Modeling of drug effect in general closed-loop anesthesia

Sander Cox
Abstract

Modeling of drug effect in general closed-loop anesthesia

Sander Cox

In medicine, anesthesia is achieved by administering two interacting drugs. Nowadays, the Depth of Anesthesia can be expressed by the Bispectral Index Scale, which is measured by an EEG. In order to make automatic closed-loop anesthesia possible with the benefits of 1) relieving the anesthesiologist from the hard task of administering optimal drug doses, 2) achieving more consistent drug effects by means of individualization, and 3) reducing side effects because of the achieved reduced overall drug administration, estimating accurate models of the effect of drug doses on the Depth of Anesthesia is essential.

The model used was a minimally parametrized PharmacoKinetic-PharmacoDynamic Wiener model. The parameters of the model were estimated using an Extended Kalman Filter, whose parameters were tuned manually. The model and filter were tested on new data from both the University of Porto and the University of Brescia. The unit of the reference data set from Porto was unknown, so in order to use the scale-dependent model an educated guess was made to convert the other data sets to a reasonable scale. Furthermore, the data from Brescia was incomplete, which could only partly be remediated. Similar tracking performances were obtained when using the new data sets compared to the reference data, however, either relatively constant estimates, or different parameter estimates for similar conditions, were typically obtained. This questions the validity of the model used and if the parameters found can be trusted. Therefore, the replication of the procedure on other complete data, and the comparison with the application of other models on the studied data, is subject for future research.
## Contents

1 Introduction 3

2 Medical background: anesthesia 4

3 Theoretical background 4

3.1 Modelling of dynamical systems 4

3.1.1 Models 4

3.1.2 Signals in models 5

3.1.3 State and state-space models 6

3.1.4 Stationary points and linearization 7

3.1.5 Disturbances 8

3.1.6 Transfer functions 9

3.1.7 PID controller 12

3.2 Pharmacokinetics and pharmacodynamics 13

3.2.1 Wiener models 15

3.3 The Kalman filter 16

3.4 The extended Kalman filter 18

4 Used model of effect of drug concentrations on BIS 21

4.1 General properties 21

4.2 The model in detail 21

4.2.1 Linear block 22

4.2.2 Nonlinear block 22

4.2.3 State-space model 23

4.2.4 Assigned values to parameters 25

5 Testing the model on data 25

5.1 Available data 25

5.2 A typical run on the old data: marsh_pat1 26

5.3 Testing on new data from Porto: Bis_1 29

5.3.1 Attempt to prevent the occurrence of complex values with constraints 30

5.3.2 Attempt with tuning 34

5.3.3 Scaling 36

5.4 Testing on data from Brescia: PZ41_1 36

5.5 Units 36

5.5.1 Calculating recommended doses 37

5.6 Running with conversion to assumed unit 39

5.6.1 60Bis_1 (converted to mg/h) 39

5.6.2 Bis1 (converted to mg/h) and tuned 43

5.6.3 3600PAZ41new tuned and with right sampling interval 46

6 Conclusion and future work 50
1 Introduction

Problem

In medicine, anesthetic drugs are administered in order to induce reversible loss of sensation and facilitate surgery. Since the depth of anesthesia (DoA) can be nowadays measured from electroencephalogram (EEG) of the patient using so-called bispectral index (BIS), automatic closed-loop anesthesia has become feasible. Apart from relieving the anesthesiologists from the routine task of adjusting the flow of anesthetic drugs to achieve a desired level of DoA, closed-loop anesthesia holds promise for more consistent drug effect because of treatment individualization and less side effects due to a reduction of the overall administered drug dose during the surgery. Systematic design of feedback control needs an accurate description of the drug concentration in, and its actual effect on the patient.

The point of departure was the article Online Nonlinear Identification of the Effect of Drugs in Anaesthesia Using a Minimal Parameterization and BIS Measurements [9], which is part of the PhD thesis of Dr. Margarida Silva. In this paper, a simpler model (than the commonly used compartment models) for anesthesia, modeling the effect of two interacting drugs on the DoA in the patient under surgery, is proposed. This minimally parametrized pharmacokinetic-pharmacodynamic (PKPD) model and the accompanying Matlab source code with the method are available and are used to model patient data from two clinical setups. The data were collected at the University of Brescia (Italy) and University of Porto (Portugal).

Goal

The aim is to mathematically model the effect of the hypnotic/amnestic agent propofol, and the analgesic drug remifentanil, that are intravenously administered in manual general anesthesia. The effect of the drugs is measured by BIS. The initial values and parameters of the model are estimated from the data. A method for parameter estimation in nonlinear statespace models, in particular the extended Kalman filter (EKF) is discussed. The EKF is the method used in the provided Matlab source code and will be applied to the available data sets from Brescia and Porto. So, a given estimation method is applied to (new) data. The algorithm and method do not have to be significantly changed, but the parameters of the EKF need tuning to the specific characteristics of the new clinical setup. An analysis of the parameter estimates and comparison of the modeling quality of the different settings is performed.

The following sections will cover the medical background, the theoretical grounds for the model, the extended Kalman filter algorithm, the model used and the clinical properties of the data, before the results of applying the modeling approach to data from the different clinical setups are discussed.
2 Medical background: anesthesia

In medicine, anesthesia is usually achieved by administering both an analgesic and a hypnotic drug. The former suppressing pain and the latter reducing consciousness. In our case the analgesic drug is propofol and the hypnotic drug is remifentanil.

The so called “depth of anesthesia” (DoA) is measured by the so called Bispectral Index Scale (BIS), which is obtained by an electroencephalogram (EEG). The BIS-measure is a percentage, ranging from 0 to 100%. At 100% the patient is completely awake and at 0% the patient is dead. It is important that the index is kept between certain values: a value that is too low results in unnecessary drug use, a longer recovery period for the patient and possibly severe side effects caused by an overdose. A value that is too high may cause a patient to regain consciousness or to not be totally pain free. The ideal DoA is between 40% and 60%. Having this in mind, the advantages of automatizing this procedure becomes apparent.

As described in Margardia M. Silva’s PhD thesis [7] anesthesia usually consists of three phases: 1) the induction phase, in which relatively high drug doses are given under a short period of time in order to quickly get the patient in a state ready for surgery, 2) a maintenance phase, in which this state is reinforced with relatively small drug doses, and 3) a recovery phase, in which the administration of the anesthetic drugs is ceased and sometimes a “counter drug” is given to neutralize the effect of these drugs if recovery seems to take too much time. The optimal start dose is given as a bolus injection and is usually estimated using individual patient properties such as weight and sex.

3 Theoretical background

3.1 Modelling of dynamical systems

For this section that covers a general theoretical background Modeling of dynamic systems by Lennart Ljung and Torkel Glad [4] was used.

3.1.1 Models

The point of modeling is to make it possible to study and understand parts of reality and link observations to patterns. A system is an object or collection of objects whose properties are of interest to study. For this system, research questions are formulated. A lot of these questions can be answered by experimentation, but in other cases this is not possible, either because it is too expensive, too dangerous, or simply because the system does not exist (yet). In those situations it is necessary to make a model of the system in order to answer the research questions. A model is a tool to making it possible to answer the research questions without having to do an experiment in “the real world”, for example on patients. Instead, experiments can be done on the model itself.

A distinction is made between mental models, verbal models, physical models and lastly mathematical models, which is the one this project is about. In mathematical models the relations between the quantities of the systems are typically described in the
form of differential equations. The mathematical model can be used to predict how the system would behave, either analytically, i.e. by analyzing the equations, or by means of simulation, i.e. a numerical experiment.

Even though simulation is safe and cheap, the quality of the model has to be good enough to be able to make accurate predictions, and answer the research questions. Apart from black box approaches, two different disciplines are at play when it comes to constructing mathematical models. The first one is the domain of expertise, i.e. the real life application area, for example anesthetics, and the second one is the knowledge of how to translate this domain knowledge into a valid mathematical model. The expert knowledge in our case is for example the known effects of propofol and remifentanil, their interaction properties, what dosages are commonly used, and so on. It is important to realize that every model has a limited domain of validity: the model is only able to make reliable predictions or claims under certain conditions. Because of these limitations models and simulations cannot totally replace experimentation.

Mathematical models can be divided into different categories based on the following properties:

- Deterministic (the same input always gives the same output) or Stochastic (there is an element of randomness).
- Dynamic (the variables of the system also depend on previous signals) or Static (the variables of the system only depend on current signals)
- Continuous time (time continuous signals) or Discrete time (measurements at specific time points, a.k.a. sampling)

A visual representation where the different logical units of a system are represented by blocks and where the interactions between the units are denoted by arrows between the blocks, is called a block diagram. Compartment models, i.e. models used for pharmacokinetics and -dynamics (section 3.2), make use of block diagrams.

3.1.2 Signals in models

Mathematical models often have both system parameters, which belong to the system, and design parameters, which belong to the method (e.g. a Kalman filter). It depends on the view of the researcher what parameters are of main interest, for example which ones are set to fixed values and which ones are estimated. A possible goal of simulating a model is to determine the best design parameters of a specific method on a specific system, or to estimate certain system parameters with (optimized) design parameters \(^1\).

The signals that the system produces are called the outputs and are sometimes of main interest. Other times, estimating parameters is of more interest. External signals influence other signals in the model but are not influenced themselves. If these external signals can be controlled, they are called input signals and

\(^1\)Marcus Björk
are denoted with \( u_i(t) \). If the external signals cannot be controlled, they are called disturbance signals.

Because of the ubiquity of computers in modern day research, essentially all signals are received, registered and processed digitally, which implies that the signals need to be sampled, i.e. measured or processed at specific times. If the time between consecutive measurements is fixed (i.e. the sampling is done uniformly) so that \( t_k = kT \), then \( T \) is called the sampling interval and \( 2\pi/T \) (in radians per second) the sampling frequency.

