Report for bioreactor project

Darren Roos

CML 732

2019-11-16

Report for bioreactor project

Darren Roos
u15041604

Department of Chemical Engineering
University of Pretoria

CML 732

2019-11-16

Report for bioreactor project

Abstract

The instrumentation, modelling and control of a bioreactor that produces fumaric acid are investigated. It is found that the pH probe drifts at a rate of $0.013\,75\,\mathrm{h}^{-1}$ and that it has a time constant of around 130 s. The delta—sigma pulse width modulator for the temperature controller is found to have better characteristic than the current implementation. A non-linear model is developed to fit data with and without ethanol production. State estimation is used to drastically improve the ability to control the reactor by allowing the system to use information from the HPLC.

Contents

List of Figures

List of Tables

1 Background

This report documents the investigation into a bioreactor. The bioreactor produces fumaric acid through the aerobic fermentation of glucose by *Rhizopus oryzae*. The aims of the project are to: characterise the pH probe that is used in the reactor; investigate the performance of an alternative pulse width modulator for use in the temperature control system; develop a framework for the simulation, control and state estimation of the reactor in Python; develop a non-linear of model of the process that is sufficient for state estimation in control; and implement state estimation to provide better controllability of the system.

1.1 System overview

There are two phases in the process: a growth phase and production phase. During the growth phase, the fungus is grown on a PVC pipe within the cylindrical fermentation vessel (see Figure 1) with high concentrations of urea in a batch process. Thereafter, in the production phase, the vessel is rinsed out and continuous fermentation begins with lower concentrations of urea. Ethanol is an unwanted by-product that is formed, but its concentration is reduced by sparging with CO_2 (Naude, 2018).

Mixing in the vessel is caused by the continuous recycling of the reactor fluid from the base to the head of the reactor and as such the system is modelled as a CSTR. The direction of the recycle switches periodically to prevent the build-up of fungus and subsequent clogging of the pump. Volume is kept constant through the use of an overflow. The overflow's fumaric acid, glucose and ethanol concentrations is measured. The fermenter is dosed with small amounts of urea during the fermentation since it has been found by Naude (2018) that this increases the rate of fumaric acid and ethanol production.

Temperature and pH are measured on the recycle line. Characteristics of the pH probe, like its accuracy, drift, and response curve, are unknown. Knowing these qualities gives an indication of the amount of confidence one should have in a reading. These are especially important when it comes to pH measurements because of the logarithmically scaled degree of sensitivity required by the instrument, and the tendency of measuring devices to drift (McMillan, 1984). pH in the reactor can be increased or decreased by dosing with a $10\,\mathrm{mol\,L^{-1}}$ NaOH or $10\,\mathrm{mol\,L^{-1}}$ HCl mixture, respectively.

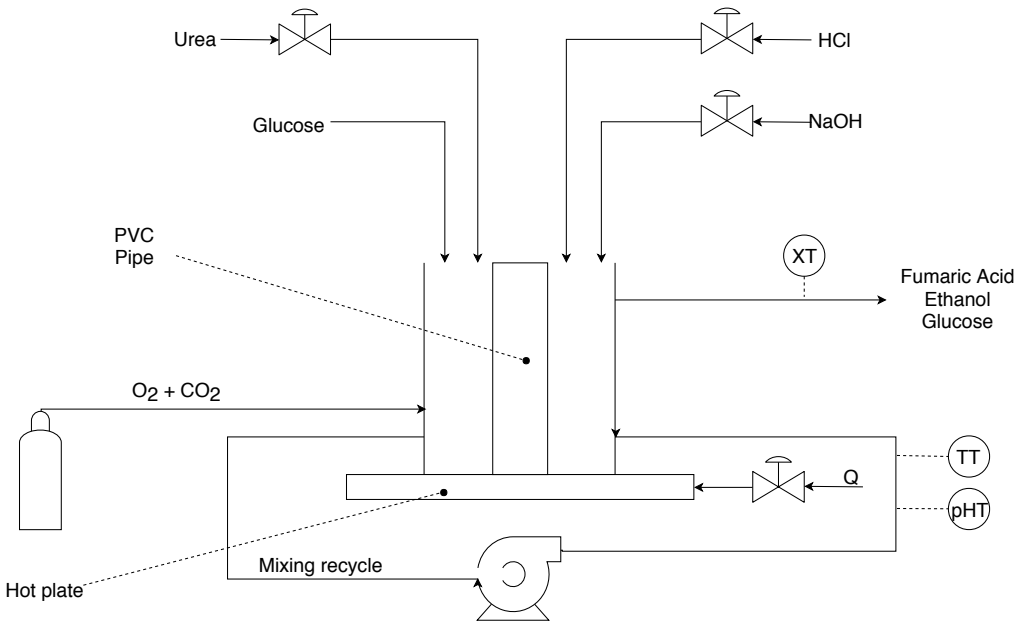

Figure 1: Basic diagram of the bioreactor indicating the important parts

The fermenter is placed on top of a hot plate in order to maintain the temperature. Unfortunately, the energy supplied to the element can only be manipulated by rotating a physical knob on the device and the control system can only turn the device on or off. As a result of this, the PID controller's output is passed thorough a pulse width modulator to the devices output.

According to Du and Robertson (2017), a pulse width modulator seeks to approximate an analog signal $f(t)$ using a digital signal. It outputs a binary signal $b(t)$ that takes on a high value $Q_{\max}$ or a low value $Q_{\min}$. For the sake of the approximation, it is important that:

$$Q_{\min} \leq f(t) \leq Q_{\max} \quad \forall t \tag{1}$$

For a given period P (not necessarily fixed), the PWM outputs $Q_{\max}$ for the first DP time and $Q_{\min}$ for the rest of the $(1 - D)P$ time, where D is the duty. During every period PWM seeks to find the duty such that:

$$\begin{aligned}\frac{1}{P}\int_t^{t+P} f(t)dt &= \frac{1}{P}\int_t^{t+P} b(t)dt \\ &= \frac{1}{P}\left[\int_t^{t+DP} Q_{\max} + \int_{t+DP}^{t+P} Q_{\min}dt\right] \\ &= DQ_{\max} + (1-D)Q_{\min}\end{aligned} \tag{2}$$

Stated more concretely, pulse width modulators output binary pulse streams such that width of the pulses cause the average value/power to approximate a desired analogue curve. It is desired to investigate the current pulse width modulation style used by the instrumentation as well as to research alternatives.

1.2 State estimation

The concentrations of fumaric acid, ethanol and glucose are not available on a continuous basis, but rather the reactor is sampled periodically (about once every 8-16 hours) and a High Performance Liquid Chromatographer (HPLC) is used to determine the concentrations. This leads to difficulties in implementing a control system which would not be able to react fast enough given the infrequent measurements.

A solution to this is to use state estimation as part of the control scheme (Hadj-Sadok and Gouze, 2001). State estimation can predict the hidden state of the concentrations, which can then be updated as samples come in from the HPLC. To do the state estimation a reliable model of the system is needed. The system is assumed has the form:

$$\begin{aligned}\dot{x} &= f(x,u) + \omega_x \\ y &= g(x,u) + \omega_y\end{aligned} \tag{3}$$

where ω_x is the state noise and ω_y is the measurement noise, f is the state transition function, and g is the state observation function. Both are assumed to be Gaussian with covariances Q_x and Q_y.

The state transition function is used to predict the future state of the system given the current state and input (Farina et al., 2016). The state observation function is used to determine the predicted system's measurable outputs given the current state (Wilken, 2015).

The state estimator that is used in this project is an Unscented Kalman Filter (UKF). A UKF is ideal because it works with non-linear functions and does not require gradient or Hessian information about the transition and observation functions. It works by sampling

the original Gaussian distribution with weighted samples called sigma points. The sigma points are chosen according to a version of scaled sigma points proposed by Merwe and Wan (2003). The point locations are given by:

$$\begin{aligned} \chi_0 &= \mu \\ \chi_i &= \mu + \sqrt{(L+\lambda)P} \quad \forall i = 1 \dots L \\ \chi_j &= \mu - \sqrt{(L+\lambda)P} \quad \forall j = L+1 \dots 2L \\ \lambda &= \alpha^2(L+\kappa) - L \end{aligned} \tag{4}$$

where α determines the spread of the points, κ is a parameter that scales the points towards the mean, and L is the dimensionality of state space. Merwe and Wan (2003) proposed the weights as:

$$\begin{aligned} \omega_0 &= \frac{\lambda}{L+\lambda} \\ \omega_i &= \frac{1}{2(L+\lambda)} \quad \forall i = 1 \dots 2L \end{aligned} \tag{5}$$

The sigma points are then passed through the nonlinear transition function. The new Gaussian distribution is fitted to the transformed sigma points.

1.3 Software

Thus far all the aims stated above have been given the relevant background, motivation and theory, except the aim regarding the development of a software framework for simulation, modelling, control and state estimation. This section aims to remedy this.

A well-developed software framework should allow the user to easily interact with the program, and should make it easy to code a new/replacement part. For example, one of the aims in this project is to make it easy for a different model of the system to be used. Object Oriented Programming (OOP) makes this almost trivial through its use of encapsulation and interfaces.

