5-2014

Design and Implementation of a Real-time Adaptive Oxygen Transfer Rate Estimator

Li Wang
Clemson University, wangliinhust@gmail.com

Design and Implementation of a Real-time Adaptive Oxygen Transfer Rate Estimator

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Li Wang
May 2014

Accepted by:
Dr. Richard Groff, Committee Chair
Dr. Timothy Burg
Dr. Sarah Harcum

Abstract

Oxygen transfer rate (OTR) is the most significant signal for aerobic bioprocess control, since most microbic metabolic activity relies on oxygen consumption. However, accurate estimation of OTR is challenging due to the difficulty of determining uncertain oxygen transfer parameters and system dynamics. This paper presents an adaptive estimator, which incorporates exhaust gas, stir speed and dissolved oxygen measurements, to predict the real-time OTR. The design of this estimator takes into account the headspace dilution effect, off-gas sensor dynamics and uncertain oxygen transfer parameters. Accurate and real-time OTR signal is derived by combining the low latency property of stir speed and dissolved oxygen signals with the high accuracy property of off-gas measurements. Proof of convergence of this adaptive estimator is provided under persistently exciting input constraint. Matlab simulation and *E. coli* fermentation experiment are provided to demonstrate the validity of the adaptive estimator. Through simulation and fermentation experiment, the estimated real-time OTR is shown to accurately track quick changes of oxygen demand in the culture. Thus, it can be applied to a variety of controls and estimation purposes, such as determining when the culture is in oxidative or overflow metabolism. Other related works on bioreactor fermentation control is also provided. A Kalman filter is developed and implemented for real-time feed rate signal estimation. Problems with OUR calculation is discussed when increased oxygen concentrations in the inlet gas and mass flow control are introduced. Two bioreactor related Matlab GUIs, i.e. fermentation GUI and display graph software, are also introduced in the appendices.

Dedication

This thesis is dedicated to my beloved parents and brother, who give me the greatest support and love through the hard time of my life.

Acknowledgments

I would like to thank my advisors Dr. Richard Groff, Dr. Timothy Burg and Dr. Sarah Harcum for their guidance in my research. Their enthusiasm and expertise in the research area has sharpened my critical thinking and professional skills. Also, I wish to thank my research teammates Matt Pepper and Ajay Padmakumar for their help and hard work in our project.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Oxygen transfer rate (OTR) is the rate at which gas phase oxygen transfers into liquid phase and becomes dissolved oxygen[11]. It is the most significant measurable online signal for aerobic bioprocess control. It can be used to quantify different physiological states of the culture by determining the related oxygen consumption[23]. The shape of the OTR signal indicates rich information about the metabolism of aerobic micro-oganisms, such as substrate limitation, oxygen limitation, product inhibition and diauxic growth, [3]. Acquiring accurate OTR signal online will greatly help monitor and determine the culture state.

The determination of real-time OTR is challenging, because it depends on the oxygen solubility, stir speed, biomass concentration, cultivation media, temperature, pressure and other factors[8]. The oxygen transfer coefficients vary from experiment to experiment. During the fermentation, the growth of cells and substrate addition change the oxygen transfer property of the culture. Thus, it is difficult to use pre-calibrated parameters for OTR calculation. Online determination of OTR with off-gas analysis gives more accurate OTR measurements[5]. However, the use of standard mass spectrometer for off-gas analysis will greatly increase the cost of experiment facility[19].

Another issue that has not been addressed is the head space dilution effect and off-gas sensor measurement lag for off-gas analysis. Due to the lag in off-gas analysis, OTR derived from this method will contain significant lag. OTR signal is often used for monitoring purpose, as it does provide data about culture health[26, 3]. In those slow monitoring applications, OTR is often

calculated periodically every 10-20 minutes[23]. Also, because of the high cost of mass spectrometer, a single mass spectrometer off-gas analyzer is often shared across multiple fermenters, which can be located 1 to 20m away from the mass spectrometer[7]. Lag effect of less than 10 minutes caused by head space dilution effect and off-gas sensor measurement dynamics is acceptable for cell growth condition monitoring. In contrast, the lag in the OTR signal makes it inappropriate for feedback control. When a control input is applied to the system, it may take more than 10 minutes to see the real response. The lag in the signal also has an filtering effect which will smooth out quick changes in the OTR signal. Quick changes in oxygen demand due to substrate feed rate increase, cell metabolism change or other interested reasons will be hidden from the user. In order to make the OTR signal suitable for feedback control and quick metabolism change detection, the lag in the OTR signal need to be accounted for.

1.2 Problem Statement

OTR is an important metabolism indicator for aerobic micro-organism fermentation. OTR is often calculated with oxygen transfer driving force and oxygen transfer coefficients. Oxygen transfer driving force is determined by the dissolved oxygen level, which can be measured by dissolved oxygen probe. Oxygen transfer coefficients are strongly related to stir speed. But they are difficult to determine and vary as culture properties change during the fermentation. The OTR signal derived from off-gas analyzer contains significant lag due to head space dilution effect and off-gas sensor measurement lag. This lag makes it not suitable for feedback control. Tracking quick changes in the cells' metabolism from filtered OTR signal is also not feasible. Thus, there is a need to design an OTR estimator that can compensate the lag in the signal and account for the uncertainty in oxygen transfer coefficients.

1.3 Literature Review

Methods for determining OTR are categorized as the oxygen transfer coefficient method or the off-gas analysis method. For the oxygen transfer coefficient method, the oxygen transfer capability is often characterized by a lumped parameter called the liquid-side volumetric mass-transfer coefficient k_La[9]. Much prior work has explored the experimental methods for determining k_La.

There are different ways to categorize the methods for OTR calculation. Based on the experiment principle, the methods can be classified as physical or chemical methods[22]. Further depending on whether the measurements are made under steady-state or transient condition, the methods can be classified as static or dynamic methods.

Physical methods directly measures the dissolved oxygen concentration of the solution over a short period of time[15]. The oxygen-containing gas is passed continuously through a bioreactor with aqueous solution. The major disadvantage of this method is that k_La is assumed constant and purely reconstructed from the noisy dissolved oxygen probe over a short period time. The chemical methods measure OTR based on the reaction rate of a homogeneous reaction. This method relies heavily on prior knowledge of the kinetics of homogeneous reactions[25]. However, the reaction parameters for this method are only valid for a specific condition, and need to be adjusted according to different catalyst concentration, pH, temperature[14]. The complexity in determining those parameters will lead to inaccuracy in OTR measurement.

The dynamic method is also based the dissolved oxygen probe measurement. The dissolved oxygen response signal is recorded after a step change of concentration in the input gas is applied[20, 6]. This method relies heavily on the accuracy and response time of the dissolved oxygen probe. Thus the oxygen probe is often calibrated and measurement dynamics is considered in the k_La calculation[12]. The static method assumes that the measurement is made under the steady-state condition, in order to simplify the oxygen transfer rate calculation[9].

Most of these methods calculate k_La when there are no organisms consuming the dissolved oxygen[17, 28]. Unfortunately, this is not true for most bioprocesses. The addition of cells will modify the oxygen transfer property of the system. Throughout the experiment, the micro-organism metabolism will generate biomass, protein and other byproducts. It is highly probable that the k_La value will drift away from the pre-calibrated value. Thus, the real-time OTR predicted from those predefined k_La values can be inaccurate.

OTR estimation for oxygen consuming system often assumes that OUR (oxygen uptake rate) and k_La remains constant for a short period of time[24]. This method is not efficient when k_La needs to be determined regularly during the experiment. It takes a relatively long time to estimate k_La once, e.g. [8] takes 30 min to obtain one estimation with 3% error. The nature of the constant OUR and k_La assumption also determines that it can not be used for online tracking of quick changes in k_La.

The off-gas analysis method directly measures the amount of oxygen transferred into the solution with an off-gas analyzer. It provides a more reliable source of real-time OTR estimation. Until recently, the high cost of mass spectrometer made off-gas analysis too expensive for most laboratories[19]. Recent advances in the sensor development offers off-gas analyzer(e.g. BlueSens Gas Sensor GmbH, Herten, Germany) with comparable accuracy to the established benchmark standard mass spectrometer. The reduced cost makes it feasible to dedicate a sensor to each bioreactor to enable more sophisticated real-time estimation and control of bioprocesses. Direct usage of off-gas sensor measurements is problematic. All off-gas sensors only measure a heavily filtered version of OTR. The dilution effect of headspace and sensor measurement dynamics exert significant lag on the OTR signal. Thus, quick changes in OTR will be filtered out and remain hidden from the user. Previous work using off-gas analysis ignored this filtering effect, because the lag in the OTR signal was acceptable for slow monitoring applications; however, the lag effect must be considered for more advanced real-time bioprocess estimation and control.

The real-time OTR estimator constructed and summarized in this work will enable more up-to-date detection of culture states and recover from quick metabolism changes that would otherwise be hidden by filtering effects. The design of the estimator takes into account the headspace dilution effect, off-gas sensor dynamics and uncertain oxygen transfer parameters. The estimated OTR signal combines the low-latency of stir speed and dissolved oxygen level signal with the accuracy of off-gas measurement. Thus, it is more suitable for feedback control and the identification of quick metabolism change.

1.4 Outline

The main goal of this thesis work is to design and implement an real-time estimator to accurately track the oxygen demand of the culture during fermentation experiments. In order to achieve this goal, a systematic research of the problem formulation, method design, implementation and improvement is conducted. This thesis reconstructs real-time OTR by modeling the dilution effect of the head space and the lag of the off-gas sensor. The prior unknown and time-vary parameter k_La is linearized by the stir speed. The parameters of the k_La linearization are adaptively updated using off-gas measurements. OTR is then calculated from stir speed and dissolved oxygen level using the adaptive k_La parameters to account for the slow variations during the entire experiment. The

thesis is divided into the following four chapters:

Chapter 2 gives a through discussion of the theoretical background of the design method. The oxygen transfer dynamics model of the bioreactor system is developed. Then, the design and convergence of the adaptive estimator are discussed in detail. Also, the persistently exciting requirement which enables the convergence of the parameters are presented. Chapter 3 explains the implementation of the adaptive OTR estimator. Both Matlab simulations and real fermentation experiments are carried out to validate and improve the performance of the estimator. The OTR estimator running on the simulated bioreactor model demonstrated excellent performance in tracking the quick changes in the OTR signal. The real fermentation experiment shows that the estimator performs as expected and fulfills the real-time tracking requirement. Chapter 4 provides information about other important and closely related work on the bioreactor. A Kalman filter is designed to derive more accurate and timely feed rate signal. Discussion about the problem with OTR calculation when different input oxygen concetration percentage and mass flow rate are also presented. Chapter 5 conclude the thesis with a summary and discussion of future works. Recommendations for future work provides several interesting extensions to the current work.

Chapter 2

Design and Methods

2.1 Overview

This chapter covers the theoretical background for the design of the adaptive OTR estimator. First, the oxygen transfer dynamics model for the bioreactor system will be introduced, specifically how head space dilution effect and sensor measurement lag affects the off-gas measurement. Second, the time varying relation between $k_L a$ and stir speed is discussed. A linearization method is adopted with two parameters that vary slowly during a fermentation experiment. Third, the design of the adaptive estimator is presented. The system is formulated into state space form with two unknown slowly time varying parameters. The adaptive estimator is proposed to simultaneously estimate the states and parameters in the system. Then, the proof of convergence for the adaptive estimator is presented. It is demonstrated that by adding two appropriate auxiliary variables into the state space form both parameter and state convergence can be achieved. Finally, the theoretical interpretation for the persistent exciting requirement, which guarantees the parameter convergence, is discussed. Practical methods designed to excite the input signals and satisfy this requirement are also proposed.

2.2 Oxygen Transfer Dynamics Model

The oxygen transfer model is presented in this section. Let b_0, b_1, b_2, b_3 denote the oxygen percentages of the gas entering the culture, coming out of the culture, in the head space of the bioreactor, and reported by the off-gas sensor, respectively. Figure 2.1 is an illustrative representation

of the oxygen transfer model variables, where V_1 is the volume of the culture and V_2 is the volume of the head space.

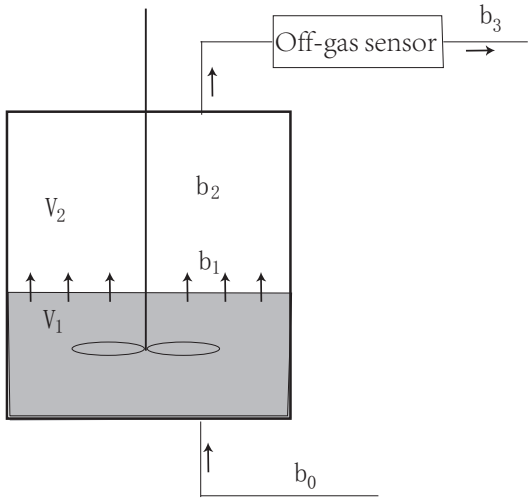


Figure 2.1: Variables for oxygen transfer model

OTR in the bioreactor is the rate of gas phase oxygen entering liquid culture and converting into dissolved oxygen. Theoretically, the OTR can be derived by calculating the oxygen percentage difference between the gas entering and exiting the culture.

$$OTR = \frac{M_f(b_0 - b_1)\rho_{o_2}}{V_1}, \tag{2.1}$$

where M_f is the mass flow rate in L_{air}/s, ρ_{o_2} is the oxygen density at $37C^o, 1\ atm$ in g/L_{oxygen}.

The off-gas sensor is used to measure the oxygen percentage coming out of the bioreactor. However, the sensor can't directly measure the off-gas percentage b_1. Instead, the sensor samples b_2 and its reading has a lag. The gas coming out of the culture first mixes with the gas in the head space of the bioreactor. This mixing process is modeled as a perfect dilution model. The dilution model can be viewed as a first order system with time constant $\tau_1 = \frac{V_2}{M_f}$. At the same time, the sensor measurement dynamics can be characterized as a first model system with time constant $\tau_2 = 120s$ by our experiment.

$$\dot{b_2} = \frac{M_f}{V_2}(b_1 - b_2) \tag{2.2}$$

$$\dot{b_3} = \frac{1}{\tau_2}(b_2 - b_3) \tag{2.3}$$

The off-gas sensor reading b_3 can be viewed as a heavily filtered version of b_1. Directly solving for b_1 in terms of b_3 will introduce noise to the signal, because it involves taking the second derivative of b_3. Let $\bar{\mathbf{x}} = [b_2, b_3]^T$, Equation (2.2) and (2.3) can be rewritten as the state space form:

$$\dot{\bar{\mathbf{x}}} = \begin{bmatrix} -\frac{M_f}{V_2} & 0 \\ \frac{1}{\tau_2} & -\frac{1}{\tau_2} \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} \frac{M_f}{V_2} \\ 0 \end{bmatrix} b_1 \tag{2.4}$$

Another way of real-time OTR determination is to consider the volumetric oxygen transfer coefficient k_La. Consider the difference between C^*, the dissolved oxygen concentration in equilibrium with the gas phase, and C, current dissolved oxygen level, as oxygen driving force for the system. OTR can be calculated with the following method.

$$\text{OTR} = k_La(C^* - C) \tag{2.5}$$

Previous literatures[2, 10, 18] also demonstrate that k_La has a strong linear relationship with stir speed.

$$k_La = \alpha_0 + \alpha_1(N - N_0) \tag{2.6}$$

$k_La$is linearized around a stir speed N_0 using a first order Taylor series expansion. By linearizing k_La near an appropriate stir speed N_0, a good approximation of k_La and trustworthy real time OTR can be derived. However, α_0 and α_1 don't remain constant during the whole fermentation process, due to the change of liquid composition and viscosity. This approximation of k_La is timely unlike other methods, because of its dependance on online sensors. Conventional methods can not account for those slowly time varying factors. The following method is to combine the Bluesens sensor measurement with the stir speed and dissolved oxygen readings to adaptively estimate α_0 and α_1. Thus an accurate real time estimation of OTR can be derived.

From Equation (2.1), (2.5) and (2.6), b_1 can be rewritten as:

$$b_1 = b_0 - \frac{V_1(C^* - C)}{M_f\rho_{o_2}}\alpha_0 - \frac{V_1(C^* - C)(N - N_0)}{M_f\rho_{o_2}}\alpha_1 \tag{2.7}$$

Combining (2.7) and (2.4), the problem can be reformulated as:

$$\dot{\bar{\mathbf{x}}} = \bar{A}\bar{\mathbf{x}} + \bar{B}\bar{\mathbf{f}}, \tag{2.8}$$

$$y = \bar{C}\bar{\mathbf{x}}, \tag{2.9}$$

$$\bar{A} = \begin{bmatrix} -\frac{M_f}{V_2} & 0 \\ \frac{1}{\tau_2} & -\frac{1}{\tau_2} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} \alpha_0 & \alpha_1 & b_0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \bar{C} = [0\ 1], \quad \bar{\mathbf{f}} = \begin{bmatrix} \frac{-V_1(C^*-C)}{V_2\rho_{o_2}} \\ \frac{-V_1(C^*-C)(N-N_0)}{V_2\rho_{o_2}} \\ \frac{M_f}{V_2} \end{bmatrix},$$

where $\bar{\mathbf{f}}$ is a vector of functions of dissolved oxygen level, stir speed, and other constant parameters. It is considered as the input signal to the system in the rearranged state space form.

The transfer function of this system is:

$$\bar{G}(s) = \bar{C}(sI - \bar{A})^{-1}\bar{B} = \frac{1}{s^2 + (\frac{M_f}{V_2} + \frac{1}{\tau_2})s + \frac{M_f}{V_2\tau_2}} \begin{bmatrix} \frac{\alpha_0}{\tau_2} & \frac{\alpha_1}{\tau_2} & \frac{b_0}{\tau_2} \end{bmatrix} \tag{2.10}$$

The observable realization of transfer function (2.10) is:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{f}, \tag{2.11}$$

$$y = C\mathbf{x}, \tag{2.12}$$

$$A = \begin{bmatrix} -\frac{M_f}{V_2} - \frac{1}{\tau_2} & 1 \\ -\frac{M_f}{V_2\tau_2} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ \alpha_0 & \alpha_1 & b_0 \end{bmatrix}, \quad C = [1\ 0], \quad \mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \frac{1}{\tau_2}\bar{\mathbf{f}},$$

where the off-gas sensor measurement y is considered as the output of the system, the input $\mathbf{f}$ to the system is a vector of functions of the dissolved oxygen level and stir speed signals.