### 3.1.3 State and state-space models

For a dynamical model, the input values at a certain time step are not enough to calculate the output: for discrete models we also need to take the inputs at previous time steps into account, and for continuous models the derivatives. The state of the system at time \( t_0 \) is the vector with the values of all internal variables (i.e. the variables that are needed to completely describe the system’s state) at time step \( t_0 \). With these state variables and the current input, the output at the next time step can be calculated iteratively. The values of intermediate time steps or derivatives are stored in the state vector. A model using its state vector for these calculations is called a state-space model.

A common way of writing a dynamical model is as a system of first-order differential equations with state variables \( (x_1(t), ..., x_n(t)) \). Written out, the system then becomes:

\[
\begin{align*}
\dot{x}_1(t) &= f_1(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)) \\
\dot{x}_2(t) &= f_2(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)) \\
&\vdots \\
\dot{x}_n(t) &= f_n(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t))
\end{align*}
\]

in vector notation: \( \dot{x}(t) = f(x(t), u(t)) \),

which has \( n \) internal variables and \( m \) input variables.

The outputs can then be calculated by the internal variables and inputs:

\[
\begin{align*}
y_1(t) &= h_1(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)) \\
y_2(t) &= h_2(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)) \\
&\vdots \\
y_n(t) &= h_n(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t))
\end{align*}
\]

in vector notation: \( y(t) = h(x(t), u(t)) \).

So in general the relations are as follows:

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
y(t) &= h(x(t), u(t))
\end{align*}
\]

which is a nonlinear state-space model.
If \( f(x, u) \) and \( h(x, u) \) are linear functions of \( x \) and \( u \), i.e., if:

\[
\begin{align*}
f(x, u) &= Ax + Bu \\
h(x, u) &= Cx + Du
\end{align*}
\]  

the model is called linear.

### 3.1.4 Stationary points and linearization

A solution \( x(t) \), for \( t \geq t_0 \), to the state model in (1) for a specific input function \( u(t) \) and initial conditions \( x_0 \) (at \( t = t_0 \)) is called a trajectory.

In some situations, the input \( u(t) \) is constant: \( u(t) \equiv u_0 \). The solution(s) (if there exist any) to

\[
\dot{x}(t) = f(x(t), u_0) = 0
\]

are denoted \( x_i^* \) (where \( i \) is just a numbering) and are called stationary solutions and \((x_0^*, u_0), (x_1^*, u_0), \ldots\) are called stationary points. The intuition behind it is that if \( \dot{x} = 0 \), there is no change and the system will stay in the same state as the initial state: the trajectory is just a constant. The output of the system will then also be constant. If trajectories that start close to the stationary point eventually end up in the stationary point, this point is called asymptotically stable. If all trajectories converge to this stationary point eventually, it is called globally asymptotically stable.

Say there is an asymptotically stable stationary solution \( x_0 \) for an input \( u_0 \). Then \( x_0 \) is dependent on \( u_0 \) and therefore also the stationary output \( y_0 \) is:

\[
y_0 = h(x_0(u_0), u_0) = g(u_0)
\]

This expression describes the stationary relationship between the constant input and corresponding stationary output. For a linear system, the time it takes for the output to get close to its new stationary output \( y_1 \) after changing \( u_0 \) to \( u_1 \) is called the time constant.

A nonlinear system can be linearized in the neighborhood of stationary solutions. For the stationary point \((x_0, y_0)\), the system then becomes:

\[
\begin{align*}
\Delta \dot{x}(t) &= A\Delta x(t) + B\Delta u(t) \\
\Delta y(t) &= C\Delta x(t) + D\Delta u(t)
\end{align*}
\]

Where \( \Delta x(t) = x(t) - x_0, \Delta u(t) = u(t) - u_0, \Delta y(t) = y(t) - y_0 \) and where \( A \) and \( B \) are the Jacobians of \( f(x, u) \) (with respect to \( x \) and \( u \) respectively) and \( C \) and \( D \) are the
jacobians of $h(x, u)$ (with respect to $x$ and $u$ respectively), evaluated at $(x_0, u_0)$. It is important to keep in mind that the linearization is only useful for points close to the stationary point, and that it is usually unclear how accurate this linear approximation actually is, although the error can be estimated. Therefore conclusions based on this approximation should be taken with care and should take the estimated error into account.

### 3.1.5 Disturbances

As explained in 3.1.2, disturbance signals are external signals that cannot be controlled. Yet these can have a significant impact on the behavior of the system, and therefore they are important to consider. It matters if the disturbance signals are known, and in that case whether they can be separately measured, or unknown.

An example of a measurable disturbance signal is that of a model of a solar-heated house, where the solar intensity (disturbance signal) together with a pump velocity (input signal) is used to model the storage temperature (output). Essentially, the input signal and disturbance signal are treated the same in the model, which leads to the following alterations of the equations in (1) presented in section 3.1.3.

\[
\dot{x}(t) = f(x(t), u(t), e(t)) \\
y(t) = h(x(t), u(t), e(t))
\]

A challenge is to describe this disturbance signal in a suitable way in order to be able to incorporate it in the model. However, an advantage is that a model can be built for the disturbance signal, separate from the dynamic model and based on direct measurements.

An example of a nonmeasurable disturbance signal is that of a model of the motion of an airplane. The forces of the air that are exerted on the plane depend not only on rudder and flap movements (inputs), but also on the wind (disturbance signal). A major difference with the previous example is that the disturbance signals are barely separately measurable: they can typically only be observed indirectly, through their effect on other measurable variables. Therefore it follows that the modeling of the disturbance signal cannot be done separately.

The more common case is when the disturbance signal is completely unknown. The disturbance signal is then usually added to the output signal in the following way:

\[
y(t) = z(t) + e(t)
\]

where $z(t)$ is the undisturbed signal and:

\[
\dot{x}(t) = f(x(t), u(t)) \\
z(t) = h(x(t), u(t))
\]

The problem of describing the disturbance signal in a suitable way remains, but $e(t)$ is now not directly measurable. Input and output signals must be used to infer $e(t)$. 

8
3.1.6 Transfer functions

This subsection is based on “Feedback systems: an introduction for scientists and engineers” [6] and [4].

A transfer function is a compact description of the input/output relation for a linear system (as described in equation (2)). It is the main tool for implementing this focus on input/output characteristics and makes it possible to understand the overall behavior of the system. The strength of the transfer function is that it provides a very convenient representation in manipulating and analyzing complex linear feedback systems. The transfer function is the Laplace transform of the output divided by the Laplace transform of the input, as explained below. ²

Consider the linear system:

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]  

(3)

The Laplace transform of the state equation equals ³:

\[
sX(s) - x_{t=0} = AX(s) + BU(s)
\]

If the initial state is chosen as \(x_{t=0} = 0\), this reduces to:

\[
X(s) = (sI - A)^{-1}BU(s)
\]

The Laplace transform of the output equation gives:

\[
Y(s) = CX(s) + DU(s)
\]

\[
\iff
\]

\[
Y(s) = C(sI - A)^{-1}BU(s) + DU(s)
\]

\[
\iff
\]

\[
Y(s) = (C(sI - A)^{-1}B + D)U(s)
\]

The Laplace transformation of (3) then gives:

\[
\begin{align*}
X(s) &= (sI - A)^{-1}BU(s) \\
Y(s) &= (C(sI - A)^{-1}B + D)U(s)
\end{align*}
\]  

(4)

²“Feedback systems” has the approach that the transfer function represents the response of the system to an exponential input: \(u = e^{zt}\) (where \(z\) is a complex number: \(s = \sigma + i\omega\)). This can be appropriate since many signals can be represented as exponentials, a sum of exponentials or by a linear combination of those. However, for this project no such assumptions about \(u(t)\) are made.

³following http://www.egr.msu.edu/classes/me851/jchoi/lecture/Lect4.pdf based on [1]
The transfer function from \( u \) to \( y \) is then the map from the input to the output:

\[
G_{yu}(s) = C(sI - A)^{-1}B + D
\]  

(5)

This mapping can be written as:

\[
Y(s) = G_{yu}(s)U(s)
\]  

(6)

An important property of the transfer function is that it is invariant to changes of the coordinates in the state space.

A transfer function is determined in the following way. Consider the linear SISO system (see section 3.2.1 for more details):

\[
d^n y \frac{d^n}{dt^n} + a_1 d^{n-1} y \frac{d^{n-1}}{dt^{n-1}} + \ldots + a_n y = b_0 \frac{d^m}{dt^m} + b_1 \frac{d^{m-1}}{dt^{m-1}} + \ldots + b_m u
\]  

(7)

where \( u \) and \( y \) as usual are input and output, respectively. Since the system is linear and differentiation in the time domain corresponds to multiplication by \( s \) of the Laplace transform, (7) can be written as:

\[
(s^n + a_1 s^{n-1} + \ldots + a_n) Y(s) = (b_0 s^m + b_1 s^{m-1} + \ldots + b_m) U(s)
\]  

(8)

This is abbreviated as:

\[
a(s) Y(s) = b(s) U(s)
\]

where \( a(s) \) is the characteristic polynomial of the ordinary differential equation. This gives:

\[
Y(s) = \frac{b(s)}{a(s)} U(s)
\]

and therefore the transfer function is:

\[
G(s) = \frac{b(s)}{a(s)}
\]  

(9)

The order of \( G(s) \) is the order of the denominator polynomial \( a(s) \). If the degree of the polynomial \( b(s) \) is less than or equal to the degree of \( a(s) \) then the model is called proper. If the degree of \( b(s) \) is strictly less than the degree of \( a(s) \), the model is called strictly proper. [3]

For a state space system with transfer function (9), the roots of the polynomial \( a(s) \) are called the poles of the system and the roots of \( b(s) \) the zeros. The poles correspond to different resonances with different dampings (modes). These modes are
then excited to different degrees for different inputs. A zero blocks the transmission of the corresponding input signals (because $Y(s) = G(s)U(s) = \frac{b(s)}{a(s)} U(s)$ assuming that $a(s) \neq 0$).