An object's interface is essentially how other programmers interact with it. If all objects that represent models of the system have the same methods with the same call signatures, then it becomes really easy to switch them out.

This kind of design is very useful in projects like those in CML, because it allows the next student to very easily improve on the work already done without having to redo all the parts.

2 Instrumentation: characterisation of the pH probe

pH measurement in this system is both important and difficult. It is important because the bacteria cannot live if the pH is too high or too low, and it is difficult because the experiments are long (often on the order of 200 h). The long experiment times leads to the problem of drift. Drift is the change in the measured value of a constant quantity over time. For example, if you have an oven at a constant true temperature of 100 °C and the temperature probe initially reads the true value, but in a week reads a couple of degrees off, then the measurement has drifted. Measurement drift in pH probes is caused by ions "leaking" into the reference electrode.

The probe is calibrated using a two point method with a pH 4 and a pH 7 buffer solution. The device then assumes a linear relationship between the pH and the measured mV value. This is illustrated in Figure 2.

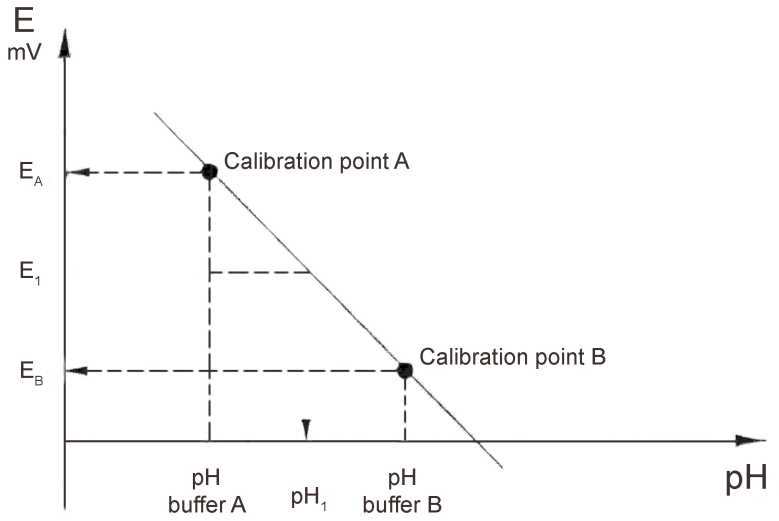

Figure 2: 2-point pH calibration

Once calibrated, the drift of the probe is then measured using the following method:

1. At $t = 0$: Measure the pH of four buffer solutions: 2, 4, 7 and 11 pH
2. Leave the probe in the pH 4 buffer solution. This is done to replicate the conditions of the bioreactor which runs at pH 5
3. At $t = 3\,\text{h}$, 8 h, and 24 h repeat the measurements of the four buffer solutions

Doing this and fitting linear drift curves gives Figure 3. The average drift is $0.013\,75\,\text{h}^{-1}$ which means that the drift over a 200 h run would be around 2.75 pH! However, the experiments only ran for 24 h and so it is unclear if the trend continues to be linear.

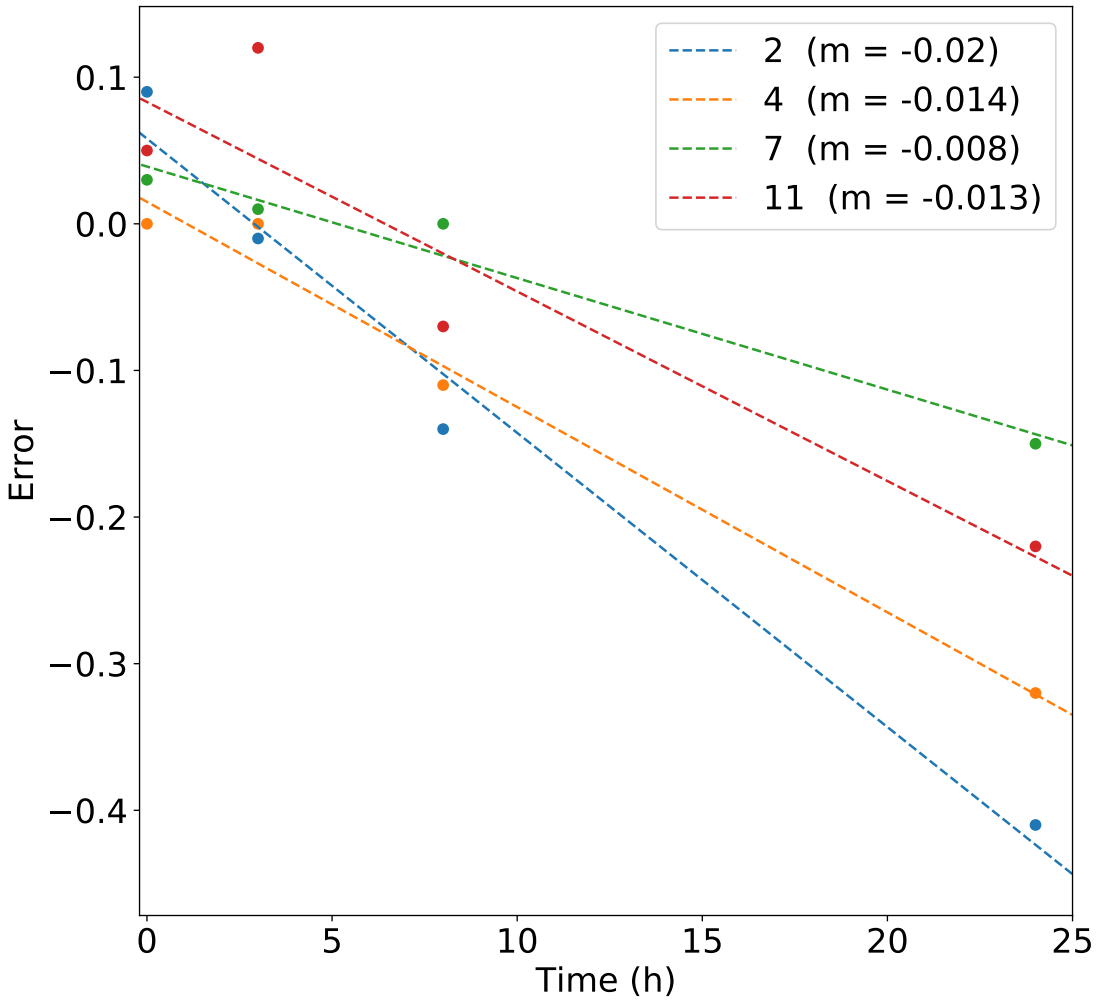

Figure 3: Drift curves for four pH values. The points of each data set has had its true value subtracted from it to make the data more comparable.

It is also important to determine the accuracy and linearity of probe. The accuracy is how close the probe measures to the true value of the system. Linearity refers to how closely the probe measures to the linear calibration curve. These characteristics can be determined by looking at the $t = 0$ values of the drift data.

The percentage error of this data is shown in Figure 4. Accuracy of the measurements is very good with the maximum error being 0.45 %. The measurement is also linear across the pH calibration range. This is seen by the fact that the errors for each pH is very low given that only two point were used for the linear calibration curve.

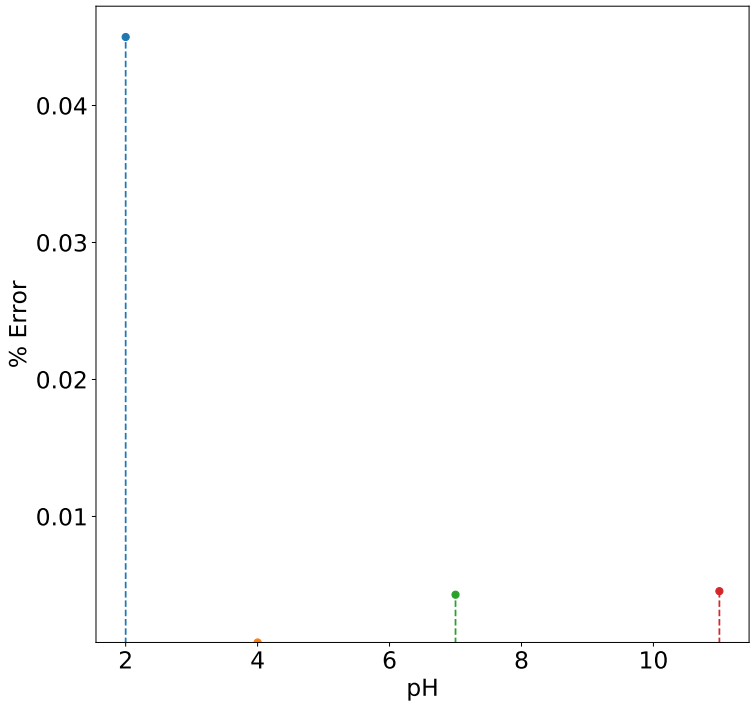

Figure 4: Percentage error of measured pH for calibrated probe

The dynamic characteristics of the pH probe need to be known as well because they affect how we should use online measurements. To this end, the time constant of the probe is determined. The probe's response is measured after calibrating the probe and then placing the probe into a buffer solution and sampling the response every second. This is done for pH 4 and 7. The results are shown in Figure 5. The estimated time constant is on the order or 130 s which is sufficiently fast enough since the pH of the reactor changes very slowly due to only small amount of fumaric acid being produced at a time (approximately $2.78 \times 10^{-5}\,\mathrm{g\,s^{-1}}$ as one can see later in Figure 8).