2.3 Time Varying k_La and N Relation

From the previous literature[2, 10, 18], it has been widely proved that k_La has a strong linear relation with N. When the stir speed increases, the oxygen transfer rate into the culture will also increase proportionally. In [2], Akesson developed a gain scheduled DO controller for finer DO

control in the bioreactor. This controller is based on a constant linearization of k_La against the stir speed N.

$$k_La(N) = \alpha(N - N_0), \tag{2.13}$$

where $\alpha = 0.92h^{-1}rpm^{-1}$ and $N_0 = 323rpm$. The linearization parameters are carefully picked from the past experiment experience on a specific bioreactor, which means that they will not be valid for experiments performed on bioreactors with different scales.

One of the advantage of using the linearization method is that both the stir speed and the dissolved oxygen level are low latency signals. By taking advantage of the low latency signals, they can compensate the lag in the off-gas sensor measurement. However, there are mainly two difficulties with the k_La linearization.

1. The linearization parameters are unknown for each new experiment. Before each experiment, a complicated characterization experiment needs to be performed to calibrate those parameters.

2. The linearization parameters are not constant throughout the entire experiment. During the fermentation, the culture grows denser over time. The added substance and the cells' metabolism can also change the oxygen transfer property of the culture. Thus those linearization parameters are changing over time. Because the bio-process has a very slow growth dynamics, those parameters should be slowly time varying.

One simple approach to fix the slowly time varying property of the linearization parameters is to perform linearization for different time intervals. Within one or two hours, those parameters can be considered as constants and fit by linear regression.

This method was verified against one of the fermentation experiment. In this experiment, constant mass flow rate and constant pressure of the system was maintained. *E. coli* fermentation can have two distinct operating conditions, a batch phase and a fed-batch phase. Typically, the cell metabolism in the two growth phases are different. Thus two time intervals in each of them are chosen to verify the performance of the fitting.

$$k_La(N) = \alpha_0 + \alpha_1(N - N_0), \tag{2.14}$$

The results of the parameter fitting for different time intervals are listed in table 2.1. It can be seen that the goodness of fit for the batch phase is very high, which indicates that linearization

Experiment stage	Time range(h)	$\alpha_0(s^{-1})$	$\alpha_1(s^{-1}rpm^{-1})$	Goodness of fit r^2
Batch	2-4	-0.8966e-3	0.3049e-3	0.9627
Fed-batch	5-9	0.0172	0.0003	0.4559

Table 2.1: k_La parameters fitting

works well in this region. However, the goodness of fit during the fed-batch phase is very poor. It indicates that the dissolved oxygen dynamics changed significantly when the cell density grow larger and glucose feeding started.

The performance of the fitting is illustrated in figure 2.2. The graph clearly demonstrated that the k_La dynamics is very simple in the batch phase and leads to good fitting. However, the k_La dynamics becomes complicated during the fed-batch phase. It miss interpreted lots of quick changes in the dynamics. Another factor that leads to the non-smooth k_La signal is the set point PID DO controller. The set point PID controller reacts purely based on the error dynamics and caused non-smooth transitions between different stir speed.

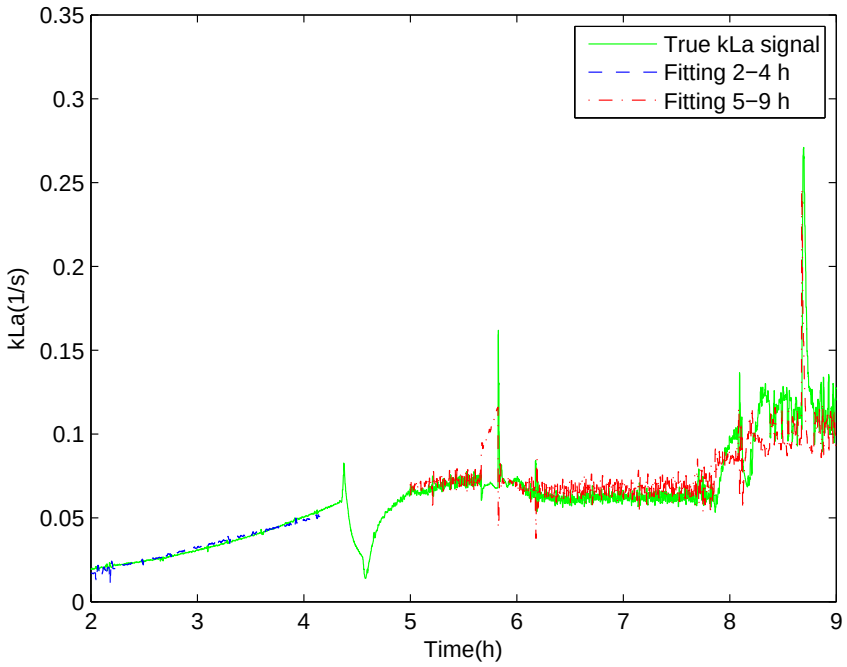


Figure 2.2: Constant k_La linearization, true k_La obtained with accurate off-gas analysis

Constant linearization is not a good method to characterize the k_La dynamics, because it can not account for fed-batch culture dynamics. Those linearization parameters are slowly time

varying during the fermentation and need to be constantly updated to achieve good approximation.

2.4 Adaptive Estimator Design

Assuming α_0 and α_1 are two slowly time varying unknown parameters, an adaptive observer of the following form can be designed to achieve simultaneous state and parameter estimation. The design of this estimator is an application of the adaptive estimation theory in [16].

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + \begin{bmatrix} 0 \\ \hat{\alpha_0} \end{bmatrix} f_0 + \begin{bmatrix} 0 \\ \hat{\alpha_1} \end{bmatrix} f_1 + \begin{bmatrix} 0 \\ b_0 \end{bmatrix} \frac{M_f}{V_2 \tau_2} + p + q, \tag{2.15}$$

$$\hat{y} = C\hat{\mathbf{x}}, \tag{2.16}$$

where $\hat{\mathbf{x}}$ and $\hat{y}$ are the estimated state and output signals, $\hat{\alpha_0}$ and $\hat{\alpha_1}$ are the estimated linearization parameters, A and C are from equation 2.11 and 2.12, p and q are auxiliary variables to compensate the parameter estimation errors. Those signals are determined with the following steps.

1. $d = [1, 4]^T$, $G(s) = [\frac{s}{d_1 s + d_2}, \frac{1}{d_1 s + d_2}]^T$;
2. $\omega(t) = G(s) f_0(t)$, $v(t) = G(s) f_1(t)$;
3. $p = [0, \ -d_2 \dot{\phi}_1 \omega_2 + d_1 \dot{\phi}_2 \omega_2]^T$, and $q = [0, \ -d_2 \dot{\psi}_1 v_2 + d_1 \dot{\psi}_2 v_2]^T$;
4. The adaptive law is chosen as $\dot{\phi} = -\gamma_0 \mathbf{e}_1 \omega$, $\dot{\psi} = -\gamma_1 \mathbf{e}_1 v$,

where $\mathbf{e}_1 = \hat{y} - y$, $\phi = [0, \hat{\alpha_0} - \alpha_0]^T$, $\psi = [0, \hat{\alpha_1} - \alpha_1]^T$.

The adaptive estimator from Equation 2.16 and 2.16 gives real-time estimates of the parameters $\hat{\alpha_0}$ and $\hat{\alpha_1}$. Real-time OTR is computed with Equation 2.5 and 2.6, except that the adaptively updated parameters $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are used.

$$\hat{\text{OTR}} = (\hat{\alpha}_0 + \hat{\alpha}_1 (N - N_0))(C^* - C), \tag{2.17}$$

2.5 Proof of Convergence

This section gives the proof of convergence for the OTR adaptive estimator. It provides the theoretical support that the designed estimator will track exactly the real trajectory of the states

and parameters. The estimation error system is first constructed by taking the difference between the real system and the estimator system. **Theorem 1** is presented to convert the error system to an equivalent error system with the same output. **Theorem 2** states that the appropriate parameter selection method will guarantee that estimation error for the equivalent error system goes to zero. Note that the proof presented in this section is a specific application of the proof for adaptive estimator provided in [16].

The estimation error dynamics can be derived by taking the difference between the estimator, represented by Equation 2.15 and 2.16, and the real system, represented by Equation 2.11 and 2.12.

$$\begin{aligned} \dot{\mathbf{e}} &= A\mathbf{e} + \phi f_0 + \psi f_1 + p + q, & (2.18) \\ \mathbf{e}_1 &= C\mathbf{e}, & (2.19) \end{aligned}$$

where $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ is the state estimation error.

Theorem 1. Let (C, A) be an observable pair, where $A \in \mathbb{R}^{2\times 2}$ is asymptotically stable. Given bounded piecewise-continuous functions of time $f_0, f_1 : \mathbb{R}^+ \to \mathbb{R}$ and $\phi, \psi : \mathbb{R}^+ \to \mathbb{R}^2$, the chosen ω, v, p, q can guarantee that the systems

$$\begin{aligned} \dot{\mathbf{e}} &= A\mathbf{e} + \phi f_0 + \psi f_1 + p + q,\ \mathbf{e}_1 = C\mathbf{e} & (2.20) \\ \dot{\epsilon} &= A\epsilon + d\phi^T\omega + d\psi^T v,\ \epsilon_1 = C\epsilon, & (2.21) \end{aligned}$$

have the same output as $t \to \infty$.

Proof. Pick a vector $d = [d_1, d_2]^T, d_1 = 1$, such that $C(sI - A)^{-1}d$ is strictly positive real. The transfer function $G(s) = [\frac{s}{d_1 s+d_2}, \frac{1}{d_1 s+d_2}]^T$, $\omega = G(s)f_0, v = G(s)f_1$. Note that s denotes the derivative operator $\frac{\mathrm{d}}{\mathrm{d}t}$.

1. When ϕ and ψ are constant vectors, choose $p = q \equiv 0$. It can be proved that:

$$\begin{aligned} C[sI - A]^{-1}\phi &= \frac{\phi_1 s + \phi_2}{k(s)}, \\ \{C[sI - A]^{-1}d\}\phi^T G(s) &= \{\frac{d_1 s + d_2}{k(s)}\}\frac{\phi_1 s + \phi_2}{d_1 s + d_2} = \frac{\phi_1 s + \phi_2}{k(s)}, \end{aligned}$$

   where $k(s)$ is the characteristic polynomial of A. Thus $C[sI-A]^{-1}\phi = \{C[sI-A]^{-1}d\}\phi^T G(s)$, similarly $C[sI - A]^{-1}\psi = \{C[sI - A]^{-1}d\}\psi^T G(s)$. Thus, the transfer function from $[f_0, f_1]^T$

to $\mathbf{e}_1$ is identical to that from $[f_0, f_1]^T$ to ϵ_1.

2. When ϕ and ψ are time varying vectors, choose $p = [0, \ -d_2\dot{\phi}_1\omega_2 + d_1\dot{\phi}_2\omega_2]^T$ and $q = [0, \ -d_2\dot{\psi}_1 v_2 + d_1\dot{\psi}_2 v_2]^T$. With the choice of d, ω, v, p and q, it can be shown that

$$\begin{aligned} & s[\phi_1 f_0 + p_1 - d_1\phi^T\omega] + [\phi_2 f_0 + p_2 - d_2\phi^T\omega] \\ = \ & \dot{\phi}_1 f_0 + \phi_1(sf_0) - \dot{\phi}_1\omega_1 - \dot{\phi}_2\omega_2 - \phi_1(s\omega_1) - \phi_2(s\omega_2) + \phi_2 f_0 - d_2\phi_1^T\omega_1 - d_2\phi_2^T\omega_2 + p_2 \\ = \ & d_2\dot{\phi}_1\omega_2 - \dot{\phi}_2\omega_2 + p_2. \end{aligned}$$

According to the definition $p_2 = -d_2\dot{\phi}_1\omega_2 + d_1\dot{\phi}_2\omega_2$, $s[\phi_1 f_0 + p_1 - d_1\phi^T\omega] + [\phi_2 f_0 + p_2 - d_2\phi^T\omega] = 0$. For any differentiable function $\phi(t)$, we can consider $C[sI - A]^{-1}[\phi f_0 + p - d\phi^T\omega]$ as the difference between the transfer functions of the two systems represented by Equation 2.20 and 2.20.

$$C[sI - A]^{-1}[\phi f_0 + p - d\phi^T\omega] = \sum_{i=1}^{2} \frac{s^{2-i}}{k(s)}[\phi_i f_0 + p_i - d_i\phi^T\omega] = 0,$$

Similarly,

$$C[sI - A]^{-1}[\psi f_1 + q - d\psi^T v] = \sum_{i=1}^{2} \frac{s^{2-i}}{k(s)}[\psi_i f_1 + q_i - d_i\psi^T v],$$

Note that case 1 is a special case of case 2, when $\dot{\phi} \equiv 0$ and $\dot{\psi} \equiv 0$. The added axillary variables p and q cancel out the effect of the dynamics of ϕ and ψ.

Since A is an asymptotic matrix, the two systems given by Eqn. 2.20 and 2.21 will satisfy $\lim_{t\to\infty} |\dot{\mathbf{e}}_1 - \dot{\epsilon}_1| = 0$ for arbitrary initial conditions.

□

Theorem 2. Let the dynamical systems be represented by the controllable and observable triple (C, A, d), where A is an asymptotically stable matrix, and $W(s) \triangleq C(sI - A)^{-1}d$ is strictly positive real. Let the elements of vectors ω and v be bounded piecewise-continuous functions. Then the origin of the system of differential equations

$$\dot{\epsilon} = A\epsilon + d\phi^T\omega + d\psi^T v, \ \epsilon_1 = C\epsilon, \tag{2.22}$$

$$\dot{\phi} = -\gamma_0\epsilon_1\omega, \tag{2.23}$$

$$\dot{\psi} = -\gamma_1\epsilon_1 v, \tag{2.24}$$

is uniformly stable in the large.

Proof. Consider the following quadratic function $V(\epsilon, \phi, \psi)$ as Lyapunov function candidate:

$$V(\epsilon, \phi, \psi) = \epsilon^T P \epsilon + \frac{1}{\gamma_0}\phi^T\phi + \frac{1}{\gamma_1}\psi^T\psi, \tag{2.25}$$

where P is a symmetric positive-definite matrix. Evaluating the time derivative of $V(\epsilon, \phi, \psi)$, we obtain

$$\dot{V} = \epsilon^T(A^TP + PA)\epsilon + 2[\phi^T\omega\epsilon^TPd - \phi^T\omega\epsilon_1] + 2[\psi^Tv\epsilon^TPd - \psi^Tv\epsilon_1]. \tag{2.26}$$

From the Lefshetz version of Kalman-Yakubovich lemma, if $W(s) \triangleq C(sI-A)^{-1}d$ is strictly positive real, then there exist symmetric positive-definite matrices P and Q that the matrix and vector equations

$$A^TP + PA = -Q,$$
$$Pd = C^T,$$

are simultaneously satisfied. Substitute into Eqn (2.26), we obtain

$$\begin{aligned}\dot{V} &= -\epsilon^TQ\epsilon + 2\epsilon^TC^T\phi^T\omega - 2\epsilon_1\phi^T\omega + 2\epsilon^TC^T\psi^Tv - 2\epsilon_1\psi^Tv \\ &= -\epsilon^TQ\epsilon \leq 0.\end{aligned} \tag{2.27}$$

Thus $\dot{V}$ is negative-semidefinite and the origin of the system is uniformly stable in the large. Since ω, v and $\dot{\epsilon}$ are bounded, we have $\lim_{t\to\infty}\epsilon = 0$. □

The proof in **Theorem 2** only guarantees that the state tracking error $\epsilon(t)$ converges to zero with the designed dynamics for $\phi(t)$ and $\psi(t)$. In order to make the parameters tracking errors go to zero, the input to the system should also satisfy the persistently exciting requirement, which will be explained in Section 2.6.

2.6 Persistently Exciting Requirement

The persistently exciting requirement means that the input should contain rich enough frequencies to excite all the modes of the dynamic system. The mathematical definition of the

persistently exciting condition is given by a autocorrelation function.

$$R_u(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{t_0}^{t_0+T} u(t)u^T(t+\tau)dt \tag{2.28}$$

Astrom and Bohlin[4] stated that if the function $R_u(\tau)$ exists and is positive definite, then the input $u(t)$ for the system is persistently exciting and the parameter estimation will converge to the true parameter value. The mathematic definition for the persistently exciting condition is not easy to verify. However, the intuition, i.e. the input signal contains rich enough frequencies[30], behind the concept can help us design such input signals to satisfy the condition.

In order to make the two unknown parameters in the system converge to the correct value, the inputs to the estimator system should satisfy the persistently exciting condition. The inputs to the estimator are f_0 and f_1.

$$f_0 = \frac{-V_1(C^* - C)}{V_2\rho_{o_2}}, \tag{2.29}$$

$$f_1 = \frac{-V_1(C^* - C)(N - N_0)}{V_2\rho_{o_2}}, \tag{2.30}$$

It can be observed that they inputs are functions of the stir speed N and the dissolved oxygen level C. Thus we can make the inputs excited by exciting both N and C. This can be achieved by designing an appropriate dissolved oxygen controller that excites both signals.

The original controller on the bioreactor is a set point controller. The set point controller uses a PID controller with pre-selected gains to control the stir speed and drive the dissolved oxygen level to a user specified value. There are several disadvantages with this set point DO controller.

1. The PID parameters are tuned for the specific bioreactor with trial and error. The tuning for those parameters are difficult, because the oxygen demand of the culture changes over time. Also, the set of optimal parameters varies from case to case.

2. The nature of the set point controller makes it not suitable for the persistently exciting condition. Because the controller tries to make the dissolved oxygen level a constant value, i.e. not exicited.

3. The PID controller covers the real metabolism of the culture and makes it difficult to analyze the actual biological meaning behind the signals. The metabolism change of the culture is

coupled with the complicated error compensation signals generated by the controller.

Due to the reasons above, we decided that the PID controller is not a suitable controller for DO control. A zig-zag DO controller is proposed to overcome the disadvantages of the PID controller. The new controller is designed to drive both the stir speed and dissolved oxygen signal up and down. It will guarantee the desired oxygen supply and also be generally suitable for different bioreactors. The detailed design and implementation of the zig-zag DO controller is given in the next chapter.

Chapter 3

Validation of the Estimator

3.1 Ecoli Metabolism and Simulink Xu Model

For the validation of the OTR estimator, both *E. coli* fermentation simulation and experiment are performed for the bioreactor. The uptake rate model developed by Xu[29] is chosen to simulate the growth of the *E. coli* during the fermentation experiment.