Some characteristics of the transfer function, like the gain and the location of the poles and zeros, determine the system completely.

The zero frequency gain (or static gain) is the magnitude of the transfer function at $s = 0$ and represents the ratio of the steady-state value of the output with respect to a step input. The system (7), assuming a constant input $u_0$ and output $y_0$, then becomes $a_n y_0 = b_m u_0$ which leads to the transfer function $G(0) = \frac{b_m}{a_n}$.

Consider now the transfer function $G(s) = C(sI - A)^{-1}B + D$. Its poles are the eigenvalues of $A$ in the state space model, because these points are where the characteristic polynomial $\lambda(s) = det(sI - A)$ equals 0 and hence $sI - A$ is noninvertible (singular). Hence the poles of a state space system only depend on matrix $A$.

Zeros are the complex numbers $s$ such that $U(s)$ can give output 0. Applying this property to the Laplace transformations of equation (3) yields:

$$s X(s) = AX(s) + BU(s)$$
$$0 = CX(s) + DU(s)$$

which can be written as:

$$\begin{pmatrix} A - sI & B \\ C & D \end{pmatrix} \begin{pmatrix} X(s) \\ U(s) \end{pmatrix} = \bar{0}$$

The zeros are the values $s$ such that

$$\begin{pmatrix} A - sI & B \\ C & D \end{pmatrix}$$

loses rank, which implies that a system where the matrix $B$ or $C$ is square and full rank does not have zeros. In practice this means that a system has no zeros if every state can be controlled independently or if the full state is measured.

A block diagram (section 3.1.1) of a system can be used to derive a transfer function for a larger part of the system by composing the transfer functions of the different blocks with block diagram algebra. Two transfer functions that are in series:

Figure 1: Two transfer functions in series

have as transfer function $G = G_2 G_1$

\[^4\text{Marcus Bjöck}\]
Two transfer functions that are in parallel:

![Parallel Transfer Functions Diagram](image)

Figure 2: Two transfer functions in parallel

have transfer function \( G = G_1 + G_2 \)

The feedback connection:

![Feedback Connection Diagram](image)

Figure 3: Two transfer functions in a feedback connection

has a combined transfer function \( G = \frac{G_1}{1 + G_1 G_2} \). This follows from:

\[
y = G_1 e = G_1 (u - G_2 y) \iff (1 + G_1 G_2) y = G_1 u \iff y = \frac{G_1}{1 + G_1 G_2} u
\]

The general idea is that the transfer functions of these simple combinations can be the building blocks of more complex systems and in their turn can used to derive more complex transfer functions.

### 3.1.7 PID controller

A PID controller has a proportional, integral and derivative part. The goal of the controller is to minimize the error between the current measured quantity and the reference signal. For this, it uses the course of this error through time, the error function \( e(t) \), and combines at every time step the current error and the integral and derivative of the error function to produce the output (which typically is connected to some action). The significance of the different parts is set by the different gains, \( K_p, K_i \) and \( K_d \) respectively. One or more gains can be 0, usually not the proportional part, resulting in a P, PI or PD controller, depending on what parts are remaining. Figure 4 visualizes this process.
It can be understood as that the proportional part takes into account the error of the present, the integral part the error of the past and the derivative part the error of the future. The output of the PID controller is mathematically expressed as:

\[ u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \]

3.2 Pharmacokinetics and Pharmacodynamics

For this part “Concepts in clinical pharmacokinetics” [2] was used. Pharmacokinetics (PK) is the study of the time course of drug absorption, distribution, metabolism, and excretion, i.e. how drug concentrations behave in different parts of the human body. The effect of a drug is usually related to its concentration at the site of action (i.e. the part of the body where it should have its effect) and therefore concentration measurements in these parts are useful. However, measuring drug concentrations at these sites is not practical and is therefore done indirectly, most often by measuring drug concentrations in blood plasma. An important assumption is that the concentrations in blood plasma are directly (and proportionally) related to the concentrations in the site of action. This property is called kinetic homogeneity.

Pharmacodynamics (PD) is about the relation between the drug concentration at the site of action and the resulting effect. It is assumed that the concentration at the site of action is the main factor that determines the intensity of the effect, but other (cellular biological) factors can also have an impact. The effectiveness of a drug, the drug potency, can be expressed in terms of the 50% effective concentration or \( EC_{50} \), which is the concentration that is needed to reach half of the maximum effect, \( E_{max} \), of the drug. A lower \( EC_{50} \) indicates a more potent drug. It should be noted that \( EC_{50} \) does not indicate other characteristics, such as the duration of the effect. An important phenomenon that should be dealt with is that the effect of some drugs can decrease over time: the patient develops a tolerance for the drug. Using the \( EC_{50} \) in a Hill function is a common way to model pharmacodynamics (see section 4.2.2).

The relationship between PK and PD is illustrated in figure 5:
The monitoring of drug concentrations has as its goal to keep a drug concentration within its *therapeutic range*: above a concentration that has an effect and below a concentration that is (possibly) toxic. The boundaries of the therapeutic range are usually expressed as a unit of *mass per volume*. Of course these boundaries are drug dependent and there is a variability in drug response between patients. Apart from patient dependent factors also drug interactions can play a role in the concentrations in the blood plasma. The task of a clinician is to select a drug, determine its dose, administer it, measure the patient’s response and the concentration of the drug in the blood plasma and adjust the dose and administration accordingly.

The behavior of drugs in the human body is very complex and therefore simplifications are needed. Commonly *compartmental models* are used, where a compartment represents a collection of organs/tissues that roughly behave as a unit. In some cases one compartment suffices, treating the whole human body as a unit, with one input (the administration of the drug) and one output (the breaking down of the drug). In other cases a two-compartment model is needed, representing the human body with as a central unit (the blood stream and highly perfused organs) that interacts with peripheral unit (e.g. fat and muscle tissue and cerebrospinal fluid). Administration and elimination takes place in the central unit. The interaction between compartments is described with *rates*. Usually it is assumed that the elimination rate is constant. A compartment model that has more than two compartments is called a *multi-compartment model*.

An example of a multi-compartment model is the very common PK model shown in figure 6.
Instead of just treating the human body as a single unit on which drugs have a direct effect, or as a central unit and a peripheral unit, it is modeled by a compartment that has an inflow and outflow of the drug and that is interacting with three other compartments, of which one is the effect compartment. The effect compartment is the unit that represents the site of action.

3.2.1 Wiener models

Pharmacokinetics and pharmacodynamics are sometimes also modeled with a Wiener model. The basic structure of a Wiener model is shown in figure 7. As in the case with a compartment model, a Wiener model can be written as a number of linear differential equations, but without the physical interpretation. Also, the pharmacodynamic part contains a nonlinearity, that in the Wiener model is represented by the nonlinear block.\textsuperscript{5}

![Wiener model diagram](image)

Figure 7: Single input single output (SISO) Wiener model

An input signal $u(t)$ is forwarded to the linear block which outputs a signal $z$ that is forwarded to the nonlinear block giving $y(t)$. The above model is a single input, single output model, but it can be expanded to a multiple input, single output (MISO) model (or even a multiple input, multiple output model if that is desired). In a MISO Wiener model the linear block consists of multiple units, as shown in figure 8.

\textsuperscript{5}Marcus Bjöck
Figure 8: Multiple input single output (MISO) Wiener model

The input signals $u_1(t), \ldots, u_n(t)$ are forwarded to their corresponding linear block units. These units output the signals $z_1, \ldots, z_n$ respectively, which will be combined into one signal: $z = C(z_1, \ldots, z_n)$. The function $C$ can for example be a weighted sum of the signals $z_1, \ldots, z_n$. The signal $z$ is then the input for the nonlinear block that in its turn outputs $y(t)$.

### 3.3 The Kalman filter

The Kalman filter is an iterative mathematical method that is used for both online and offline state estimation. It combines a prediction (estimate) with measurements and averages them, depending on the estimated error in the prediction and the estimated error of the measurements in a particular way, producing an updated estimate. Hence, the Kalman filter is usable in models with noise. The Kalman filter is the best linear filter and gives an optimal combination of previous estimates and measurements [5]. Since the data will be sampled, the Kalman filter will be explained for the discrete time case.

Essentially, at every iteration the Kalman gain, current estimate and the new uncertainty in the estimate are calculated 6. The following equation gives an intuitive explanation of the process:

$$\text{Estimate}(t) = \text{Estimate}(t-1) + \text{KalmanGain} \times (\text{Measurements} - \text{Estimate}(t-1))$$

(11)

The Kalman gain is a measure for assigning more weight to the estimate or to the measurements. If the error in the estimate is bigger, the measurements should have more impact, if the error in the measurements is bigger, the estimate should, as achieved by:

$$\text{KalmanGain} = \frac{\epsilon_{\text{estimate}}}{\epsilon_{\text{estimate}} + \epsilon_{\text{measurements}}}$$

(12)

---

6Special Topics - The Kalman Filter (7 of 55) The Multi-Dimension Model 1 [https://www.youtube.com/watch?v=cD7WkbAIL0](https://www.youtube.com/watch?v=cD7WkbAIL0)
We can visualize the role of the Kalman filter in the diagram in figure 9.

\[ \Pi x(t) \]
\[ KF \]
\[ \hat{x}(t+1) \]

**Figure 9: Schematic representation of the Kalman filter**

It shows that there is a process \( \Pi \) with a state \( x \), which changes with time \( t \): \( \Pi x(t) \). This process produces at a specific time \( t \), for giving inputs \( u(t) \), and distorted by the process error \( v(t) \), an output \( y(t) \). The Kalman filter \( KF \) uses the input plus the output of the process distorted by measurement noise \( e(t) \) and estimates the state of the dynamic system at time \( t : \hat{x}(t) \). It can be seen as the Kalman filter gradually filtering out the uncertainty of the process and the noise of the measurement. That is why it is called a filter.