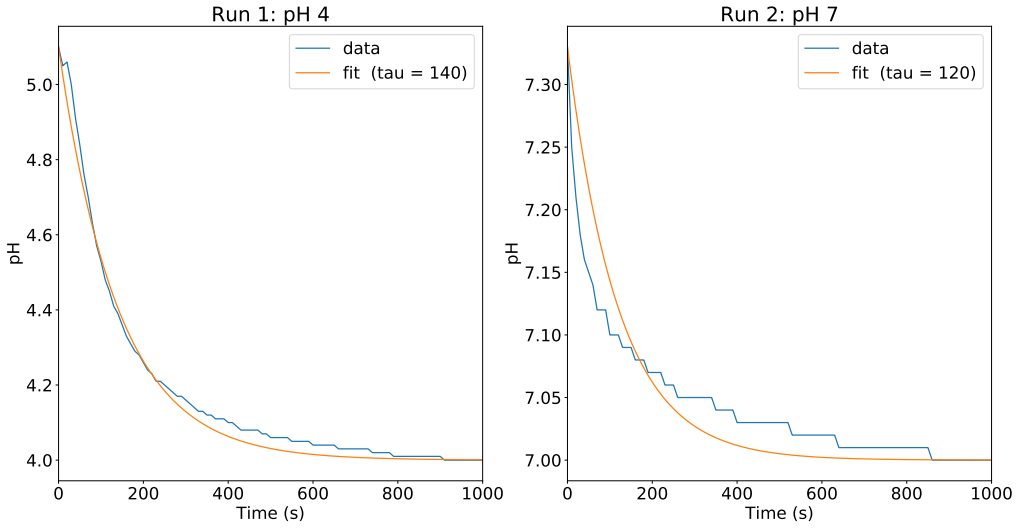

Figure 5: Dynamic response of the pH probe to step excitements

3 Instrumentation: pulse width modulator

For this subtask, the aim is to investigate the performance of the current pulse width modulator in use for the temperature controller against another pulse width modulator design.

The current PWM operates by selecting a period P and sampling the desired power output Q at the start of the period. It then selects the duty D such that the average power delivered for the period is the same as Q.

The alternate PWM is a delta—sigma design. It works by integrating the error between the desired signal and the binary signal. When the magnitude of the error surpasses some tolerance value E, the binary signal changes state.

3.1 Experimental set-up

Both modulators are given the input signal

$$u(t) = \frac{1}{2} + \frac{1}{4}\sin\left(\frac{t}{10}\right) \tag{6}$$

Performance is characterised by the squared error of the time averaged power of the output of the analog input u and the binary output q:

$$R = \int_0^\infty \left[\frac{1}{T}\int_0^T \left(u(t) - q(t)\right)^2 dt\right] dT \tag{7}$$

The performance is also weighted by the average time between state changes in the binary signal. Thus, in order for the test to be fair, the value of P for the current PWM and the value of E of the delta—sigma PWM should be chosen such that both signals have similar numbers of state changes per second. Figure 6 shows an example of such a run. The R values are 11.84 and 3.03 for the current and delta—sigma modulators, respectively.

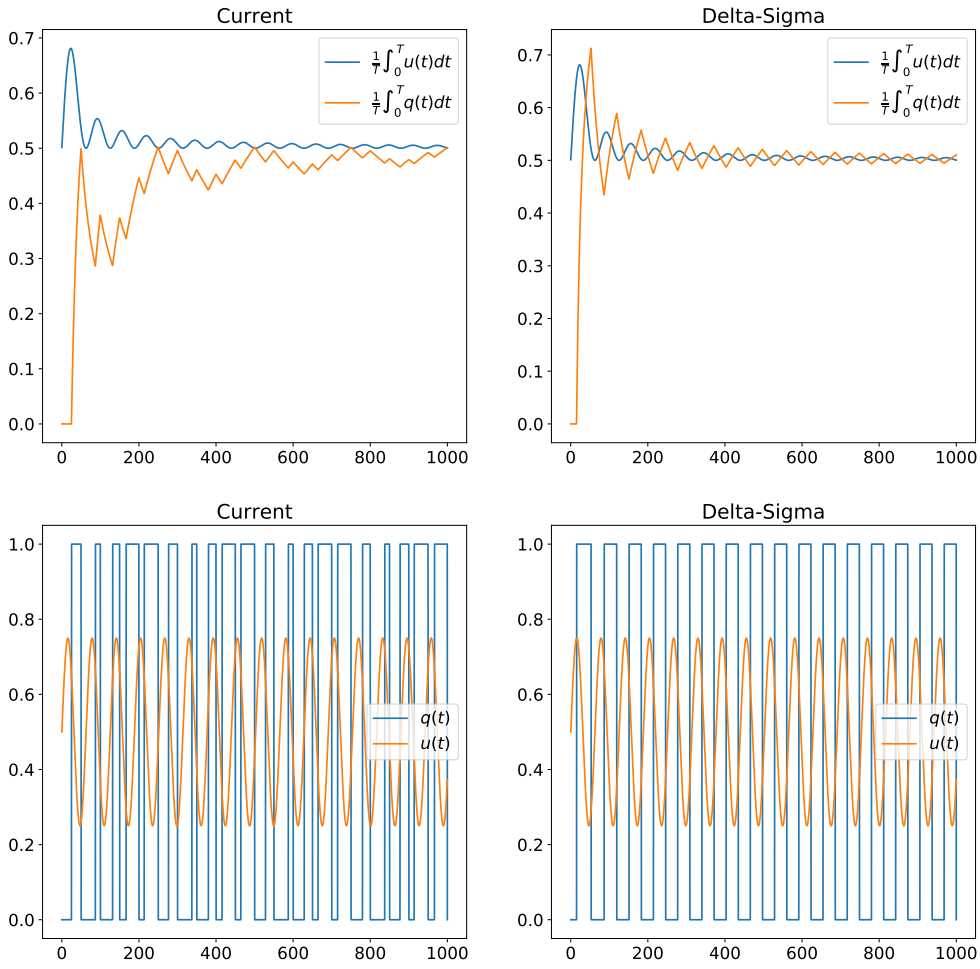

Figure 6: A single run of both modulators. $P = 38$ and $E = 4$. Both have a frequency of switching of 0.053 Hz

3.2 Results and discussion

Values for R and the state change frequency are found for several values of P and E. The result is shown in Figure 7. The figure makes it very clear that the delta—sigma modulator outperforms the current modulator at every frequency.

Figure 6 gives an indication as to why this is the case. One can see that the delta—sigma modulator's output is more regular (which is what one would expect given that the input signal is cyclic), and that it's "response" is faster.

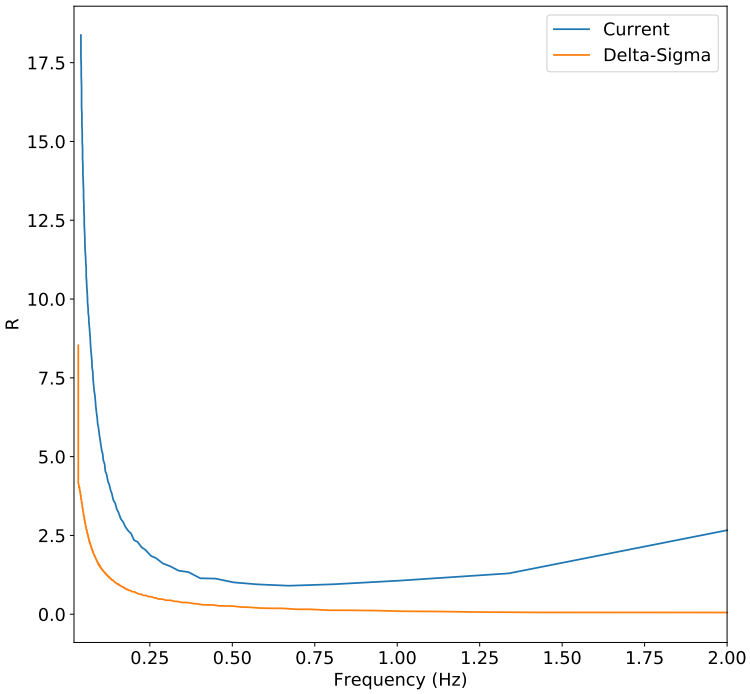

Figure 7: The error values R for both modulators as a function of the state switching frequency

4 Modelling

The model's shape for the system is developed for the data shown in Figure 8 and Figure 9, however, the parameters are chosen based only on the data in Figure 8 and the data in Figure 9 is used for comparison. It is important to note that the two sets of data are from completely different operating regimes. The data in Figure 8 are from a run where there is sufficient glucose for ethanol production, while the data in Figure 9 shows a run where there is insufficient glucose for ethanol production.

