, then $Z \in \mathbb{R}^{7\times 14}$.

Finally, with the previous definitions, the modified augmented system is capable of performing model predictive control for the plant output and can be written as a function of the current augmented state and the predicted input changes over the specified prediction horizon:

$$Y = Wx_a(k) + Z\Delta U \tag{33}$$

This is the predictive plant model and takes as input the necessary information for making inference about how the output will change relative to a new input. Ultimately, this modified plant model will inform the optimizer about what control action is required to reach the desired output.

A.3 Optimization

An optimization strategy needs to be implemented to assist in computing the optimal input sequence for the actual plant. The following section discusses the optimization problem that is solved to realize this controller by outlining the cost function and the linear matrix inequality constraints that will be imposed on the optimal input.

A.3.1 Cost Function

Like with any optimization problem, the model predictive control strategy has its own associated cost function which revolves around the decision variable ΔU. Implementing the model predictive control strategy aims to minimize this exact cost function.

$$J(\Delta U) = \frac{1}{2}(r_p - Y)^T Q(r_p - Y) + \frac{1}{2}\Delta U^T R \Delta U \tag{34}$$

The actual cost function is quadratic and has two terms that are associated with the predicted output Y, and its difference from the reference signal r_p, and with the predicted input changes ΔU. The $Q \in \mathbb{R}^{7\times 7}$ and $R \in \mathbb{R}^{14\times 14}$ matrices will accordingly penalize deviations from the desired output r_p and large changes in the input. Note that:

$$r_p = \begin{bmatrix} y_{ref}(k) & y_{ref}(k+1) & \cdots & y_{ref}(k+N_p) \end{bmatrix}^T \tag{35}$$

Furthermore, to solve this optimization problem the gradient of the cost function needs to be calculated with respect to the decision variable ΔU.

$$\nabla J(\Delta U) = -(r_p - Wx_a - Z\Delta U)^T QZ + \Delta U^T R \tag{36}$$

Evaluating the gradient is essential to determining the optimal solution via the gradient descent method. This will be discussed in more detail in Section –.–.–.

A.4 Constraints

While model predictive control is a powerful controller/observer based technique, it may compute an optimal input sequence that is infeasible for the plant or does not properly align with the physical system dynamics. For the anesthesia delivery system, the two inputs are flow rates of propofol and remifentanil. Unless there is another drug injected at the same flow rates to counteract

propofol and remifentanil, which is not modelled in this system, there is no possible way for the control designer to remove the already infused drug from the patient. In other words, a negative control action, in this case a backwards flow rate, does not have proper meaning for this particular anesthesia delivery system model.

It is this understanding and interpretation of the current anesthesia delivery system which acts as the motivation for wanting to impose constraints on the input of the model predictive controller. These constraints can be synthesized with linear matrix inequalities and are accordingly included in the computation of the optimal input sequence. There are two types of input constraints that we want to implement on this optimization problem.

A.4.1 Input Amplitude

The first set of constraints that are going to be imposed for this model predictive controller are on the amplitude of the control action. This was discussed previously in the motivation for input constraints, but essentially we are looking to keep all control action values positive as a backwards flow rate would not be feasible for the physical system.

The generalized form of amplitude constraints on the control action at the current time step k for the desired output can be expressed as:

$$u_i^{min} \leq u_i(k) \leq u_i^{max} \quad \text{for} \quad i = 1, 2, \ldots, m \tag{37}$$

To express the generalized form over the prediction horizon N_p, in other words for all the elements in ΔU, we note the relation between the current current input, the previous input, and the change in the input.

$$\begin{aligned} u(k) &= u(k-1) + \Delta u(k) \\ &= u(k-1) + \begin{bmatrix} I_m & O_m & \cdots & O_m \end{bmatrix} \Delta U \end{aligned} \tag{38}$$

It follows that the next input is expressed in a similar form from the elements of the input changes over the prediction horizon.

$$\begin{aligned} u(k+1) &= u(k) + \Delta u(k+1) \\ &= u(k-1) + \Delta u(k) + \Delta u(k+1) \\ &= u(k-1) + \begin{bmatrix} I_m & I_m & \cdots & O_m \end{bmatrix} \Delta U \end{aligned} \tag{39}$$