In equation form the model becomes:

\[ x(t + 1) = Ax(t) + Bu(t) + v(t) \]  \hspace{1cm} (13)
\[ y(t) = Cx(t) + e(t) \]  \hspace{1cm} (14)

where \( x(t + 1) \) is the new state vector, \( x(t) \) the current, \( y(t) \) the output, \( u(t) \) is the input vector, \( v(t) \) process noise and \( e(t) \) measurement noise. \( A, B \) and \( C \) are matrices (recall the linear model of section 3.1.3, \( D \) is set to zero).

The dynamical system starts with initial values \( \hat{x}(0) = \hat{x}_0 \). The process error \( v(t) \) and measurement error \( e(t) \) have an expected value of 0:

17
\[ E\{v(t)\} = 0 \]
\[ E\{e(t)\} = 0 \]
and covariance matrices \( R_1 \) and \( R_2 \), respectively. The difference between the estimate and real state can be proven to have an expected value of 0:
\[ E\{\hat{x} - x\} = 0 \]

The covariance matrix of the error of the estimate is called \( P(t) \) and is estimated with the matrix \( A \) and the covariance matrix of the process error \( R_1 \) by:
\[ \hat{P}(t) = A^T P(t - 1) A + R_1 \quad (15) \]
This covariance matrix is then used together with the covariance matrix of the measurement error \( (R_2) \) to get to the Kalman gain \( (K) \):
\[ K(t) = P(t) C^T \left[ CP(t)C^T + R_2 \right]^{-1} \quad (16) \]
If the variance of the estimation error \( (P(t)) \) is small compared to the variance of the measurement error \( (R_2) \), which means that (assuming a bias of 0) the estimate is close to the true value, the value of \( K \) is small. On the other hand, if \( P(t) \) is big compared to \( R_2 \), it means the estimate is not so close to the true value and we want to adjust the estimate more crudely, which is reflected in a bigger value for \( K \). The estimate of the next state is then as follows:
\[ \hat{x}(t + 1) = A\hat{x}(t) + Bu(t) + K(t) \left[ C\hat{x}(t) - y(t) \right] \quad (17) \]
where \( C\hat{x}(t) - y(t) = \hat{y}(t) - y(t) \) is the difference between the estimated output and the measured output, which, depending on \( K \), is given more or less weight for the estimate of \( x(t + 1) \). After this, \( P(t) \) is updated with:
\[ P(t) = (I - KC)\hat{P}(t) \quad (18) \]
and is used for the next iteration.

### 3.4 The extended Kalman filter

The extended Kalman filter is used in situations where the system is nonlinear. The functions of the state vector \( f \) and the functions of the input vector \( h \) could be any differentiable functions:
\[ x(t + 1) = f(x(t), u(t)) + v(t) \quad (19) \]
\[ y(t) = h(x(t)) + e(t) \quad (20) \]
In the linear case, the Kalman filter has been proven to gradually improve the state estimate until stationary conditions are reached. In the extended Kalman filter there is no such guarantee, but we assume that if we repeatedly linearize the system around the current working point and use the standard Kalman filter, this would be good enough.

Torsten Söderström [10](p 245) initially uses the equations:

\[ \begin{align*}
    x(t+1) &= f(t, x(t)) + g(t, x(t))v(t) \\
    y(t) &= h(t, x(t)) + e(t)
\end{align*} \tag{21} \tag{22} \]

As T. Söderström remarks, this equation can be extended with an input signal \( u(t) \):

\[ \begin{align*}
    x(t+1) &= f(t, x(t), u(t)) + g(t, x(t))v(t)
\end{align*} \tag{23} \]

We now want to locally approximate the system with a linearization, or more precisely: compute the filter gain by linearizing the nonlinear model. We therefore substitute \( f, g \) and \( h \) with an expansion of a first order (i.e. linear) Taylor series around estimates of \( x(t) \). For \( f \) and \( g \) this is around the most recent estimate: \( \hat{x}(t|t) \). For \( h \) this is around the predicted state: \( \hat{x}(t|t-1) \), because the measurement \( y(t) \) has to be compared with its predicted value to obtain a correction term to the filter. [10](p 246)

\[ \begin{align*}
    f(t, x(t)) &\approx f(t, \hat{x}(t|t)) + F(t)(x(t) - \hat{x}(t|t)) \\
    g(t, x(t)) &\approx G(t) \\
    h(t, x(t)) &\approx h(t, \hat{x}(t|t-1) + H(t)(x(t) - \hat{x}(t|t-1)),
\end{align*} \]

where:

\[ \begin{align*}
    F(t) &= \frac{\partial f(t, x)}{\partial x}|_{x=\hat{x}(t|t)} \\
    G(t) &= g(t, x)|_{x=\hat{x}(t|t)} \\
    H(t) &= \frac{\partial h(t, x)}{\partial x}|_{x=\hat{x}(t|t-1)}
\end{align*} \]

In words, this means that \( f \) is approximated by evaluating \( f \) at the estimate \( \hat{x} \) and adding the difference between the true \( x \) and the estimate multiplied by a Jacobi matrix. The smaller the difference, the smaller the effect. Note that \( f \) and \( h \) are nonlinear functions that need to be linearized by their respective Jacobi matrices, while the noise \( v(t) \) is a \textit{linear} function of \( g \), so it can be estimated by just plugging in the estimated \( x \) in the function \( g \).

This gives rise to the following linear system:
\[ x(t + 1) = F(t)x(t) + G(t)v(t) + \tilde{u} \]
\[ y(t) = H(t)x(t) + e(t) + w(t) \]
\[ \tilde{u}(t) = f(t, \hat{x}(t|t) - F(t)\hat{x}(t|t)) \]
\[ w(t) = h(t, \hat{x}(t|t-1)) - H(t)\hat{x}(t|t-1) \]

(24)

In words, \( \tilde{u} \) is the linearization error of \( F \) and \( w(t) \) the linearization error of \( H \). The next state is then approximated by the previous state multiplied by a Jacobi matrix, plus the process error, plus the linearization error. The output \( y(t) \) is then described in terms of a current state multiplied by a Jacobi matrix and two additional terms: the measure error and the linearization error. Note that \( \tilde{u}(t) \) has nothing to do with \( u(t) \) (the input vector). The resemblance is just a matter of notation.

By applying the Kalman filter, the following formulas are obtained:

Using
\[ \hat{x}(t|t) = \hat{x}(t|t-1) + K(t) [y(t) - H(t)\hat{x}(t|t-1) - w(t)] \]
and
\[ w(t) = h(t, \hat{x}(t|t-1)) - H(t)\hat{x}(t|t-1) \]
we get:

\[ \hat{x}(t|t) = \hat{x}(t|t-1) + K(t) [y(t) - h(t, \hat{x}(t|t-1))] \]  
(Comp. simple Kalman (17): \( \hat{x}(t + 1) = A\hat{x}(t) + Bu(t) + K(t)(C\hat{x}(t) - y(t)) \))

\[ \hat{x}(t + 1|t) = F(t) \hat{x}(t|t) + \tilde{u}(t) = f(t, \hat{x}(t|t), u(t)) \]  
(26)
(Note the inclusion of the input \( u(t) \))

\[ K(t) = P(t|t-1) H^T(t) [H(t)P(t|t-1)H^T(t) + R_2(t)]^{-1} \]  
(27)
(Compare simple Kalman (16): \( K(t) = P(t) C^T [C P(t) C^T + R_2]^{-1} \))

\[ P(t|t) = P(t|t-1) - K(t) H(t) P(t|t-1) \]  
(28)
(Compare simple Kalman (18): \( P(t) = (I - K C) \hat{P}(t) \))

\[ P(t + 1|t) = F(t) P(t|t) F^T(t) + G(t) R_1 G^T(t) \]  
(29)
(Compare simple Kalman (15): \( \hat{P}(t) = A^T P(t-1) A + R_1 \))
4 Used model of effect of drug concentrations on BIS

4.1 General properties

M.M. Silva et al. [9] use a dynamical, stochastic, discrete time model with input signals (see section 3.1.1) for the anesthesia of a patient:

\[ x(t + 1) = f(t, x(t), u(t)) + g(t, x(t))v(t) \]

(equation 23 repeated for convenience)

A linear function for the input is used, which turns the above equation into:

\[ x(t + 1) = f(t, x(t)) + Bu(t) + g(t, x(t))v(t) \] (30)

The estimate of \( x(t + 1) \) then becomes:

\[ \hat{x}(t + 1|t) = f(t, \hat{x}(t|t)) + Bu(t) \] (31)

Note that these above two equations are special cases of the equations (23) and (26) which is a consequence of the choice for a linear function for the input signal. The input signal is not (necessarily) constant.

The input \( u(t) \) are the infusion rates of the administered drugs propofol and remifentanil. The output value \( y(t) \) is the measured BIS. The disturbance signals (which are unknown) consist of the noise in the process (the process error: \( v(t) \)) and the noise in the measurement (measurement error: \( e(t) \)) (see 3.3 and 3.4). Input and output signals must be used to infer the measurement noise (recall 3.1.5).

4.2 The model in detail

The following MISO Wiener model is used:

![Figure 10: Multiple input single output (MISO) Wiener model](image)

21
There are two input signals, the infusion rates of both drugs, and one output signal, the BIS value.

### 4.2.1 Linear block

The linear block units are:

\[
\hat{X}_\text{prop}(s, \alpha) = \frac{k_1 k_2 k_3 \alpha^3}{(s + k_1 \alpha)(s + k_2 \alpha)(s + k_3 \alpha)} R^\text{prop}(s) \tag{32}
\]

\[
\hat{X}_\text{remi}(s, \eta) = \frac{l_1 l_2 l_3 \eta^3}{(s + l_1 \eta)(s + l_2 \eta)(s + l_3 \eta)} R^\text{remi}(s) \tag{33}
\]

where \(\hat{X}_\text{prop}(s, \alpha)\) is the Laplace transform of the output from the model linear dynamic part for propofol \(\hat{x}_\text{prop}(t, \alpha)\) and \(R^\text{prop}(s)\) is the Laplace transform of the input signal \(r^\text{prop}(s)\), which is the concentration of the administered drug. And similarly for remifentanil. The parameters \(k_1, k_2, k_3, l_1, l_2,\) and \(l_3\) are set to fixed values. This means the pole locations \(-k_1 \alpha, -k_2 \alpha\) and \(-k_3 \alpha\) of (32) and \(-l_1 \eta, -l_2 \eta\) and \(-l_3 \eta\) of (33) depend respectively on \(\alpha\) and \(\eta\), which are chosen as the unknown parameters.