Simulated fermentation experiment is important for the design of advanced control and estimation techniques. Because the real fermentation experiment often lasts for more than twelve hours or even several day. The simulated fermentation greatly reduced the time required to develop new techniques for the bioreactor system. By choosing appropriate parameters for the model, the simulation result can closely emulate the real bioreactor system.

The Xu uptake rate model reflects the all the biological pathway that leads to the production of biomass. The mathematic interpretation of the model is listed as follows:

$$\frac{dX}{dt} = \mu X - \frac{F}{V}X, \tag{3.1}$$

$$\frac{dS}{dt} = -qSX - \frac{F}{V}(S - S_{in}), \tag{3.2}$$

$$\frac{dA}{dt} = (qA_p - qA_c)X - \frac{F}{V}A, \tag{3.3}$$

$$\frac{dV}{dt} = F - F_{sample}, \tag{3.4}$$

$$OUR = qOX, \tag{3.5}$$

$$\mu = (qS_{ox} - q_m)Y_{X/S,ox} + qS_{of}Y_{X/S,of} + qA_cY_{X/A}, \tag{3.6}$$

Symbols	Parameter	Unit
qS	Glucose fluc	$g/(g \cdot h)$
qA_p	Acetate production flux	$g/(g \cdot h)$
qA_c	Acetate consumption flux	$g/(g \cdot h)$
qO	Oxygen flux	$g/(g \cdot h)$
qS_{ox}	Glucose flux in oxidative	$g/(g \cdot h)$
q_m	Maintenance flux	$g/(g \cdot h)$
qS_{of}	Glucose flux in overflow	$g/(g \cdot h)$
$Y_{X/S,ox}$	g X produced per g S in oxidative phase	g/g
$Y_{X/S,of}$	g X produced per g S in overflow phase	g/g
$Y_{X/A}$	g X produced per g A	g/g

Table 3.1: Xu uptake rate model coefficients

where X is the biomass concentration, S is the substrate concentration, A is the acetate concentration, V is the volume of the culture, OUR is the oxygen uptake rate, μ is the biomass growth rate and F is the substrate feeding rate. The other coefficients are listed in the table 3.1.

The Xu uptake rate model captures the biomass production process as an exponential growth process with growth index μ. μ is then decomposed into the growth contributed by glucose consumption and acetate consumption.

The *E. coli* has three different phases of metabolism, i.e. oxidative, overflow and metabolite consumption illustrated in Fig. 3.1 as μ_1, μ_2 and μ_3[29]. In the oxidative metabolism phase, the cells process glucose aerobically without producing harmful byproduct. When excessive amount of glucose exists in the environment, the cells go to overflow metabolism, processing glucose anaerobically while producing growth inhibitor(acetate). The metabolite consumption processes acetate when glucose supply becomes limited. In the Xu uptake rate model, the $Y_{X/S,ox}$ and $Y_{X/S,of}$ terms characterize the growth rate contributed by oxidative metabolism and overflow metabolism respectively.

The oxygen uptake rate is also included in the model. During the oxidative metabolism phase, the cells acquire oxygen to process glucose. However, when the cells enter overflow metabolism, part of the glucose is converted into acetate. This part does not use oxygen to proceed. The consumption of acetate also requires oxygen.

Given appropriate uptake rate parameters, the Xu uptake rate model can mimic the dynamics of different culture phases. The reference parameters were retrieved from different *E. coli* fermentations in the related literature. Thus, the Xu model can be used as a benchmark model for the simulations to validate proposed control algorithms.

The Xu update rate model is implemented in Matlab Simulink for the purpose of control algorithm validation. The six dynamic equations are included to simulate the biomass, substrate, acetate, culture volume, oxygen consumption and biomass growth rate during the fermentation experiment. At the same time, the dissolved oxygen dynamics, which involves stir speed, dissolved oxygen concentration and $k_L a$, are implemented in the model.

This Simulink Xu model is a powerful tool for prototyping dissolved oxygen controllers, glucose feeding controllers, and the OTR estimator. An *E. coli* fermentation experiment often takes more than 12 hours to complete. The fault in the controller design might lead to model crash and data loss during the experiment. The Simulink Xu model provides a testbed to quickly validate controllers.

An OPC simulation server is created to simulate all the different fermentation variables on the actual bioreactor system. By properly setting up the OPC simulation server, the controllers can communicate with the simulated Xu model through OPC communication. The advantage of implementing the extra layer of OPC communication is that the controller can be switched back and forth between the real bioreactor system and the simulated Xu model. This greatly reduces the time needed for controller development and minimizes the fault due to model conversion.

3.2 Simulated Estimator with Feed Rate Pulse and Different Flow Rate

The tracking performance of the adaptive estimator is first evaluated with Matlab simulation. A model developed in[29] is adopted to simulate substrate and oxygen uptake rate by generating carbon dioxide and acetate. The Xu model consists of the dynamic equations of biomass, substrate, acetate, oxygen, growth rate and pH. It accurately emulates the metabolism of the cells and serves as a good model for controller development.

The principle of the growth maximizing controller is to maximize oxidative metabolism while avoiding the overflow phase[27]. Thus, it would be beneficial to determine online whether the culture enters overflow metabolism or not. OUR (Oxygen Uptake Rate) is the rate at which cells take dissolved oxygen in the culture and use it for metabolism reactions. It is a good indicator for overflow because it plateaus when cells enter overflow phase.

In the simulation, a five-minute glucose feed rate pulse is executed to generate an OUR pulse.

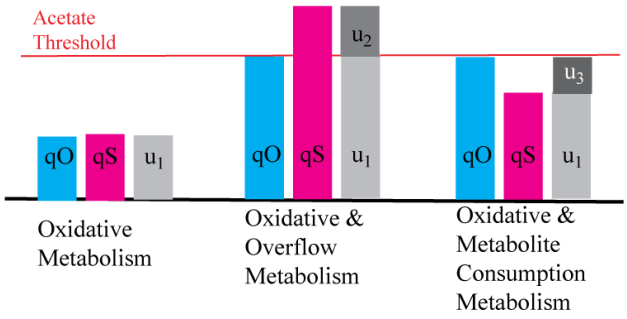
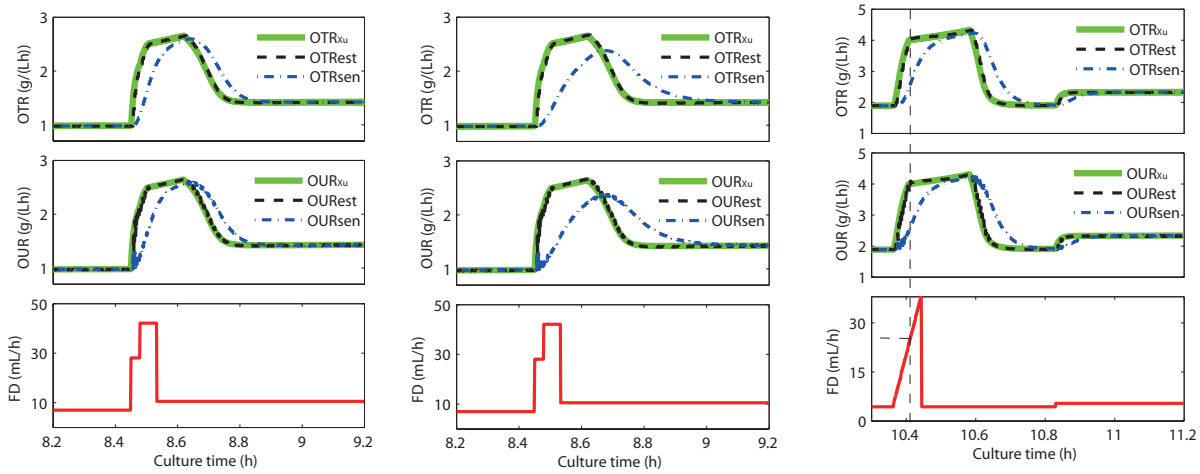


Figure 3.1: *E. coli* metabolic phases

As shown in Fig. 3.2a, OTR_{Xu} spikes immediately after the pulse and then plateaus, which indicates that the cells enter overflow metabolism. The estimator successful tracks the sudden changes in OTR_{Xu}, while the OTR_{sen} calculated directly from the off-gas sensor experiences significant lag and fails to capture the plateau region of OTR. The estimated OTR is more accurate and up to date compared with traditional methods. Note that OUR directly characterizes the metabolism of the cells. However, the OTR we estimated is not equivalent to OUR of the cells. Instead, OUR can be calculated with Eq. (3.7). Because the magnitude of $\dot{C}$ is negligible compared with OTR and OUR, it can be observed that OTR $\approx$ OUR according to Fig. 3.2a.

$$\dot{C} = \text{OTR} - \text{OUR}, \tag{3.7}$$

The filtering effect of the head space will be more significant when the head space is larger and/or mass flow rate is slower, such as would be observed with mammalian cell cultures. The time constant of the headspace dilution model is $\tau_1 = \frac{V_2}{M_f}$. When the mass flow rate M_f decreases from 3 Lpm to 1 Lpm, τ_1 will be three times larger, i.e. the lag in OTR signal will be more significant. Another fermentation simulation with $M_f = 1$ Lpm is performed to verify the performance of the estimator under lower mass flow rate. As illustrated in figure 3.2b, the estimator tracks the real OTR with high accuracy, while the OTR signal directly calculated from the off-gas sensor experiences significant lag. More importantly, OTR_{sen}, OTR computed directly from off-gas sensor, completely missed the plateau region, which indicates the cells enter overflow metabolism. Fermentation under low mass flow rate is the common case for mammalian cells, which require much less oxygen consumption than *E. coli* .



(a) Simulated glucose pulse with M_f=3 Lpm (b) Simulated glucose pulse with M_f=1 Lpm (c) Simulated ramped pulse with M_f=3 Lpm

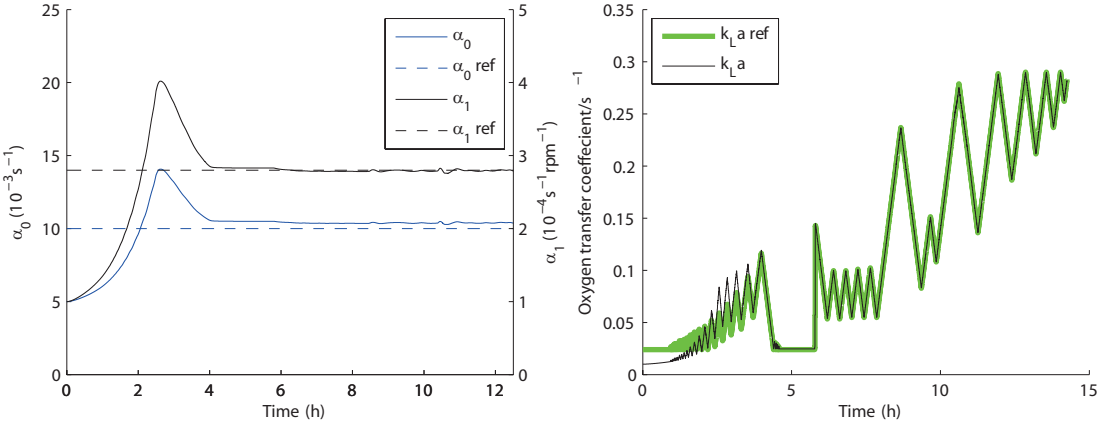
Figure 3.2: Simulated OTR estimation, where OTR_{Xu} is the simulated OTR signal from Xu model, OTR_{est} is the OTR signal reconstructed by the estimator, OTR_{sen} is the OTR signal directly calculated from the off-gas sensor reading. Similarly, OUR_{Xu} is the simulated OUR signal from Xu model, OUR_{est} is the OUR signal calculated from OTR_{est}, OUR_{sen} is the OUR signal calculated from OTR_{sen}. FD is the glucose feed rate during the fermentation.

A more interesting potential application of the estimator is shown in Fig. 3.2c. When ramped glucose feed rate is executed, OTR_{Xu} gradually increases and reaches the maximum value. By comparing the real-time feeding profile and OTR signal, the estimator can be used to identify the exact glucose feed rate at which the cells enter overflow metabolism. In Fig. 3.2c, the cells enters overflow at the feed rate of approximately 21 mL/h. By performing the slope pulse periodically during the fermentation, the maximum feed rate can be identified periodically and be used for maximizing growth control.

It was also verified that the estimated parameters converged to the correct value. As shown in figure 3.3a, the two linearization parameters α_0 and α_1 gradually converge to the reference values at the 4th hour. In figure 3.3b, estimated k_La accurately tracks the k_La reference value despite the quick changes in k_La. The estimator is validated to achieve both state and parameter tracking.

3.3 Bioreactor System Setup

A Sartorius Biostat B 5L glass vessel bioreactor shown in Fig. 4.1 is used for the *E. coli* fermentation experiment. A Digital Control Unit (DCU) interfaces with the electronic devices like



(a) α_0 and α_1 tracking

(b) simulated k_La

Figure 3.3: Performance of k_La tracking

pH sensor, DO sensor, temperature sensor, stir motor and pumps on the bioreactor. The estimator model running on Matlab 2012a is updating every 5 seconds with the new data collected from different sensors. It communicates with the DCU through an OPC server, with the Bluesens off-gas sensor through another OPC server, with two balances through serial communication, with mass flow controller through UDP communication. In the experiment, *E. coli* MG1655 pTV1GFP (cite strain) was cultured for 12 hours in minimal media. The bioreactor was inoculated at an optical density of 0.5 OD with a growth rate controlled at $0.25h^{-1}$. The volume of the culture was 1.67 L. The mass flow of the input gas was kept at 3 L/min by the mass flow controller.

The details for bioreactor software setup is presented in Section 4.2.

3.4 Off-gas Sensor and DO Sensor Characterization

3.4.1 Off-gas Sensor Characterization

In order to implement the adaptive OTR estimator on the system, the time constant τ_2 for the off-gas sensor needs to be identified. In the oxygen transfer dynamic system, the sensor measurement dynamics is modeled as a first-order system. The bioreactor operates at different mass flow rate for different culture or during different phase of fermentation. Thus, the effect of different mass flow rate should also be studied. In this fermentation experiment, the BlueInOne off-gas analyser was adopted.

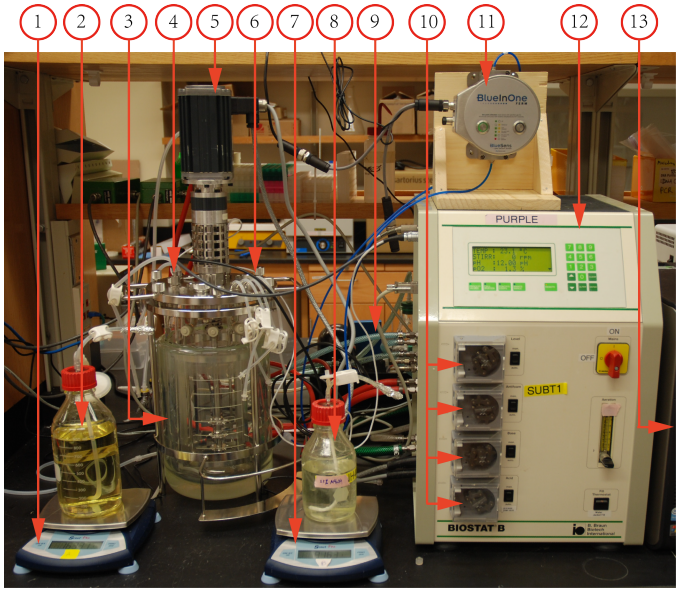


Figure 3.4: Bioreactor system setup: 1:glucose balance; 2:glucose bottle; 3:bioreactor vessel; 4:DO probe; 5:stir motor; 6:pH probe; 7:base balance; 8:base bottle; 9:mass flow controller; 10:feeding pumps(from up to down: acid, glucose, antifoam, base); 11:bluesens off-gas sensor; 12:DCU; 13:xPC target.

The characterization experiment was a study of the delay and response time of the sensor, τ_2. Previous publications[1] characterized response time and measurement delay of the sensor, but for a different model and at a much smaller flow rate (0.040 vs 3 L/min (lpm)). The input gas, b_0 , was connected directly to the input of the BlueInOne sensor, bypassing the stirred-tank vessel; this allowed for measurement of only the sensor dynamics, free from gas transportation delay. As shown in Fig. 3.5, the composition and flow of the input gas was varied in the experiment. The input gas was switched between pure nitrogen ($GASMX = 100\%, b_0 = 0\%$) and atmospheric air ($GASMX = 0\%, b_0 = 20.96\%$) in 10 minute increments. The response time (54 seconds) is the time constant for the sensor, i.e., the time for the sensor to reach 63% of its final value. The mass flow rate for the input gas is then varied between 1 lpm and 8 lpm. The data showed that the response time did not change with increasing gas flow, indicating the measuring chamber in the BlueInOne was being flushed well above maximum requirements.

The data for determining the time constant of the BlueInOne sensor is listed in table 3.2. Since the sensor measurement dynamics is modeled as first-order dynamics. When a step input is applied to the system, the time it takes for the sensor to reach 63% of the final value is equal to the time constant τ_2 of the system. According to the experimental data, the time constant of the sensor

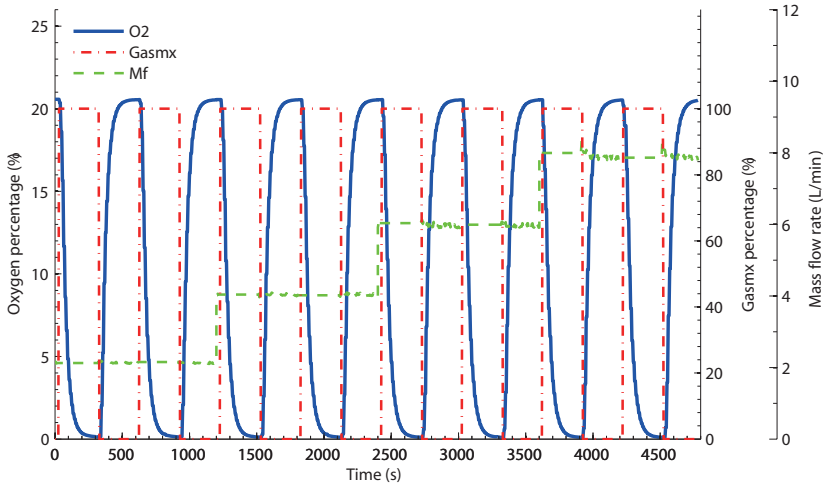


Figure 3.5: Bluesens time constant identification

remains approximately the same under different mass flow rate(1 lpm - 10 lpm). Thus, it is valid to use the same time constant $\tau_2 = 54s$ for our application.

