Model-Based Conductivity Control of Fluid Composition

Måns Fällman

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
ISRN LUTFD2/TFRT--5992--SE
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2016

Abstract

The purpose of this thesis was to reduce the start-up and restart time of a hemodialysis machine by targeting time consuming control loops. This was done by investigating different techniques of modelling to find the best combination of versatility and conformity to the process. Finding the best controller was done by evaluating different controllers in simulations with the model in MATLAB/Simulink by comparing closed loop performance and robustness. The results were verified by implementing the controller on a LabView realtime hardware setup.

Acknowledgements

I would like to thank all the employees at Baxter for the warm welcome and for the good company during these months, especially Anders Wallenborg and Evan Foy for all your guidance, help and motivation. A huge thanks to Tore Hägglund, my advisor at the department of automatic control for your assistance during this project. The biggest thanks goes to Michael Pettersson, my advisor at Baxter. This would not have been possible without your efforts and support. You have been a great colleague and a true mentor.

Contents

Nomenclature

Notion	Description	Unit
K_A	Conductivity of A concentrate	[mS/cm]
K_B	Conductivity of B concentrate	[mS/cm]
τ_A	Time delay for A concentrate	[s]
τ_B	Time delay for B concentrate	[s]
Q	Main fluid flow	[l/s]
V_1	Volume of upper first chamber	[l]
V_2	Volume of lower first chamber	[l]
V_3	Volume of upper second chamber	[l]
V_4	Volume of lower second chamber	[l]
Pu_A	A-concentrate pump flow	[ml/min]
Pu_B	B-concentrate pump flow	[ml/min]
h	Sample rate	[s]

Abbreviation	Description
RO	Reverse Osmosis
TTT	Time To Treatment
HDF	Hemodiafiltration
CAD	Computer Aided Design
PID	Proportional Integral Derivative
LQG	Linear Quadratic Gaussian
MPC	Model Predictive Control
DLL	Dynamic-Link Library
PXI	PCI eXtensions for Instrumentation
I2C	Inter-Integrated Circuit
AMIGO	Approximate M-constrained Integral Gain Optimization

1

Introduction

Dialysis refers to the procedure of removing waste products and excess fluid from the blood stream. This is the main form of treatment for both acute and chronic renal failure. The most common [Jain et al., 2012] dialysis procedure is called hemodialysis where blood is withdrawn from the body and filtered externally which can be seen in Figure 1.1.

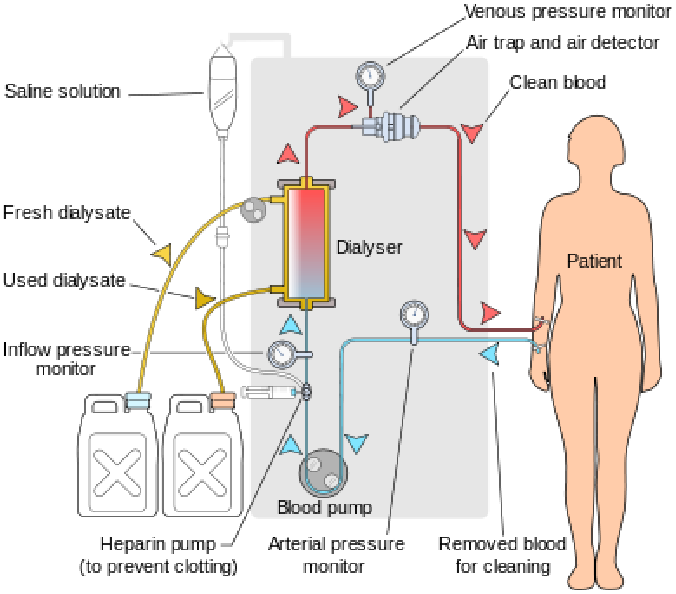

Figure 1.1 Schematic of a hemodialysis treatment

The filter is called a dialyzer and can be described as two chambers separated by a semi-permeable membrane that allows for smaller particles, such as electrolytes, urea and water to pass, but stops larger ones such as proteins and blood cells. With blood in one of the chambers the other side of the filter is filled with a dialysate. Depending of the fluid compositions in the dialysate, some of the smaller particles will diffuse from the blood to the dialysate. This filters out certain waste products by not adding them to the dialysate. The most basic hemodialysis machines usually include two types of concentrates. A-concentrate is a mixure of electrolytes to match the content of the patients blood. Their effect on the conductivity when mixed with water is well known and their solubility in water is not temperature dependent. The other component i called B-concentrate and usually only includes bicarbonate, (HCO_3^-) which works as a pH buffer. Unlike the A-concentrate, the solubility of the B-concentrates is temperature dependent which complicates the modelling. The A-and B-concentrates can not be added at the same point in the fluid path due to the risk of bicarbonate precipitation. This could be avoided by premixing the concentrates, although this introduces a major risk as it provides a good environment for bacterial growth. Both A and B concentrates are available as solution or a dry product where the machine adds water.

The quality of hemodialysis treatments varies a lot around the world, some countries prefer a simpler and cheaper procedure whereas some prefer a more complex treatment. This could be a more specific concentrate control or HDF treatment. The hemodialysis machines seen in Figure 1.2 differ in hardware due to their targeted market.

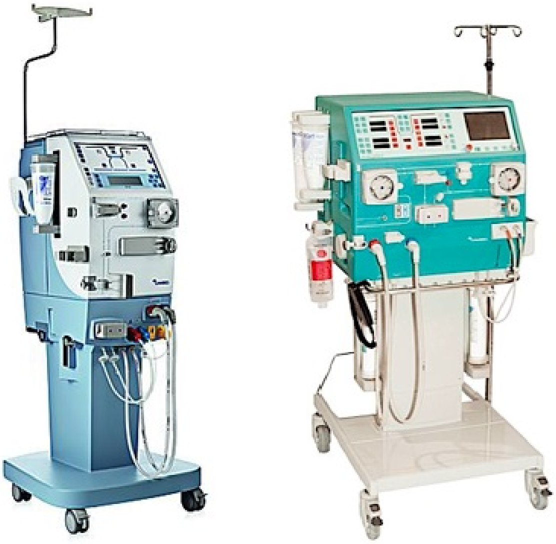

Figure 1.2 A low-end and a high-end dialysis machine from Gambro

A typical dialysis treatment usually takes around 4 hours. At three treatments per week this turns into a very time consuming activity, that is why a lot of time is spent on trying to decrease the time for overhead activities such as starting and restarting. One of the most time consuming steps during start-up is to reach an acceptable fluid composition where the concentrate control is included. The start-up or restart time is referred to as TTT (Time To Treatment) and the goal is to minimize TTT.

Project goal

The goal of this master thesis project was to reduce the TTT for a hemodialysis machine by developing a better conductivity controller. To reach the goal, a model of the process was first derived. Three controllers were then designed and compared using Matlab/Simulink. The three controllers were PID, LQG, and MPC. The LQG controller was found most suitable, and this controller was implemented in LabView and tested on a dialysis machine test setup.

2

Modelling

When working with model based control, the first thing needed is of course a model of the process. This was obtained by investigating physical properties and collecting data from the process. Some prior knowledge of the process was present due to earlier work at Baxter.

2.1 Physical setup

The fluid preparation of a dialysis machine seen in Figure 2.1 works similar to an assembly line. RO-water[1] is heated, degassed and concentrates are mixed in.

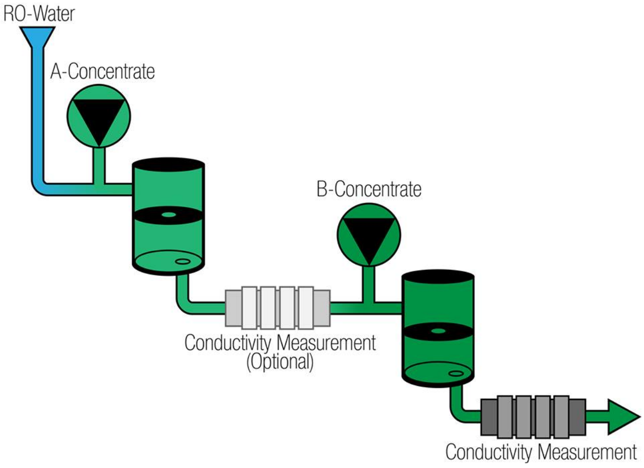

Figure 2.1 A simplistic schematic of the setup

[1] Reverse osmosis water. Desalinated water by overcoming the osmotic pressure

Before treatment can begin all of these steps have to pass a function check to ensure a safe treatment. This is referred to as getting a "green fluid path" which is indicated by the dialysis machine. If any of the controlled values deflect for a specified time, the machine will automatically pause the treatment until it achieves an acceptable fluid decomposition.

The concentrate pumps are piston type pumps powered by stepper motors. This entails an almost constant volumetric flow for each revolution. This adds a disturbance in the output as the pump only extracts concentrate for less than half of the period. In order to combat these disturbances a mixing chamber is added as a low pass filter after each infusion point. The mixing chamber is separated into two smaller chambers by a divider in order to achieve a greater attenuation for high frequency disturbances. After the last mixing chamber a conductivity measurement cell is used to measure the feedback signal. The conductivity measurement cell provides some internal filter but only for anti-aliasing.

2.2 Model

Each subchamber acts as a first order low pass filter for the concentrate added into the main flow. The conductivity in each subchamber could therefore be interpreted as a state. As the chambers are always filled up and the main flow set constant during the start up sequence, this turns into a simple and linear mixing problem seen in Figure 2.2.