### 4.2.2 Nonlinear block

In the article, the input for the nonlinear block is defined as the weighted sum

\[
\hat{z}(t) = \hat{U}_\text{remi}(t, \eta) + m \hat{U}_\text{prop}(t, \alpha) \tag{34}
\]

where

\[
\hat{U}_\text{prop}(t, \alpha) = \frac{\hat{x}_\text{prop}(t, \alpha)}{EC_{50}^\text{prop}} \tag{35}
\]

\[
\hat{U}_\text{remi}(t, \eta) = \frac{\hat{x}_\text{remi}(t, \eta)}{EC_{50}^\text{remi}} \tag{36}
\]

and where \(x_c^\text{remi}(t)\) and \(x_c^\text{prop}(t)\) are the effect concentrations, which are normalized with respect to their concentration at half the maximal effect (\(EC_{50}^\text{remi}\) and \(EC_{50}^\text{prop}\)). These normalization constants are fixed and specific to a drug (see section 3.2 for drug potency).

The following equation for the output of the nonlinear block was derived from the generalized Hill equation:

\[
\hat{y}(t) = \frac{y_0}{(1 + (\hat{U}_\text{remi}(t, \eta) + m \hat{U}_\text{prop}(t, \alpha))^\gamma} \tag{37}
\]

Here \(\hat{y}(t)\) is the BIS-estimate at time \(t\) and \(y_0\) is the initial BIS-value (i.e. the value corresponding to the normal awake condition, before the administration of the drugs). In addition to \(\alpha\) and \(\eta\), \(m\) and \(\gamma\) are the parameters chosen to be unknown.
4.2.3 State-space model

The transfer function

\[
\hat{X}_{\text{prop}}(s, \alpha) = \frac{k_1k_2k_3\alpha^3}{s^3 + (k_1 + k_2 + k_3)\alpha s^2 + (k_1k_2 + k_2k_3 + k_1k_3)\alpha^2 s + k_1k_2k_3\alpha^3} R_{\text{prop}}(s)
\]

(38)

which is a strictly proper transfer function since the degree of the numerator of the fraction is smaller than the denominator (recall section 3.1.6), was converted to a state space representation in controllable canonical form\(^7\):

\[
\dot{\hat{x}}_{\text{prop}}(t) = A_{\text{prop}} \hat{x}(t) + B_{\text{prop}} u(t)
\]

\[
\dot{\hat{x}}_{\text{prop}}(t) = \begin{pmatrix}
0 & 1 & 0 & \cdots \\
0 & 0 & 1 & \cdots \\
-k_1k_2k_3\alpha^3 & -(k_1k_2 + k_2k_3 + k_1k_3)\alpha^2 & -(k_1 + k_2 + k_3)\alpha & \cdots \\
0 & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots \\
\end{pmatrix} \hat{x}(t) + \begin{pmatrix}
0 \\
0 \\
1 \\
0 \\
1 \\
1 \\
\end{pmatrix} u(t),
\]

(39)

and the output:

\[
L(t) = \tilde{C} \hat{x}(t) = (k_1k_2k_3\alpha^3 \ 0 \ 0) x(t)
\]

(40)

The analogue holds for remifentanil and combining both results into a joint system matrix yields:

\[
A(\alpha, \eta) =
\begin{pmatrix}
0 & 1 & 0 & \cdots \\
0 & 0 & 1 & \cdots \\
-k_1k_2k_3\alpha^3 & -(k_1k_2 + k_2k_3 + k_1k_3)\alpha^2 & -(k_1 + k_2 + k_3)\alpha & \cdots \\
0 & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 \\
\cdots & 0 & 1 & 0 \\
\cdots & 0 & 0 & 1 \\
-l_1l_2l_3\eta^3 & -(l_1l_2 + l_2l_3 + l_1l_3)\eta^2 & -(l_1 + l_2 + l_3)\eta
\end{pmatrix}
\]

\(^7\text{https://en.wikipedia.org/wiki/State-space_representation}\)
and

\[ C' = \begin{pmatrix} k_1 k_2 k_3 \alpha^3 & 0 & 0 & l_1 l_2 l_3 \eta^3 & 0 & 0 \end{pmatrix} \]

By taking into account (34), (35) and (36) and choosing another ordering of the elements in the state vector, the vector used in the article is obtained:

\[ C(\theta) = \begin{pmatrix} 0 & 0 & m \frac{k_1 k_2 k_3 \alpha^3}{EC_{50}^{prop}} & 0 & 0 & \frac{l_1 l_2 l_3 \eta^3}{EC_{50}^{remi}} \end{pmatrix} \]

where

\[ \hat{z}(t) = C(\theta)\hat{x}(t) \]

Since we have two input signals, \( B^{prop} \) has to be composed with the corresponding matrix for remifentanil. Another choice of ordering then leads to the matrix used in the article:

\[ B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T \]

\( B \) has to be multiplied with \( u(t) \), which contains the infusion rates, which is called \( r(t) \) in the article and which contains both the rate of propofol and remifentanil:

\[ r(t) = \begin{pmatrix} r^{prop}(t) & r^{remi}(t) \end{pmatrix}^T \]

The above gives the continuous-time representation:

\[ \dot{x}(t) = A(\alpha, \eta)\dot{x}(t) + B r(t) \]
\[ \hat{z}(t) = C(\theta)\dot{x}(t) \tag{41} \]

Because of the dimensions of the matrices \( A, B \) and \( r(t) \), \( \dot{x} \) is a column vector with 6 entries, the first three related to the linear part of propofol, the last three of remifentanil. In order to be able to use the discrete time EKF, the model in 41 is discretized.

From 4.2.1 and 4.2.2 it follows that the model has four system parameters that are to be estimated:

\[ \theta = (\alpha \ \eta \ m \ \gamma)^T \]

The state space of the discretized model is augmented with these four parameters, which leads to the following ten state variables:

\[ \hat{x} = (\hat{x}_{prop1} \ \hat{x}_{prop2} \ \hat{x}_{prop3} \ \hat{x}_{remi1} \ \hat{x}_{remi2} \ \hat{x}_{remi3} \ \hat{\alpha} \ \hat{\eta} \ \hat{m} \ \hat{\gamma})^T \]

(remember that \( m = 1 + \beta \))
4.2.4 Assigned values to parameters

The following parameters have been assigned the following values:

- $k_1$ and $l_1$ are set to 1
- $k_2$ and $l_2$ are set to 2
- $k_3$ and $l_3$ are set to 3
- $y_0$ is set to 97.7 due to monitor restrictions.

The parameters $k_1$, $k_2$, $k_3$, $l_1$, $l_2$ and $l_3$ are determined over a large database of patient data. Also $EC_{50}^{prop}$ and $EC_{50}^{remi}$ are set to fixed values, but these values are not made explicit in [9] nor is their unit mentioned. This unit inconclusiveness gives rise to problems if these values are to be used on different data sets with different units. It is not mentioned in [9] to what database these parameters were set. Instead, a reference is made to [8] but it is not mentioned there either.

5 Testing the model on data

5.1 Available data

The data was provided by the universities of Brescia and Porto. The set-up of Porto made use of a PID controller (recall 3.1.7) while in the case of Brescia the drugs were administered manually by an anesthesiologist monitoring the patient (so also in closed-loop). Porto used a sample time of 5 s; Brescia used 1 s. Each file corresponds to a specific surgery of a patient and has (at least) three sets of numbers: the BIS-values, the infusion rates for propofol and the infusion rates for remifentanil. The available files were:

**Brescia** (sample time 1 s)
- $PZ41_1.mat$
  - The infusion rate of the drugs are in $ml/h$ and the volumes are in $ml$. The drug concentrations are 20 $mg/ml$ for propofol and 10 $μg/ml$ for remifentanil.
- $PAZ41new.mat$ (a slightly corrected file of $PZ41_1$, see section 5.4)
  - The infusion rates of the drugs are in $mg/s$ for propofol and $μg/s$ for remifentanil. The drug concentrations are 20 $mg/ml$ for propofol and 10 $μg/ml$ for remifentanil.

**Porto** (sample time 5 s)
- $marsh\_pat1.mat$ through $marsh\_pat7.mat$ (part of BIS Urology, old files)
  - The infusion rate units are unknown
• *schnider* _pat1.mat* through *schnider* _pat7.mat* (part of BIS Urology, old files)
  – The infusion rate units are unknown

• *Bis*_1.*mat* and *Bis*_2.*mat* (new files)
  – The infusion rates of the drugs are in mg/min

• *data* _HA_1.*mat* through *data* _HA_5.*mat* (for cross checking in section 5.5.1)
  – The infusion rates of the drugs are in mg/min

The model and implementation will be tested on PAZ41new and Bis_1.

5.2 A typical run on the old data: *marsh* _pat1*

In order to gain understanding of the process, the program with the implementation of the EKF that was provided was first run on data that it was said to work on for the chosen parameters at section 4.2.4. This data was *marsh* _pat1*. Here the other parameters were:

- \( P_0 = diag(100^2, 10^2, 1^2, 100^2, 10^2, 1^2, 10^2, 100^2, 100^2) \)
- \( R_1 = diag(10^2, 10^2, 0.1^2, 10^2, 10^2, 0.1^2, 10^2, 10^2, 30^2, 30^2) \)
- \( R_2 = 10^{10} \)
- \( EC_{50}^{\text{prop}} = 80 \)
- \( EC_{50}^{\text{remi}} = 65 \)

Figure 11 plots the infusion rates of propofol and remifentanil at every time step (as present in the data).