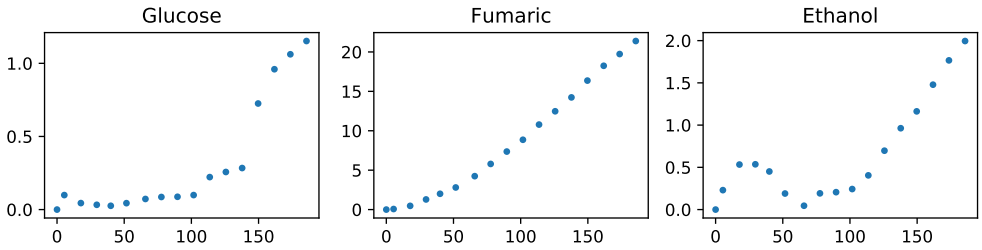

Figure 8: HPLC samples for bioreactor run with ethanol production

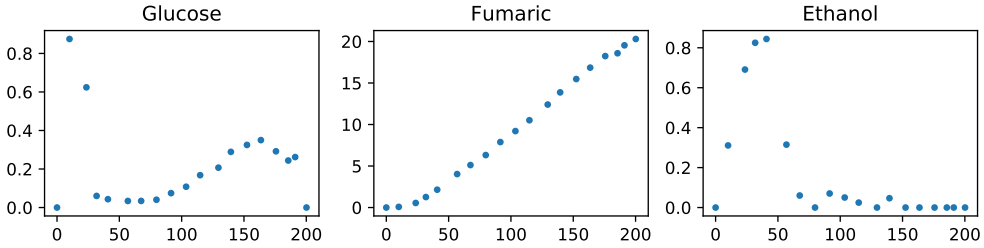

Figure 9: HPLC samples for bioreactor run without ethanol production

4.1 Model equations

Equations #1 through #9 in Table 1 [1] represent ordinary CSTR mole balances with the relevant terms set to zero for glucose, biomass, fumaric acid, ethanol, carbon dioxide, oxygen, nitrogen, hydrochloric acid and sodium hydroxide, respectively. Equations #12 and #13 are the volume balances (density is assumed constant due to low concentrations) for the liquid and gas phases, respectively. Equations #10 and #11 represent CSTR mole balances for modelled enzymes.

Enzyme Y represents the enzyme that exists within the bacteria from the growth phase that continues ethanol production. This enzyme causes the initial rise in ethanol concentration in both the run that has sufficient glucose for ethanol production and the one that does not. This is represented by the rate seen in Equation #35 that depends on the

[1] We will use #n to represent equations from Table 1

concentration of Y. Enzyme Z is an enzyme that helps model the conversion of ethanol to fumaric acid. It causes the decrease in ethanol production, however, it is important to note that the bacteria compensates for the conversion of fumaric acid from ethanol by producing less fumaric acid from glucose. This is represented by the rate seen in Equation #37 that depends on the concentration of Z. The normal increase of ethanol due to the presence of glucose is mathematically described in Equation #36. It is important to note that this mechanism is not fully understood and so this aspect of the model is built to fit the data rather than by using first principles.

Equations #14 through #24 represent the equations relating the concentration to the mole values. An important aspect to note in the concentrations is in the biomass concentration. There are unknown amounts of biomass in the reactor from the growth phase. There is no way to measure the amount of biomass until after the entire run has completed. This is because measurement requires drying the bacteria which kills them. Thus, when one runs the model simultaneously with a run one must guess/infer the biomass (more on this in the state estimation section).

Five bacterial reactions are modelled, excluding the enzymatic reaction that converts ethanol to fumaric acid for now. These reactions are seen in Equation 8. The first two reactions ($r_{FA,f}$ and $r_{E,f}$) are fermentation reactions and show the incomplete breakdown of glucose to fermentation products. The third reaction (r_{TCA}) condenses the relevant reactants and product of the TCA cycle into a single reaction. The fourth reaction (r_{resp}) takes place in the mitochondria of the bacteria where cellular respiration occurs. The last reaction (r_{bio}) shows how the bacteria produces more of itself by combining smaller molecules into larger ones that allow it to undergo mitosis.

$$\begin{aligned}
\text{Glucose} + 2\,\text{CO}_2 + 6\ \text{ATP} &\rightarrow 2\ \text{FA} + 2\,\text{H}_2\text{O} && \textit{(Fermenetation)} \\
\text{Glucose} &\rightarrow 2\ \text{Ethanol} + 2\,\text{CO}_2 + 2\ \text{ATP} && \textit{(Fermenetation)} \\
\text{Glucose} &\rightarrow 6\,\text{CO}_2 + 12\ \text{NADH} + 4\ \text{ATP} && \textit{(TCA cycle)} \\
\text{NADH} + \frac{1}{2}\text{O}_2 &\rightarrow \frac{7}{3}\text{ATP} && \textit{(Resperation)} \\
\text{Glucose} + \gamma\ \text{ATP} &\rightarrow 6\ \text{Biomass} + \alpha\text{CO}_2 + \beta\ \textit{NADH} && \textit{(Anabolism)}
\end{aligned} \tag{8}$$

The rates for $r_{FA,f}$ and $r_{E,f}$ are determined from concentration values as seen in Equations #25 and #26, respectively. The $\frac{C_G}{0.01+C_G}$ term seen in both equations is known as a Monod term, which is used to decrease the rate when the concentration of glucose s low. The subtraction of the r_z term in Equation #25 is done to take into account the bacteria compensating for fumaric acid production from ethanol by reducing fumaric acid

production from glucose.

Using the rates of the reactions in Equation 8, one can determine the rates of the individual components as is done for glucose, biomass, fumaric acid, ethanol, carbon dioxide and oxygen in Equations #27 through #33, respectively.

Biomass requires energy to reproduce (as seen by r_{bio}), but it also requires energy to maintain its current state. This energy requirement θ can be dependent on the concentrations in the cell as seen in Equation #34. This together with Equation 8 gives rise to the energy and redox balances in Equations #40 and #41.

The pH of the system adds more complexity and non-linearity to the system. This system has the presence of a weak diprotic acid (fumaric acid), which makes pH calculations more complex. In Equations #42 through #46, on sees the equilibrium relationships of the different compounds. The normal format of an acid equilibrium equation is

$$\begin{aligned} K &= \frac{C_{\text{H}^+} C_{A^-}}{C_{A,\text{undissolved}}} \\ C_A &= C_{A,\text{undissolved}} + C_{A,\text{dissolved}} \\ C_{A,\text{dissolved}} &= C_{A^-} \end{aligned} \tag{9}$$

but for the sake of brevity, these have been combined where relevant into

$$K = \frac{C_{\text{H}^+} C_{A^-}}{C_A - C_{A^-}} \tag{10}$$

Equation #47 is the charge balance for the system. Because we are working in a neutral charge system, the number of moles of positive and negative ions must be equal. Equation #48 is the familiar pH calculation.

The energy effects of the system are taken into account in Equation #49. One can see a directly proportional relationship between the heater input and the rate of change of the temperature of the system. While in reality there are other delays (like the plate needing to be heated up), these are considered negligible. There is also a directly proportional relationship between the rate f change of the system's temperature and the difference between the system's temperature and the surrounding temperature. This takes into account convection heat losses and any minor conduction heat losses due to contact with rubber pipes, etc.

Table 1: Model equations

#	Equations	Inputs	Outputs	Parameters
Mole balances				
1)	$\frac{dN_G}{dt} = F_{G,in}C_{G,in} - F_{out}C_G - r_G C_X V$	$F_{G,in}$, $C_{G,in}$, F_{out}	N_G, C_G, r_G, C_X, V	
2)	$\frac{dN_X}{dt} = r_X C_X V$		N_X, r_X	
3)	$\frac{dN_{FA}}{dt} = -F_{out}C_{FA} + r_{FA}C_X V$		N_{FA}, C_{FA}, r_{FA}	
4)	$\frac{dN_E}{dt} = -F_{out}C_E + r_E C_X V$		N_E, C_E, r_E	
5)	$\frac{dN_{\mathrm{CO_2}}}{dt} = F_{\mathrm{CO_2},in}C_{\mathrm{CO_2},in} - F_{out,g}C_{\mathrm{CO_2}} + r_{\mathrm{CO_2}}C_X V$	$F_{\mathrm{CO_2},in}$, $C_{\mathrm{CO_2},in}$, $F_{out,g}$	$N_{\mathrm{CO_2}}$, $C_{\mathrm{CO_2}}$, $r_{\mathrm{CO_2}}$	
6)	$\frac{dN_{\mathrm{O_2}}}{dt} = F_{\mathrm{O_2},in}C_{\mathrm{O_2},in} - F_{out,g}C_{\mathrm{O_2}} - r_{\mathrm{O_2}}C_X V$	$F_{\mathrm{O_2},in}$, $C_{\mathrm{O_2},in}$	$N_{\mathrm{O_2}}$, $C_{\mathrm{O_2}}$, $r_{\mathrm{O_2}}$	
7)	$\frac{dN_N}{dt} = F_{N,in}C_{N,in} - F_{out}C_N - \delta r_{\mathrm{X}} C_X V$	$F_{N,in}$, $C_{N,in}$	N_N, C_N	δ
8)	$\frac{dN_{\mathrm{HCl}}}{dt} = -F_{out}C_{\mathrm{HCl}}$		N_{HCl}, C_{HCl}	
9)	$\frac{dN_{\mathrm{NaOH}}}{dt} = F_{\mathrm{NaOH},in}C_{\mathrm{NaOH},in} - F_{out}C_{\mathrm{NaOH}}$	F_{NaOH}, $C_{\mathrm{NaOH},in}$	N_{NaOH}, C_{NaOH}	
10)	$\frac{dN_z}{dt} = -190 * r_z C_X V$		N_z, r_z	
11)	$\frac{dN_y}{dt} = -95 * r_y C_X V$		N_y, r_y	
12)	$\frac{dV}{dt} = F_{G,in} + F_{N,in} + F_{M,in} + F_{\mathrm{NaOH},in} - F_{out}$	$F_{M,in}$		
13)	$\frac{dV_g}{dt} = F_{\mathrm{CO_2},in} + F_{\mathrm{O_2},in} - F_{out,g}$		V_g	