When continuing with this pattern over for time steps $k, k+1, \ldots, k+N_p-1$, the predicted inputs, previous inputs, and input changes over the prediction horizon can be related through a system of equations:

$$\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} u(k-1) + \begin{bmatrix} I_m & O_m & \cdots & O_m \\ I_m & I_m & \cdots & O_m \\ \vdots & & \ddots & \vdots \\ I_m & I_m & \cdots & I_m \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_p-1) \end{bmatrix} \tag{40}$$

Assigning some of the vectors and matrices with the following definitions can further simplify the process in rewriting the input amplitude constraints.

$$U = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} \qquad \tilde{E} = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} \qquad \tilde{H} = \begin{bmatrix} I_m & O_m & \cdots & O_m \\ I_m & I_m & \cdots & O_m \\ \vdots & & \ddots & \vdots \\ I_m & I_m & \cdots & I_m \end{bmatrix} \tag{41}$$

We are able to determine the exact sizing of the previously defined vectors and matrices based on Sections –.–.– and –.–.–. Knowing these dimensions, it can be calculated that $U \in \mathbb{R}^{14}$, $\tilde{E} \in \mathbb{R}^{14}$, and $\tilde{H} \in \mathbb{R}^{14\times 14}$.

Therefore, we are able to express now:

$$U = \tilde{E}u(k-1) + \tilde{H}\Delta U \tag{42}$$

Referring back to the generalized form of the amplitude constraints on the control action, this can be expressed for the vector U of predicted inputs as:

$$U^{min} \leq U \leq U^{max} \tag{43}$$

The vectors for the minimum and maximum input amplitudes are in $\mathbb{R}^{28}$ and given the following definitions.

$$U^{min} = \begin{bmatrix} u_1^{min} \\ u_2^{min} \\ \vdots \\ u_1^{min} \\ u_2^{min} \end{bmatrix} \quad \text{and} \quad U^{max} = \begin{bmatrix} u_1^{max} \\ u_2^{max} \\ \vdots \\ u_1^{max} \\ u_2^{max} \end{bmatrix} \tag{44}$$

Combining the modified generalized form with the previously derived expression relating the predicted inputs to the previous input and the predicted input changes yields a more comprehensive expression of the input amplitude constraint.

$$\begin{aligned} \begin{bmatrix} -U \\ U \end{bmatrix} &\leq \begin{bmatrix} -U^{min} \\ U^{max} \end{bmatrix} \\ \begin{bmatrix} -\tilde{E}u(k-1) - \tilde{H}\Delta U \\ \tilde{E}u(k-1) + \tilde{H}\Delta U \end{bmatrix} &\leq \begin{bmatrix} -U^{min} \\ U^{max} \end{bmatrix} \\ \begin{bmatrix} -\tilde{H} \\ \tilde{H} \end{bmatrix} \Delta U &\leq \begin{bmatrix} -U^{min} + \tilde{E}u(k-1) \\ U^{max} - \tilde{E}u(k-1) \end{bmatrix} \end{aligned} \tag{45}$$

Again, we are able to apply the following definitions to the above block matrices.

$$H = \begin{bmatrix} -\tilde{H} \\ \tilde{H} \end{bmatrix} \qquad E = \begin{bmatrix} -U^{min} + \tilde{E}u(k-1) \\ U^{max} - \tilde{E}u(k-1) \end{bmatrix} \tag{46}$$

Based on previous derivations, we are able to determine that $H \in \mathbb{R}^{28\times 14}$ and $E \in \mathbb{R}^{28}$ which gives a sense of the computational intensity by including these constraints.

Finally, these definitions allow us to formalize the input amplitude constraints over the prediction horizon in a linear matrix inequality that takes the form:

$$H\Delta U \leq E \tag{47}$$

Implementing this linear matrix inequality into the optimizer for the model predictive controller will constrain the computed inputs to lie between the specified range of the control designer. This is just one of two input constraints that need to be considered for the model predictive controller.