Figure 12 plots for every data point of infusion rates the output of the linear block unit of both propofol and remifentanil, normalized with their respective half effect “concentrations” (EC\(_{50}\)). This plot gives an indication of their respective contributions. The amplitude of the output of each linear block is in the same range as the amplitude of its input, because its static gain is one. Therefore, the output of each block should be, just as its input, in the vicinity of the corresponding half effect concentration (EC\(_{50}\)), because a much lower value would have no effect and a much higher value would indicate an overdose. In other words: it should be inside its therapeutic range (recall section 3.2). It follows that the output of both linear block units, normalized by the respective EC\(_{50}\)s, should roughly have an order of 1, to which the shown graph complies. The sample interval is 5 seconds.

Figure 13 plots the estimates of the unknown model parameters at every time step. All parameter estimates clearly change over time.
Figure 11: The infusion rates of propofol and remifentanil from marsh_pat1 plotted against time.

Figure 12: Outputs of the linear block of propofol (red) and remifentanil (blue) for every data point of marsh_pat1. The sample interval is 5 seconds.

Figure 14 plots the estimated BIS-value with the measured BIS-value at every time step. It can be seen that the estimated values follow the measured values, but do so quite roughly: the graph of the real BIS-values show bigger variations than the graph of the estimated values. This is the desired outcome. The curve of the BIS-estimates should show the underlying BIS-values, without being distorted by the measurement noise. So one wants the estimated BIS-values to follow the measurements, but not too closely: ideally the error shall not be correlated.

In order to be able to better evaluate the results, another was created where the difference between measured and estimated bis-value was plotted, which will be included in the following sections.
Figure 13: Estimates of the different parameters plotted against time (marsh.pat1)

Figure 14: The BIS-values (magenta dashed line) and BIS-estimates (blue line) of marsh.pat1 plotted against time.
5.3 Testing on new data from Porto: Bis_1

The program was run on new data from Porto, which was Bis_1. Here the other parameters were:

- \( P_0 = \text{diag}(10000 \ 10^2 \ 1^2 \ 100^2 \ 10^2 \ 1^2 \ 10^2 \ 1^2 \ 100 \ 10^2) \)
- \( R_1 = \text{diag}(100 \ 10^2 \ 0.1^2 \ 10^2 \ 10^2 \ 0.1^2 \ 10^2 \ 10^2 \ 400 \ 30^2) \)
- \( R_2 = 2 \cdot 10^9 \)
- \( EC_{50}^{\text{prop}} = 50 \)
- \( EC_{50}^{\text{remi}} = 30 \)

A problem that occurred was that the program returned complex values. This was indicated by Matlab but could also be suspected by the appearance of the first figure (figure 15), which is symptomatic for the emergence of complex values:

![Figure 15: Outputs of the linear block of Propofol (red) and Remifentanil (blue) for every data point of Bis1 giving rise to the suspicion that values had turned complex.](image)

Matlab’s debugger led to the iteration in the end fase of the execution where suddenly the calculation of the matrix \( H \) turned into a matrix with complex values. More exactly, when calling the function to calculate the jacobi matrix, all the entries suddenly became complex. A suggestion was to manually convert accidentally obtained complex vectors into real ones. The imaginary parts were very close to zero so the idea was that it would not make a difference when ignored. This did not work in the end, since even though \( H \) (see section 3.4) is forced to be real valued, the vector with all ten state estimates \( x[t] \) happens to become complex valued anyway.

---

8 according to Alexander Medvedev
9 From now on using \( \text{tf} = \text{isreal}(H) \) as a break point
10 by Alexander Medvedev
The following part of the bigger expression that is used to calculate the state estimates

\[(x_0 k_2 k_3 x_3^3 / C_{50prop} + I_2 I_3 x_8 x_6 / C_{50rem})^x_{10}\]

with its parameters plugged in:

\[(6 x_9 x_7 x_3 / 50 + 6 x_8 x_6 / 30)^x_{10}\]

(42)

turns complex. This happens when the base of the expression turns negative (a negative number to a rational power turns complex). The base can only become negative if the first terms gets negative and has bigger absolute value than the second (positive) term or if the second term gets negative and has a bigger absolute value than the first (positive) term, or if both terms are negative. It was discovered that \(x_3, x_7\) and \(x_8\) are positive throughout the running of the program, \(x_9\) turns negative in an iteration after the change to complex values, and most interestingly \(x_6\) gets negative long before the change to complex values and alternates between negative and positive values throughout the running of the program.

By inspecting the different state values during run time it was discovered that more \(x\)-values attain negative values from very early on. Only \(x_3, x_7\) and \(x_8\) were positive throughout the running. Surprisingly, this does not give rise to problems before the end phase, where the complex numbers originated.

The reason why some \(x\)-values become negative, could lie in the settings of the EKF (treated in section 5.3.2) or in the scaling of the data (treated in section 5.3.3). Before going into these topics, it is investigated if the problem can be solved with constraints.

5.3.1 Attempt to prevent the occurrence of complex values with constraints

Positivity constraint

The first approach is to put a constraint in the program that the state estimates can never be negative. This was done by altering the assignment of the state values from

\[x_1=x(1);\]

to

\[x_1= \max (0, x(1));.\]

and similarly for \(x_2\) through \(x_{10}\). Even though this measure prevented negative values and the first figure does not show behavior of complex values anymore, it resulted also in the warning

**Warning:** Matrix is singular to working precision.

which points to problems with inverting the matrix in the equation of determining the Kalman gain in equation (27). The implementation used the Matlab function `inv()` and
Figure 16: Outputs of the linear block of Propofol (red) and Remifentanil (blue) for the first twelve data point of Bis1 with a positivity constraint.

A suggestion\(^\text{11}\) was to use the function `pinv()` (pseudo inverse) instead. The use of the pseudo inverse in combination with the positivity constraint resulted in the following error message:

```
Error using svd
Input to SVD must not contain NaN or Inf.
Error in pinv (line 18) [U,S,V] = svd(A,'econ');
```

Apart from the singularity warning, the first figure (figure 16) also looks very different: instead of having two heavily oscillating graphs as in figure 12, it shows two smooth monotonically increasing curves and only plots the first 12 time steps. Furthermore, all the parameters seem to have been constant during these time steps as shown in figure 17. And finally, the output of the filter is not visible anymore in figure 18. Zooming in revealed that the bis-estimates were only plotted during the first minute, complying with twelve time steps with a sample time of five seconds.

The reason for these observations is that an entry of the state vector \((x_1)\) turns negative in iteration 12 and is then (by the constraint) set to zero. In following calculations this entry causes divisions by zero, resulting in NaN-entries. Subsequent calculations with these NaN-entries result in more NaN-entries. This explains the thrown warning and the incomplete graphs: a matrix with NaN entries cannot be inverted and a NaN as an output cannot be plotted.

\(^{11}\)by Marcus Björk
Figure 17: Parameter estimates for data points of the first 12 time steps of Bis1 with a positivity constraint. All parameters seem to have been constant throughout these time steps.

Figure 18: The BIS-values (magenta dashed line) of Bis1 with a positivity constraint plotted against time. The BIS-estimates (blue line) have vanished because the values were NaNs that were not plotted by Matlab. Only a blue dot in the upperleft corner can be observed.
Another constraint on the state estimates that was tried was the absolute value. This resulted in no warnings about complex values, nor warning about singular matrices. The figures of the parameter estimates, BIS-measurements and -estimates, and infusion rates had no detectable strange behavior, but in the figure of linear block contributions (figure 19) the curve of remifentanil (blue) has become very small (compare figure 12). Also, the plot in figure 20 of the difference between measured and estimated BIS values showed a discrepancy indicating that the filter is maybe not working properly.

In this case using the pseudo inverse did not result in error messages, but did not change the figures either.
5.3.2 Attempt with tuning

The extended Kalman filter needs to be tuned in order to give satisfying results. The tuning parameters are $P_0$, $R_1$ and $R_2$. $P_0$ is the initial estimate of the covariance matrix. $R_1$ corresponds to the process noise and $R_2$ to the measurement noise, which says something about how much one relies on the model or on the data. $R_1$ is a vector with values corresponding to the uncertainty of every state; $R_2$ is a scalar. The ratio between $R_2$ and a state’s entry in $R_1$ will determine how fast this state will change and consequently whether the estimate $\hat{y}$ follows the measurements poorly, smoothly or too closely. To some degree one should be able to affect a state individually with tuning.

The idea was that different settings of the filter might also solve the problem of arising negative $x$-values and thereby prevent the emergence of complex values.

Since the imaginary values appear so late, it is probably not helpful to alter the matrix with initial values $p_0$, since its effect decreases over time.

The way the new estimate is calculated:

$$\hat{x}_{t+1} = \hat{x}_t + K_t(y(t) - \hat{y}(t))$$

points to the conclusion that a too big number is subtracted and that therefore the Kalman gain is too strong. This is probably a consequence of that the variance of the noise of the measurements is bigger than expected. In order to reduce the effect, the
initial values of $R_2$ can be increased in equation (27) (repeated for convenience):

$$K(t) = P(t|t-1) H^T(t) [H(t)P(t|t-1)]H^T(t) + R_2(t)]^{-1}$$

First it was attempted to remediate the occurrence of negative concentrations by altering $R_2$. Significantly increasing or decreasing this value had no positive effect. Hereafter, a more directed approach was taken in the form of altering the entry in the diagonal $R_1$ matrix corresponding to the x-values that turned negative. Special attention was given to $x_6$ since that was the variable in the exponential expression (42) that turned negative first.

Tuning the different entries in $R_1$ only managed to delay the emergence of complex numbers, so altering $R_1$ and $R_2$ did not eliminate the problem of arising complex values. Also, negative values appeared frequently when the program was run on marsh_pat1 so this property does not appear to be of crucial importance.