#	Equations	Inputs	Outputs	Parameters
	Concentrations			
14)	$C_G = \frac{N_G}{V}$			
15)	$C_X = \frac{N_X}{V}$			
16)	$C_{FA} = \frac{N_{FA}}{V}$			
17)	$C_E = \frac{N_E}{V}$			
18)	$C_{\mathrm{HCl}} = \frac{N_{\mathrm{HCl}}}{V}$			
19)	$C_{\mathrm{NaOH}} = \frac{N_{\mathrm{NaOH}}}{V}$			
20)	$C_z = \frac{N_z}{V}$		C_z	
21)	$C_y = \frac{N_y}{V}$		C_y	
22)	$C_N = \frac{N_N}{V}$			
23)	$C_{\mathrm{CO_2}} = \frac{N_{\mathrm{CO_2}}}{V_g}$			
24)	$C_{\mathrm{O_2}} = \frac{N_{\mathrm{O_2}}}{V_g}$			
	Basic rates			
25)	$r_{FA,f} = 15 \times 10^{-3} \frac{C_G}{0.01+C_G} - 0.5 r_Z$		$r_{FA,f}$	
26)	$r_{E,f} = 15 \times 10^{-3} \frac{C_G}{0.01+C_G} - 0.5 r_Z$		$r_{E,f}$	
27)	$r_{\mathrm{bio}} = k_b C_N$		r_{bio}	k_b
28)	$r_G = -r_{FA,f} - r_{\mathrm{TCA}} - r_{E,f} - r_{\mathrm{bio}}$		r_{TCA}	
29)	$r_X = 6 r_{\mathrm{bio}}$			
30)	$r_{FA} = 2(r_{FA,f} + 0.5 r_z)$			
31)	$r_E = (r_{1i} + r_{2i} + r_d)(\frac{C_G}{0.01+C_G})$		$r_{1i} r_{2i}$, r_d	
32)	$r_{\mathrm{CO_2}} = -2 r_{FA,f} + 6 r_{\mathrm{TCA}} + 2 r_{E,f} + \alpha r_{\mathrm{bio}}$			α
33)	$r_{\mathrm{O_2}} = 0.5 r_{\mathrm{resp}}$		r_{resp}	
34)	$r_{\theta,f} = (\theta)(\frac{C_G}{1 \times 10^{-3}+C_G})$		$r_{\theta,f}$	θ

#	Equations	Inputs	Outputs	Parameters
	Enzymatic rates			
35)	$r_{1i} = k_1 C_y$			k_1
36)	$r_{2i} = k_2$			k_2
37)	$r_d = k_3 C_z$			k_3
38)	$r_z = r_d + r_{2i}$			
39)	$r_y = r_d + r_{1i}$			
	ATP Balance			
40)	$r_{FA,f} + \gamma r_{\text{bio}} + r_{\theta,f} = 4r_{\text{TCA}} + \frac{7}{3} r_{\text{resp}} + 2r_{E,f}$			γ
	NADH Balance			
41)	$12 r_{\text{TCA}} + \beta r_{\text{bio}} = r_{\text{resp}}$			β
	pH calculations			
42)	$K_{FA} = \frac{C_{\text{H}^+} C_{FA^-}}{C_{FA} - C_{FA^-}}$		C_{H^+}, C_{FA^-}	K_{FA}
43)	$K_{FA^-} = \frac{C_{\text{H}^+} C_{FA^{2-}}}{C_{FA^-}}$		$C_{FA^{2-}}$	K_{FA^-}
44)	$K_{\text{HCl}} = \frac{C_{\text{H}^+} C_{\text{Cl}^-}}{C_{\text{HCl}} - C_{\text{Cl}^-}}$		C_{Cl^-}	K_{HCl}
45)	$K_{\text{NaOH}} = \frac{C_{\text{Na}^+} C_{\text{OH}^-}}{C_{\text{NaOH}} - C_{\text{Na}^+}}$		C_{Na^+}, C_{OH^-}	K_{NaOH}
46)	$K_w = C_{\text{H}^+} C_{\text{OH}^-}$			K_w
47)	$C_{\text{H}^+} + C_{\text{Na}^+} = C_{FA^-} C_{FA^{2-}} + C_{\text{Cl}^-} + C_{\text{OH}^-}$			
48)	$pH = -\log C_{\text{H}^+}$		pH	
	Temperature calculations			
49)	$\frac{dT}{dt} = 4.5Q - 0.25(T - T_{amb})$	Q, T_{amb}	T	

4.2 Model fits

Once the dynamics have been modelled, the next step is to fit the parameters to the data. This is done by looking up realistic values for parameters that have them: α, θ, γ and β all have values taken from our third year bioreaction engineering course. The values for the equilibrium constants are just physical constants and so were looked up. The values for the k_i parameters were tuned to give the desired shapes. Final values that are used can be seen in Table 2.

Table 2: Parameters for nonlinear model

Parameter	Value
δ	0.2
α	0.1
θ	0.1 $\frac{\text{mol ATP}}{\text{mol X}}$
k_1	3.1304×10^{-3}
k_2	1.1598×10^{-3}
k_3	3.2609×10^{-3}
γ	1.8 $\frac{\text{mol ATP}}{\text{mol bio}}$
β	0.1 $\frac{\text{mol NADH}}{\text{mol bio}}$
pK_{FA}	-3.03
pK_{FA^-}	4.44
pK_{HCl}	8.08
pK_{NaOH}	0.56
pK_w	-14

Figure 10 shows the response of the model to the input data that produced the HPLC samples shown. The curve fits the fumaric acid and ethanol data very well. It does not provide a very good fit of the glucose data, but does have the general shape correct. It is very difficult to get very accurate fits because the flux model of the bacteria is unknown. However, the model is good enough to be used for state estimation and control.

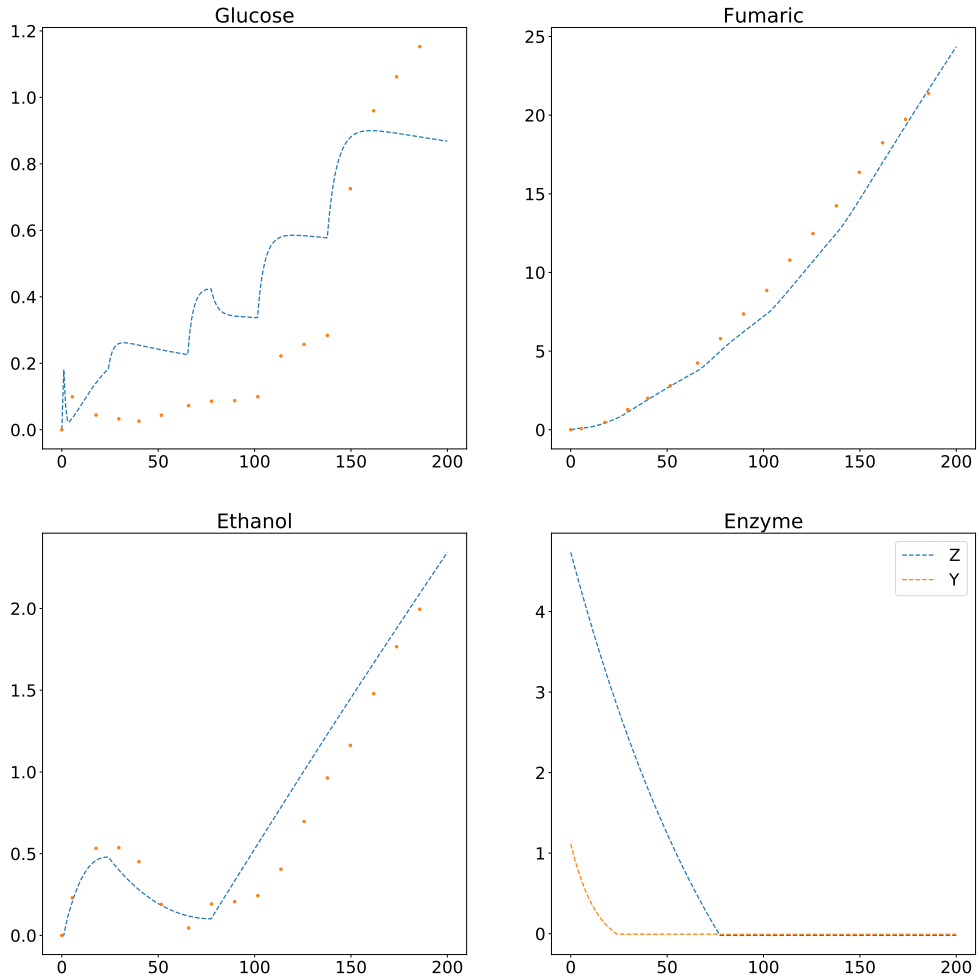

Figure 10: Model and HPLC samples for bioreactor run with ethanol production

The model is compared against the data from the case that has no ethanol production. Results are seen in Figure 11. The model fits the fumaric acid data very well, but does not predict the lack of ethanol production. This is because the model does not take into account the different regimes of the bacteria. At the time of writing, Reuben Swart (in the Department of Chemical Engineering), is doing his Master's on the concentration of glucose at which the bacteria stop producing ethanol. The model does still predict the general shape of the glucose curve again. From the above results, the model can be seen to be imperfect, but good enough for control purposes.