A.4.2 Input Rate of Change

The second set of constraints that are going to be imposed are on the rate of change of the control action. The primary motivation of including this constraint is to minimize, and in some cases induce, sharp changes in the control action and keep the flow rates at a gradual increase or decrease throughout a given patient procedure.

The generalized form of rate of change constraints on the control action at the current time step k for the desired output can be expressed as:

$$\Delta u_i^{min} \leq \Delta u_i(k) \leq \Delta u_i^{max} \quad \text{for} \quad i = 1, 2, \ldots, m \tag{48}$$

Following similar steps as for when we derived the input constraints, the generalized form of the rate of change constraints can be expressed for the $\Delta u(k)$ of predicted input changes as:

$$\Delta U^{min} \leq \Delta u(k) \leq \Delta U^{max} \tag{49}$$

The vectors for the minimum and maximum input rate of changes are in $\mathbb{R}^2$ and given the following definitions.

$$\Delta U^{min} = \begin{bmatrix} \Delta u_1^{min} & \Delta u_2^{min} \end{bmatrix}^T \quad \text{and} \quad \Delta U^{max} = \begin{bmatrix} \Delta u_1^{max} & \Delta u_2^{max} \end{bmatrix}^T \tag{50}$$

Combining the modified generalized form with the previous minimum and maximum input change definitions creates a refined expression of the input rate of change constraint.

$$\begin{bmatrix} -I_m \\ I_m \end{bmatrix} \Delta u(k) \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix} \tag{51}$$

It follows that the input changes at the next time step $k+1$ through the prediction horizon N_p can be expressed in an identical manner.

$$\begin{bmatrix} -I_m \\ I_m \end{bmatrix} \Delta u(k+1) \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$
$$\vdots \tag{52}$$
$$\begin{bmatrix} -I_m \\ I_m \end{bmatrix} \Delta u(k+N_p-1) \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$

When continuing with this pattern, the predicted input changes can be bounded by the minimum and maximum rate of change through a system of equations:

$$\begin{bmatrix} -I_m & O_m & \cdots & O_m & O_m \\ I_m & O_m & \cdots & O_m & O_m \\ O_m & -I_m & \cdots & O_m & O_m \\ O_m & I_m & \cdots & O_m & O_m \\ \vdots & & & & \vdots \\ O_m & O_m & \cdots & O_m & -I_m \\ O_m & O_m & \cdots & O_m & I_m \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_p-1) \end{bmatrix} \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \\ -\Delta U^{min} \\ \Delta U^{max} \\ \vdots \\ -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix} \tag{53}$$

Assigning some of the vectors and matrices with the following definitions can further simplify the process in rewriting the input rate of change constraints.

$$G = \begin{bmatrix} -I_m & O_m & \cdots & O_m & O_m \\ I_m & O_m & \cdots & O_m & O_m \\ O_m & -I_m & \cdots & O_m & O_m \\ O_m & I_m & \cdots & O_m & O_m \\ \vdots & & & & \vdots \\ O_m & O_m & \cdots & O_m & -I_m \\ O_m & O_m & \cdots & O_m & I_m \end{bmatrix} \qquad C = \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \\ -\Delta U^{min} \\ \Delta U^{max} \\ \vdots \\ -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix} \tag{54}$$

Again, we are able to determine the exact sizing of the previously defined vectors and matrices based on Sections –.–.– and –.–.–. Knowing these dimensions, it can be calculated that $G \in \mathbb{R}^{28\times 14}$ and $C \in \mathbb{R}^{28}$.

Finally, these definitions allow us to formalize the input rate of change constraints over the prediction horizon in a linear matrix inequality that takes the form:

$$G\Delta U \leq C \tag{55}$$

Adding this linear matrix inequality into the optimizer for the model predictive controller will constrain the minimum and maximum allowable differentials between control actions. This is the second of two input constraints that need to be considered.