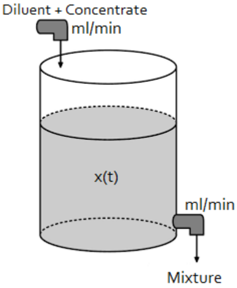

Figure 2.2 A simple illustration of a mixing problem

In a mixing problem [Olsson and Rosen, 2005, p. 53], a mixture of diluent and concentrate is added to a chamber initially filled with the diluent. The mixture is extracted at the same speed as the intake resulting in a constant volume. The rate of change for the concentration inside can be described as the difference between injected and extracted concentration. By assuming a homogeneous mixture and the conservation of mass, Equation 2.1 can be used to describe the change of concentration

$$V \cdot \dot{c} = Q_{in} \cdot c_{in} - Q_{out} \cdot c = Q_{in}(c_{in} - c) \tag{2.1}$$

where V is the chamber volume, c_{in} is the injected concentration, c is the concentration inside the chamber, Q_{in} is the injected flow rate and Q_{out} is in extracted flow rate. Equation 2.1 is only valid for a constant volume which is achieved when Q_{in} equals Q_{out}. As the four tanks are connected in series, the unified system would be the convolution of the dynamics of each chamber. The state space form is more intuitive for systems of higher degree, each subchamber accounts for one row in the state space equation. The state space representation can be seen in Equation 2.2

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.2}$$

where A is the system matrix, B is the input matrix, C is the output matrix, D is the feedtrough matrix, x is the state vector, which is the respective volumes conductivity measured in mS/cm and y is the same as the last state in x and is the measured output in mS/cm. The output y, is sometimes referred to as C2 as this is the name of the sensor.

$$\begin{gathered} A = \begin{bmatrix} -\frac{Q}{V1} & 0 & 0 & 0 \\ \frac{Q}{V2} & -\frac{Q}{V2} & 0 & 0 \\ 0 & \frac{Q}{V3} & -\frac{Q}{V3} & 0 \\ 0 & 0 & \frac{Q}{V4} & -\frac{Q}{V4} \end{bmatrix} \\ B = \begin{bmatrix} \frac{Q}{V1}K_A & 0 \\ 0 & 0 \\ 0 & \frac{Q}{V3}K_B \\ 0 & 0 \end{bmatrix} \\ C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ D = \begin{bmatrix} 0 & 0 \end{bmatrix} \\ u = \begin{bmatrix} Pu_A \\ Pu_B \end{bmatrix} \end{gathered} \tag{2.3}$$

Q is the main fluid flow, V_{1-4} are the volumes of each subchamber, K_A and K_B are the steady state gains of A- and B-concentrate, respectively. Pu_A and Pu_B are the volumetric flow of concentrates in ml/min. Although the state space form is the most useful the transfer function is also sometimes needed. The transfer function for each input is calculated as

$$H(s) = C(sI - A)^{-1}B + D \tag{2.4}$$

which results in the two transfer functions.

$$H(s) = \begin{bmatrix} \frac{\frac{Q^4 K_A}{(V_1 V_2 V_3 V_4)}}{(s+\frac{Q}{V_1})(s+\frac{Q}{V_2})(s+\frac{Q}{V_3})(s+\frac{Q}{V_4})}, & \frac{\frac{Q^2 K_B}{(V_3 V_4)}}{(s+\frac{Q}{V_3})(s+\frac{Q}{V_4})} \end{bmatrix} \begin{bmatrix} Pu_A \\ Pu_B \end{bmatrix} \tag{2.5}$$

To obtain the dynamics of the system, the flow and chamber volumes have to be estimated. During start up the main flow is set to $500ml/min$. The volumes of the sub-chamber in the mixing chambers are harder to find. The total chamber volume can be estimated by weighing the chamber before and after it has been filled and then visually inspect or measure the relative distance to the divider. Another way is checking CAD-files of the chambers. It is also important to verify the volumes both by measuring them physically and the dynamics that they impose. Both chambers have a volume of $60ml$ and their dividers are identically placed. Table 2.2 shows the volumes estimated by measuring the offset of the divider.

Chamber	**Volume** [l]
V_1	0.0399
V_2	0.0219
V_3	0.0384
V_4	0.0216

Table 2.1 Estimated volumes in each subchamber

Without having an estimate of the stationary gain we can still use Matlab/Simulink to simulate the step response.

```
Q = 0.5/60; %Main flow
V1 = 0.0399; %Subchamber volumes
V2 = 0.0295;
V3 = 0.0384;
V4 = 0.0216;
Ka = 1;
Kb = 1;

A = [-Q/V1 0 0 0;
     Q/V2 -Q/V2 0 0;
     0 Q/V3 -Q/V3 0;
     0 0 Q/V4 -Q/V4];

B = [Q/V1*Ka 0;
    0 0;
    0 Q/V3*Kb;
    0 0];

C = [0 0 0 1];
step(ss(A,B,C,[0 0]))
```

The step responses are shown in Figure 2.3

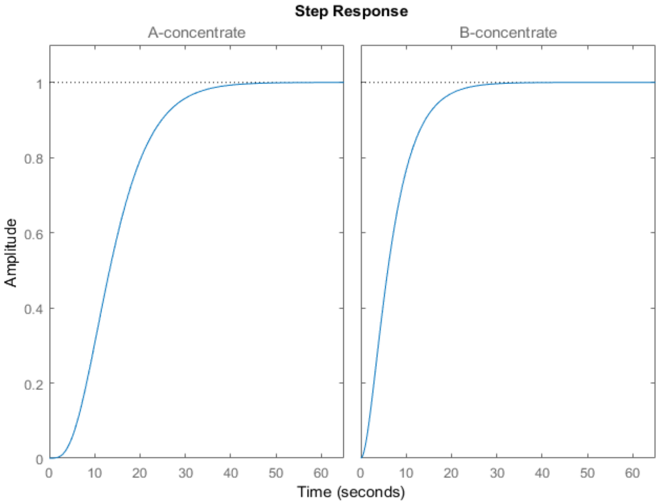

Figure 2.3 Step response from each input

2.3 Delays

Due to the physical distance between the concentrate injection points and conductivity measurements a time delay is present. This time delay is due to the flow of water and thereby also inversely proportional to the flow. The time delay is present due to both the tubing and the fact that the mixing chambers do not have a stirrer. Excluding the fact that the concentration in each chamber is non homogeneous, the time delays could be estimated by measuring the length L and area of the tubing A

$$\tau = \frac{L \cdot A}{Q} \tag{2.6}$$

where Q is the main flow rate and τ is the time delay from the injection point to the measuring point.

The best way to estimate the time delay is by looking at the delayed step response on the actual system in Figure 2.4.

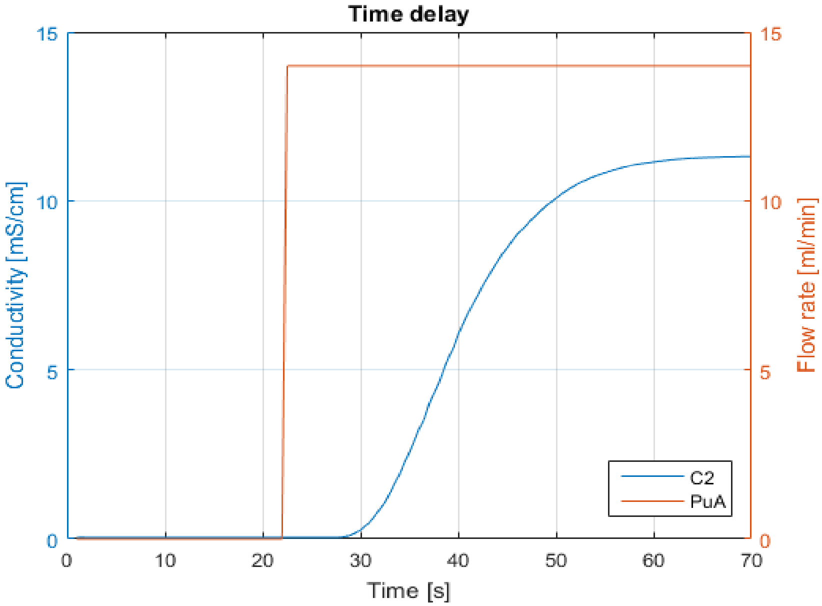

Figure 2.4 Time delay for A-concentrate seen as the difference between the input step and output change

The time delays are shown in Table 2.2 but the controller still needs to be robust enough to handle deviations. Deviations can originate from different tubing lengths inside, machine configurations or replacement components. The actual delay was only interesting in order to verify the controllers stability. Testing the controller by altering the time delay around the actual delay showed that the controller could maintain stability for a changed time delay. This is interesting as the total length of tubing is subject to change and this means that the controller would not need returning.