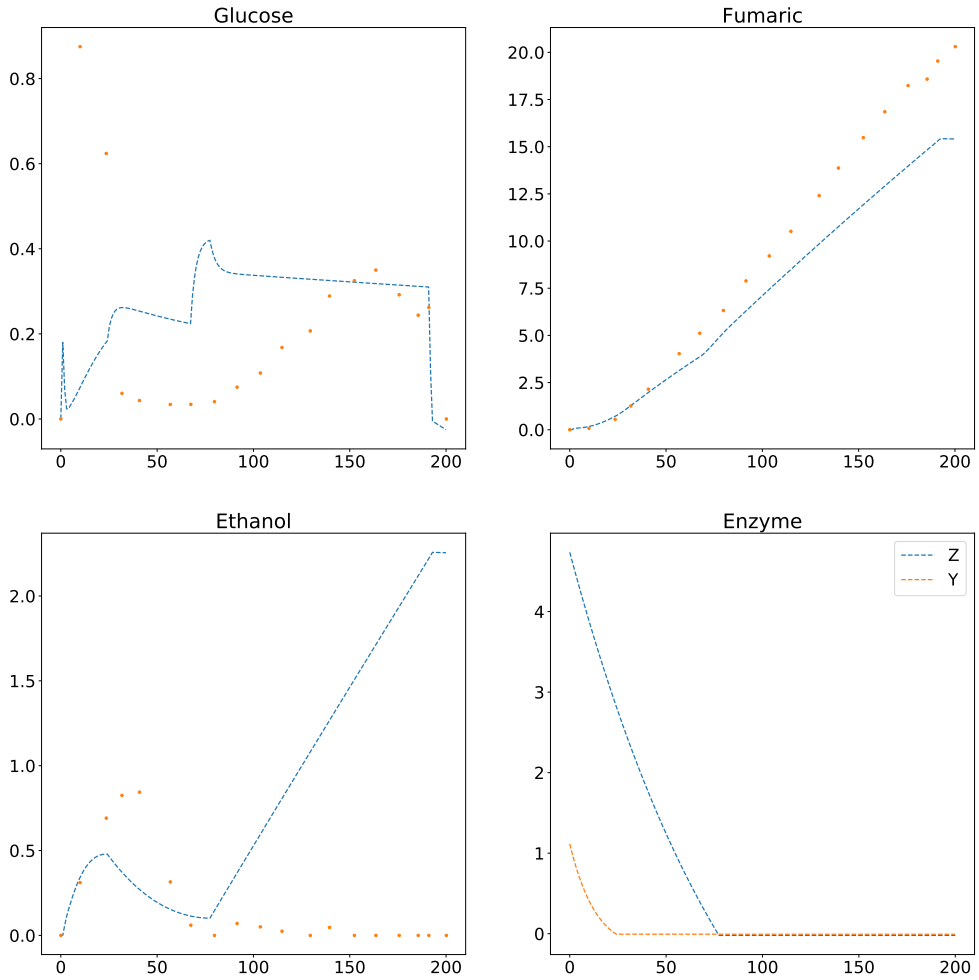

Figure 11: Model and HPLC samples for bioreactor run without ethanol production

5 Control: state estimation

Now that we have a model of the system, we can proceed to build the state estimator for use in the control system. An Unscented Kalman Filter (UKF) works as shown in Figure 12. It selects a set of points and weights (together known as sigma points) to approximate the current Gaussian. Then these points are put through the state transition function f (g in the figure) and used to approximate the new Gaussian. It is important to note that the red line of the actual Gaussian is unknown and so one uses the sigma points to predict it.

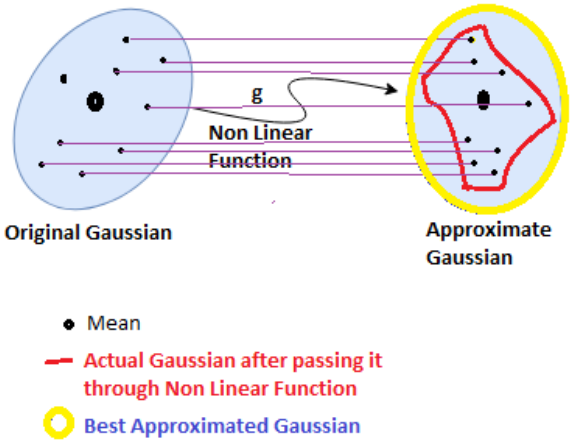

Figure 12: Visual representation of how the UKF works (Chadha, 2018)

As mentioned before, the sigma points are chosen according to a version of scaled sigma points proposed by Merwe and Wan (2003). Their method is shown in Equation 4 and Equation 5.

However, they did not realise that for any $\alpha \leq 1$, their weights no longer work since they give negative weights for ω_0. As far as the author of this document could tell, this is not stated anywhere by the original authors. Thus, for this work (where it is required that $\alpha \leq 1$), the weights are changed to be the unscaled versions:

$$\begin{aligned} \omega_0 &= \frac{\kappa}{L+\kappa} \\ \omega_i &= \frac{1}{2(L+\kappa)} \quad \forall i = 1 \ldots 2L \end{aligned} \tag{11}$$

The filter continues in from the sigma points into the familiar pattern of a prediction step and an update step. The prediction step us used to estimate the future values of the states using the present values. It works by mapping the sigma points χ_i into the updated space by means of the state transition function $f(\chi_i)$. The mean and covariance of the

new points is calculated. It is important to remember to add the uncertainty created by the state/model noise. This process is captured mathematically as

$$\begin{aligned} \mathcal{Z}_i &= f(\chi_i, u_{k+1}) \\ \mu_{k+1|k} &= \sum_{i=0}^{2n} \omega_i \mathcal{Z}_i \\ Q_{k+1|k} &= Q_x + \sum_{i=0}^{2n} \omega_i (\mathcal{Z}_i - \mu_{k+1})(\mathcal{Z}_i - \mu_{k+1})^T \end{aligned} \tag{12}$$

The updated step proceeds by moving the sigma points into the measurement space:

$$\begin{aligned} \hat{\mathcal{Z}}_i &= g(\chi_i, u_{k+1}) \\ \hat{\mu}_{k+1|k} &= \sum_{i=0}^{2n} \omega_i g \hat{\mathcal{Z}}_i \\ \hat{Q}_{k+1|k} &= Q_y + \sum_{i=0}^{2n} \omega_i (\hat{\mathcal{Z}}_i - \hat{\mu}_{k+1|k})(\hat{\mathcal{Z}}_i - \hat{\mu}_{k+1|k})^T \end{aligned} \tag{13}$$

In order to calculate prediction error, we need to calculate the cross-correlation between sigma points in state space and sigma points in the measurement space:

$$\hat{R}_{k+1|k} = \sum_{i=0}^{2n} \omega_i (\chi_i - \mu_{k+1|k})(\hat{\mathcal{Z}}_i - \hat{\mu}_{k+1|k})^T \tag{14}$$

This is then used to construct the Kalman gain

$$K = \hat{R}_{k+1|k} \hat{Q}^{-1}_{k+1|k} \tag{15}$$

which is used for the completion of the update step:

$$\begin{aligned} \mu_{k+1|k+1} &= \mu_{k+1|k} + K(z - \hat{\mu}_{k+1|k}) \\ Q_{k+1|k+1} &= (I - K\hat{R}_{k+1|k}) Q_{k+1|k} \end{aligned} \tag{16}$$

where z is the measurement/observation. A working, tested and robust UKF implementation in a Python package called `filterpy` is used in the software. An adjusted class for the updated sigma point method is implemented using the `filterpy` version as a starting point.

All states in the model in Table 1 are predicted. The measurement space consists of the three HPLC concentrations of the glucose, fumaric acid and ethanol. It is important that the covariances (and hence the standard deviations) of the state and measurement noise is chosen well. These values should provide realistic insight into the uncertainty of these values. These values are shown in Table 3.