A.4.3 Combining Input Constraints

Having derived the two linear matrix inequalities for the control action amplitude and rate of change, we need to synthesize both expressions into a single linear matrix inequality that accounts for both input constraints.

$$\begin{aligned} H\Delta U &\leq E \quad \text{(amplitude)} \\ G\Delta U &\leq C \quad \text{(rate of change)} \end{aligned} \tag{56}$$

Since both constraints include the vector ΔU of input changes over the prediction horizon N_p, these expressions can be combined into a larger block linear matrix inequality denoted by the following:

$$\begin{bmatrix} H & O_{28\times 14} \\ O_{28\times 14} & G \end{bmatrix} \begin{bmatrix} \Delta U \\ \Delta U \end{bmatrix} \leq \begin{bmatrix} E \\ C \end{bmatrix} \tag{57}$$

Like with all the previous derivations, assigning new definitions to these block matrices can simplify the combined inequality constraints for the gradient descent algorithm which will be discussed in Section –.–.–.

$$\Theta = \begin{bmatrix} H & O_{28\times 14} \\ O_{28\times 14} & G \end{bmatrix} \qquad \Pi = \begin{bmatrix} E \\ C \end{bmatrix} \tag{58}$$

Additionally, $\Delta\mathbf{U} = \begin{bmatrix} \Delta U & \Delta U \end{bmatrix}^T \in \mathbb{R}^{28}$ will represent the independently combined predicted input changes associated for each individual constraint. These definitions have $\Theta \in \mathbb{R}^{56\times 28}$ and $\Pi \in \mathbb{R}^{56}$. The sizes of these block matrices and vectors further show the computational intensity of imposing input constraints for our model predictive controller.

These variables result in the following simplified linear matrix inequality that encompasses all desired input constraints:

$$\Theta\Delta\mathbf{U} \leq \Pi \tag{59}$$

This finalized LMI will be be an input to the optimizer and will be used for calculating the optimal input changes. While clearly computationally expensive, the inclusion of this LMI will ultimately assist in modelling a more accurate anesthesia delivery system.

A.4.4 Input Change Equality

There is one extra constraint that can be imposed on this optimization problem. A constraint can be setup to ensure that the values associated with each predicted set of input changes for each input constraint are equivalent.

Suppose we have the two sets of predicted input changes ΔU_1 for the amplitude constraint and ΔU_2, for the rate of change constraint. The desired result of this constraint can be expressed as:

$$\Delta u_1(k+i) = \Delta u_2(k+i) \quad \text{for} \quad i = 0, 1, \ldots, N_p - 1 \tag{60}$$

The general expression above can be rewritten with identity matrices for both sets of predicted input changes concatenated on top of one another.

$$\begin{bmatrix} I_{14\times 28} & -I_{14\times 28} \end{bmatrix} \begin{bmatrix} \Delta U_1 \\ \Delta U_2 \end{bmatrix} = 0 \tag{61}$$

Here again, we assign definitions to the block matrices to further simplify the equality constraints for the gradient descent algorithm during optimization.

$$\Xi = \begin{bmatrix} I_{14} & -I_{14} \end{bmatrix} \tag{62}$$

Similar to the inequality constraints, $\Delta \mathbf{U} = \begin{bmatrix} \Delta U_1 & \Delta U_2 \end{bmatrix}^T \in \mathbb{R}^{28}$ represents the independently combined predicted input changes for each individual constraint. This definition has $\Xi \in \mathbb{R}^{14\times 28}$ which continues to show the high computational cost of placing constraints on our model predictive controller.

This variable substitutions results in the following simplified linear matrix inequality imposed on the input constraints over the prediction horizon:

$$\Xi \Delta \mathbf{U} = 0 \tag{63}$$

With all the LMIs defined, they can be set as inputs to the optimizer for calculating our desired, optimal input changes. These inequality and equality constraints will help further model an accurate anesthesia delivery system with appropriate propofol and remifentanil flow rates.

A.5 Realization

Even though there are optimization tools such as CVX, or even built-in MATLAB toolboxes for MPC, the methodology used to implement this model predictive controller lies in the gradient descent algorithm. More specifically, a first-order Lagrangian algorithm is applied for the inequality and equality constraints. This section looks to outline the exact calculations and mathematical methods used to compute the optimal input sequence to the anesthesia delivery system.