3.4.2 Dissolved Oxygen Level Calibration

In order to get accurate dissolved oxygen level during the fermentation, the dissolved oxygen probe needs to be calibrated. The calibration process includes a 0% and a 100% reference calibration. Detailed discussion about the procedure of dissolved oxygen level calibration is given in section 4.2.4.

3.5 Estimator Implementation with Feed Rate Pulses

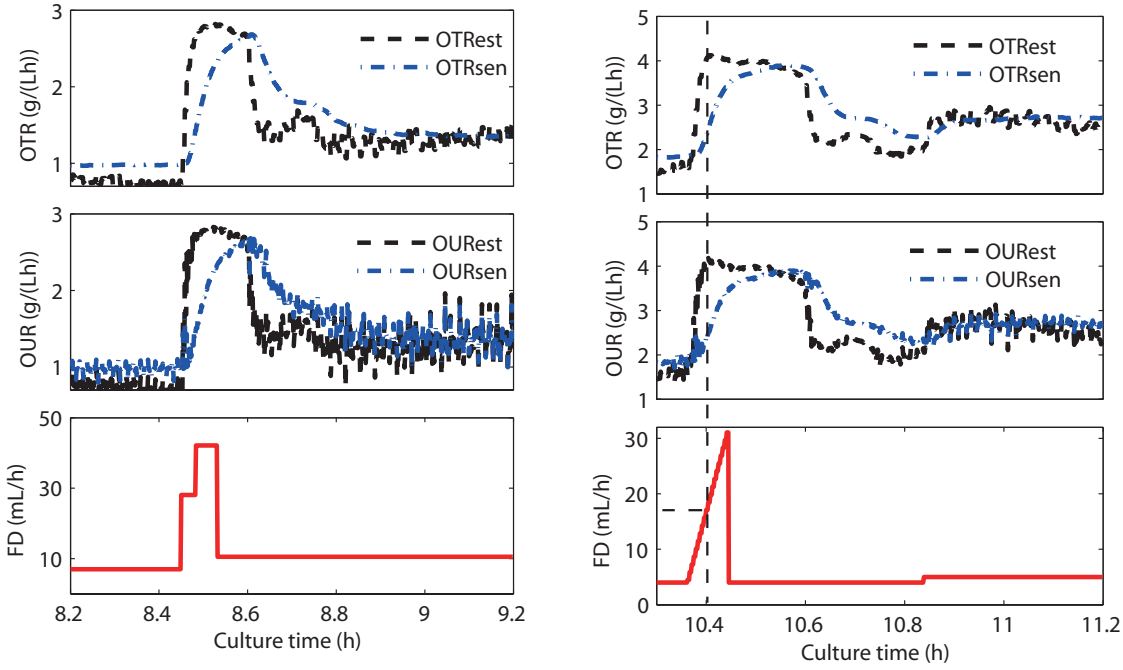
The simulated fermentations have provided enough information to understand the construction and tuning of the estimator. Several real *E. coli* fermentation experiments were carried out to validate the performance of the adaptive OTR estimator. The same feed rate pulses as the simulation were performed during the *E. coli* fermentation experiment. As shown in Figure 3.6a, a five minute pulse is applied to the culture. The response of the OTR_{est} is almost immediate while the peak in OTR_{sen} does not occur until after the five minute pulse is over. This validates that OTR_{sen} calculated directly from the off-gas sensor measurement leads to significantly lagged OTR signal, which is inappropriate for control and estimation purpose. However, OTR_{est} successfully compensates for the filtering effects and responses immediately to the metabolism changes. The plateau region

No.	Mass flow rate(lpm)	$t_{63\%}(s)$	Direction of change
1	2	53	falling($20.96\% \rightarrow 0\%$)
2	2	52	rising($0\% \rightarrow 20.96\%$)
3	2	53	falling($20.96\% \rightarrow 0\%$)
4	2	56	rising($0\% \rightarrow 20.96\%$)
5	4	56	falling($20.96\% \rightarrow 0\%$)
6	4	55	rising($0\% \rightarrow 20.96\%$)
7	4	56	falling($20.96\% \rightarrow 0\%$)
8	4	55	rising($0\% \rightarrow 20.96\%$)
9	6	56	falling($20.96\% \rightarrow 0\%$)
10	6	56	rising($0\% \rightarrow 20.96\%$)
11	6	56	falling($20.96\% \rightarrow 0\%$)
12	6	55	rising($0\% \rightarrow 20.96\%$)
13	8	54	falling($20.96\% \rightarrow 0\%$)
14	8	53	rising($0\% \rightarrow 20.96\%$)
15	8	56	falling($20.96\% \rightarrow 0\%$)
16	8	55	rising($0\% \rightarrow 20.96\%$)

Table 3.2: $t_{63\%}$ of Bluesens sensor under different mass flow rate

seen in OTR_{est} indicates that the culture entered overflow metabolism. In overflow metabolism, the oxidative metabolism becomes saturated with glucose and any excess glucose absorbed by the cells is anaerobically converted to acetate, which can inhibit growth at high concentrations (2 g/L). The profile of OTR_{est} indicates that the oxidative metabolism could process glucose at a higher rate, thus the feed rate could be increased significantly.

A ramped glucose feeding pulse was executed to explore the potential of using this adaptive OTR estimator to determine the optimal feed rate. As shown in Figure 3.6b, a five minute ramped glucose pulse is applied to the culture. The corresponding OTR_{est} signal increases to a peak value and remains around that value for the next twelve minutes. When the excessive glucose in the culture generated by the pulse is consumed, OTR_{est} falls back to normal feeding profile level. It can be read from the figure that OTR_{est} peaks at approximately 17 mL/h glucose feed rate, which indicates that cells will go to overflow when feed rate exceeds 17 mL/h at this specific culture time. This provides a potential method to adaptively determine the optimal glucose feed rate during the fermentation. By performing the feed rate pulses periodically during the fermentation experiment, the optimal feed rate can be identified and used to modify the current feeding profile.



(a) experimental pulse

(b) experimental slope

Figure 3.6: Experimental OTR estimation

Actual fermentation experiment implementation shows that the adaptive estimator successfully tracks the quick metabolism changes in the culture. It also inherits the accuracy of off-gas analysis. The adaptive OTR estimator fulfills the design requirements. The estimated low-latency and accurate OTR signal can be used for control and estimation applications. The oxygen consumption behavior of *E. coli* in actual and simulated experiments is very similar, which also validates that the simulated Xu model gives an good analogy of real *E. coli* metabolism.

3.6 PID vs. Zig-zag DO Controller

In order to make the unknown parameters α_0 and α_1 converge to the true value, the inputs to the estimator need to be persistently exciting. Persistently exciting means that the inputs to the system need to contain rich enough signals to excite all the modes of the system[30]. A zig-zag DO control strategy, illustrated in Fig. 3.7, is implemented to satisfy the persistent excitation requirement. The controller commands the stir speed to keep on increasing or decreasing with $\Delta = 5$ rpm discrete steps every 20 seconds, while maintaining the DO level within a given range. This control

strategy guarantees that both the oxygen demand and the persistently excitation requirements are satisfied.

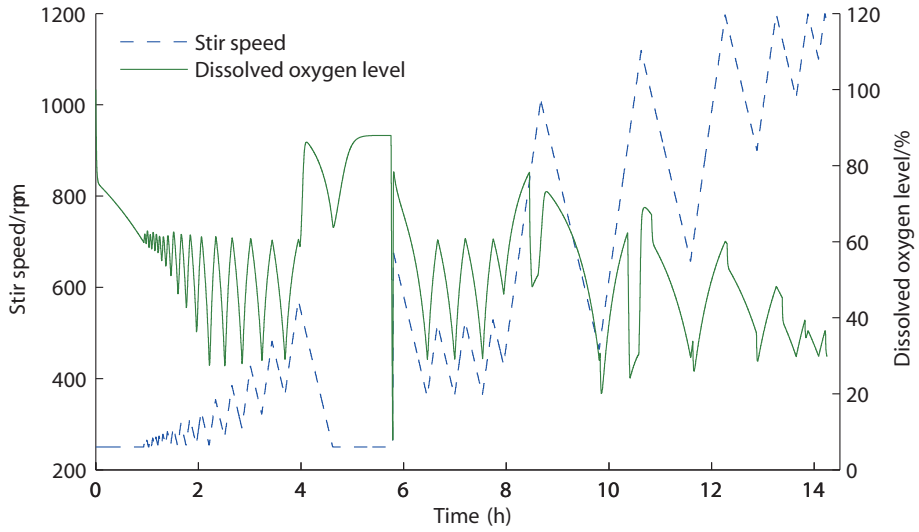


Figure 3.7: Simulated *E. coli* Fermentations using Xu model with constant step size zig-zag DO controller

Another advantage of using the zig-zag DO controller is that it keeps the DO signal clean and easy to interpret. The PID DO controller tries to keep the DO level at a given set point. During the experiment, the stir speed fluctuates quickly to compensate the dynamics of DO level. The resulting signal is always fast changing and noisy. *E. coli* oxygen demand variations due to metabolism and feed rate changes are hidden by the complicated controller behaviour. In contrast, the zig-zag DO controller always keeps the signal clean and readable. It is much more helpful for analyzing the underlying oxygen demand of *E. coli* .

The preliminary design of the zig-zag DO controller can be represented as the following equations.

$$N(t+1) = N(t) + k(t)\Delta, \tag{3.8}$$

$$k(t+1) = \begin{cases} k(t) & \text{if } DO_l < DO(t) < DO_h \\ 1 & \text{if } DO(t) < DO_l \\ -1 & \text{if } DO(t) > DO_h \end{cases}, \tag{3.9}$$

where $\Delta = 5\ rpm$ is the unit stir speed change every time step, $N(t)$ is the stir speed at time step

t, DO_l and DO_h are the desired lower and upper limits of dissolved oxygen level. $k(t)$ takes the value of 1 or -1 to control the direction of stir speed change. DO signal triggers the change of $k(t)$ so that stir speed increases when the DO level falls below DO_l, decreases when the DO level goes above DO_h and maintains the same trend when the DO level is within the desired range. Note that the time step for this controller is chosen as 15s because of the 15s sample time limitation of OPC communication.

The preliminary design of DO controller is simulated with the Xu model in a 14.5 hours fermentation. It can be observed from Figure 3.7 that the zig-zag controller keeps the signal excited and also maintains the dissolved oxygen level within the desired range. The simulated fermentation contains two phases, batch phase and fed-batch phase. During the first 5.7 hours, the glucose feeding pump is turned off. The *E. coli* grows purely with the limited nutrient exists from the beginning of the fermentation. At the 4th hour, the culture runs out of glucose. Thus the oxygen demand decreases significantly, and the DO level jumps to almost 100 percent level. In response to the DO change, the controller decreases the stir speed to the lowest value(250 rpm). When fed-batch phase starts at 5.7th hour, the glucose feeding pump starts working and the oxygen demand increases. The controller increases the stir speed to keep up with the oxygen demand.

The constant step zig-zag DO controller also satisfies the persistently exciting criteria. As illustrated in fig. 3.8, the estimated OTR signal gradually converges to the real OTR signal at the 4th hour and keeps the good tracking performance until the end of the fermentation experiment. The estimator converges slowly during the first four hours, because the oxygen demand at the beginning of the fermentation is low and the stir motor can only operate at low stir speed. The limited changes in the stir speed and DO level lead to the slow convergence speed. In the fed-batch phase, the glucose feeding rate increases by every 1% steps due to the limitation of the pump. It is shown in the figure that every step change of the glucose feeding rate leads to a step change in OTR. The estimator tracks exactly the quick step changes in OTR, while the value calculated directly from the off-gas sensor has significant lag and fails to capture those quick changes. There are also two glucose feeding pulses(t = 8.4h and t = 10.3h) applied to the culture during the fed-batch phase. The estimator successfully compensates the lag in the system and accurately tracks the quick changes in the OTR signal.

The overall performance of the constant step DO controller is satisfactory. However, there are some drawbacks in the preliminary design. The unit stir speed change is picked according to

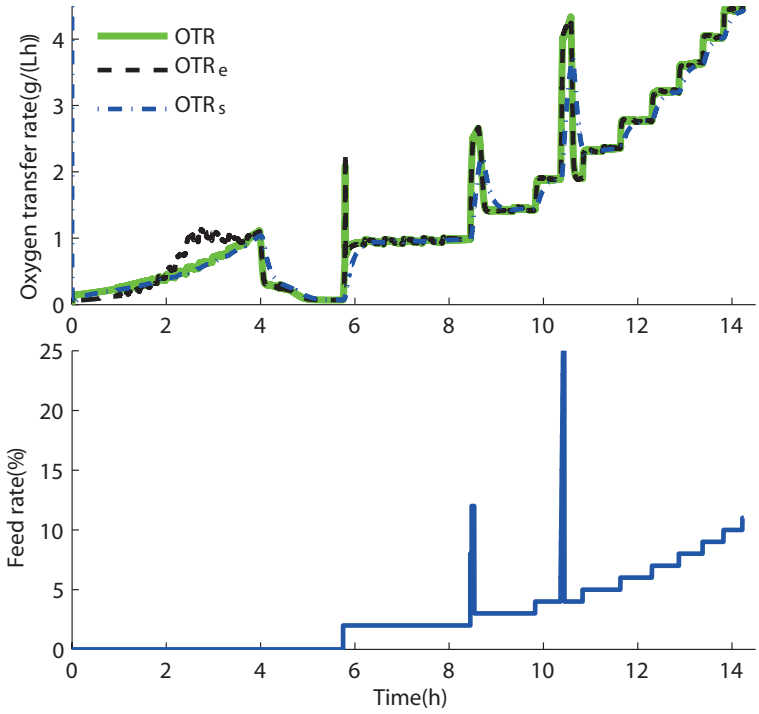


Figure 3.8: Simulated OTR tracking of *E. coli* fermentation with Xu model: OTR is the simulated signal, OTR_e is the estimated OTR, OTR_s is the OTR calculated directly from the off-gas sensor

prior knowledge and remain fixed throughout the entire experiment. This is problematic when the same method is applied to different fermentations with different bio-reactors. The unit stir speed change might become too small or too large, which will lead to lack of dissolved oxygen in the culture or too much disturbance in the signal. Thus, there is a need to design a universal DO controller that works in different fermentation experiments.

An improved variable step zig-zag DO controller is proposed. The unit stir speed change is modified into a function $\Delta(t)$, which varies depending on the change of DO level. The persistently exciting input requirement puts a constraint on the input signals, which are functions of DO level and stir speed. Thus, the controller is mainly designed to excite both the DO level and stir speed.

Parameters	$DO_h(\%)$	$DO_l(\%)$	γ_0	γ_1	$\delta_h(\%)$	$\delta_l(\%)$
Values	60	30	0.2	0.8	5	1

Table 3.3: Recommended parameters for the zig-zag DO controller

The change of DO level is chosen to be the criteria for determining the step size of the stir speed.

$$N(t+1) = N(t) + k(t)f(k(t))\Delta(t), \tag{3.10}$$

$$k(t+1) = \begin{cases} k(t) & \text{if } DO_l < DO(t) < DO_h \\ 1 & \text{if } DO(t) < DO_l \\ -1 & \text{if } DO(t) > DO_h \end{cases}, \tag{3.11}$$

$$f(k(t)) = \gamma_0 k(t) + \gamma_1, \tag{3.12}$$

$$\Delta(t+1) = \begin{cases} \Delta(t) & \text{if } \delta_l < \|DO(t) - DO(t-3)\| < \delta_h \\ \Delta(t)/2 & \text{if } \|DO(t) - DO(t-3)\| > \delta_h \\ 2\Delta(t) & \text{if } \|DO(t) - DO(t-3)\| < \delta_l \end{cases}, \tag{3.13}$$

In the improved zig-zag controller, $\Delta(t)$ varies according to the DO level change in 1 min($\|DO(t) - DO(t-3)\|$, time step size is 15s). If the DO level change is smaller than δ_l, which means the signal is not excited enough, $\Delta(t)$ will be doubled. If the DO level change is larger than δ_h, which means the DO level is fluctuating too fast, $\Delta(t)$ will be halved. Otherwise, $\Delta(t)$ remains the same with the previous sample time. In this way, $\Delta(t)$ can adaptively adjust itself according to the varying need of dissolved oxygen in the culture. With the same amount of change in stir speed, the decrease in DO level is often faster than the increase in DO level. The $f(k(t))$ term is added to compensate for the different rates of change for the increase and decrease in DO level. The recommended values for those parameters are given in table 3.3.

Another *E. coli* fermentation simulation is performed under the same condition with the previous simulation. The constant step size zig-zag controller is replaced with the variable step size zig-zag controller to test its validity. The simulation result is illustrated in fig. 3.9. Compared with fig. 3.7, there are several changes in the behavior of the DO controller. The step size of the controller is time varying according to the oxygen demand of the culture. The controller takes smaller steps during the batch phase of the fermentation, while taking larger steps in the latter part of the fed-batch phase. It can be observed that the DO level varies faster at the beginning and slower by the end in fig. 3.7. This is because when the biomass increases, the oxygen demand also increases.

It becomes more and more difficult to drive the DO level up and down with the same stir speed step size. But the variation of the DO level stays almost the same throughout the fermentation in fig. 3.9. It can be concluded that the variable step size DO controller is more robust against the variations in oxygen demand.