Table 3: Covariance of state and measurement noises

State	Covariances	Standard deviation
N_G	1×10^{-6}	$0.18\,\mathrm{g\,L^{-1}}$
N_X	1×10^{-3}	$0.78\,\mathrm{g\,L^{-1}}$
N_{FA}	1×10^{-5}	$0.37\,\mathrm{g\,L^{-1}}$
N_E	1×10^{-4}	$0.46\,\mathrm{g\,L^{-1}}$
$N_{\mathrm{CO_2}}$	1×10^{-5}	$3.16 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$
$N_{\mathrm{O_2}}$	1×10^{-5}	$3.16 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$
N_N	1×10^{-5}	$0.19\,\mathrm{g\,L^{-1}}$
N_{HCl}	1×10^{-5}	$0.11\,\mathrm{g\,L^{-1}}$
N_{NaOH}	1×10^{-5}	$0.13\,\mathrm{g\,L^{-1}}$
N_Z	1×10^{-2}	$0.1\,\mathrm{mol\,L^{-1}}$
N_Y	1×10^{-2}	$0.1\,\mathrm{mol\,L^{-1}}$
V	1×10^{-5}	$3.16 \times 10^{-3}\,\mathrm{L}$
V_g	1×10^{-5}	$3.16 \times 10^{-3}\,\mathrm{L}$
T	1×10^{-1}	$0.32\,^{\circ}\mathrm{C}$
Measurement	**Covariance**	**Standard deviation**
N_G	1×10^{-12}	$1.8 \times 10^{-4}\,\mathrm{g\,L^{-1}}$
N_{FA}	1×10^{-12}	$1.16 \times 10^{-4}\,\mathrm{g\,L^{-1}}$
N_E	1×10^{-12}	$4.6 \times 10^{-5}\,\mathrm{g\,L^{-1}}$

The value of the glucose content as seen in Figure 8 and Figure 9 is around $0.5\,\mathrm{g\,L^{-1}}$, thus a standard deviation of $0.18\,\mathrm{g\,L^{-1}}$ aptly expresses the uncertainty we see in Figure 11. Similar arguments are made for the fumaric acid and ethanol values of $0.78\,\mathrm{g\,L^{-1}}$ and $0.37\,\mathrm{g\,L^{-1}}$, respectively. The amount of biomass in most runs varies between $0.8\,\mathrm{g\,L^{-1}}$ and $2\,\mathrm{g\,L^{-1}}$. Since we are very uncertain of the value in this range a value of $0.78\,\mathrm{g\,L^{-1}}$ is appropriate. The low level of uncertainty expressed in the enzyme values are due to the fact that they were tuned to give good fits, and so we are confidant in those values. Similarly, the volumes of the gas and liquid phases remain more or less constant, and so we are not uncertain about their values. The measurement standard deviations are small because we assume that they are very accurate and contain very little noise.

Three very important aspects regarding the implementation of the state estimator in this system: firstly, the prediction step needs to be done frequently so that an online estimate exists; secondly, the update steps do not happen ar regular intervals; and thirdly, when an update occurs it contains information about the system some time in the past due to the delay between sampling the system and receiving the HPLC results.

Fortunately, the design of the UKF in `filterpy` easily handles the first two cases by allowing the user to call the predict and update steps independently. The third case is solved by storing the values of $\mu_{k|k}$ and $Q_{k|k}$, and then when an update occurs, the user can enter the time at which the sample was taken. The software then backdates the update to the correct time by resetting the state estimator's $\mu_{k|k}$ and $Q_{k|k}$ at the time of the sample and performs the update. It then performs all the necessary prediction steps up until the current time.

The results of the runs with and without ethanol production with state estimation are shown in Figure 13 and Figure 14, respectively. The addition of state estimation makes a huge difference to the predictions in the run without ethanol production. It is clear that using the state estimated mean in a controller would be much better than using the model.

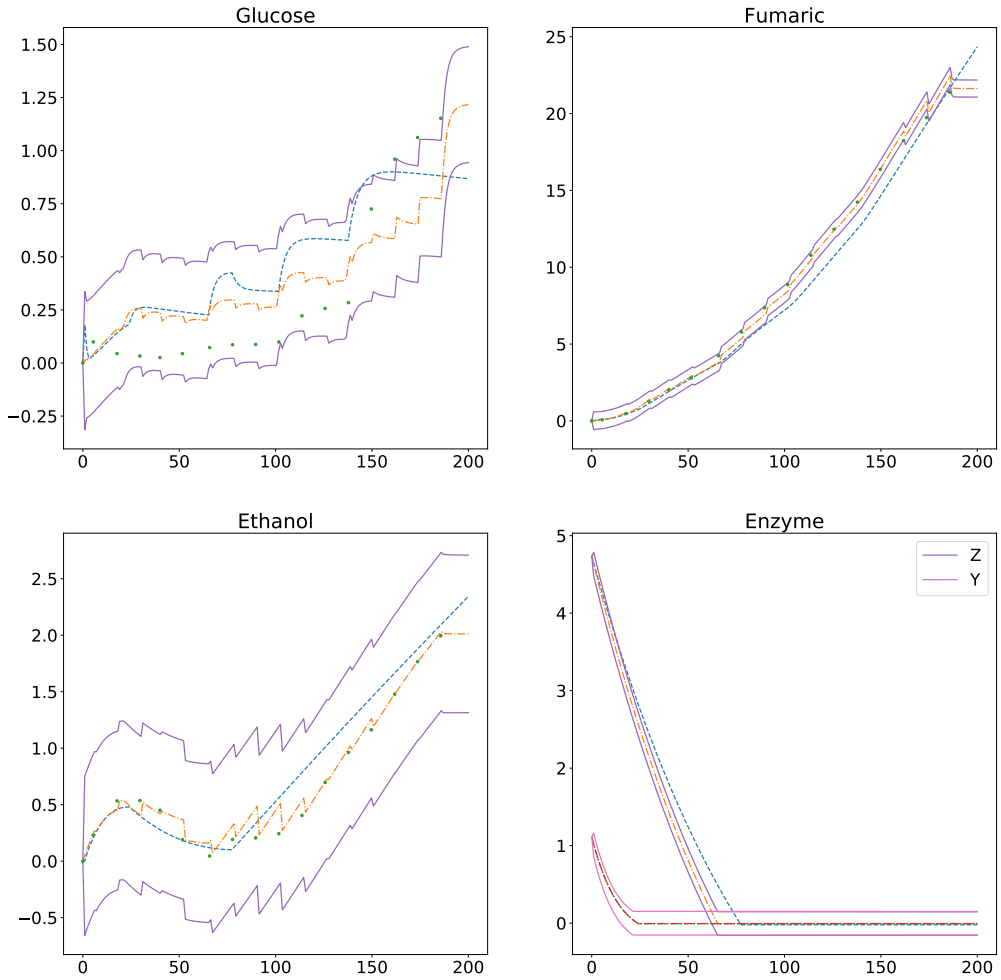

Figure 13: Model, HPLC samples and 90 % confidence intervals of state estimation average for bioreactor run with ethanol production. The solid lines represent the boundaries of the confidence region and the dotted lines represent the model predictions without any state estimation. The dotdash lines represent the mean of the state estimator. In the enzyme graph the C_Z and C_Y confidence lines are distinguished in the legend.