	Delay [s]
τ_a	4.51
τ_b	2.04

Table 2.2 Time delays for each input

There is no way to incorporate a time delay in state space representation so the transfer function is instead used.

$$H(s) = \begin{bmatrix} \dfrac{\frac{Q^4 K_A}{(V_1 V_2 V_3 V_4)}}{(s+\frac{Q}{V_1})(s+\frac{Q}{V_2})(s+\frac{Q}{V_3})(s+\frac{Q}{V_4})} e^{-s\tau_A}, & \dfrac{\frac{Q^2 K_B}{(V_3 V_4)}}{(s+\frac{Q}{V_3})(s+\frac{Q}{V_4})} e^{-s\tau_B} \end{bmatrix} \begin{bmatrix} Pu_A \\ Pu_B \end{bmatrix} \tag{2.7}$$

2.4 Verification

Disabling the PID controller and setting a fixed pump speed on the real process, produces a step response. This was done for both A- and B-concentrates, one at a time. By simulating the model with the same fixed input it can be compared with the process output. The Matlab system identification toolbox is a great tool for comparing processes and models. The following Matlab script compares the real process output y_data to the model output when subjected to u_data as input.

```
Q = 0.5/60; %Main flow
V1 = 0.0399; V2 = 0.0295;
V3 = 0.0384; V4 = 0.0216;   %Subchamber volumes
Ka = 0.8431; Kb = 0.1328;   %SS Gain

A = [-Q/V1 0 0 0;
    Q/V2 -Q/V2 0 0;
    0 Q/V3 -Q/V3 0;
    0 0 Q/V4 -Q/V4];
B = [Q/V1*Ka 0;
    0 0;
    0 Q/V3*Kb;
    0 0];
C = [0 0 0 1];

compare(y_data,sim(ss(A,B,C,0),u_data))
```

Figure 2.5 and Figure 2.6 show step responses for the two models.

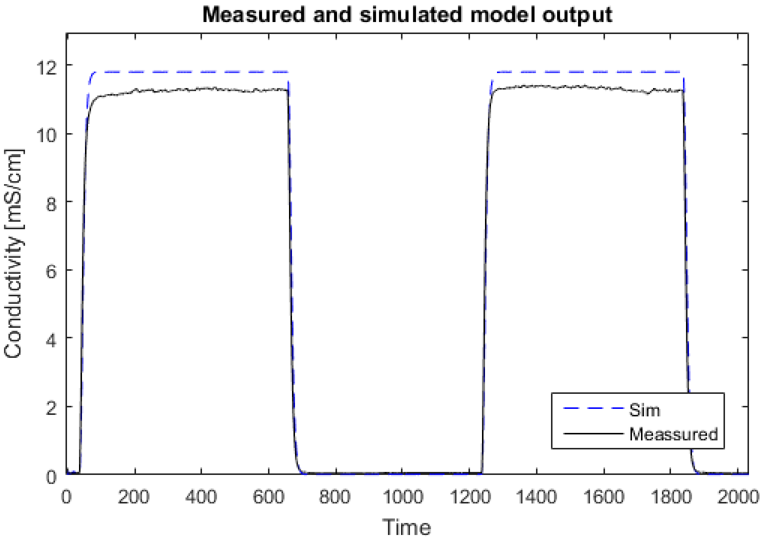

Figure 2.5 First comparison of process and model for A-concentrate

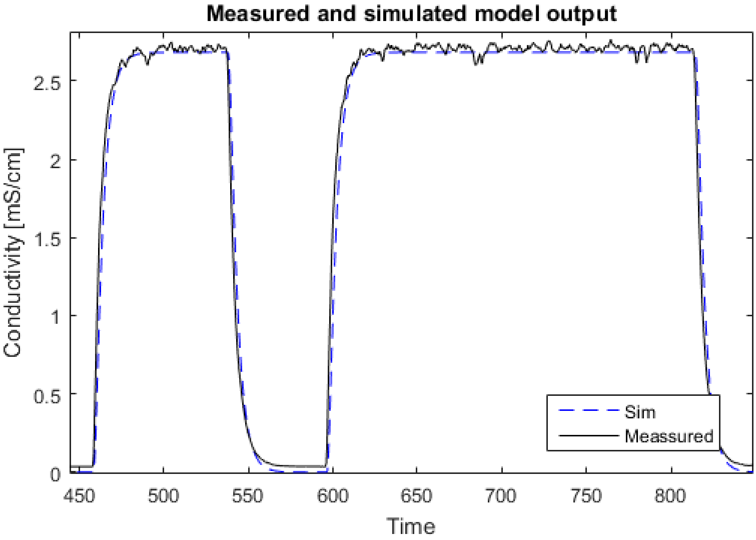

Figure 2.6 First comparison of process and model for B-Concentrate

By varying the model volumes to fit the step response to the process step response, the effective volume of the chambers can be estimated. Due to disadvantageous shapes of the chambers, the volumes might behave as smaller than they were previously measured to be. This would be an indication that the mixing chambers might need to be redesigned.

With a fixed pump speed the steady state gain can be calculated as the mean output divided by the mean pump rate seen in Equation 2.8.

$$\frac{y_{\text{mean}}}{u_{\text{mean}}} = \text{Gain}_{\text{ss}} \tag{2.8}$$

Table 2.3 shows the steady state gains calculated using this method.

Ka $[\frac{mS/cm}{ml/min}]$	**Kb** $[\frac{mS/cm}{ml/min}]$
0.8431	0.1328

Table 2.3 Steady state gain calculated by the empirical method

It is also important to verify the empirical gains with the calculations based on the molar conductivity of the electrolytes. For A concentrate this is dependent on the different electrolytes included and for B concentrate this is dependent on the temperature. The exact calculations are a bit outside the scope for this thesis work. In testing values were taken from an existing conductivity calculation script.

Offline estimations

The Matlab system identification toolbox provides a number of tools for offline estimation. Although the models produced by the different identification methods usually have a high conformity with the real process they do not offer the true states. Whereas each state in the physical model is related to a physical variable, the pseudo-states of the estimated model are there to achieve a process of the same degree but not related to any physical variable. Even though they performed better than physical modelling, the improvement was only around 1% which can be seen in Figure 2.7 and 2.8. These models often included complex conjugated poles which of course are not present in this system.

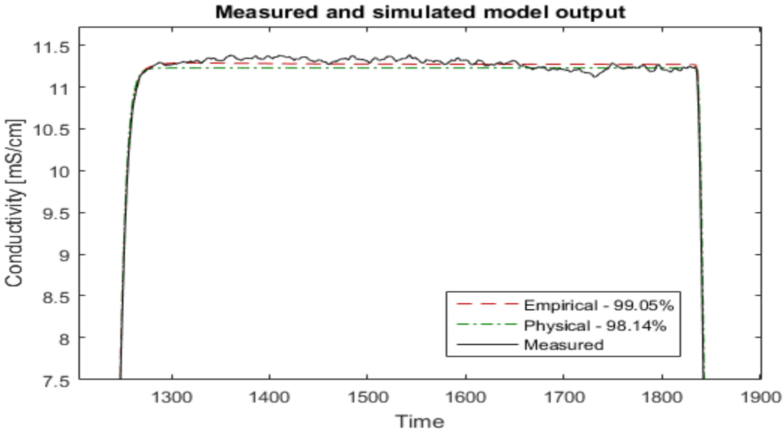

Figure 2.7 Comparison of process, physical model and empirical model for A-

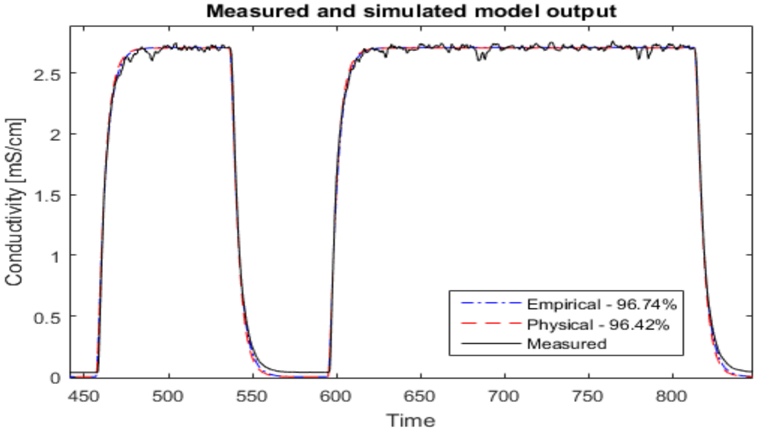

Figure 2.8 Comparison of process, physical model and empirical model for B-Concentrate

3

Simulation

Once the model was deemed fitting, work on a controller simulation started. Each concentrate was tested one at a time by altering the B matrix. When testing A-concentrates the second column of B was removed and vice versa.

3.1 Simulink

The process was made as an individual block in Simulink to simplify comparison between controllers. Running different controller set-ups using a so called "model reference" is a convenient way to quickly compare results. The model in Figure 3.1 includes a saturation to not exceed the maximum allowed pump speed, and a transport delay.

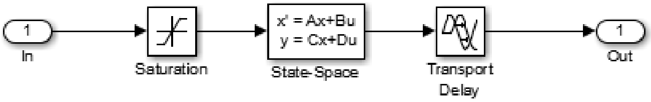

Figure 3.1 Simulink model of the process

The process still relied on values from the Matlab workspace for all of the state space matrices, saturation levels and transport delay. While the goal was to reduce the start-up time, the controller still had to be able to handle changes in the main flow during treatment. The time variant state space representation can be seen in

Equation 3.1

$$\begin{aligned}\dot{x}(t) &= A(Q)x(t) + B(Q)u(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{3.1}$$

where A, B and C matrices are defined in Equation 2.3. The state space block of Simulink does not support variable system matrices so instead it was designed by implementing the matrix element calculations with simple Simulink blocks as seen in Figure 3.2.