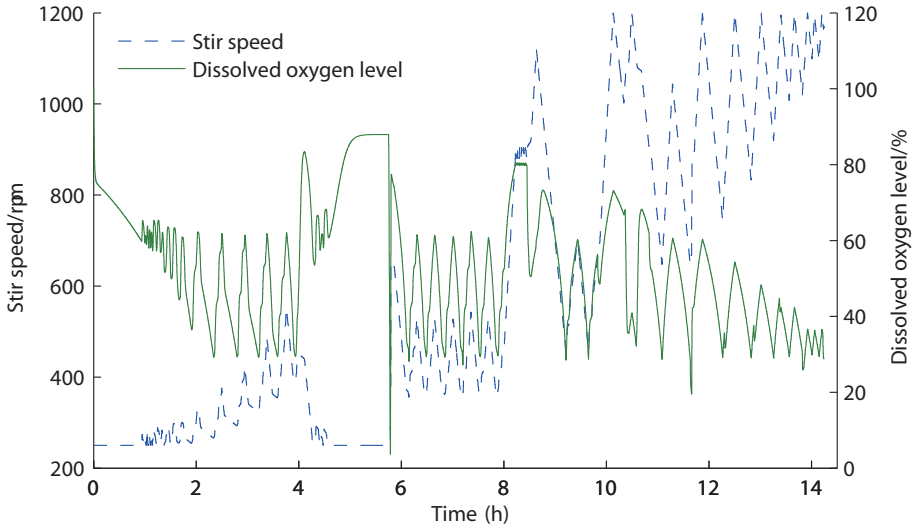


Figure 3.9: Simulated *E. coli* fermentation with Xu model using variable step size zig-zag DO controller

The constant step size DO controller is implemented in a real *E. coli* fermentation experiment. The stir speed step size is picked to be $\Delta = 5$ rpm from previous fermentation experience. The stir speed and DO level signal is shown in fig. 3.10. The controller successfully maintains the DO level within desired range and also keep both signals excited. The overall trend of the signals behave the same way as the simulated signals, except that the practical oxygen dynamics adds some fluctuations to the DO level signal. Note that at the 5th hour, the cells had consumed all the glucose in the batch phase. Thus, oxygen demand decreased shapely with the dissolved oxygen level spiked up. In response to the DO change, the stir speed decreased to the lowest allowable value. Even under the minimum stir speed, the culture still contained too much dissolved oxygen compared with the upper limit DO_h of the DO controller.

The variable step zig-zag DO controller is also implemented in the fermentation experiment. The difference from the previous experiment is that the source of nutrient changed from glucose to xylose. The growth rates of *E. coli* for feeding glucose and xylose are quite different. Thus the

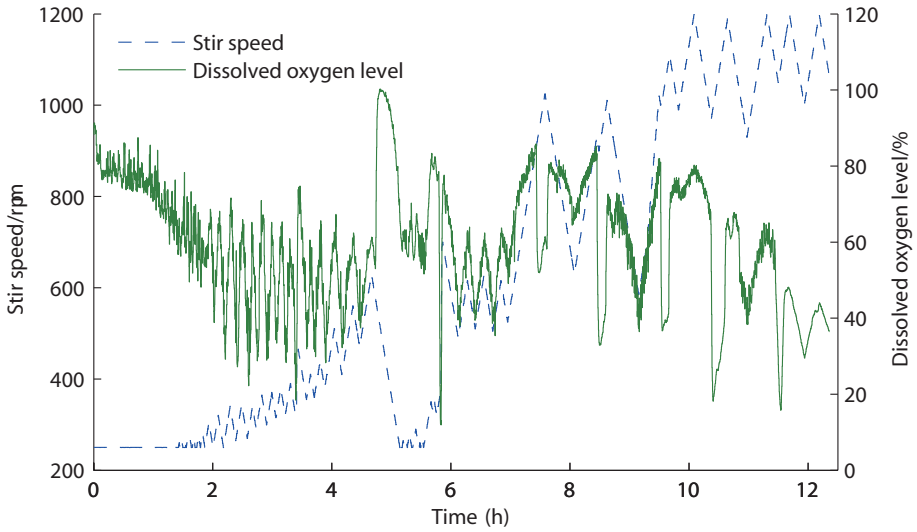


Figure 3.10: Experimental *E. coli* fermentation using constant step zig-zag DO controller

different oxygen demand of *E. coli* determines that using same constant step DO controller is not valid to keep the dissolved oxygen level excited and within range. Figure 3.11 shows the experimental DO dynamics with variable step zig-zag DO controller. It can be observed that *E. coli* grows much faster in the current feeding profile. The stir speed reaches 1100 rpm within just 6 hours. With the different oxygen demand, the variable step zig-zag DO controller adaptively adjusted the stir speed step size and make the dissolved oxygen signal excited and within desired range. It is validated that the variable zig-zag DO controller still functions correctly in different fermentations.

The zig-zag DO controller drives the DO level up and down throughout the fermentation experiment. There is a concern that varying dissolved oxygen level might harm the normal growth of the cells. In order to compare the effect of varying oxygen level, two fermentation experiments with different DO controllers were performed. As shown in figure 3.12, the proposed DO controller was verified against industrial standard PID DO controller. The proposed DO controller aimed at driving DO level up and down in the range of 30% to 60%, while the PID controller aimed at keeping DO level at 30%. Biomass concentration was sampled every half an hour. With the same inoculation at the beginning of the fermentation and same growth rate feeding profile during the fermentation, both experiments achieved approximately the same amount of biomass concentration by the end of the fermentation. Thus, it can be concluded that the proposed zig-zag controller does not have negative effect on the biomass growth as long as sufficient dissolved oxygen is supplied.

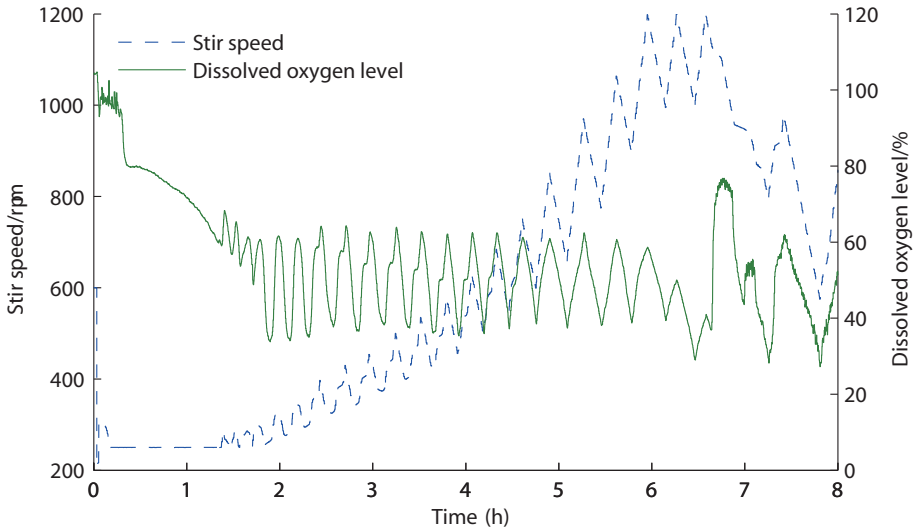


Figure 3.11: Experimental *E. coli* xylose fermentation using variable step zig-zag DO controller

3.7 Validity of k_La Linearization

In the modeling of OTR, the linearization of k_La, i.e. $k_La = \alpha_0 + \alpha_1(N - N_0)$, is performed based on the assumptions that k_La and N has a strong linear relationship, and α_0 and α_1 are slowly time varying.

In order to validate this assumption, the scatter plot of the k_La against the stir speed N during the actual fermentation experiment is shown in Fig. 3.13a. The overall trend of the plot shows that k_La and N has a strong linear relation within the operating range of 250 rpm to 1200 rpm. The data are shown with different colors and markers for every two hour interval. As the fermentation progressed, the curve gradually shifted upwards. This happens because the culture composition gradually changes over time. The trajectories of α_0 and α_1 are illustrated in fig. 3.13b. It is clearly illustrated that α_0 and α_1 gradually vary during the 12 hours fermentation experiment. The parameters can be approximated as constants within short time intervals. The parameters drift significantly over the long time period of the fermentation. This drift has never been captured online for oxygen consuming system before. This is one of the main contributions of the adaptive estimator. Thus, the proposed slowly time varying k_La and N linearization model gives a reasonable approximation of k_La.

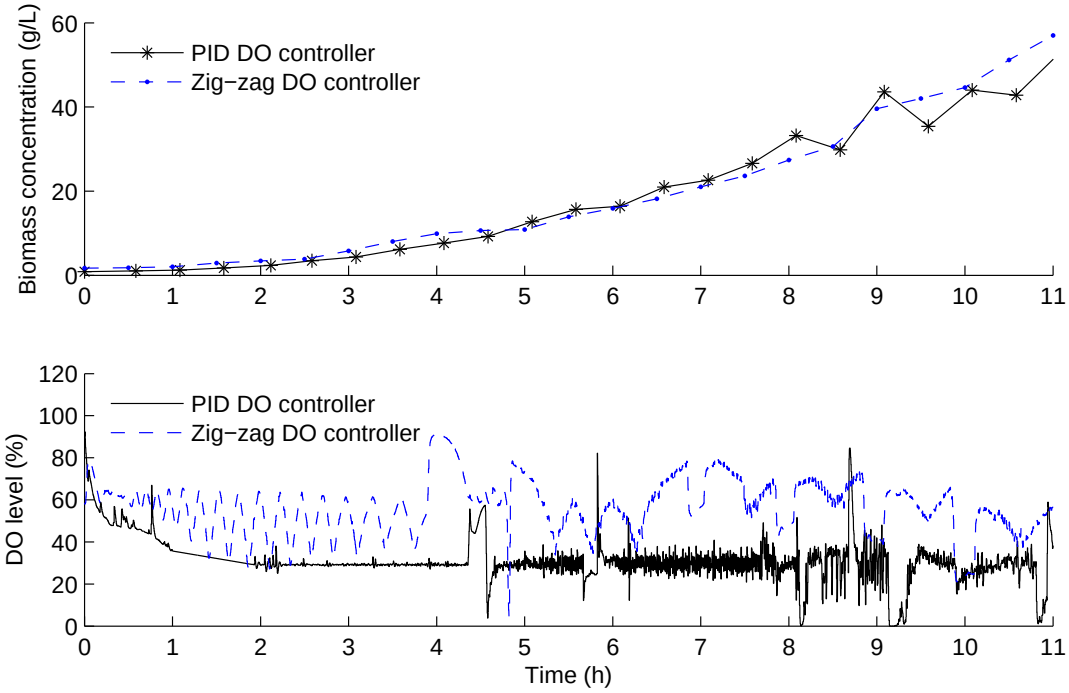
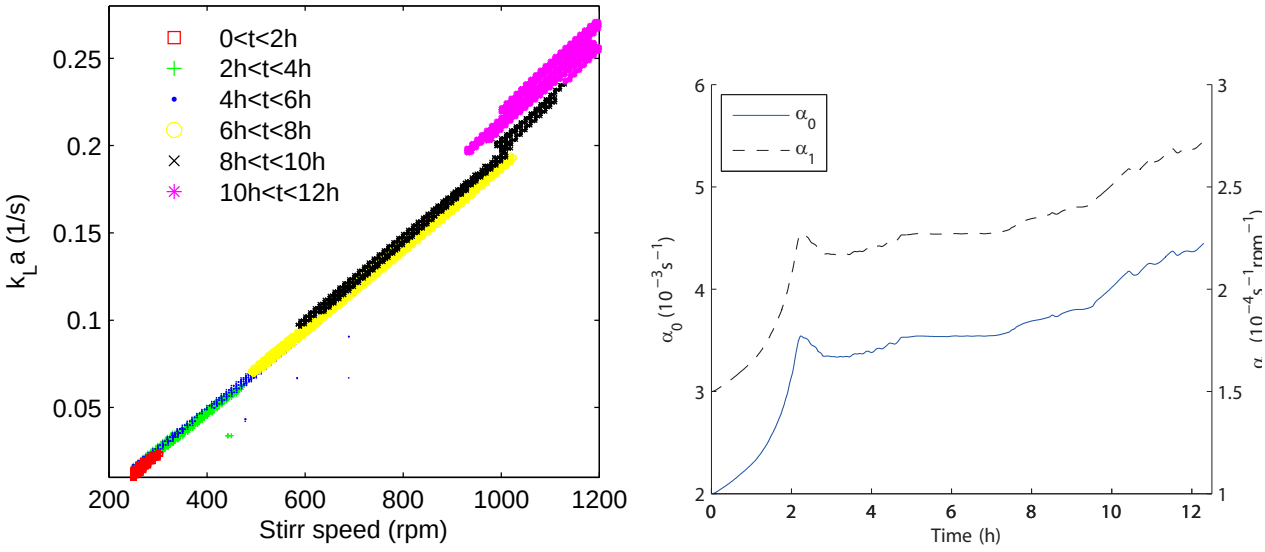


Figure 3.12: Experimental *E. coli* fermentations with different DO controllers



(a) Scatter Plot of Estimated kLa vs Stir speed

(b) α_0 and α_1

Figure 3.13: Experimental validation of $k_L a$ linearization

Chapter 4

Other Related Work

This chapter presents three bioreactor related problems. First, a Kalman filter is developed for feed rate estimation by combining measurements from two different sensors. Two Kalman filter designs are proposed and compared with the low-pass filter. Second, the software set up for the bioreactor system is introduced. The construction of bioreactor data acquisition and control system with serial communication, UDP communication, and OPC server/client is discussed. Finally, problems with OUR calculation when different Gasmx percentage and mass flow rate are used are presented. Error source diagnose is performed and possible solution is proposed.

4.1 Kalman Filter for Bio-reactor Feed Rate Estimation

4.1.1 Problem Formulation

The bio-reactor substrate feed rate estimation problem is considered in this section. The challenge is that industrial bio-reactor feeding pump has a poor resolution and lots of fluctuation. At the same time, there is always air bubbles in the line, which is difficult to track and evaluate. Sometimes, the substrate is volatile (e.g. ammonia solution) and makes the air bubble problem worse. All these factors make the pump command an unreliable indicator of the actual substrate feed rate. Getting real-time and accurate substrate feed rate is significant for the control of bio-reactor.

In order to get the actual substrate feed rate, a digital scale was added to the system to weigh the bottle of substrate. The digital scales are connected to the computer through serial

communication with a resolution of 0.1 gram. The bio-reactor substrate feeding setup is illustrated in figure 4.1.

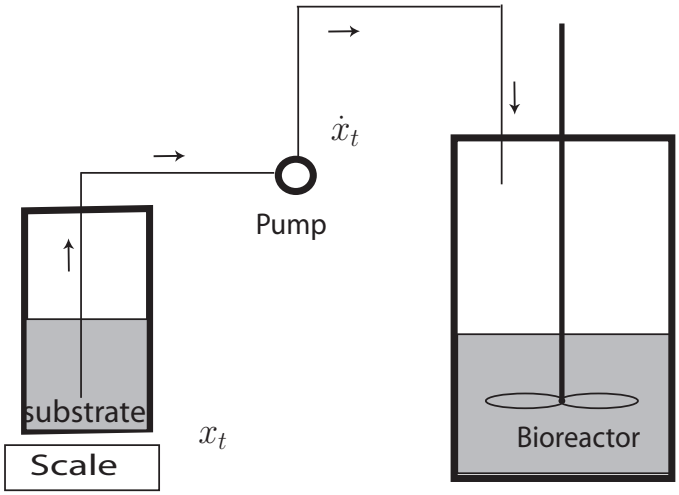


Figure 4.1: Bio-reactor substrate feeding setup, where x is the weight of the substrate bottle, $\dot{x}$ is the feed rate of substrate from the bottle

One common approach is to take the derivative of the scale readings and filter the data with a low pass filter ($\omega_c = 1/120$). The actual substrate feed rate can be derived in this way. However, the low pass filter introduces serious first-order lag into the system. The filtered feed rate has a poor real time property and can't be used for the real-time control of the bio-reactor.

The objective of this part is to combine the scale data and pump command data to get a good estimation of the feed rate. The feed rate estimation should be both fast and accurate. A Kalman filter is implemented and compared with the low pass filter result.

The core idea of the Kalman filter is to construct an appropriate Kalman gain K_t to weigh the model prediction with the sensor measurement[13]. Kalman filter performs a continuous cycle of prediction and updates. The main loop of the filter is described as follows.

1. Time update (prediction with process model)

   (a) Predict next state

$$X_{t,t-1} = \Phi X_{t-1,t-1} \tag{4.1}$$

(b) Predict next state covariance

$$S_{t,t-1} = \Phi S_{t-1,t-1}\Phi^T + Q \tag{4.2}$$

2. Measurement update (correction with sensor measurement)

(a) Calculate Kalman gain

$$K_t = S_{t,t-1}M^T|MS_{t,t-1}M^T + R|^{-1} \tag{4.3}$$

(b) Update state

$$X_{t,t} = X_{t,t-1} + K_t(Y_t - MX_{t,t-1}) \tag{4.4}$$

(c) Update state covariance

$$S_{t,t} = |I - K_t M|S_{t,t-1} \tag{4.5}$$

4.1.2 Constant velocity substrate feeding model

In the Kalman filter setup, a constant velocity substrate feeding model is chosen. The scale reading is considered as the measurement of the weight of the substrate bottle x. And the pump command is considered as the measurement of substrate feed rate $\dot{x}$.

1. State transition equation

$$X_{t+1} = \Phi X_t + \begin{bmatrix} 0 \\ N(0,\sigma_a^2) \end{bmatrix}, \tag{4.6}$$

where $X_t = [x, \dot{x}]^T$, $\Phi = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix}$, T is the sample time, $N(0,\sigma_a^2)$ is the Guassian dynamic noise.

2. Measurement equation

$$Y_t = MX_t + \begin{bmatrix} N(0,\sigma_{n1}^2) \\ N(0,\sigma_{n2}^2) \end{bmatrix}, \tag{4.7}$$

where $M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $N(0,\sigma_{n1}^2)$ and $N(0,\sigma_{n2}^2)$ are the Guassian measurement noise.

3. Covariance matrixes

   (a) Dynamic noise covariance matrix

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \tag{4.8}$$

   (b) Measurement noise covariance matrix

$$R = \begin{bmatrix} \sigma_{n1}^2 & \sigma_{n1,n2} \\ \sigma_{n2,n1} & \sigma_{n2}^2 \end{bmatrix} \tag{4.9}$$

   (c) State estimation covariance matrix

$$S = \begin{bmatrix} \sigma_x^2 & \sigma_{x,\dot{x}} \\ \sigma_{\dot{x},x} & \sigma_{\dot{x}}^2 \end{bmatrix} \tag{4.10}$$

The Kalman filter is implemented with the method given by Eq 4.1 - 4.5. The filter is then tuned by varying the ratio of the dynamic noise covariance to the measurement noise covariance.

4.1.3 Step velocity substrate feeding model

Because of the poor resolution of the substrate feeding pump, the pump is actually operating with discrete feed rate. Most of the time, the feed rate is almost constant. When feed rate command changes, it discretely jumps up to another value. Thus a step velocity substrate feeding model is designed for the Kalman filter.