A.5.1 Modified Cost Function

First, let us restate the cost function associated with this optimization problem for our model predictive controller:

$$J(\Delta \mathbf{U}) = \frac{1}{2}(\mathbf{r_p} - \mathbf{Y})^T \mathbf{Q}(\mathbf{r_p} - \mathbf{Y}) + \frac{1}{2}\Delta \mathbf{U}^T \mathbf{R} \Delta \mathbf{U} \tag{64}$$

It is important to note the inclusion of the bold variables in the cost function. This indicates that the cost function, and subsequently each matrix and vector variable in it, needs to be modified

to account for the multiple inequality constraints.

Essentially all the bold matrix and vector variables are now block matrices and concatenated vectors of the original variables. The following definitions will clarify the structure of these variables as they are implemented for this algorithm.

For the reference output and calculated output over the prediction horizon:

$$\mathbf{Y} = \begin{bmatrix} Y \\ O_{7\times1} \\ O_{7\times1} \\ Y \end{bmatrix} \qquad \mathbf{r_p} = \begin{bmatrix} r_p \\ O_{7\times1} \\ O_{7\times1} \\ r_p \end{bmatrix} \tag{65}$$

For the output deviation and input change weight matrices:

$$\mathbf{Q} = \begin{bmatrix} Q & O_{21\times21} \\ O_{21\times21} & Q \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} R & O_{14\times14} \\ O_{14\times14} & R \end{bmatrix} \tag{66}$$

Second, it follows that we need to consider a modified expression for the gradient of the cost function given the inequality and equality constraints on this optimization problem:

$$\nabla J(\Delta\mathbf{U}) = -(\mathbf{r_p} - \mathbf{W}\mathbf{x_a} - \mathbf{Z}\Delta\mathbf{U})^T\mathbf{Q}\mathbf{Z} + \Delta\mathbf{U}^T\mathbf{R} \tag{67}$$

The variables that need modification for the cost function gradient expression are the modified augmented system and input matrices:

$$\mathbf{W} = \begin{bmatrix} W & O_{21\times21} \\ O_{21\times21} & W \end{bmatrix} \qquad \mathbf{Z} = \begin{bmatrix} Z & O_{21\times14} \\ O_{21\times14} & Z \end{bmatrix} \tag{68}$$

When examining the composition of these new block matrix variables we see that $\mathbf{Y}, \mathbf{r_p} \in \mathbb{R}^{28}$ and $\mathbf{Q}, \mathbf{R}, \mathbf{W}, \mathbf{Z} \in \mathbb{R}^{28\times28}$. With these dimensions, it can also be determined that $\nabla J(\Delta\mathbf{U})^T \in \mathbb{R}^{28}$. Understanding the dimensions of the modified cost function will ensure that our results from the gradient descent algorithm are consistent.

A.5.2 Gradient Descent Algorithm

Finally, we are able to implement the gradient descent algorithm as the optimizer for our model predictive controller. This algorithm will compute for each time step what is the optimal input that will get our system closest to the desired output. After completing all the necessary derivations, we are able to outline the finalized optimization problem:

$$\begin{aligned} \min_{\Delta\mathbf{U}} \quad & J(\Delta\mathbf{U}) = \frac{1}{2}(\mathbf{r_p} - \mathbf{Y})^T\mathbf{Q}(\mathbf{r_p} - \mathbf{Y}) + \frac{1}{2}\Delta\mathbf{U}^T\mathbf{R}\Delta\mathbf{U} \\ \text{s.t.} \quad & \Theta\Delta\mathbf{U} \leq \Pi \quad \text{where} \quad g(\Delta\mathbf{U}) = \Theta\Delta\mathbf{U} - \Pi \\ & \Xi\Delta\mathbf{U} = 0 \quad \text{where} \quad h(\Delta\mathbf{U}) = \Xi\Delta\mathbf{U} \end{aligned} \tag{69}$$

Since we are minimizing with respect to the decision variable $\Delta\mathbf{U}$, it follows that the gradient descent algorithm needs to make iterative computations to find $\Delta\mathbf{U}^*$. Without imposing the inequality and equality constraints, the above optimization problem can be written as:

$$\Delta\mathbf{U}^+ = \Delta\mathbf{U} - \alpha\nabla J(\Delta\mathbf{U})^T \tag{70}$$

To compute the next, optimal value of $\Delta\mathbf{U}$, the algorithm requires a predefined step size α as well as the gradient of the cost function $\nabla J(\Delta\mathbf{U})^T$. As a reminder, the computed $\Delta\mathbf{U}^+ \in \mathbb{R}^{28}$ is a vector of the predicted input changes over the prediction horizon $N_p = 7$ for $m = 2$ inputs.