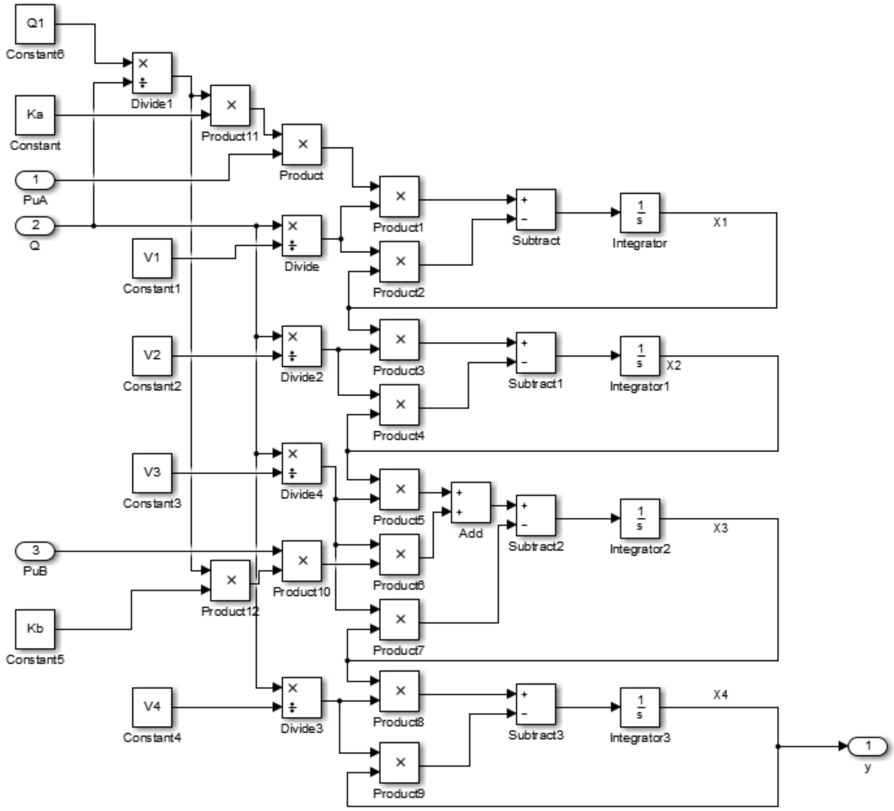

Figure 3.2 Simulink model of the time variable process

The time variable model was used to test model deviations accounted from a changed main flow. The model uses the three inputs *PuA*, *PuB* and *Q* to define the time variable model.

3.2 Controller

Three different controller set-ups were evaluated in this phase. As the goal was a controller that is feasible for implementation in a dialysis machine the computation complexity was somewhat important. The different controllers were of various complexities which had to be included in the decision-making as the target hardware is shared by other time consuming events. The three controller algorithms tested were PID, LQG and MPC.

PID

The PID parameters were computed by the AMIGO method found in [Åström and Hägglund, 2006] which uses the transfer function obtained in Equation 2.7. The PID parameters used can be seen in Table 3.1.

Parameter	A	B
K	0.848	9.18
T_i	11.22	5.328
T_d	3.756	1.302

Table 3.1 PID parameters generated from the transfer function in 2.7

The original PID parameters served as a baseline when comparing the other controllers, although the more optimized PID parameters was tested to get an indication how much improvements can be achieved by optimization. The PID algorithm can be seen in Equation 3.2

$$u = K(-y + \frac{1}{T_i}\int e\,dt - T_d\frac{dy}{dt}) \tag{3.2}$$

The Simulink model for the PID controller can be seen in Figure 3.3

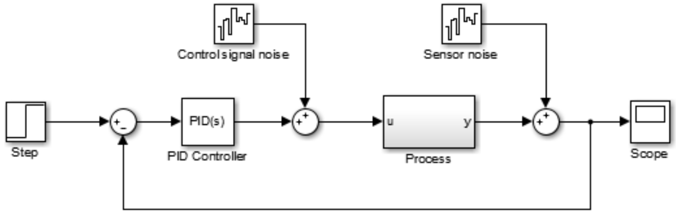

Figure 3.3 Simulink model of the PID setup

Figure 3.4 shows the PID closed loop step response for A- and B-concentrate respectively. Their respective setpoints are the default concentrate levels to verify that the pumps are not over excited.

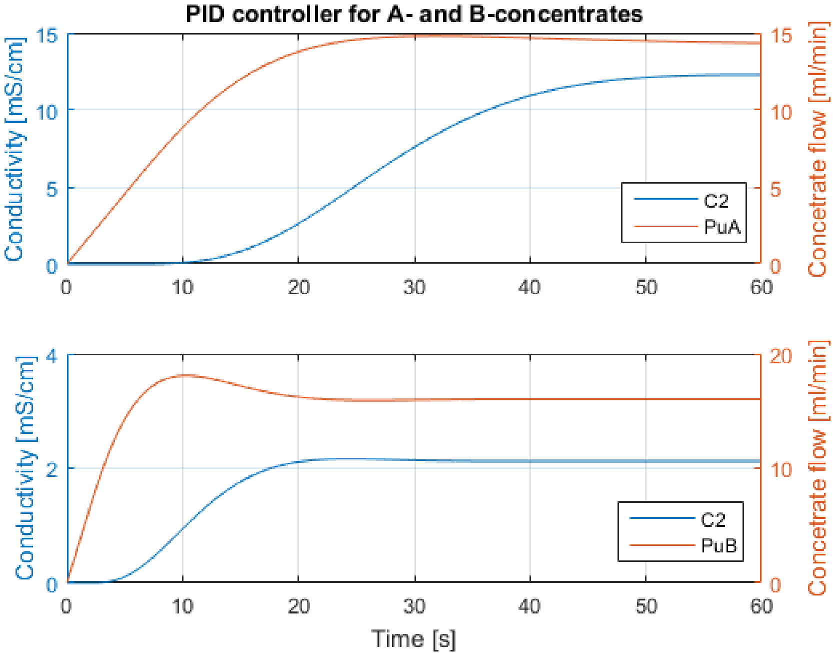

Figure 3.4 Simulated closed loop performance of a PID

LQG

The linear quadratic gaussian controller tested can be seen in Figure 3.5. It uses two key components to operate, a linear quadratic integral controller to generate the control signals and a Kalman filter to estimate the both the non-measured and measured states.

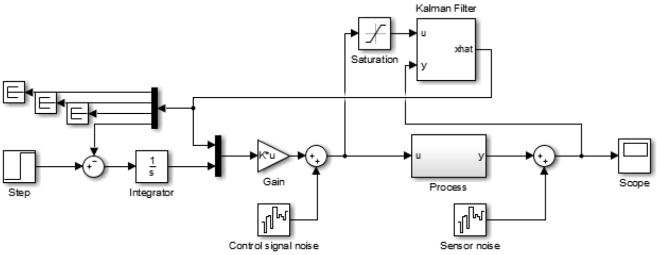

Figure 3.5 Simulink model of the LQG setup

Kalman filter: This is based on the Matlab/Simulink source which is in turn based on the principles of [Franklin et al., 1997]. The state of the system represented in the state space form is shown in Equation 3.3,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Gw(t) \\ y(t) &= Cx(t) + Du(t) + Hw(t) + v(t) \end{aligned} \tag{3.3}$$

where A, B, C and D, defined in Equation 2.3, can be estimated by the Kalman filter in Equation 3.6. $u(t)$ is the input, $w(t)$ is the process white noise and $v(t)$ is the measurement white noise with both zero mean.

$$\begin{aligned} E[w(t)] = E[v(t)] &= 0 \\ E[w(t)w^T(t)] &= Q(t) \\ E[v(t)v^T(t)] &= R(t) \\ E[w(t)v^T(t)] &= N(t) \end{aligned} \tag{3.4}$$

$Q(t)$ is the process covariance, $R(t)$ the measurement covariance and $N(t)$ the process and measurement noise cross covariance. The goal is to estimate $\hat{x}$ which minimises the state estimation error covariance $P(t)$ seen in the Equation 3.5.

$$P = E[(x - \hat{x})(x - \hat{x})^T] \tag{3.5}$$

The Kalman filter is then the optimal solution described by the following equations which minimizes P.

$$\begin{aligned} L(t) &= P(t)C^T(t) + \bar{N}(t) \\ \dot{P}(t) &= AP(T) + P(t)A^T + \bar{Q}(t) - L(t)\bar{R}(t)L^T(t) \\ \dot{\hat{x}}(t) &= Ax(t) + Bu(t) + L(t)(y(t) - C\hat{x}(t) - Du(t)) \end{aligned} \tag{3.6}$$

$\bar{R}(t)$, $\bar{Q}(t)$ and $\bar{N}(t)$ are defined as

$$\begin{aligned}\bar{Q}(t) &= GQ(t)G^T \\ \bar{R}(t) &= R(t) + HN(t) + N^T(t)H^T + HQ(t)H^T \\ \bar{N}(t) &= GQ(t)H^T + N(t)\end{aligned} \tag{3.7}$$

When G equals the identity matrix I, $\bar{Q}(t)$ turns into the process covariance matrix. H specifies the feedtrough of process noise to the measurement noise. As no feedtrough is present, H equals zero and the measurement noise covariance $\hat{R}(t)$ is only dependant on $R(t)$. $\bar{N}(t)$ is process and measurement noise cross covariance matrix.