1. State transition equation

$$X_{t+1} = \Phi X_t + \begin{bmatrix} 0 \\ u_t \end{bmatrix} + \begin{bmatrix} N(0, \sigma_{a1}^2) \\ N(0, \sigma_{a2}^2) \end{bmatrix}, \tag{4.11}$$

where $X_t = [x, \dot{x}]^T$, $\Phi = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix}$, T is the sample time, $N(0, \sigma_{a1}^2)$ and $N(0, \sigma_{a2}^2)$ are the Guassian dynamic noise. u_t is a discrete pulse signal which is constructed with the derivative

of the pump command. It represents the instantaneous changes of the pump command.

2. Measurement equation

$$Y_t = MX_t + N(0, \sigma_n^2), \tag{4.12}$$

where $M = [1\ 0]$, $N(0, \sigma_n^2)$ is the Guassian measurement noise.

The implementation of the Kalman filter based on the step velocity substrate feeding model is similar to Eq 4.1 - 4.5. The only difference is that control signal is added to the state prediction Eq 4.1.

$$X_{t,t-1} = \Phi X_{t-1,t-1} + \begin{bmatrix} 0 \\ u_t \end{bmatrix} \tag{4.13}$$

4.1.4 Experimental results of Kalman filter

Both of the two Kalman filter designs are tested against the experimental data to evaluate the performance of the filters. The result of the constant velocity Kalman filtering is compared with the inaccurate pump command and low pass filtered value in Fig. 4.2.

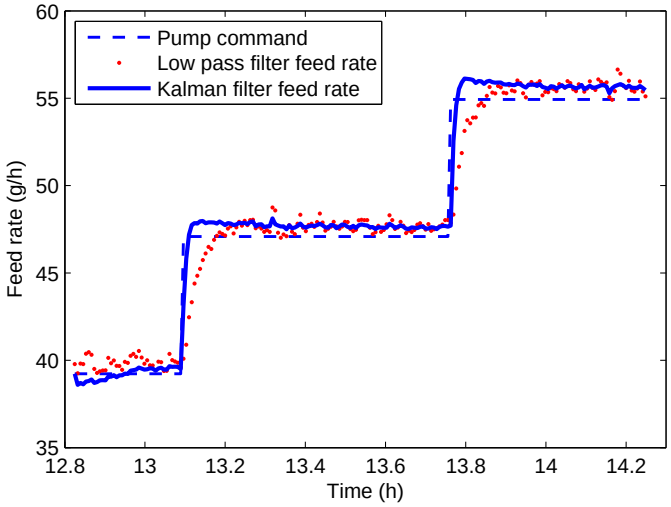


Figure 4.2: Filter 1: Feed rate comparison

It can be concluded from the figure that Kalman filtered feed rate converges to the same feed rate derived from the low pass filter. At the same time, the Kalman filtered feed rate has a much smaller lag compared with the low pass filter. The real-time and accurate property of the Kalman filter makes it a good choice for substrate feed rate estimation.

Parameters	Baud rate	Data bits	Parity	Stop bits	Byte order	Flow control
Values	2400	7	none	1	LittleEndian	hardware

Table 4.1: Serial configuration for balances

The result of the Kalman filter based on the step velocity model is illustrated in Fig. 4.3. It can be concluded from the figure that the step velocity substrate feeding model has even better real-time property and same accuracy. The feed rate estimation goes to new feed rate at almost exactly the same moment when the pump command changes.

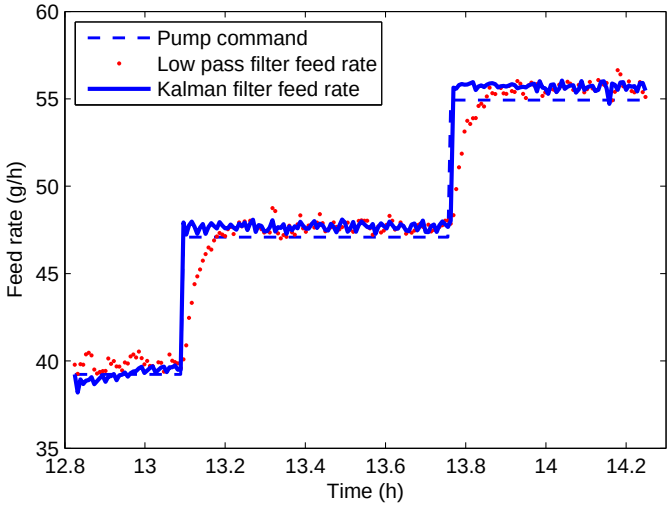


Figure 4.3: Filter 2: Feed rate comparison

4.2 Bioreactor Software Setup

4.2.1 Serial communication with the balances

The Sartorius Biostat B 5L bioreactor comes with four peristaltic pumps. The feed rate of the pump is not very accurate, because of the fluctuation of the motor, bubbles in the tubing, inaccurate calibration and other factors. In order to improve the pump feed rate signal accuracy, two balances are added to the system. As illustrated in fig. 4.1, object 1 and object 7 are the two balances used to weigh the base and glucose bottles. The balances have a resolution of 0.1 g. The real-time readings of the balances can be accessed through serial communication. The parameters for the serial port configuration are listed in table 4.1.

The Simulink model running on the main computer is collecting the balance data through serial communication every 5s. However, false readings of the balances are observed during the experiment. As illustrated in fig 4.4, the balance reading jumps back to zero every several minutes. The cause of this problem is then determined with a separate experiment.

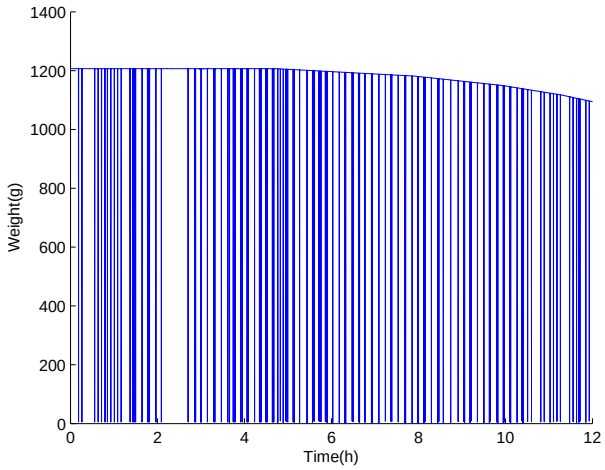


Figure 4.4: Balance false reading

Before the explanation of the source of error. We briefly introduce the concept of pseudo real-time simulation. In the OPC toolbox of the Simulink model, there is a option called pseudo real-time simulation. When this option is selected, the model execution time will match the system clock as closely as possible by slowing down the simulation properly. However, when the computing power of the computer is occupied by other processes, the simulation will become slower than the real system. This case is called pseudo real time violation.

The experiment is performed by keeping an object with constant weight of 125g on the balance, and collecting data for the whole system. The simulation model is running every 1s. As shown in fig. 4.5, the pseudo real-time criteria remains above zero most of the time, and the balance reading remains constant. However, whenever pseudo real-time violation occurs, the pseudo real-time criteria goes below zero and the balance reading is corrupted. There is no easy fix to this problem. The only measure to take is to guarantee that the computing resource is always larger than the simulation demand.

The pseudo real-time violation doesn't occur very frequently. So the collected data is smoothed to remove those wrong readings. By taking pointwise difference of the data and ap-

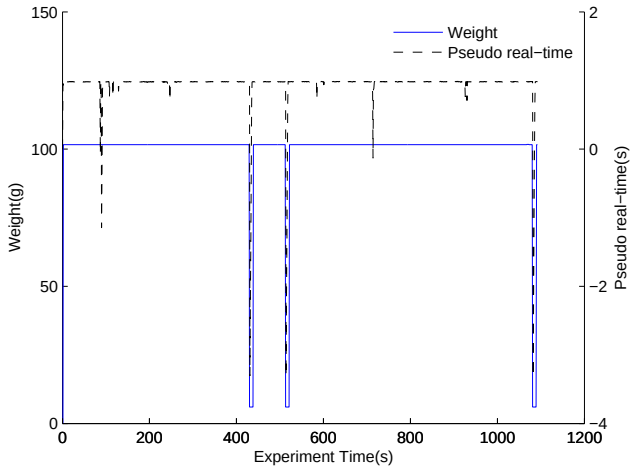


Figure 4.5: OPC real-time violation

plying a low pass filter, the actual feed rate of the pump can be derived. The comparison between the commanded feed rate and the actual feed rate is shown in fig 4.6. Note that by taking pointwise difference and applying low pass filter, extra noise and lag are introduced to the signal. A more accurate method using a Kalman filter is presented in chapter 4. This method is especially useful when volatile solution is pumped into the bio-reactor.

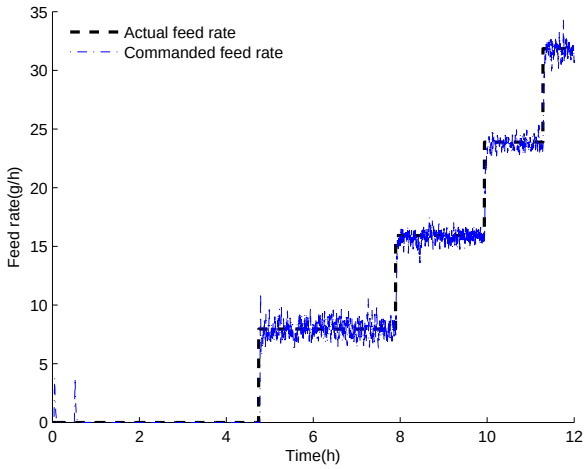


Figure 4.6: Feed rate calculated from balance data

4.2.2 UDP communication with the mass flow controller

In order to calculate the real-time OTR of the system, there is a need to measure the real-time mass flow of the input air. In this bioreactor system, the OMEGA FMA2000 mass flow controller is used to control and sense the real-time flow rate of the input air into the bioreactor. The setpoint signal and output signal for the mass flow controller are analog voltage signals(0-5 VDC). Matlab Simulink is used to construct the model of the bioreactor control system. xPC target, which is a real-time software environment from Mathworks that runs on a separate workstation, acts as a bridge to transfer data between the Simulink model and the mass flow controller. The xPC target is running in the stand-alone mode and exchanges data with the host PC through UDP communication. The terminal board of xPC target uses two analog I/O ports to send setpoint signal and receive output signal from the mass flow controller. The overall structure of the mass flow control and data acquisition system is illustrated in fig. 1.

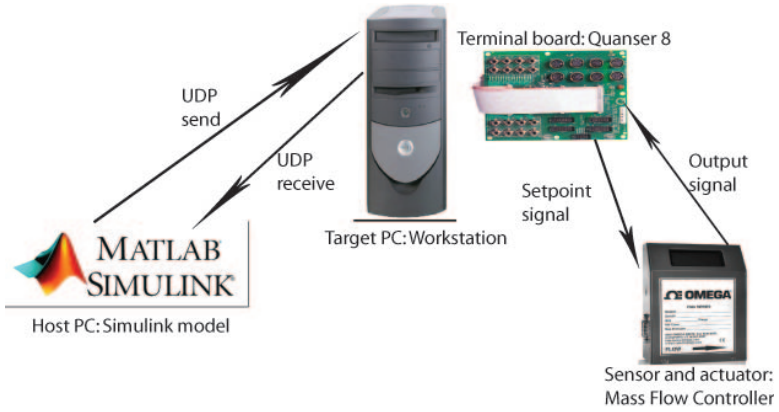


Figure 4.7: Mass flow controller and xPC target

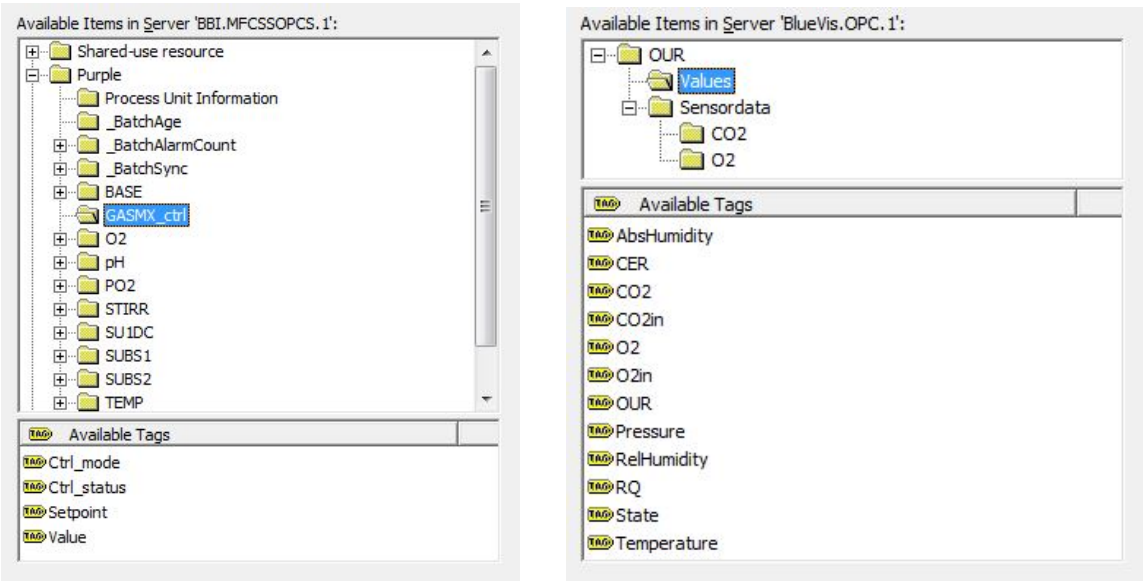
4.2.3 OPC server and client

OPC is a data access standard, which follows the client/server approach. The OPC servers are communication drivers and can provide access to process values. Different variables are assigned to different item IDs. The OPC clients subscribe to groups of item IDs in order to transmit and receive data with the actual devices.

The BioStat B bioreactor system uses Digital Control Unit(DCU) to run the actual bioreactor. The MFCS/win software can communicate with the DCU by reading and writing to different variables. The MFCS is a closed OPC server/client system. The variables from the off-gas sensor

can also be accessed from a off-gas sensor OPC server. Matlab Simulink contains a OPC toolbox, which provides the capability to communicate with OPC enabled devices. The Simulink model can act as OPC clients to subscribe to the MFCS OPC server and off-gas sensor OPC server. The Simulink model can communication with most of the devices and perform control and data acquisition through the OPC servers. Apart from the communication functions, the OPC toolbox can also enable pseudo real-time simulation. In this way, the OPC block will slow down the simulation so that it runs at the rate with the system clock.

The item IDs on the two OPC servers for MFCS(BBI.MFCSSOPCS.1) and off-gas sensor(BlueVis.OPC.1) are shown in fig. 4.8a and fig. 4.8b. The Simulink model can easily subscribe to different tags on the two OPC servers, in order to send commands and receive sensor signals.



(a) OPC server variables for MFCS

(b) OPC server variables for off-gas sensor

Figure 4.8: OPC server variables

4.2.4 Dissolved Oxygen Probe Calibration

The purpose of the dissolved oxygen level calibration is threefold: 1.conduct calibration for the dissolved oxygen (DO) sensor and provide reliable sensor reading for real fermentation experiment. 2. Examine the effectiveness of DO level calibration GUI. Look for future improvement. 3. Collect experimental DO level data. Analyze and characterize the process of air dissolving into the solution.

A DO level calibration GUI in fig. 4.9 is designed and tested in matlab. The experimental version of this GUI is used to guide the user through the calibration process. Calibration steps are as follows:

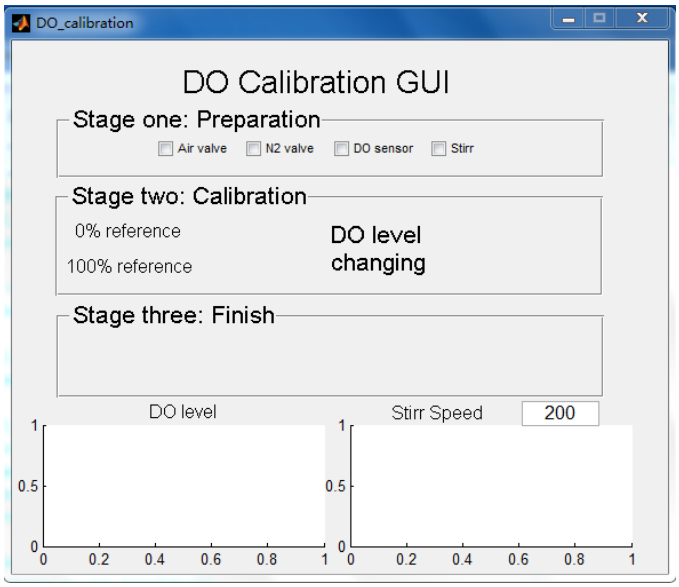


Figure 4.9: DO calibration GUI

1. OPC server setting Stage: Add PO2 variable into OPC server, so that we can read dissolved oxygen level from the OPC client. If it is already there, skip this step.

2. Preparation stage:

   (a) Turn on air valve and make sure that the inputting air flow rate indicator on the DCU controller is set to 3.0. When this operation is finished, check the checkbox for air valve.

   (b) Change the air source to nitrogen on the MFCS. This corresponds to setting GASMX_ctrl at remote auto mode with set point 100. Turn on nitrogen gas valve and gradually adjust the flow rate knob of the compressed nitrogen gas. The required flow rate is quite small compared with the capacity of the compressed nitrogen gas tank. Thus, the adjustment for flow rate knob should be very delicate. Make sure that the flow rate indicator on the DCU controller settles at 3.0 and check the checkbox for N2 valve.

   (c) Correctly install the DO sensor on the head plate of the bio-reactor. Make sure that we can get valid sensor reading from the DO sensor and change the DCU controller to the

DO level calibration mode. When this operation is finished, check the checkbox for DO sensor. Note that if DO sensor was disconnected from the amplifier for more than 10 minutes, it should be polarized for 1 hour before use.

(d) Correctly install the motor on the top of the bio-reactor, make sure that the shaft is center to avoid excessive wobbling. Real time stirr speed can be read from the GUI and adjusted from the editable text box. When the stir configuration is ready, check the checkbox for Stir.

3. Calibration stage

(a) When all the four preparation checkboxes are checked, the GUI will enter calibration stage. Input button for pure nitrogen gas as 0% reference will pop-up. Pressing this button will set GASMX_ctrl at remote auto mode with set point 100.

(b) In the experiment, the DO level will start to drop until it reaches equilibrium point. The settling of DO level is detected by the GUI and a calibration button will pop-up to indicate the flatting of DO level curve. Confirm 0% calibration on the DCU controller and also confirm on the GUI to finish 0% reference calibration. (note that there is also auto detection on DCU controller, by manually confirm 0%, you will override the DCU controller)

(c) When 0% reference calibration is done, input gas button will pop-up for 100% reference. Pressing this button will set GASMX_ctrl at remote auto mode with set point 0. This corresponds to pure air input.