Another approach was to take a closer look at the data that was fed into the program (Bis_1). With inspection, nothing anomalous could be found: the data showed an overall “smooth” behavior: there were no outliers among the data points. The values for the propofol concentrations were generally between 2 and 4, apart from the start dose, a few points where it was around 20, or sections where it was 0. But those points were not close to the point where complex values arose and moreover they were already occurring early on. However, comparing figure 21 to figure 11 reveals a discrepancy: the plots of propofol and remifentanil in figure 21 are not on the same scale, indicating there could be an issue with scaling.

Figure 21: The infusion rates of propofol and remifentanil from Bis_1 plotted against time.
5.3.3 Scaling

The next step was to take a look at the data (marsh_pat1 etc.) to which the algorithm was tuned. Inspecting this it showed there was a big difference in scale between the two data sets, the second having numbers in the range of multiple hundreds. This raises the question what units are used in the different data sets. It was tried to scale the data points from (Bis_1) into to the range of the data points from marsh_pat1. Roughly this meant multiplying the propofol concentrations by 100 and the remifentanil concentrations by 1000. This indeed prevented the occurrence of complex numbers: neither Matlab threw a warning, nor did the program stop at the conditional break point. More information about the exact use of units is needed to convert the numbers more precisely. In the paper the unit mg/ml was used, but the graphs carry the description ml/h. The unit question will be treated in section 5.5.

5.4 Testing on data from Brescia: PZ41_1

Inspection of the PZ41_1 data showed that it does not fit the structure of an induction phase and maintenance phase (as explained in section 2). The two data files of PZ41_1 that were inspected were Pompe.Propofol that contained infusion rates in ml/h and Pompe.Vol_prop that contained volumes in ml. The first one showed a very steady behavior: the infusion rate is around 20 ml/h most of the time: from just after the start to almost the end of the data set. Then, after a short transition, it is 10 for a period of time and then practically turns to zero. The second data file showed a steadily increasing number from 0 to 29. This data probably corresponds to the accumulated administered drug volume. The numbers seem to increase at a very steady pace, being in accordance with the former data. Throughout the data there are no outliers, so there do not seem to be an initial doses and there are no intermittent infusions.

This oddity was reported to the responsible person in Brescia and it turned out there was an error with the program that took care of the measurements: it did not save the drug infusions from the bolus mode (only from the maintenance mode). A new, reconstructed data set (PAZ41new) was received where the initial bolus was calculated with help of the volume data. However, no intermittent doses were present and the maintenance dose was of a different order of magnitude in comparison to the old data set.

5.5 Units

It turned out that the data was collected from three different hospitals with three different protocols and therefore different units. Information about the used units has been received from Brescia, but only a vague guess for the used units in the old Porto data was received. In order to determine in what unit the BIS Urology files are delivered, the approach was to, for the respective drugs, look up the recommended bolus doses (used in the induction phase) per kilogram body weight and their units, and with the
help of the patient information, in particular the patient’s weight, calculate the actual recommended initial dose. Since these numbers are quite standard, the data should resemble the calculated recommended initial dose. This recommended dose is converted to different units and these are compared to the induction doses of the data, in an attempt to infer if the unit is volume (ml) per time unit, or mass (mg or µg) per time unit and whether the time unit is second, minute or hour. In case of volume per time unit, even the concentration is needed (mg/ml or µg/ml) to make conversion between mass and volume possible.

5.5.1 Calculating recommended doses

Table 1 shows the found information with respect to the recommended propofol dosages.

Table 1: Recommended propofol dosages. “IVP q10sec” means “intravenous push every ten seconds” and “ASA III/IV” are categories by the American Society of Anesthesiology.

<table>
<thead>
<tr>
<th>Induction</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;55 years: 40 mg IVP q10sec until onset (2.25 mg/kg)</td>
<td>&lt;55 years old: 0.1-0.2 mg/kg/min IV</td>
</tr>
<tr>
<td>&gt;55 years or debilitated or ASA III/IV: 20 mg IVP q10sec until onset (1.15 mg/kg)</td>
<td>&gt;55 years old or debilitated or ASA III/IV: 0.05-0.1 mg/kg/min IV</td>
</tr>
</tbody>
</table>


The patient statistics comprise five fields: a number for the patient’s sex, the age, the weight, the height and a fifth number of which the meaning was unclear. Patient number 1 of the marsh study (marsh_pat1) had the numbers 1.0, 43.0, 63.0, 170.0, 51.72

which means they is under 55 years of age. This would mean that the induction onset is somewhere between 126 and 157.5 mg with an initial dosage of 240 mg/min. The rate during maintenance would then be 6.3 - 12.6 mg/min.
Table 2: Values for different dose types per data set

<table>
<thead>
<tr>
<th></th>
<th>Induction phase dose</th>
<th>Maintenance doses</th>
<th>Intermittent doses</th>
</tr>
</thead>
<tbody>
<tr>
<td>marsh_pat1 (prop_vel)</td>
<td>555.5556 (long)</td>
<td>166.9444 - 78.5833</td>
<td>585.5278 - 2318.2</td>
</tr>
<tr>
<td>Bis_1 (up)</td>
<td>179.1117 (short)</td>
<td>7.9783 - 1.4783</td>
<td>12.8017 - 22.1750</td>
</tr>
<tr>
<td>PZ41_1 (Pompe.Propofol)</td>
<td>? 22.10000</td>
<td>20.1000</td>
<td>none</td>
</tr>
<tr>
<td>PAZ41new (prop)</td>
<td>3.3330</td>
<td>0.1228 - 0.1006</td>
<td>none</td>
</tr>
<tr>
<td>data_HA_1</td>
<td>200-188.32 (very short)</td>
<td>19.3667 - 2.1000</td>
<td>21.2967</td>
</tr>
</tbody>
</table>

The Swedish medical products agency\textsuperscript{12} has similar recommendations for Propofol, only having a slightly lower bound for both induction phase and maintenance phase and slightly higher for intermittent doses:

- induction: 20-40 mg every tenth second until 1.5-2.5 mg/kg
- maintenance: 4-12 mg/kg/h
- intermittent: 25-50 mg,

The recommended induction dose of 120-240 mg/min and maintenance dose of 3.15/6.3-12.6 mg/min were converted to different units. The values of PAZ41new, PZ41_1, BIS_1 and dataHA_1 all fit the recommendations when converted to their stated units. However, the numbers of marsh_pat1 do not comply with the recommendations. The value 555 would fit best in the range of 360-720 ml/h (with a concentration of 20 mg/ml) for the induction phase, but then values of 78-166 are much too high for the recommended 9.45/18.9-37.8 ml/h for maintenance. If the unit would be ml/h for a concentration of 10 mg/ml then 555 would be significantly less than the recommended range of 720-1440 ml/h and values of 78-166 would be significantly more than 18.9/37.8-75.6. It has to be concluded that the proportion between induction and maintenance dose does not follow the recommendations. The maintenance dose should be about one twentieth of the induction dose for this patient’s weight. This proportion varies somewhat according to the patient’s weight, but even for heavier patients the induction and maintenance doses are too close to each other. All seven marsh_pat files showed the same pattern. Maybe this has something to do with the nature of the surgery, but this information can probably not be retrieved anymore and Margarida M. Silva recommended not to rely on this data. Therefore it is decided to use it only for illustrative purposes.

The patient of the Brescia data has the following statistics:

**Sex:** 1, **Age:** 28, **Weight:** 90, **Height:** 190, **Ts:** 1

which would imply that the total induction onset is somewhere between 180 and 225 mg with an initial dose of 240 mg/min. The maintenance dosage should then be 9-18 mg/min.

\textsuperscript{12}Låkemedelsverket
Since the data from Brescia is, according to personal communication, received in \( ml/h \), conversion to \( mg/ml \) requires to know the used concentrations. However, there is confusing information about the used concentrations. Personal communication stated that the concentrations were \( 20\, mg/ml \) for Propofol and \( 10\, \mu g/ml \) for Remifentanil, but in the data file, the statistics read \( \text{ConcProp: 20 and ConcRemi: 50} \).

Because of the lack of information about BIS Urology, it is decided not to use those files for calibrating the program or determining the used unit. The most likely scenario at this point is that the program is calibrated for input in \( mg/h \) (according to Margarida Silva), that there is something significantly abnormal about the BIS Urology files and that the label of the \( y \)-axis of the graph plotting the drugs doses, stating the unit is \( ml/h \) is a typo.

5.6 Running with conversion to assumed unit

Converting the data of PAZ41new (multiplying by 3600) and Bis_1 (multiplying by 60) to \( mg/h \) does not result in any complex numbers. And below follows the analysis of these rescaled data sets.

5.6.1 60Bis_1 (converted to mg/h)

The parameters \( R_1, R_2, P_0, EC_{50}^{\text{prop}}, EC_{50}^{\text{remi}} \) were initially as in section 5.3. Running the program on the converted data with these parameters led to figures 22-25. The blue curve in figure 22 shows that the normalized linear output of remifentanil is too small (it is not roughly of order 1 as explained in section 5.2). This will be dealt with in the next section (section 5.6.2).

Figure 23 shows the parameter estimates. Unlike figure 13, not all four parameter estimates vary (a lot): at least one seems to have been constant. Changing the corresponding entries in \( R_1 \) did not resolve this problem.

The curve of the BIS-estimates in figure 24 follows the BIS-measurements too closely, which means it does not show the underlying structure of the BIS-measurements but is influenced too much by the measurement noise. This can also been inferred from figure 25, where the error between the estimates and measurements are too close to zero. In an attempt to make the estimate less influenced by the measurement noise, the covariance matrices of the process error (\( R_1 \)) and measurement error (\( R_2 \)) as well as the initial covariance matrix of the error of the estimate (\( P_0 \)) were changed. The biggest impact seemed to result from changing \( R_2 \) from \( 2 \cdot 10^8 \) to \( 2 \cdot 10^{11} \).