Linear quadratic integral controller: The LQG is based on the principles of [Young and Willems, 1972].
For the continuous time state space system seen in Equation 3.8 with input $u(t)$,

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned} \tag{3.8}$$

the linear quadratic integral controller minimizes the cost function $J(u)$ defined in Equation 3.9 but also rejects constant offset error by integrating the feedback error.

$$J(u) = \int_0^\infty z^T(t)Qz(t) + u^T(t)Ru(t) + 2z^T(t)Nu(t)\,dt \tag{3.9}$$

Q is the state weight matrix, R the control signal weight and N is state and control cross term weight. z is the augmented state vector which includes the integrated error. Including the reference value r into the equation adds an integral action to the controller.

$$\begin{aligned}\dot{z}(t) &= \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} z + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r = A_a z + B_a u \\ y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} = C_a z(t)\end{aligned} \tag{3.10}$$

The state feedback gain vector K is the infinite horizon optimal gain for the system in Equation 3.8. The control signal u is generated by state feedback control and can be seen in Figure 3.2.

$$u = -K \begin{bmatrix} x \\ x_i \end{bmatrix} = -Kz \tag{3.11}$$

By solving the continuous time algebraic Riccati equation below, the solution can then used to calculate the optimal state feedback K for the augmented system.

$$\begin{aligned}A_a^T X + XA_a - (XB_a + N)R^{-1}(B_a^T X + N^T) + Q = 0 \\ K = R^{-1}(B_a^T X + N^T)\end{aligned} \tag{3.12}$$

Here A_a and B_a is defined in Equation 3.10. Q, R and N are the state, control signal and cross term weight matrices. X is the solution to the continuous time algebraic Riccati equation.

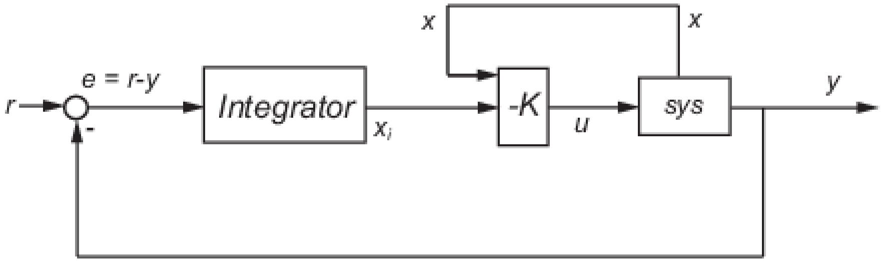

Figure 3.6 Schematic of the state feedback controller

The following Matlab code calculates the optimal feedback gain used in the simulations below.

```
Q = 0.5/60; %Main flow
V1 = 0.0399; %Subchamber volumes
V2 = 0.0295;
V3 = 0.0384;
V4 = 0.0216;
Ka = 0.8431;    %SS Gain
Kb = 0.1328;
A = [-Q/V1 0 0 0;
    Q/V2 -Q/V2 0 0;
    0 Q/V3 -Q/V3 0;
    0 0 Q/V4 -Q/V4];
B = [Q/V1*Ka 0;
    0 0;
    0 Q/V3*Kb;
    0 0];
Q = [1 0 0 0 0;
    0 1 0 0 0;
    0 0 300 0 0;
    0 0 0 1 0;
    0 0 0 0 4];
R = 0.5;
N = 0;
[K,S,e] = lqi(A,B,Q,R,N)
```

Figure 3.7 shows the LQG closed loop step response for A- and B-concentrate respectively and also their respective control signals to verify that the pumps are not over excited. The set-points in the simulations are the default concentrate levels for a standard treatment.

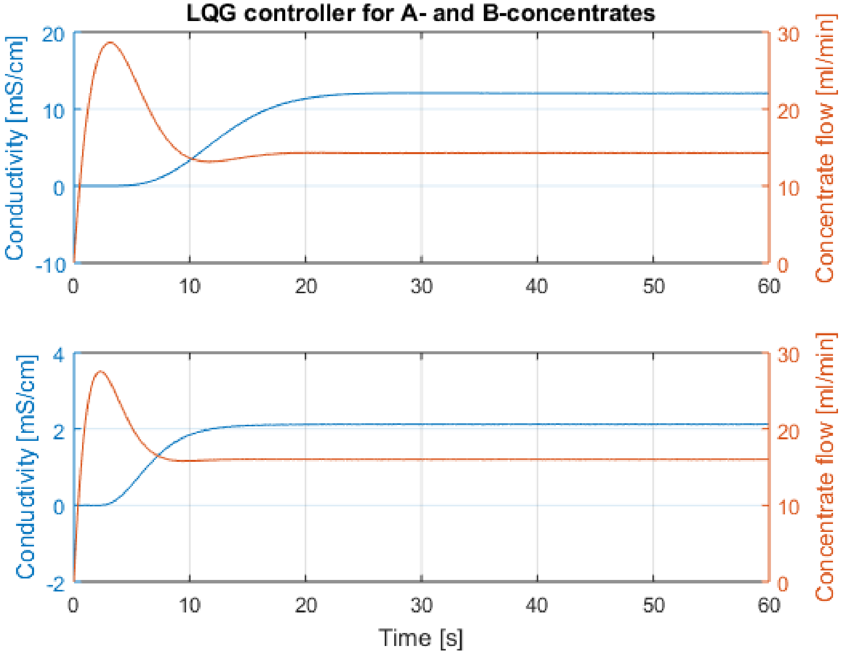

Figure 3.7 Simulated closed loop performance of a LQG

In Figure 3.5, the estimated output $\hat{y}$ is used instead of the real feedback y to integrate the error. This was done to reduce overshoot on reference step change. The Kalman filter does not include the time delay so it is effectively used as a predictor to counteract the effect of the time delay. This solution bears some resemblance to an Otto Smith predictor.

MPC

The MPC controller was evaluated to see how much the performance could increase with a more advanced algorithm. Simulink MPC toolbox was used to speed up evaluation, with the accompanying MPC tuner which made the tuning quick. Figure 3.8 shows the Simulink model for MPC setup with the extra constraints for input and output.

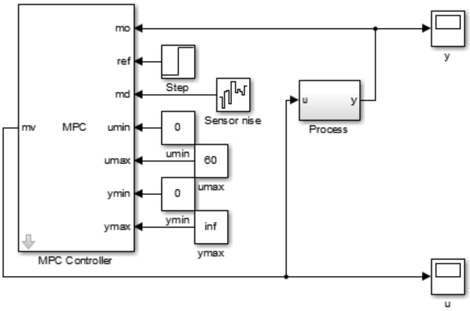

Figure 3.8 Simulink model of the MPC setup

The largest advantage in the MPC scenario is the ability to set the prediction horizon longer than the time delay. Here it is also important that the model is accurate.

For the discrete time state space

$$\begin{aligned} x[n+1] &= Ax[n] + Bu[n] + w[n] \\ y[n] &= Cx[n] + v[n] \end{aligned} \tag{3.13}$$

with the white noise disturbances $w[n]$ and $v[n]$ with zero mean and variance Q and R respectively, the internal state estimator is on the form

$$\hat{x}[n|n] = \hat{x}[n|n-1] + K(y[n] - \hat{y}[n]) \tag{3.14}$$

with the filter gain K. The corresponding one step ahead predictor is

$$\begin{aligned} \hat{x}[n+1|n] &= A\hat{x}[n|n] + Bu[n] + w[n] \\ \hat{y}[n] &= C\hat{x}[n] + v[n] \end{aligned} \tag{3.15}$$

By using the results for the one step predictor, the second step can be predicted and so on. The depth of the predictions is called the prediction horizon. With the state predictions calculated the optimal control signal can then be calculated using a finite horizon optimal control procedure. The depth of the predicted control signals is called the control horizon and is always less than the prediction horizon. Only the first value is used and the rest are discarded as the next iteration generates new. With these predictions the MPC algorithm allows for constraint violations to be predicted and avoided.

There are different versions of MPC algorithms and Simulink does not state which they have used so this is more of a general explanation of an MPC[Johansson, 2008]. An informative way to describe the MPC algorithm is shown in Figure 3.9

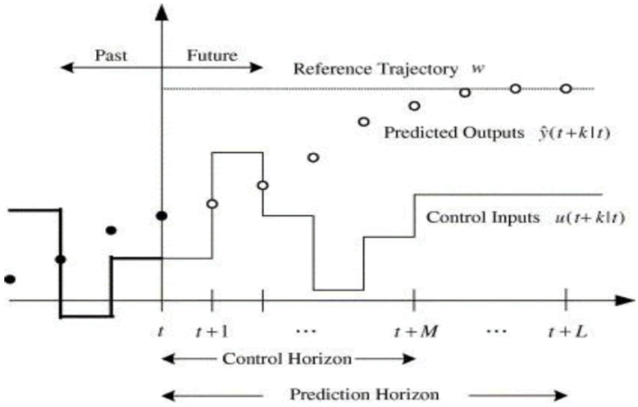

Figure 3.9 A graphic explanation of a single MPC iteration

Figure 3.10 shows the MPC closed loop step response for A- and B-concentrate respectively and also their respective control signals. The control signal is saturated by the controller when it goes higher than $60ml/min$.

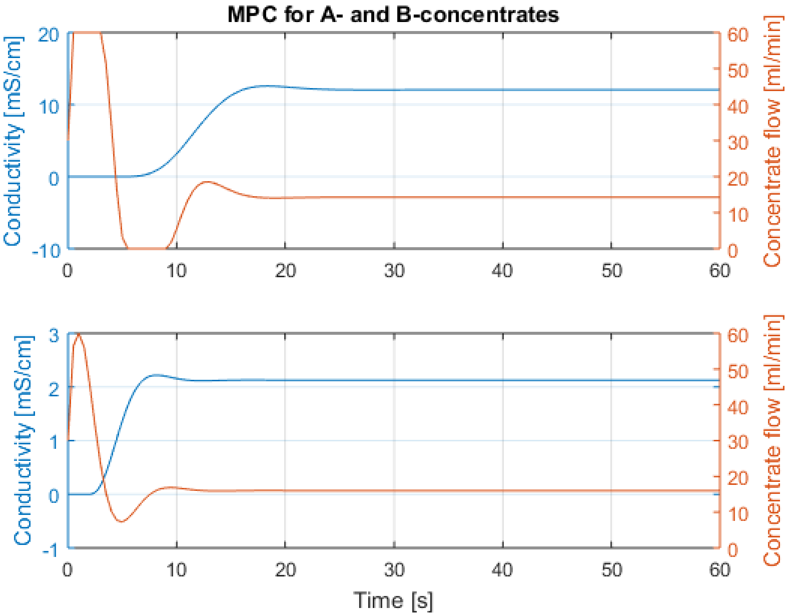

Figure 3.10 Simulated closed loop performance of an MPC

Simulation results

By comparing the closed loop step responses of the simulated systems it is apparent that the PID controlled performed a lot worse than the two other alternatives for the A-concentrate which is easily seen in Figure 3.11.