(d) In the experiment, the DO level will start to rise until it reaches equilibrium point. The settling of DO level is detected by the GUI and a calibration button will pop-up to indicate the flatting of DO level curve. Confirm 100% calibration on the DCU controller and also confirm on the GUI to finish 100% reference calibration. (note that there is also auto detection on DCU controller, by manually confirm 100%, you will override the DCU controller)

4. Finishing stage

(a) When the calibration for both 0% and 100% DO levels are ready, a finish button will pop up. Click on this finish bottom will stop the simulation. GASMX_ctrl and STIRR will

be automatically set to remote auto mode with set point zero. Experiment data for DO level and Stirr speed are recorded into data.mat file.

(b) If there is no alarm or warning, then you are ready to start the fermentation experiment. If there is an alarm, it means that calibration is invalid. Please follow the instruction and redo the calibration.

The results of the experiment are as follows: 1. In the previous experiment using matlab simulation, the GUI is successfully used to guide the user through the entire calibration process. Simulation result is provided in Fig. 2. All the calibration steps should follow the standard procedure of DO sensor calibration. 2. The algorithm for detecting the flattening of DO level reading should work correctly. When the DO level readings stabilize at 0% or 100% for the specified period of time, the GUI will send out correct notification. 3. The experiment data for DO level and Stirr speed should be displayed in real time on the GUI. By the end of the experiment, those value are correctly saved into the data.mat file. Apart from verifying the validity of the DO calibration GUI, there are

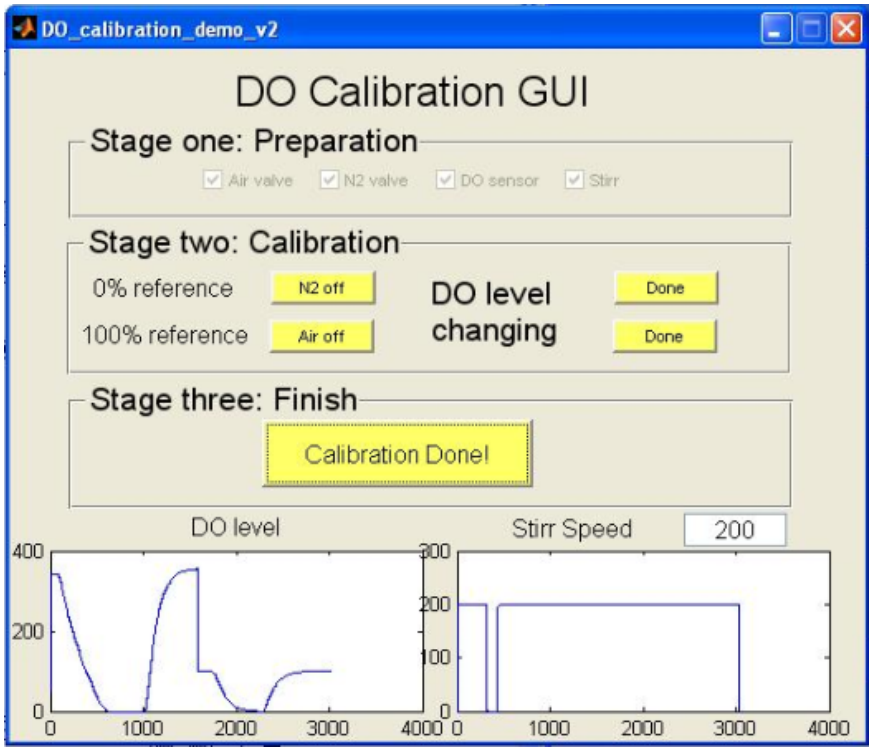
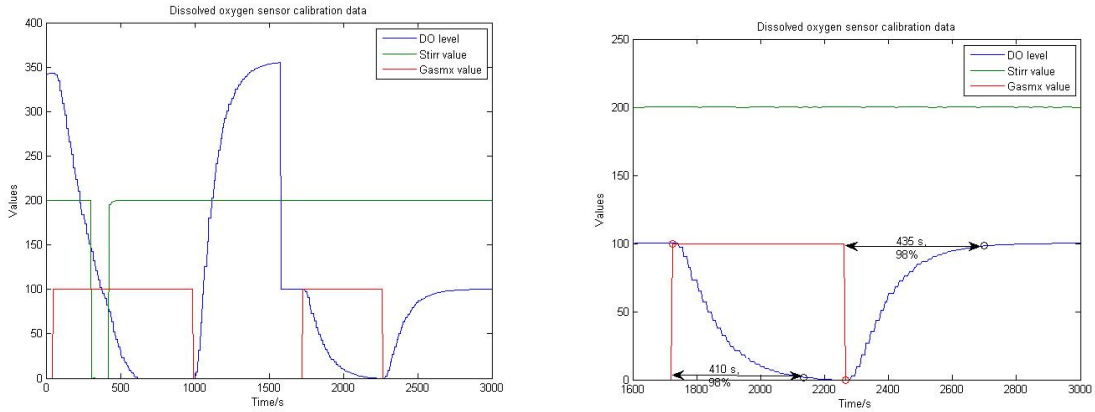


Figure 4.10: DO calibration GUI after experiment

also some interesting information about the oxygen transfer dynamics. When the Gasmx percentage

is switching back and forth between 0% and 100%, the dissolved oxygen dynamics behaves like a first order system. The time constant($\tau = t_{95\%}/3 = 140s$) of the first order system is identified from the DO signal in figure 4.11b.

This time constant τ is the accumulative effect the dissolved oxygen probe measurement time constant τ_s and the oxygen transfer time constant τ_t. From the manual for the dissolved oxygen prob, we can find that the prob measurement time constant $\tau_s = 8 - 15s$. Thus, the oxygen transfer time constant should be the dominant effect in governing the dissolved oxygen curve. It also validate that the dissolved oxygen prob is fast enough to capture the dynamics of the dissolved oxygen level.



(a) DO calibration curve

(b) Time constant analysis

Figure 4.11: DO calibration experimental result

4.3 Problems with OUR Calculation Under Different Gasmx Percentage and Mass Flow Rate

Using the Bluesens measurement to derive the real-time OUR signal is an important part of the oxygen demand analysis. In order to achieve higher cell density by the end of the fermentation, higher mass flow rate and oxygen percentage are needed. However, we meet several problems when different Gasmx percentages and different mass flow rates are introduced into the system. This section focuses on the diagnosis and possible solutions to the problem.

4.3.1 Bioreactor oxygen supply system

The bioreactor air flowchart is illustrated in figure 4.12. Atmospheric air comes from one hose and goes into the DCU. Pure oxygen comes from a oxygen tank and goes into the DCU. Inside the DCU, there is a gas mixing mechanism which switches between the two input gases (air and pure oxygen) with specified percentage. This percentage is given the tag of Gasmx in the system. It is implemented by controlling a value to switch between air and pure oxygen every 5 seconds.

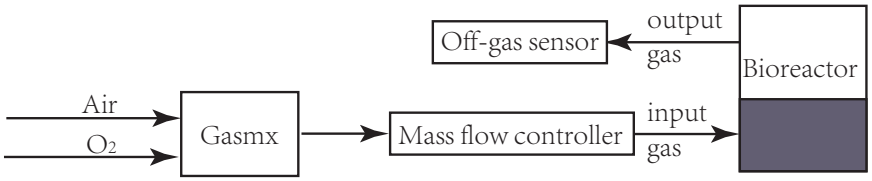


Figure 4.12: Bioreactor air flowchart

Equation 4.14 and 4.15 are applied to calculate the online OUR value. In this calculation, it is assumed that the input mass flow rate is approximately the same with the output mass flow rate.

$$\text{OTR} = \frac{M_f(b_0 - b_1)\rho_{o_2}}{V_1}, \tag{4.14}$$

$$\text{OUR} = \text{OTR} - \frac{dC}{dt}, \tag{4.15}$$

where M_f is the mass flow rate, ρ_{o_2} is the oxygen density at $37C^o, 1\ atm$, C is the dissolved oxygen concentration.

4.3.2 Online OUR miscalculation due to Gasmx changes

The off-gas analysis method is adopted for the online calculation of OUR signal. However, problems happen when different gasmx values and mass flow rates are used in the system. Theoretically, the oxygen demand and carbon dioxide production of the culture will not change when the Gasmx value is changed. Because the culture only interacts with the dissolved oxygen inside the culture. Thus, the metabolism of the cells will not change instantaneous when Gasmx changes.

A 21 hours long *E. coli* fermentation experiment is performed to test the online O2 transfer rate(OTR) and CO2 transfer rate(CTR) calculation method. During the fermentation experiment, the Gasmx percentage was increased by 10 percent steps whenever the stir speed was saturated

and can not follow up with the oxygen demand. The continuous metabolism of the cells is verified by the continuous CTR curve in figure 4.13. However, the OTR curve is shifted down by some value each time the Gasmx percentage changes. Theoretically, the percentage difference between the input oxygen($O2_{in}$) and off-gas oxygen($O2_{out}$) should be preserved. This is not the case in the experimental plot. The unreasonable OTR curve indicates that certain parameter is not accounted for when Gasmx percentage changes. Further analysis and experiments should be conducted to diagnose the problem. Note that the spikes in the OTR signal are causes by transient variation of system pressure due to Gasmx changes.

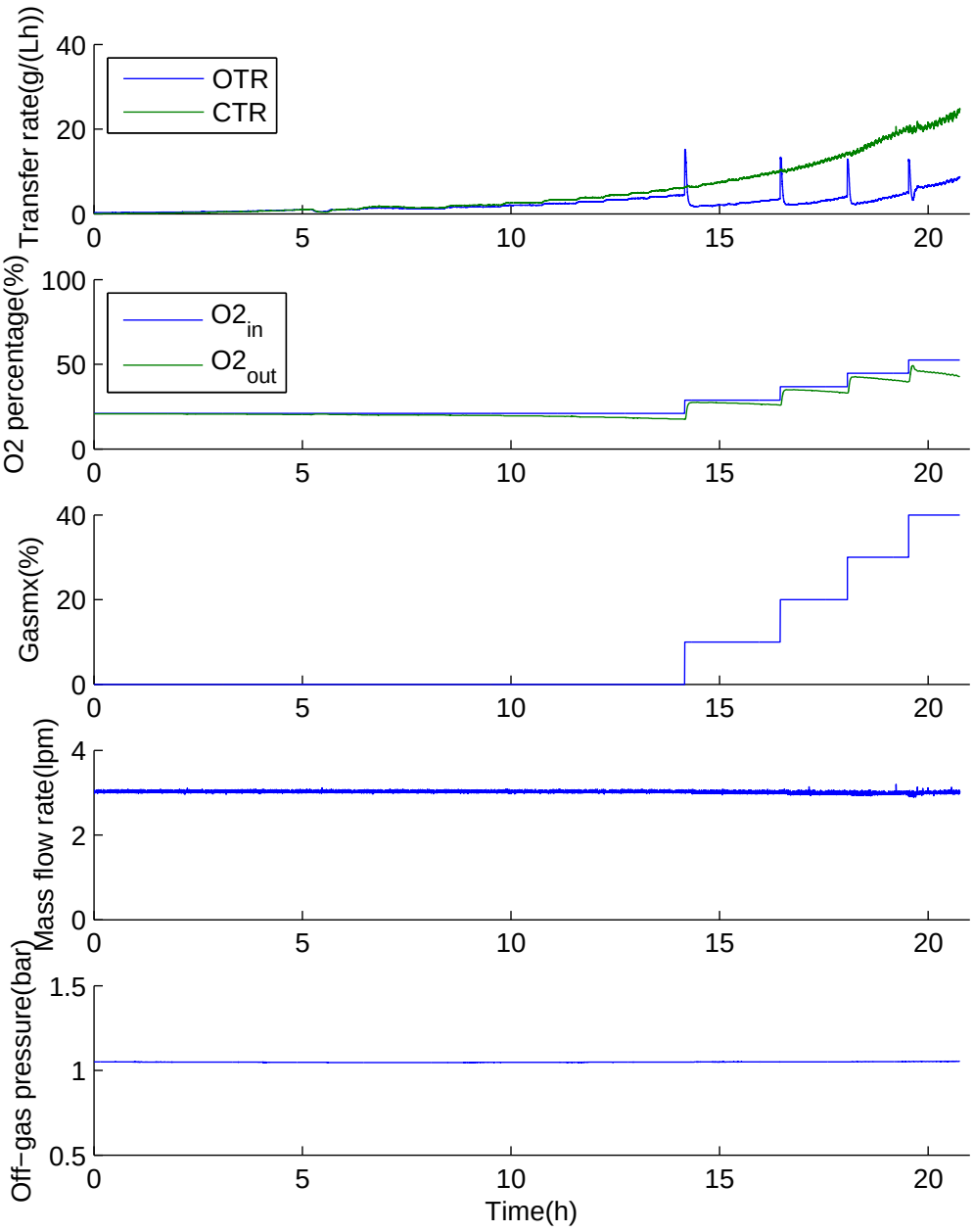


Figure 4.13: Experimental O2 transfer rate(OTR) and CO2 transfer rate(CTR)

4.3.3 Miscalculation error source diagnose

Several methods are attempted to diagnose and correct this problem. At first, a Gasmx calibration experiment is performed to make sure that the Gasmx control is working in a correct manner. For this experiment, the bioreactor is connected in the loop with 2L of water. The stir speed is set to 700 rpm. The mass flow rate is set to 6 lpm. Before this experiment, the O2 tank and atmospheric air supply pressure is equilibrated to minimize the fluctuation due to gas switch. During the experiment, the gasmx percentage is increased by 10 percent steps every 15 minutes. This 15 minutes time interval will give enough time for the oxygen measurement to settle to a constant value. The result of the experiment is illustrated in figure 4.14. It can be observed that the off-gas measurement value gradually settles to the expected value, which indicates that the gasmx mechanism is function correctly.

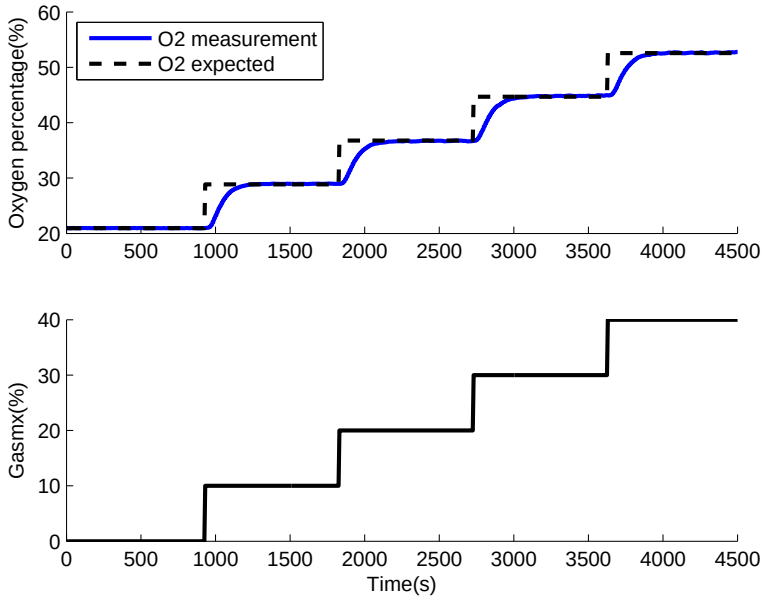


Figure 4.14: Oxygen percentage with different Gasmx values

The expected oxygen percentage values and the measured oxygen percentage values are listed in table 4.2. The relative error for the gasmx mechanism is within 0.5%. Thus, we can trust that the oxygen percentage expected is the same with the measured value, and the gasmx percentage provides a valid conversion to oxygen input percentage.

The mass flow controller only measures and controls the gas flow rate at the input end of

Gasmx value(%)	O2 Expected(%)	O2 Measured(%)	Relative error(%)	$\tau_{63}(s)$
10	28.86	28.95	0.3	120
20	36.52	36.7	0.5	120
30	44.67	44.85	0.4	120
40	52.58	52.63	0.1	120

Table 4.2: Oxygen percentage with different Gasmx values

the bioreactor. It is often assumed that the input flow equals the output flow for simplicity. This assumption is widely used in the OTR related literature. However, the amount of gas consumed by the culture does not necessarily equal the amount of gas produced. An improved method based on nitrogen conservation is adopted for more accurate OTR calculation[21].

$$\text{OTR} = \frac{M_f \rho_{o_2}}{V_1}(b_0 - b_1 \frac{1 - b_0 - P^{in}_{CO_2} - P^{in}_{H_2O}}{1 - b_1 - P^{out}_{CO_2} - P^{out}_{H_2O}}), \tag{4.16}$$

where $P^{in}_{CO_2}$ and $P^{in}_{H_2O}$ are the CO_2 and H_2O percentage in the input gas. $P^{out}_{CO_2}$ and $P^{out}_{H_2O}$ are the CO_2 and H_2O percentage in the output gas. It is assumed that the total amount of nitrogen remains unchanged in the input and output gas component. There is no sensor to measure the off-gas flow rate. In this method, the off-gas flow rate is approximated by scaling the input gas flow rate with the nitrogen percentage.

Both $P^{in}_{CO_2}$ and $P^{in}_{H_2O}$ are assumed to be zero. Because atmospheric air and pure oxygen contains less than 0.1% CO_2. Also the input air goes through a filter which contains desiccant and removes all the vapor. Apart from CO_2, O_2 and H_2O, all the other inert gas components are considered as N_2 for simplicity. The off-gas sensor can also be used to measure and calculate the percentage of $P^{out}_{CO_2}$ and $P^{out}_{H_2O}$.

The nitrogen conservation method is applied on the same experiment data. The OTR calculation results are illustrated in figure 4.15. It can be observed that the compensated method makes the calculation even worse. OTR becomes negative when the Gasmx is turned on. Further investigation is needed to get correct OTR signal.