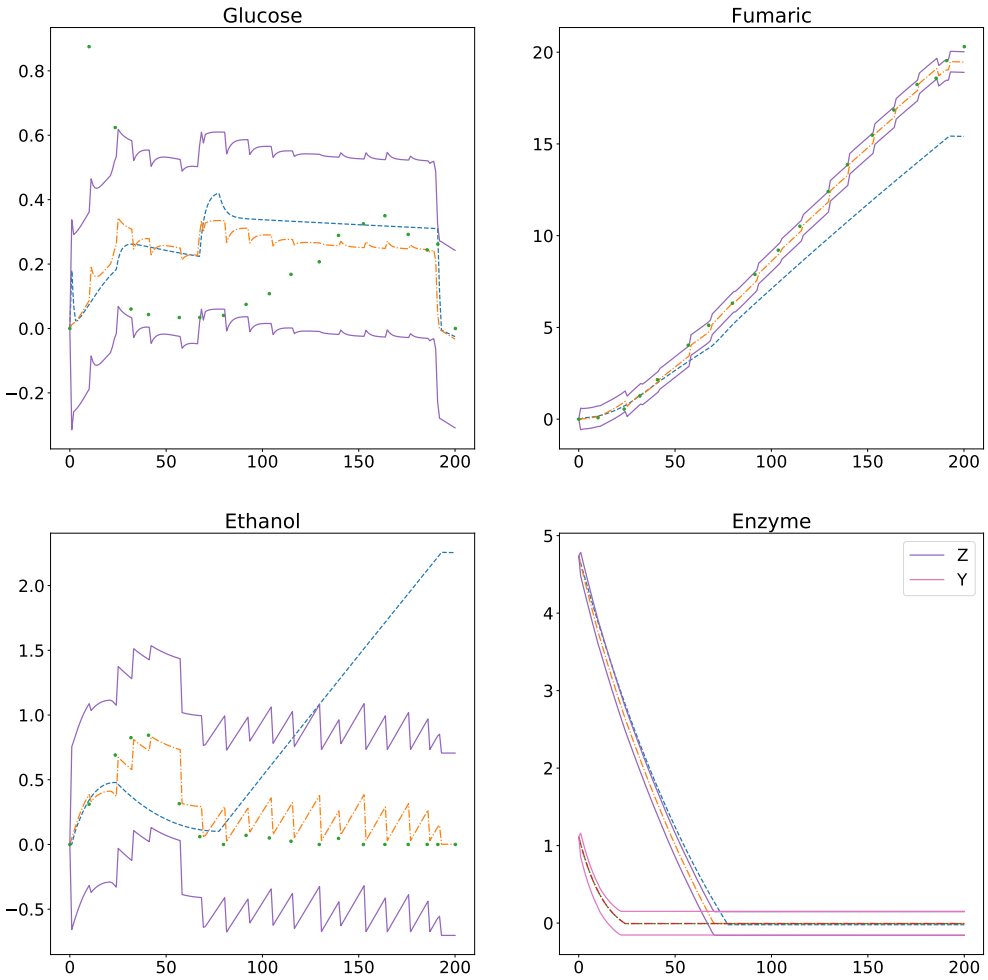

Figure 14: Model, HPLC samples and 90 % confidence intervals of state estimation average for bioreactor run without ethanol production. The solid lines represent the boundaries of the confidence region and the dotted lines represent the model predictions without any state estimation. The dotdash lines represent the mean of the state estimator. In the enzyme graph the C_Z and C_Y confidence lines are distinguished in the legend.

6 Software framework

This section aims to give a brief, high-level overview of the software developed for this project. Figure 15 shows the relationships in the code diagrammatically. The central element is the simulation code. It serves as the main hub for information by storing and providing data that is relevant for the time series data. It receives information about inputs which it passes to the model. It takes the outputs from the model and passes them to the state estimation system. Finally, it also sends its data to plotting code for display to the user.

It is important ot note the mirroring of the fake and Labview code. The fake code is used to test that the system is working and the Labview code is used to interface with the actual system through Labview. The state updater and input objects have the same interface which make it easy to change them out for one another without changing the other code at all.

The state updaters serve to allow for the backdating of HPLC data and perform the update step in the state estimation process. The sigma point object provides sigma point locations and weights to the state estimator. The state estimator class performs the prediction steps in the state estimation system.

The model object contains the non-linear model of the system that is found in Table 1. It interfaces with the simulation system to receive inputs from the input objects and returns outputs that are used in state estimation and plotting.

There are two methods for plotting the data: online or offline. Online plotting is a vital aspect because one wants to be able to see live results of the simulation and state estimation. However, when testing the system the offline plotting code is much faster.

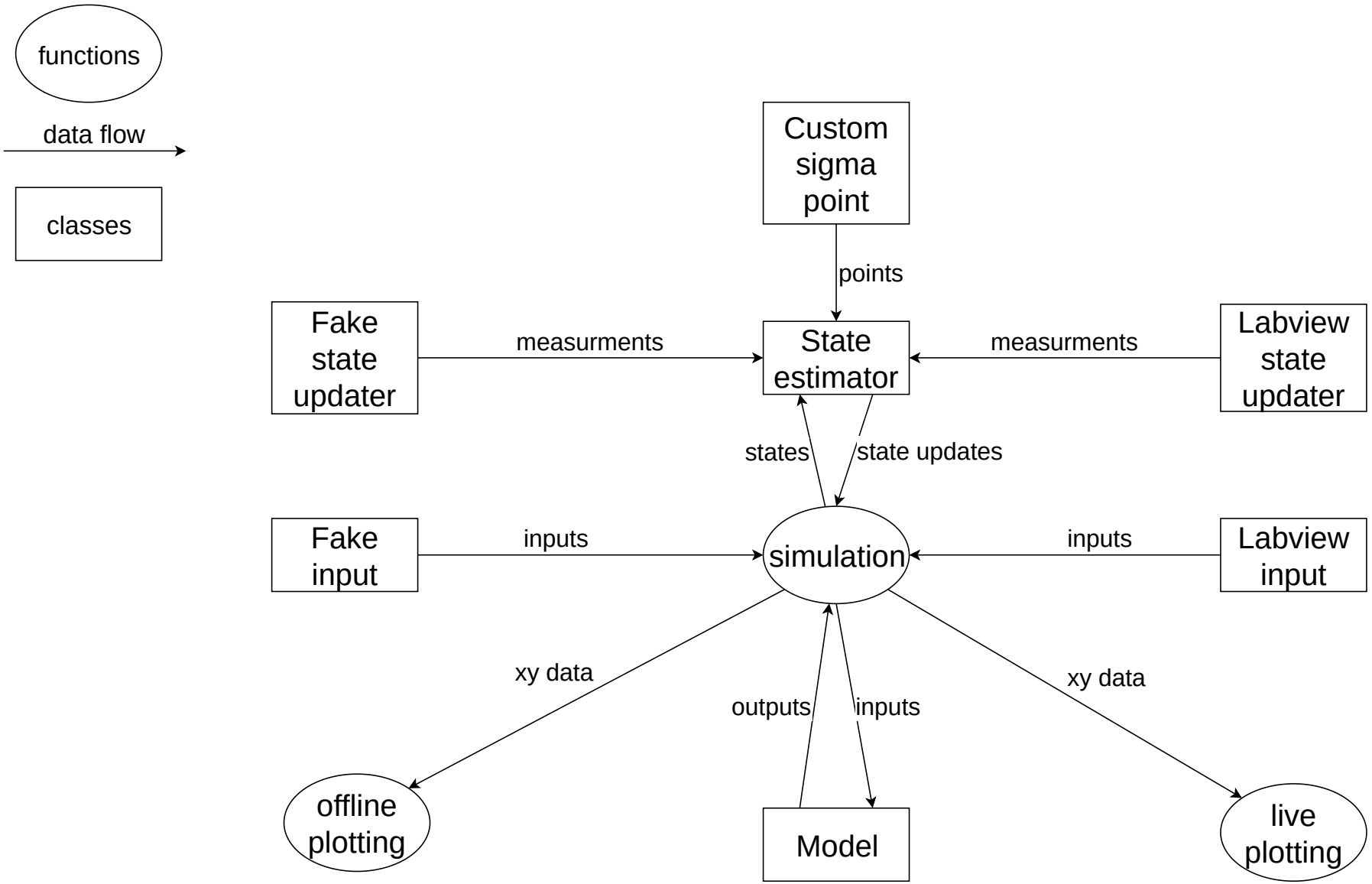

Figure 15: High level overview of the developed software

7 Conclusions and future work

This project has met its aims: drift, accuracy, linearity and dynamics of the pH probe have been found. More work can be done to repeat these experiments and perform longer drift experiments. There is also much future work to be done in the characterisation of the pumps, temperature probe and gas analysers.

The current pulse width modulator of the system was compared to an alternative and it was found that the pulse width modulator can be improved. These improvements were then implemented.

A non-linear model of the system was developed and used for state estimation. Future work would be to improve the model by researching the exact metabolic pathways (flux model) and enzymes that are at work in the model. Another method would be to characterise the different regimes of the bacteria and look at creating a growth model and a no ethanol model.

The state estimation system drastically improves the predicted outputs and future work would look into a control philosophy for the reactor: what needs to be controlled and how to control it. This posses many challenges since a significantly better understanding of the bacteria would be needed before this can occur.

The software that was developed allows for easy upgrading and replacement of the different parts. Future work should include translating the current Labview interface to Python as this would allow more advanced techniques to be applied very simply as the current Labview code is messy and difficult to maintain.

References

Chadha, H (2018). "The Unscented Kalman Filter: Anything EKF can do I can do it better!" In: URL: https://towardsdatascience.com/the-unscented-kalman-filter-anything-ekf-can-do-i-can-do-it-better-ce7c773cf88d.

Du, R and P Robertson (July 2017). "Cost-Effective Grid-Connected Inverter for a Micro Combined Heat and Power System". In: *IEEE Transactions on Industrial Electronics* 64.7, pp. 5360–5367. DOI: 10.1109/TIE.2017.2677340.

Farina, M, L Giulioni, and R Scattolini (2016). "Stochastic linear Model Predictive Control with chance constraints - A review". In: *Journal of Process Control* 44, pp. 53–67.

Hadj-Sadok, M and J Gouze (2001). "Estimation of uncertain models of activated sludge processes with interval observers". In: *Journal of Process Control* 11.3, pp. 299–310. ISSN: 0959-1524. DOI: https://doi.org/10.1016/S0959-1524(99)00074-8. URL: http://www.sciencedirect.com/science/article/pii/S0959152499000748.

McMillan, G (1984). "pH control: A magical mystery tour". In:

Merwe, R van der and E Wan (2003). "Sigma-Point Kalman Filters for Probabilitic Inference in Dynamic State-Space Models". PhD thesis. OGI School of Science and Engineering.

Naude, A (2018). "Fumaric and Malic Acid Production with Immobilised Rhizopus Oryzae". PhD thesis. University of Pretoria.

Wilken, S (2015). "Computationally efficient formulation of stochastic dynamical control within the context of switching probabilistic graphical models". Univesity of Pretoria.