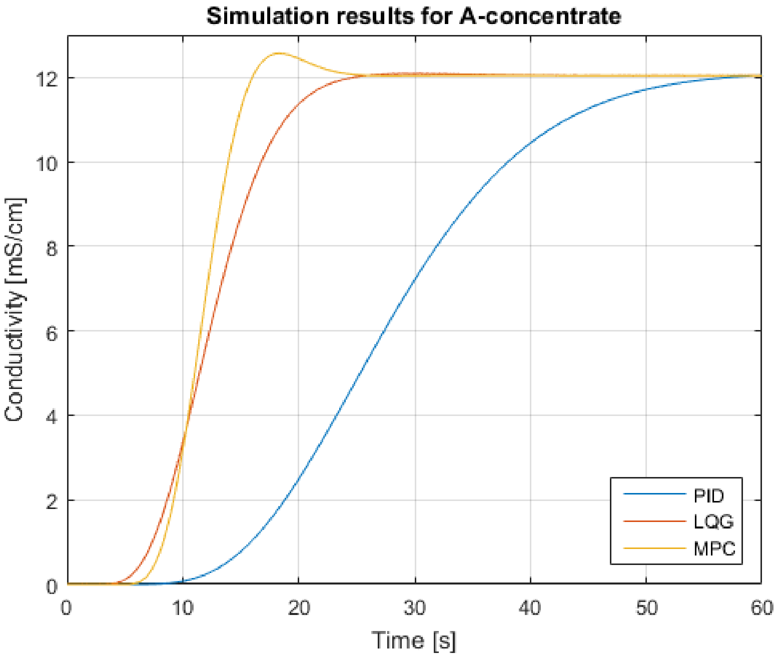

Figure 3.11 Simulated closed loop performance for A-concentrate

Figure 3.12 shows the closed loop performance of the three different controller for the B-concentrate. The difference in performance between the controllers are not as great as in the previous case. This might be an indication that the performance increase available of a more advanced algorithm is dependent on the time constant and time delay of the process. If this assumption is correct it might be better to reduce the size of the first mixing chambers and place the infusion point closer to measurement cell and instead use a PID controller.

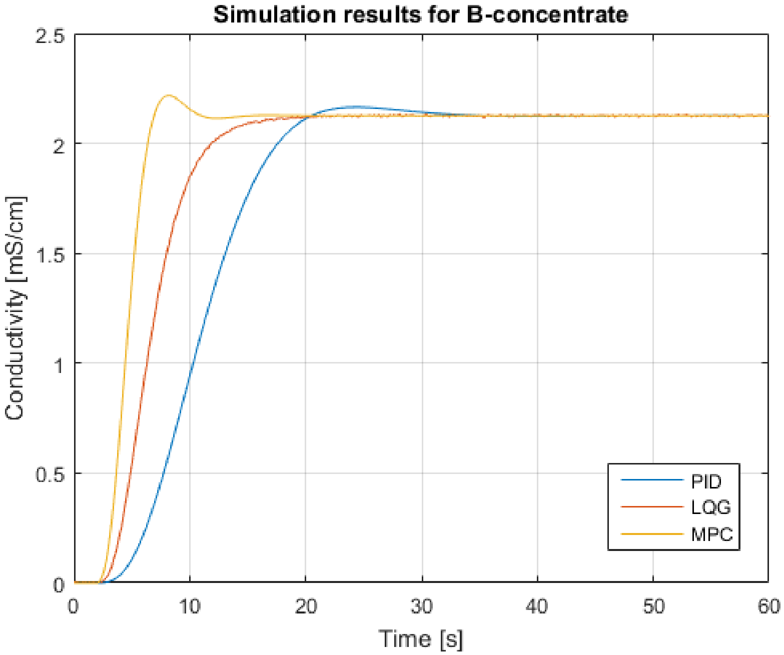

Figure 3.12 Simulated closed loop performance for B-concentrate

Due to the little difference in performance between MPC and LQG, the latter was implemented because of the simpler algorithm.

4

Implementation

The real time target is a National Instruments PXI system [NI, 2015] seen in Figure 4.1. These systems do not support either Matlab script or Simulink in a native environment which means that the controller had to be rewritten in Labview. Labview is a graphic programming language from National Instruments similar to Simulink but lacks functions for advanced automatic control. Another possibility would be to code generate from Simulink and include the controller as DLL in Labview, but this approach would impede the possibility to probe the controller while running.

Figure 4.1 A National Instruments PXIe, similar to the one used in the project

4.1 Discretization

In order to run on a physical system the model in Equation 2.3 had to be discretized. The conductivity measurement cell in Figure 2.1 is connected to the PXI chassis by an I2C bus and due to communication bus congestion the sample rate h is limited to 0.5 s. This rather slow sampling speed does not lead to any implications as the smallest time constants of the system are at least a factor 4 larger.

The discrete time state representation of the continuous time state space in Equation 2.2 is shown in the Equation 4.1.

$$\begin{aligned} x[n+1] &= \Phi x[n] + \Gamma u[n] \\ y[n] &= Cx[n] + D[u] \end{aligned} \tag{4.1}$$

where Φ is of the same size as A and Γ is the same size as B. The Φ and Γ matrices are functions of the sample time h instead of continuous time. Equation 4.2 shows the procedure of producing both Φ and Γ.

$$\begin{aligned} \Phi &= e^{Ah} = I + hA + \frac{(hA)^2}{2!} + \cdots \\ \Gamma &= \int_0^h e^{Ah} B \, \mathrm{d}x = A^{-1}(\Phi - I)B \end{aligned} \tag{4.2}$$

The discretization seen in Equation 4.2 is based on the principles of [Glad and Ljung, 2003]. The resulting matrices are shown in Equation 4.3.

$$\begin{aligned} \Phi &= \begin{bmatrix} 0.9008 & 0 & 0 & 0 \\ 0.1642 & 0.8267 & 0 & 0 \\ 0.0090 & 0.0935 & 0.8972 & 0 \\ 0.0006 & 0.0089 & 0.1660 & 0.8246 \end{bmatrix} \\ \Gamma &= \begin{bmatrix} 0.0836 & 0 \\ 0.0076 & 0 \\ 0.003 & 0.0137 \\ 0 & 0.0013 \end{bmatrix} \\ C &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ D &= \begin{bmatrix} 0 & 0 \end{bmatrix} \end{aligned} \tag{4.3}$$

4.2 Controller

From the results in the simulations, the LQG and MPC performed equally well in terms of closed loop response. The computational complexity for the MPC controller is higher so the LQG controller was deemed most fitting for the application. As in the simulations the LQG controller is comprised of a Kalman filter observer and a state feedback controller but when running on a physical setup, the discrete time alternatives had to be used.

Discrete Kalman filter: This work is based on the Matlab/Simulink source which is in turn based on the principles of [Franklin et al., 1997]. Using the discrete time state space model seen in Equation 4.4, the discrete Kalman filter in Equation 4.6 is obtained.

$$\begin{aligned} x[n+1] &= \Phi x[n] + \Gamma u[n] + Gw[n] \\ y[n] &= Cx[n] + Du[n] + Hw[n] + v[n] \end{aligned} \tag{4.4}$$

which is the discrete time alternative of the state space system in Equation 3.3 with the input $u[n]$, $w[n]$ and $v[n]$ is the discrete time white noise of the process respectively measurement where are

$$\begin{aligned} E[w[n]] &= E[w[n]] = 0 \\ E[w[n]w^T[n]] &= Q[n] \\ E[v[n]v^T[n]] &= R[n] \\ E[w[n]v^T[n]] &= N[n] \end{aligned} \tag{4.5}$$

where $Q[n]$ is the process covariance, $R[n]$ is the measurement covariance and $N[n]$ is process and measurement cross covariance. The state estimator form can be seen in Equation 4.6.

$$\hat{x}[n+1|n] = \Phi\hat{x}[n|n-1] + \Gamma u[n] + L[n](y[n] - C\hat{x}[n|n-1] - Du[n]) \tag{4.6}$$

The gain $L[n]$ is calculated trough the discrete algebraic Riccati equation

$$\begin{aligned} L[n] &= (\Phi P[n]C^T)(CP[n]C^T + \bar{R}[n])^{-1} \\ P[n+1] &= \Phi(P[n] - P[n]C^T(CP[n]C^T + \bar{R}[n])^{-1}CP[n])\Phi^T + \bar{Q}[n] \\ P[0] &= E[(x[0] - \hat{x}[0])(x[0] - \hat{x}[0])^T] \end{aligned} \tag{4.7}$$

where $\bar{R}[n]$ is the measurement noise covariance matrix and $\bar{Q}[n]$ is the process noise covariance matrix defined in the Equation 4.8.

$$\begin{aligned} \bar{Q}[n] &= GQ[n]G^T \\ \bar{R}[n] &= R[n] + HN[n] + N^T[n]H^T + HQ[n]H^T \\ \bar{N}[n] &= G(Q[n]H^T + N[n]) \end{aligned} \tag{4.8}$$

The G matrix is used to convert the scalar process covariance $Q[n]$, into a process variance matrix $\bar{Q}[n]$. As expected the process noise covariance is the same for each state which makes G the identity matrix of the same size as Φ. The H matrix describes how the feedtrough disturbance from $Q[n]$ affects the measurement noise covariance matrix and as no feedtrough is present H is zero. These are not the usual abbreviations or implementation that is taught in automatic control but the way that Matlab/Simulink implemented it. The process noise covariance and measurement noise covariance are measurable disturbances but instead using them to control the Kalman filter gain is useful for rejecting the time delay. On a reference value change a low Kalman filter gain is favourable to reduce oscillations due to time delay. After a while it is instead better to use a higher Kalman filter gain for finer tuning because a constant offset is not tolerable.