We can modify the above expression to include constraints by adding the Jacobian matrices of each constraint:

$$\Delta\mathbf{U}^+ = \Delta\mathbf{U} - \alpha(\nabla J(\Delta\mathbf{U})^T + \Theta^T\mu + \Xi^T\lambda) \tag{71}$$

Notice that the gradient step size α is now multiplied through by the summation of the gradient of the cost function, Θ, for the input amplitude and rate of change constraints, and Ξ, for the equality constraints of the prediction horizon.

The Jacobian matrices also need to be multiplied by Karush-Kuhn-Tucker multipliers for optimization. These are updated throughout the gradient descent algorithm as well:

$$\begin{aligned} \mu^+ &= [\mu + \beta g(\Delta\mathbf{U})]_+ \\ \lambda^+ &= \lambda + \beta h(\Delta\mathbf{U}) \end{aligned} \tag{72}$$

The optimization multipliers are computed by evaluating the inequality and equality constraints at the current values of $\Delta\mathbf{U}$ and multiplying by a gradient step size β. Consequently, $\mu \in \mathbb{R}^{56}$ and $\lambda \in \mathbb{R}^{14}$ for the proper computation of optimal input changes.

A.5.3 MATLAB

With the gradient descent algorithm outlined by the previous derivations and mathematical theory, we can realize this algorithm by using MATLAB. The following steps are performed by MATLAB to solve the optimization problem posed by model predictive control:

1. Define reference signal length N and reference signal values z_{ref}.
2. From time steps $k = 1$ to $k = N - 1$ do the following:
   (a) Assign reference signal values from the current time step k to the prediction horizon $k + N_p$ to the augmented reference signal vector $\mathbf{r_p}$.
   (b) For a given number of gradient descent algorithm iterations do the following:
      i. Evaluate the cost function gradient:

$$\nabla J(\Delta\mathbf{U}) = -(\mathbf{r_p} - \mathbf{W}\mathbf{x_a} - \mathbf{Z}\Delta\mathbf{U})^T\mathbf{Q}\mathbf{Z} + \Delta\mathbf{U}^T\mathbf{R}$$

ii. Evaluate the inequality constraints:

$$g(\Delta\mathbf{U}) = \Theta\Delta\mathbf{U} - \Pi$$

iii. Evaluate the equality constraints:

$$h(\Delta\mathbf{U}) = \Xi\Delta\mathbf{U}$$

iv. Compute the predicted input change:

$$\Delta\mathbf{U}^{+} = \Delta\mathbf{U} - \alpha(\nabla J(\Delta\mathbf{U})^{T} + \Theta^{T}\mu + \Xi^{T}\lambda)$$

v. Update the Karush-Kuhn-Tucker multipliers:

$$\mu^{+} = [\mu + \beta g(\Delta\mathbf{U})]_{+}$$
$$\lambda^{+} = \lambda + \beta h(\Delta\mathbf{U})$$

(c) Using the augmented (Φ, Ψ, Ω), calculate the following:

i. Predicted input:

$$u^{+} = u + \Delta\mathbf{U}(1)$$

ii. Augmented state:

$$x_a^{+} = \Phi x_a + \Psi\Delta\mathbf{U}(1)$$

iii. Observed state:

$$y = \Omega x_a^{+}$$

iv. Predicted output:

$$z = f(y)$$

No built-in toolboxes need to be downloaded, and the entire algorithm can be sythesized with the use of `for` loops and `if`,`else` statements. As seen in the derivation, the matrix multiplications make the entire algorithm computationally expensive. Reducing the number of iterations of the gradient descent algorithm that are performed for each time step reduces the computational intensity. However, the algorithm will not converge as well to the optimal solution.

A GitHub repository contains all the code used for analyzing this system and synthesizing a model predictive controller: https://github.com/pmunar15/Anesthesia-Delivery-Control.git