A peculiar finding was that two different sets of values for \( P_0 \) produced similarly smooth estimates, as shown in figure 26 and figure 27, but resulted in very different estimations of \( m \): if \( P_0 \) is as in (43), \( m \) is estimated around 0.6 (figure 28) and if \( P_0 \) is as in (44), \( m \) is estimated around 1 (figure 29). (\( R_2 = 2 \cdot 10^{11} \) and \( R_1, EC_{50}^{\text{prop}}, \) and \( EC_{50}^{\text{remi}} \) as before.)
\[
P_0 = \text{diag} \left( 10^6 \ 10^4 \ 10^2 \ 10^6 \ 10^4 \ 10^2 \ 10^4 \ 10^6 \ 10^5 \right) \quad (43)
\]

\[
P_0 = \text{diag} \left( 10^4 \ 10^2 \ 10^0 \ 10^4 \ 10^2 \ 10^0 \ 10^2 \ 10^4 \ 10^3 \right) \quad (44)
\]

So, given a fixed scaling, it apparently is possible to find different parameter estimates with different settings of the EKF. It is therefore unclear how to determine if a solution is valid.

Figure 22: Outputs of the linear block of propofol (red) and remifentanil (blue) for every data point of Bis1 converted to mg/h. The blue line is close to zero. The sample interval is 5 seconds.

Figure 23: Parameter estimates for every data point of Bis1 converted to mg/h.
Figure 24: The BIS-values (magenta dashed line) of Bis1 converted to mg/h plotted against time. The BIS-measurements are mostly covered by the BIS estimates.

Figure 25: The difference between the measured and estimated BIS values ($y - \hat{y}$) for Bis1 converted to mg/h.
Figure 26: The BIS-values (magenta dashed line) and BIS-estimates (blue line) of Bis1 converted to mg/h plotted against time with $P_0$ as in equation (43).

Figure 27: The BIS-values (magenta dashed line) and BIS-estimates (blue line) of Bis1 converted to mg/h plotted against time with $P_0$ as in equation (44).
Figure 28: Parameter estimates for every data point of Bis1 converted to mg/h where $P_0$ is as in equation (43).

Figure 29: Parameter estimates for every data point of Bis1 converted to mg/h where $P_0$ is as in equation (44).

5.6.2 Bis1 (converted to mg/h) and tuned

In order to get the normalized linear output of both propofol and remifentanil around order 1 (as explained in section 5.2), the half effect concentrations were changed. This led to the following parameters:
Figure 30 shows that the normalized linear output in general is in the desired range for both drugs, except for the very beginning of the surgery, where it is rather high for propofol. However, this might be acceptable for the induction phase where typically high dosages are administered. Figure 31 shows the parameter estimates. Again, not all four seem to vary as much as in figure 13. Changing the corresponding entries in $R_1$ did not resolve this problem.

Figure 32 shows the BIS-measurements and estimates. Except for the beginning, the estimates follow the measurements smoothly, indicating a good estimate. This is in compliance with figure 33 that shows the difference between the measurements and estimates: the difference is restricted, indicating the estimates are not far off, but at the same time not too small, indicating the estimates are not influenced too much by measurement noise. More tuning of $P_0$ or using the values of $P_0$ in [9] did not improve the approximation in the beginning.

$P_0 = diag(10^4 \ 10^2 \ 10^0 \ 10^4 \ 10^2 \ 10^0 \ 10^2 \ 10^4 \ 10^3)$

$R_1 = diag(1 \ 10^4 \ 0.1^2 \ 10^2 \ 0.1^2 \ 200 \ 10^2 \ 30^2 \ 30^2)$

$R_2 = 2 \cdot 10^{11}$

$EC_{50}^{\text{prop}} = 50$

$EC_{50}^{\text{remi}} = 1$

$13 P_0 = diag (100^2 \ 10^2 \ 1^2 \ 100^2 \ 10^2 \ 1^2 \ 10^2 \ 10^2 \ 10^2 \ 10^2)$
Figure 31: Parameter estimates for every data point of Bis1 converted to mg/h

Figure 32: The BIS-values (magenta dashed line) and BIS-estimates (blue line) of Bis1 converted to mg/h plotted against time.
Figure 33: The difference between the measured and estimated BIS values \((y - \hat{y})\) for Bis1 converted to mg/h.

### 5.6.3 3600PAZ41new tuned and with right sampling interval

The data of PAZ41new was converted to mg/h, the sample time of 1 s was taken into account and the half effect concentrations were changed for the same reason as in section 5.6.2. This led to the following parameters:

- \(P_0 = diag \begin{pmatrix} 100^2 & 10^2 & 1^2 & 100^2 & 10^2 & 1^2 & 10^2 & 10^4 & 10^4 \end{pmatrix}\)
- \(R_1 = diag \begin{pmatrix} 100 & 10^2 & 0.1^2 & 10^2 & 10^2 & 0.1^2 & 10^2 & 10^2 & 20^2 & 30^2 \end{pmatrix}\)
- \(R_2 = 2 \cdot 10^{11}\)
- \(EC_{50}^{prop} = 500\)
- \(EC_{50}^{remi} = 30\)

Figure 34 shows the infusion rates of propofol and remifentanil as present in the data. Except for the very beginning (that corresponds to values that were added manually), the graphs shows no peaks (compare figure 11). This is because the bolus doses were not registered by the equipment of the Brescia set-up, as explained in section 5.4. This lack of variation makes it harder to draw solid conclusions since the filter needs some variation to work properly.

Figure 35 shows the normalized output of the linear blocks, which roughly is in the desired range. Only in the beginning there is a very high amplitude for remifentanil, but this may be acceptable for the induction phase.
An important and worrying observation is that the parameter estimates of $\alpha$ and $\eta$ in figure 36 are flipped compared to the estimates for the Bis1 data in figure 31. This is unexpected since one expects all patients to comply to the same model, and that parameters only differ slightly between patients. The situation where parameter estimates develop significantly differently between patients is not in line with this. Also, not all four parameters seem to vary as much as in figure 13: some seem to be roughly constant. Changing the corresponding entries in $R_1$ did not resolve this problem.

Figure 37 shows the BIS-measurements and -estimates. Just as the BIS-estimates for the Bis1 data in figure 32, it shows a poor approximation in the beginning of the process and this could not be resolved by more tuning of $P_0$ or using the recommended entries for $P_0$ in [9]. However, whereas the curve in figure 32 displays a smooth development, the curve in figure 37 displays a more sharp-cornered irregular pattern, indicating that these estimates are influenced more severely by measurement noise. This is in compliance with figure 38 that shows that the difference between the measurements and estimates is quite small (and smaller than for Bis1 in figure 33). Maybe more tuning of $R_1$ will be able to improve the approximation, but this result was the best result obtained within the scope of this project.

Figure 34: The infusion rate of propofol and remifentanil from PAZ41new (converted to mg/h and tuned) plotted against time.
Figure 35: Normalized outputs of the linear block of propofol (red) and remifentanil (blue) for every data point of PAZ41new (converted to mg/h and tuned). The sample time is 1 s.

Figure 36: Parameter estimates of PAZ41new (converted to mg/h and tuned) plotted against time.
Figure 37: The BIS-values (magenta dashed line) and BIS-estimates (blue solid line) of PAZ41new (converted to mg/h and tuned) plotted against time.

Figure 38: The difference between the measured and estimated BIS values ($y - \hat{y}$) for PAZ41new (converted to mg/h and tuned).
6 Conclusion and future work

The uncertainty about the used unit in the BIS Urology files caused a lot of trouble and this uncertainty could not be resolved using international dosage standards. Therefore, eventually an educated guess was made about the unit in which the data of Brescia and Porto should be provided to the program. In addition, with trial and error, the $EC_{50}$s were set to values that produced a sensible scale in the normalized output of the linear block. This guessing of the right scaling of the data and the right values for the half effect concentrations made it harder to draw reliable conclusions.

The EKF was tuned by varying the initial covariance matrix $P_0$, and the covariance matrices of the process noise and measurement noise ($R_1$ and $R_2$ respectively). At best, this resulted in fairly good BIS-estimations, but at the same time in parameter estimates that were constants over time. Addressing this issue by directly changing the corresponding entries in $R_1$ did not result in more varying parameter estimates. The state space entries related to the linear part of propofol and remifentanil (i.e. the first six entries) did not seem to play a role at all and changing a parameter entry in $R_1$ did not straightforwardly affect the estimate of the corresponding parameter. This is not in accordance with general properties of tuning (as treated in section 5.3.2) and this finding questions how useful further tuning of $R_1$ is, and also how adequate the used model is or how adequate the set parameters are.

Another worrying finding was that different settings of the EKF, while resulting in comparable tracking performances, could lead to different parameter estimates. This raises the question what solution is correct and how to determine this. Assuming the scaling of the data is somewhat accurate, this points out the limitations of the used model: apparently it is not able to produce a unique solution for the used data. Also, the finding that parameter estimates could differ significantly between patients, questions the used model. Maybe the parameter estimates could be quite different between patients, but reasonably they should be in the same order of magnitude and should play a comparable role in the model. Flipped parameters as described in 5.6.3 do not comply with this. Another finding that questions the used model was that altering $P_0$ did not manage to improve the approximation in the beginning of the process. A different model may therefore be more adequate.

Another difficulty was that the data of Brescia did not include bolus dosages. This has a negative implication on the validity of the results, because one needs to have some variation in the input for the algorithm to produce reliable estimates. This is because if the input signals are constant, then the output signal will also become constant, which makes it impossible to learn about the dynamics of the system.

Since the results question the validity of the used model, future work should include: the replication of the procedure on other data, the replication on data from Brescia that does include all administered bolus doses, and the comparison with the application of other models (e.g. compartment models) on the studied data.

The long term goal is to make automatic closed-loop anesthesia possible with the benefits of relieving the anesthesiologist from the hard task of administering optimal...
drug doses, achieving more consistent drug effects by means of individualization, and reducing side effects because of the achieved reduced overall drug administration.

References