Discrete linear quadratic integral controller: Implemented in a similar fashion to the continuous time linear quadratic controller in the simulations which is based on the principals of [Young and Willems, 1972]. For the discrete time state space system,

$$\begin{aligned} x[n+1] &= Ax[n] + Bu[n] \\ y[n] &= Cx[n] \end{aligned} \tag{4.9}$$

the linear quadratic integral controller provides constant offset rejection and at the same time minimizes the cost function $J[u]$.

$$J[u] = \sum_{n=0}^{\infty} (z^T Qz + u^T Ru + 2z^T Nu) \tag{4.10}$$

where Q, R and N are the weights for the state, control signal and cross term. z is the state vector x with an added integrated state which is achieved by augmenting the state space matrices. The reference value r is added to introduce integral action to the controller.

$$z[n+1] = \begin{bmatrix} \Phi & 0 \\ -Ch & 1 \end{bmatrix} z[n] + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u[n] + \begin{bmatrix} 0 \\ h \end{bmatrix} r[n] = \Phi_a z[n] + \Gamma_a u[n] \tag{4.11}$$

h is process sample time which is used to form the discrete time integrator. The optimal state feedback gain on infinite horizon is calculated by

$$K = (\Gamma_a^T X \Phi)(\Gamma_a^T X \Gamma_a + R)^{-1} \tag{4.12}$$

where X is the solution to the discrete time algebraic Riccati equation.

$$\Phi_a^T X \Phi_a - X + \Phi_a^T X \Gamma (\Gamma_a^T X \Gamma_a + R)^{-1} \Gamma_a^T X \Phi_a + Q = 0 \tag{4.13}$$

The schematic setup can be seen in Figure 3.2. The feedback is taken from the estimator instead of the real process. This was done to reduce the start up time but this also introduces some issues with constant offset rejections. To generate the control signals to the pumps, the feedback gain matrix is multiplied by the estimated states seen in Equation 4.14.

$$\begin{bmatrix} PuA \\ PuB \end{bmatrix} = -Kz[n] \tag{4.14}$$

Control signal switching

As there is only one output from the process only one pump can be controlled at a time. While one pump is regulated by feedback the other pump runs at a constant speed by feedforward. In order to switch between the two modes, the gain feedback matrix K is changed. By setting a row in K to zeros it disables the feedback which can be seen in Equation 4.15.

$$\begin{aligned} K_A &= \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ K_B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -(k_3+k_4) & k_3 & k_4 & k_5 \end{bmatrix} \end{aligned} \tag{4.15}$$

One thing to note is that the B-concentrate only affects the last two states, as the infusion point is between the first and second chamber in Figure 2.1. The second element in the gain feedback matrix K_B assures that no offset on the control signal is present due to A-concentrates already added. When changing from A to B concentrate, the state vector $\hat{x}$ will be non-zero due to the influence of A concentrate. This means that the state feedback controller will start at a negative control signal (which is saturated) until the integrator can catch up. With the B-concentrate only affecting the last two states, the presence of electrolytes in the fluid path acts as a load disturbance. But by subtracting the offset from the control signal the load disturbance is dealt with without having to wait for the integrator to wind up. The pump running at constant speed still has to feed the signal to the Kalman estimator in order to properly track the states.

4.3 Results

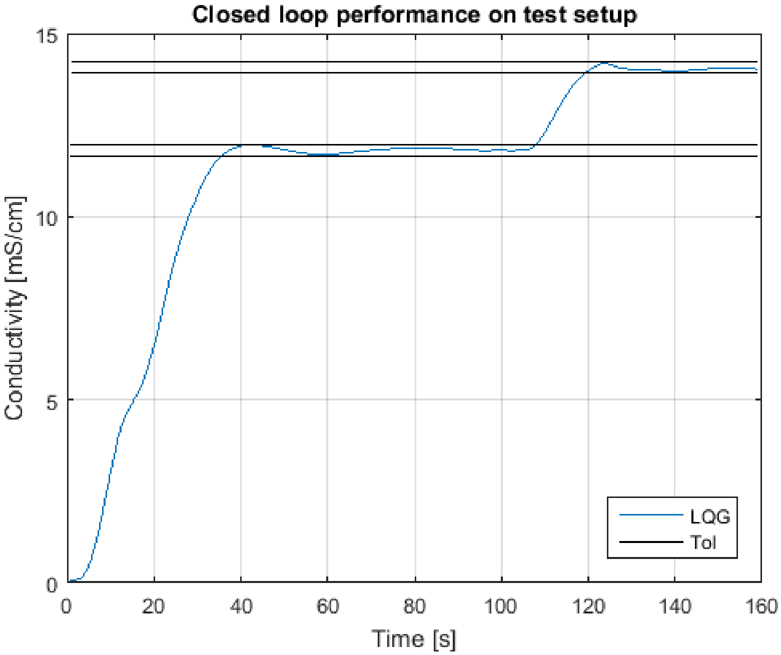

Figure 4.2 Comparsion of real closed loop performance

The start up sequence for the dialysis machine can be seen in Figure 4.2. Starting with the A-concentrate, it has to reache the reference value within a 1% tolerance. It then waits for 45 seconds as a safety check to verify that the right type of concentrate is used before B-concentrate starts. After this has been attained the B-concentrate pump starts and when the conductivity is stable treatment can begin. The LQG controller did it in around 36s and 14s for the A- and B-concentrate respectively. These results are relativity close to the results in the simulations. The increase in closed loop performance in the more advanced controller is most striking for the A-concentrate which is likely due to the longer time delay and larger time constants.

4.4 Model variations

Although the model performed well when tested, real world usage often adds more challenges. The ones stated below are the two major problems that were encountered while testing.

Flow changes

When treatment has started it is sometimes necessary to change the main flow. As a rule of thumb the main flow is usually set to twice the blood flow. For the pump in open loop control this only entails a scaling relative to new flow set-point. For the feedback controlled pump this results in a step change although the set-point is still the same. The Kalman filter does not account for a variable flow, but by scaling the control signal fed into the Kalman filter would trick the state estimator that the concentrations were changing and this would in turn make the state feedback controller react. In Figure 4.3 the green line represents the main fluid flow that changes from the nominal $500 ml/min$ to $800 ml/min$ and then down to $300 ml/min$, the blue line show how the feedback is effected and the red and yellow shows the unscaled control signal. The scaling of the control signal is the relation between the nominal starting flow and the new flow set-point. Even though the dynamics and time delay change it is still handled by the robustness of the controller seen in Figure 4.3 . This could be solved by using an extended Kalman filter (EKF) instead.

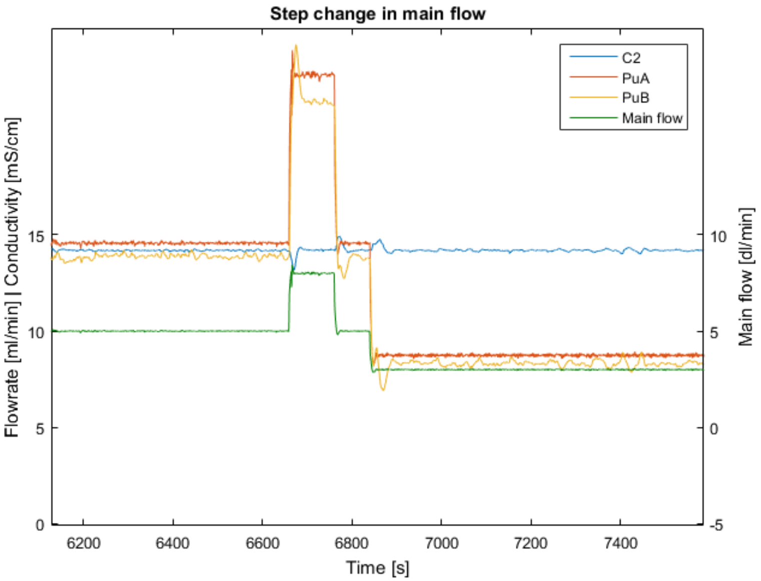

Figure 4.3 A change from 500 ml/min to 800 ml/min and then to $300 ml/min$

Start-up priming

Another issue related to model deviations is the priming sequences. The dialysis machine is always empty before treatment begins. The goal of priming is to fill the pick-up tubes with concentrate. The pick-up tubes are used to transport concentrate from the container to the pump. Independent of the type of concentrates used (either a dry product or a premixed can be used), it has to be primed before the controller can start. This is done by running the pumps at a fixed speed until the measurement cell can detect a small but significant change in conductivity which can be seen in Figure 4.4. So for the first part of the priming no electrolytes are injected and the Kalman filter will estimate wrongly. This is compensated by abusing the fact that the Kalman filter gain in Equation 4.7 is dependent on the relation of the process noise covariance Q and the measurement noise covariance R. By temporarily reducing the measurement noise covariance R more trust is put into the measurement than into the model and this allows the Kalman filter to quickly re-converge as the output from the process will still be close to zero. This means that independent of the pick up tube length or priming time it will not affect the start-up time.

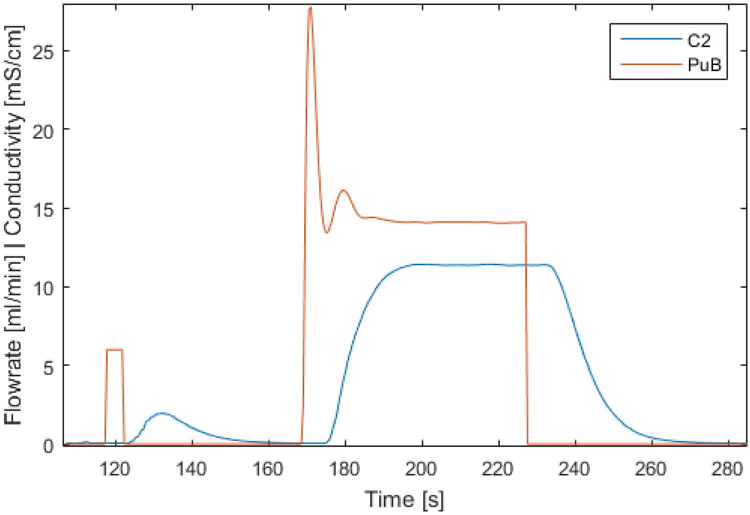

Figure 4.4 A priming sequence followed by a controller start

4.5 Alternative configurations

After some initial testing it was decided to remove the first mixing chamber on some configurations to cut costs. This does, however, cause a problem with the controller, reducing the system to a second order differential equation. The simple solution would be to produce different code packages to each configuration but this would in turn be problematic during production. Instead the controller was modified to alter the B matrix depending on configuration.

$$B = \begin{bmatrix} \frac{QK_A}{V1} & 0 \\ 0 & 0 \\ 0 & \frac{QK_B}{V3} \\ 0 & 0 \end{bmatrix} \to \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{QK_A}{V3} & \frac{QK_B}{V3} \\ 0 & 0 \end{bmatrix} \tag{4.16}$$

The changed state space equation becomes.

$$\hat{x}(t) = \begin{bmatrix} -\frac{Q}{V1} & 0 & 0 & 0 \\ \frac{Q}{V2} & -\frac{Q}{V2} & 0 & 0 \\ 0 & \frac{Q}{V3} & -\frac{Q}{V3} & 0 \\ 0 & 0 & \frac{Q}{V4} & -\frac{Q}{V4} \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{QK_A}{V3} & \frac{QK_B}{V3} \\ 0 & 0 \end{bmatrix} u(t) \tag{4.17}$$

The change alters the model to make the first two states uncontrollable with no input, setting them to zero removes their impact and effect on the control signal as they will never change.

Some configurations also make use of another conductivity measurement cell. It is placed between the first and second chamber and can be seen as greyed out in Figure 2.1. This allows for running both A and B concentrate in feedback control. This change affects the C matrix in Equation 2.3

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \to \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

resulting in the state space system

$$\begin{aligned} \hat{x}(t) &= \begin{bmatrix} -\frac{Q}{V1} & 0 & 0 & 0 \\ \frac{Q}{V2} & -\frac{Q}{V2} & 0 & 0 \\ 0 & \frac{Q}{V3} & -\frac{Q}{V3} & 0 \\ 0 & 0 & \frac{Q}{V4} & -\frac{Q}{V4} \end{bmatrix} x(t) + \begin{bmatrix} \frac{QK_A}{V1} & 0 \\ 0 & 0 \\ 0 & \frac{QK_B}{V3} \\ 0 & 0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \end{aligned} \tag{4.19}$$

This in turn, adds another integral state to the state vector z which is defined in Equation 3.10. Control signal separation is done by altering the state feedback gain K.

$$\begin{bmatrix} Pu_A \\ Pu_B \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 & k_{15} & 0 \\ 0 & -(k_{23}+k_{24}) & k_{23} & k_{24} & 0 & k_{26} \end{bmatrix} z \tag{4.20}$$

This setup is only needed when running dry product concentrates as the concentration can change depending on internal mixing ratios.

4.6 Code generation

After successfully operating on a test setup of a dialysis machine at Baxter's facilities it was deemed suitable to test on a real dialysis machine. The hardware in a dialysis machine does not support either Labview or Matlab/Simulink. So once again a choice had to be made concerning rewriting or auto-generate code. Both Labview and Matlab/Simulink offer C code generators but previous testing at Baxter had revealed that the Labview C coder has some issues. Matlab/Simulink offers a few different alternatives for code generation which have been used by the company previously to produce C code for other control loops with good results. Simulink coder provides a code generation advisor to help with optimization and compilation which turned out to be a quite useful feature. Some modifications of the Simulink model had to be done i.e. replacing source and sink blocks with inputs and outputs, declaring their respective input and output category. Generating code is favourable from a designer stand point but not as popular with programmers. The model can be seen in the Figure 4.5.

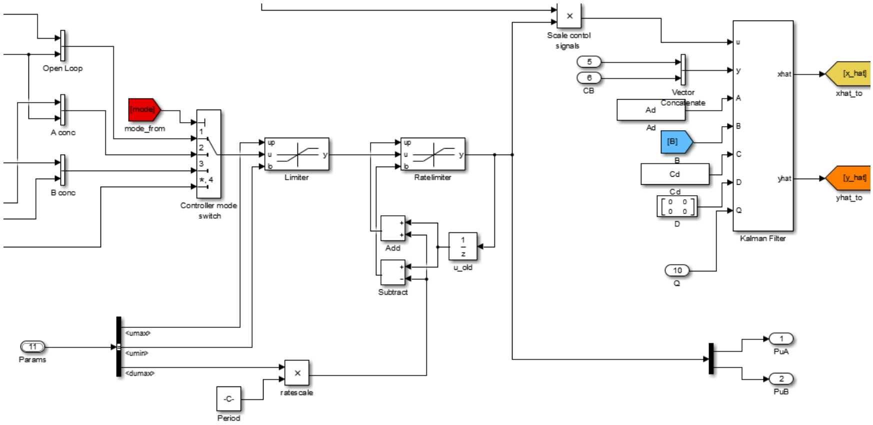

Figure 4.5 Part of the simulink model used for code generation

5

Conclusions

A model based conductivity controller is well suited for use in a dialysis machine. The cascading nature of the process means that a good model can be achieved in a short time by dividing it into first order systems. This is only one of several time consuming processes in dialysis machines but if the same work would be done for other systems it could possibly save more time.

An LQG controller performs similar to other more advanced algorithms like MPC in simulations. The speed-up it offers comes at the cost of robustness, where it is more sensitive to model deviations than simpler alternatives such as a basic PID.

A lot of time in this thesis work was spent on Labview programming to verify the results from the simulations on a test setup. By instead using a hardware target supporting Simulink Realtime would possibly speed up development by sizeable amount and from there on generate the embedded code for production.

As there is no baseline value to compare the results to it is hard to specify the total speed-up, but from the results of the simulations a PID properly tuned would be around 1 minute slower. 1 minute might not sound like a lot compared to the length of the treatment but the cost of a dialysis is highly related to the start-up as it is the only time when the dialysis machine needs an operator.

Bibliography

Åström, K. J. and T. Hägglund (2006). *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709.

Franklin, G. F., J. D. Powell, and M. L. Workman (1997). *Digital Control of Dynamic Systems (3rd Edition)*. Addison-Wesley. ISBN: 0201820544. URL: `http://www.amazon.com/Digital-Control-Dynamic-Systems-Edition/dp/0201820544%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0201820544`.

Glad, T. and L. Ljung (2003). *Reglerteori : flervariabla och olinjära metoder*. Studentlitteratur, Lund. ISBN: 9789144030036.

Jain, A., P Blake, P Cordy, and A. Garg (2012). “Global trends in rates of peritoneal dialysis.” *Journal of the American Society of Nephrology (JASN)* **23**:3, 533 –544 12p. ISSN: 1046-6673. URL: `\href{http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com.ludwig.lub.lu.se/login.aspx?direct=true&db=ccm&AN=108166614&site=eds-live&scope=site}`.

Johansson, R. (2008). *Predictive and Adaptive Control*. Department of Automatic Control, Lund University.

NI (2015). *National Instruments pxie-1078 real time system*. URL: `http://sine.ni.com/nips/cds/view/p/lang/sv/nid/209253` (visited on 01/03/2016).

Olsson, G. and C. Rosen (2005). *Industrial Automation: applications, structures and systems*. Media-Tryck, Lund University.

Young, P. C. and J. Willems (1972). “An approach to the linear multivariable servomechanism problem†”. *International journal of control* **15**:5, pp. 961–979.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS

Date of issue
June 2016

Document Number
ISRN LUTFD2/TFRT--5992--SE

Author(s)
Måns Fällman

Supervisor
Michael Pettersson, Baxter
Tore Hägglund, Dept. of Automatic Control, Lund University, Sweden
Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle
Model-Based Conductivity Control of Fluid Composition

Abstract

The purpose of this thesis was to reduce the start-up and restart time of a hemodialysis machine by targeting time consuming control loops. This was done by investigating different techniques of modelling to find the best combination of versatility and conformity to the process. Finding the best controller was done by evaluating different controllers in simulations with the model in MATLAB/Simulink by comparing closed loop performance and robustness. The results were verified by implementing the controller on a LabView realtime hardware setup.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-45

Recipient's notes

Security classification

http://www.control.lth.se/publications/