A comparison of the gas components in the input and output gas is listed in table 4.3. The off-gas sensor does not directly measure the vapor percentage $P^{out}_{H_2O}$, instead it measures the relative humidity RH in the air-water mixture. The simple conversion between $P^{out}_{H_2O}$ and RH can

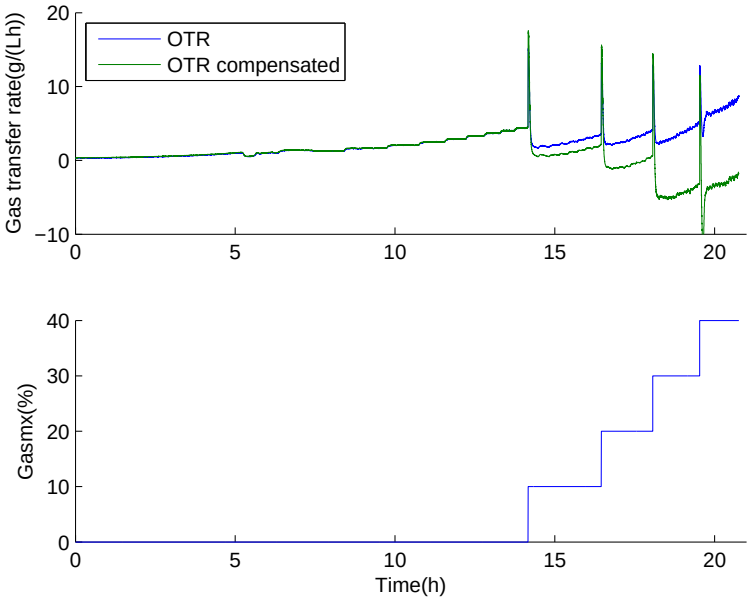


Figure 4.15: Experimental OTR and Compensated OTR

be implemented as follows.

$$P_{H_2O}^{out} = RH \frac{\phi_{H_2O}}{\phi_{off}}, \tag{4.17}$$

where ϕ_{H_2O} is saturated water vapor pressure at 36.5 °C, ϕ_{off} is the pressure of the off-gas from the bioreactor. During the fermentation experiment, RH fluctuates around 23.7%.

From table 4.3, it can be concluded that $Mf_{in} < Mf_{out}$. Because the volumetric nitrogen percentage is higher in the input flow than the output flow. The total quantity of nitrogen should remain the same. This might lead to problem with the OTR calculation, due to the assumption of equal input and output flow. By comparing the nitrogen percentages before and after Gasmx change, it can be concluded that the cells produce more off-gas when the Gasmx is increased. This is problematic, because the *E. coli* metabolism should not change suddenly with different input oxygen percentage.

4.3.4 Proposed solution to correct miscalculation

To be able to use different Gasmx percentage and mass flow rate is important for extending the fermentation time and achieving high biomass density. However, the problem in OTR calculation can not be fully diagnosed and corrected with the current set of sensors. The easiest way to fix the

Gas component	O_2(%)	CO_2(%)	N_2(%)	H_2O(%)
Input gas(0% Gasmx)	21	0	79	0
Output gas(0% Gasmx)	17.6	3.4	73.2	5.8
Input gas(10% Gasmx)	28.9	0	71.1	0
Output gas(10% Gasmx)	27.5	3.5	63.2	5.8

Table 4.3: Input and output gas component analysis when Gasmx switches from 0% to 10%

calculation error is by adding another flow meter to the output gas. The new flow meter can also be used to verify how accurate the compensation method works.

Chapter 5

Conclusions and Recommendations

This thesis presents the design and implementation of an adaptive estimator to accurately track the real-time OTR of aerobic bioprocess. The estimator is motivated by the need to acquire a low latency OTR signal for advanced aerobic bioprocess estimation and control. In order to compensate for the lag in the signal, the head space dilution effect and off-gas sensor lag are included into the oxygen transfer dynamic system. The low latency stir speed and dissolved oxygen signals are also incorporated into the model. By combining the accuracy of the off-gas analyser and the low latency property of the dissolved oxygen dynamics, the designed adaptive estimator can achieve both high accuracy and low latency.

The convergence of this adaptive estimator is proved under the condition of persistent exciting input. A constant step zig-zag DO controller was implemented to excite the signals and satisfy the convergence requirement. Another more general variable step zig-zag DO controller was proposed to enable it to work with different dissolved oxygen demand.

The adaptive estimator was verified specifically against the *E. coli* fermentation. A Simulink model that simulates the *E. coli* metabolism is developed based on the Xu *E. coli* model. This model accurately represents how the *E. coli* takes in substrate and oxygen to produce biomass, while generating carbon dioxide and acetate. The OTR estimator was first implemented on this Simulink model. Simulated glucose feed rate pulses, which lead to quick changes in OTR and OUR, were generated during the fermentation. Simulation results show that the estimated OTR signal accurately compensates for the lag and tracks the simulated OTR signal. The application of this estimator for overflow metabolism detection was also discussed.

The OTR estimator is implemented on a real bioreactor system. The hardware and software setup for the bioreactor system is elaborated in the experiment chapter. The Sartorius bioreactor is integrated with two balances, mass flow controller and off-gas sensor to form the complete fermentation system. A fermentation control and data acquisition model is implemented in Simulink. The model communicates with different sensors and actuators through serial, UDP and OPC communication. Fermentation experiments using similar parameters with the simulation were performed. The experiment results show that the OTR estimator provides high accuracy and low latency OTR signal. It can also be used to detect the overflow metabolism phase when a glucose feed rate pulse is applied to the culture. The OTR estimator accurately tracked the *E. coli* metabolism and will be an integral part in the development of maximizing controllers for oxidative metabolism.

In addition to the adaptive OTR estimator, a glucose feed rate estimator based on Kalman filter is designed and implemented. The accuracy of the peristaltic pump is affected by the fluctuation of the pump and air bubbles in the tubing. The estimator improves the accuracy of the feed rate signal by appropriately combining the pump command with the balance data. The result outperforms the previous method of low-pass filtering the balance data. This method has a greater advantage when volatile solutions such as ammonia are used in the pipe.

Several Matlab based softwares for the bioreactor fermentation are presented in the appendices. The fermentation experiment GUI interacts with the user to display fermentation variables and send user commands to the bioreactor in real-time. The display graph software uses the JFreeChart Java extension to visualize multiple fermentation variables. It enables the easy comparison and analysis across different online and offline variables. The fermentation experiment database integrates all the online and offline data from 11 different experiments. It enables easy access to the past experiment data. The database can also be exported into excel files according to user requirement.

The work in this thesis enables the high accuracy and low latency tracking of OTR signal during the fermentation process. It has several interesting extensions for the future work.

1. The OTR estimator can be used to detect the overflow metabolism phase of the culture by identifying the OUR plateau region. By using a slowly increasing feed rate ramp pulse, it can be used to determine the exact feed rate at which the cells enter overflow. During the experiment, the controller can regularly determine the maximizing feed rate and set it to the feeding profile. Thus, the controller can adaptively adjust and achieve the best growth rate.

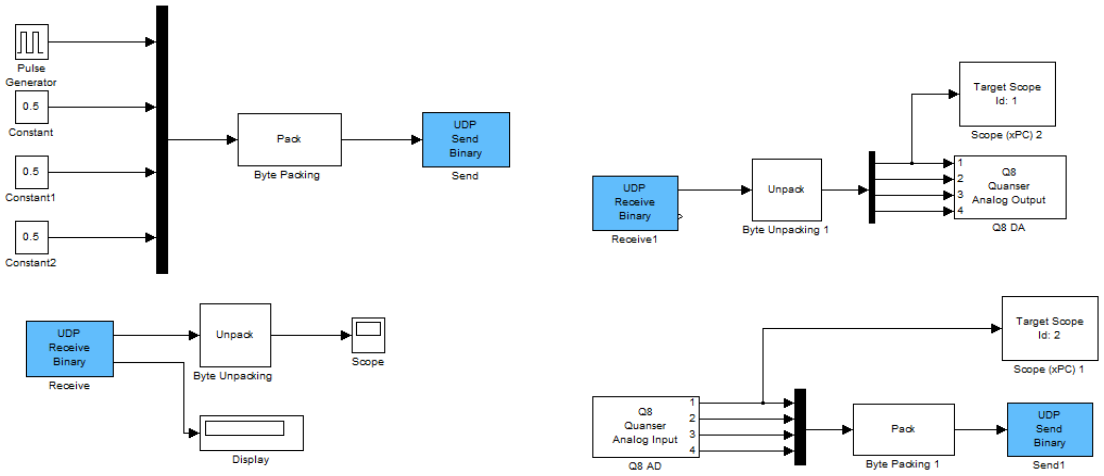
2. In this work, the OTR estimator is only tested against the *E. coli* fermentation for a specific bioreactor. The estimator can be applied to different micro-organism fermentation with different scale of bioreactor. For example, the mammalian cell fermentation requires much lower mass flow rate than *E. coli* . Thus, the lagging effect in the OTR signal will be more significant. If the mass flow rate is small enough, the gas transfer delay should be taken into consideration. Different scales of bioreactor could also lead to different headspace dilution effects.

3. The gas transfer rate modeling method can also be applied to other gas components. For example, by adding a dissolved carbon dioxide sensor, a similar adaptive estimator can be designed to estimate the carbon dioxide production rate.

4. The display graph software is an offline data visualizing tool. It is not implemented online because of the high computational resource requirements, which might lead to a pseudo real-time violation during the fermentation experiment. It still remains to create a computational efficient online multi-axis display program.

Appendices

Appendix A Simulink Code



(a) Host PC model (b) Target PC model

Figure 1: xPC target Simulink model

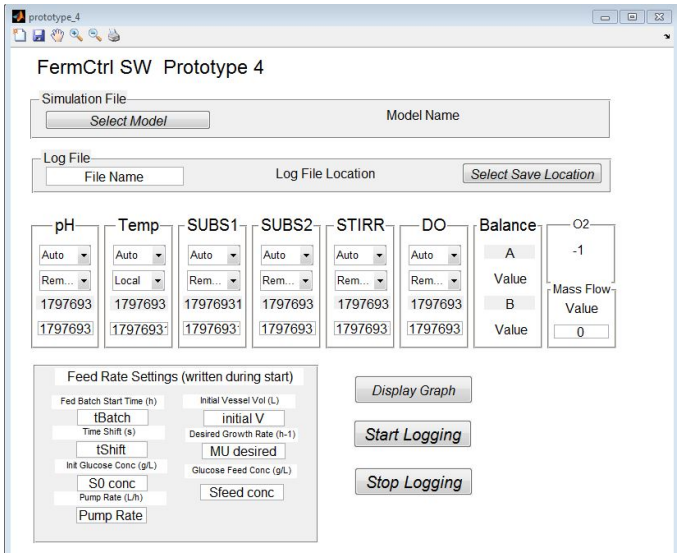


Figure 2: Fermentation user interface

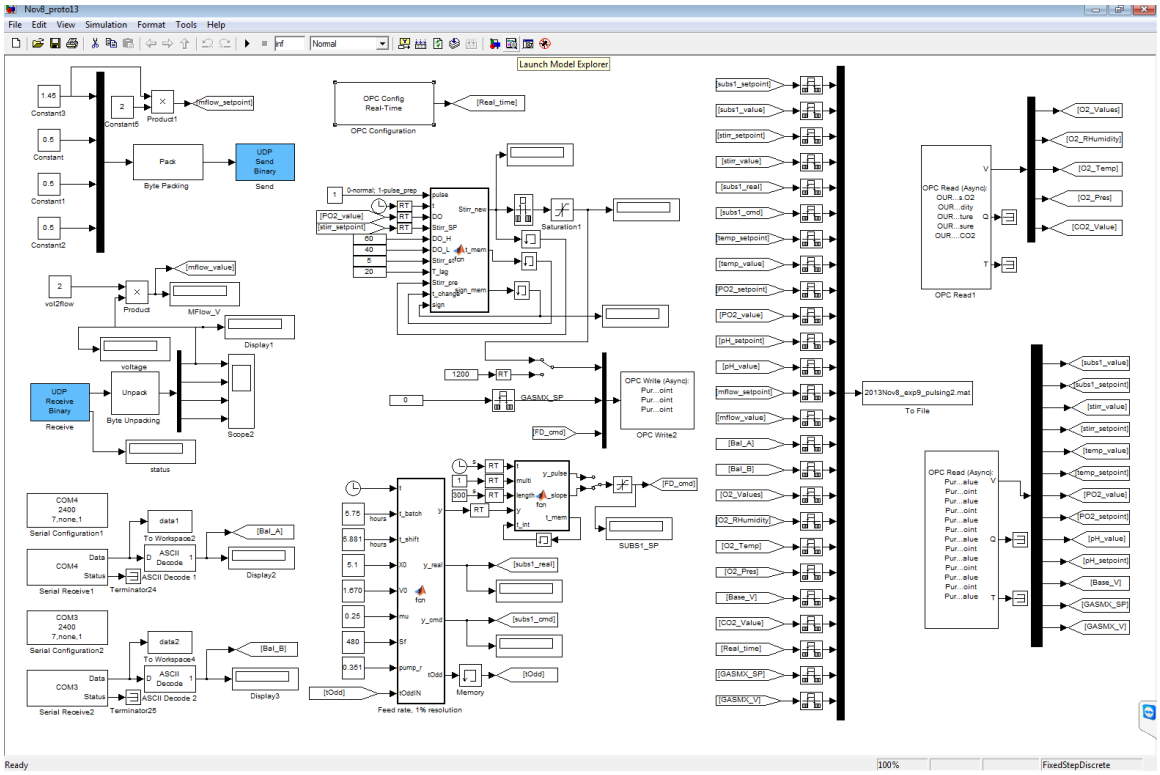


Figure 3: Fermentation experiment Simulink model

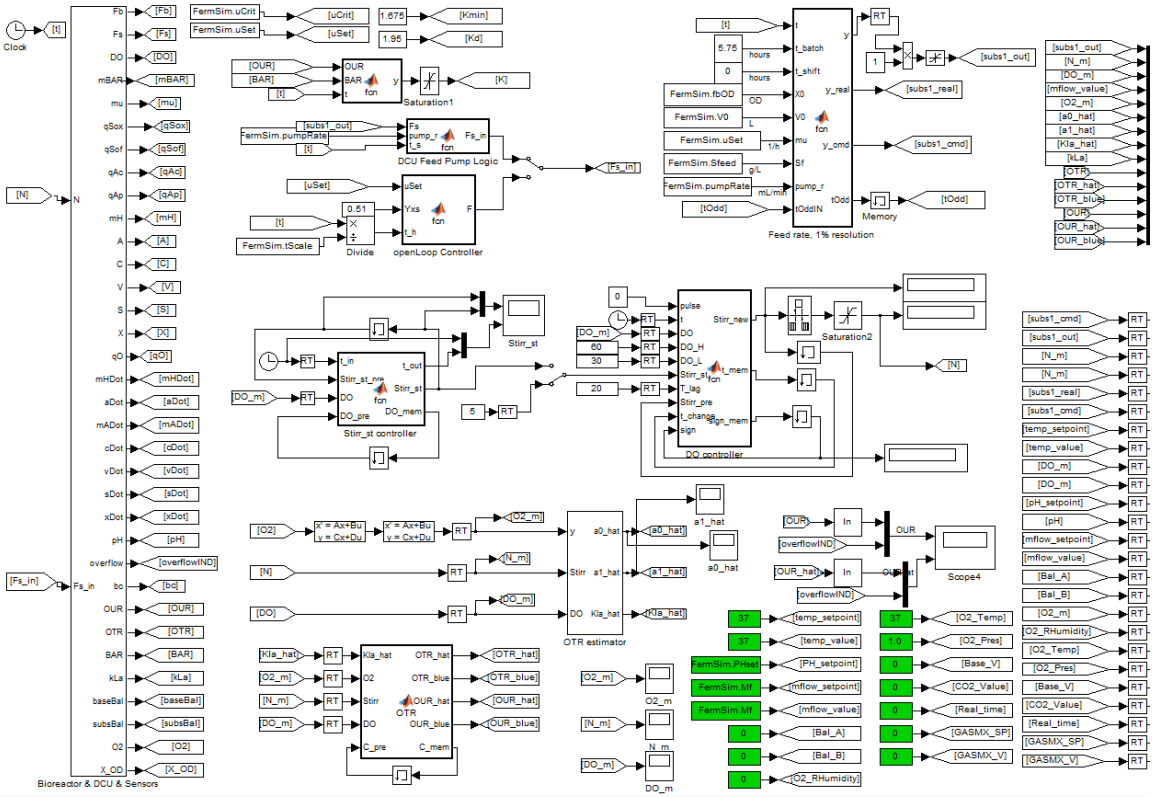


Figure 4: Simulated Xu model

Appendix B Diagrams

Eleven different *E. coli* fermentation experiments have been performed on the bioreactor system. Because our understanding of the fermentation process is gradually building up on previous experience. Different experiments are performed under different conditions.

The substrate used for the first ten experiments are glucose. And the last one is using xylose. The dissolved oxygen sensor has gone through several changing. At the beginning, efforts are made to tune the PID parameters to achieve more smooth DO control. Then, the PID controller is considered too noisy and making the stirring motor overheat. A step DO controller is used by simply step up the stir speed whenever the DO demand increases. This controller gives much smoother signal that are easy to comprehend. Later, a constant step zig-zag controller is used to satisfy the persistently exciting requirement of the signal. In the end, an improved variable step zig-zag controller is implemented to fit the needs of different bioreactors. Among all the experiments, there is one experiment with BlueSens data logging issue. The Gasmx mechanism and mass flow controller are used in several of the experiment to extend the fermentation time and achieve higher biomass production.

All those different configurations of the experiments give different perspectives to *E. coli* fermentation experiment. Table 1 is a summery of the experiment information for the eleven fermentation experiments.

Exp no.	Date	substrate	DO control	BlueSens	Gasmx	Mf	offline measurement
1	130719	glucose	PID	✓	×	✓	✓
2	130725	glucose	PID	✓	×	✓	✓
3	130808	glucose	PID	✓	✓	✓	✓
4	130830	glucose	step	✓	✓	×	✓
5	130916	glucose	step	✓	✓	×	✓
6	130920	glucose	step	✓	✓	×	✓
7	131015	glucose	constant zig-zag	×	×	×	✓
8	131101	glucose	constant zig-zag	✓	×	×	✓
9	131107	glucose	constant zig-zag	✓	×	×	✓
10	131218	glucose	constant zig-zag	✓	×	×	✓
11	140325	xylose	variable zig-zag	✓	✓	×	×

Table 1: Experiment variable information

In order to better manage the data acquired from those experiments, a Matlab fermentation database has been created. The fermentation database contains both online and offline data from the experiment. A detailed description of the variables is given in Table 2. This database provides easy access to all the fermentation experiment data. When conducting data analysis across different experiments, they can be easily accessed by changing the experiment index. Also, the database can be exported into excel spread sheet according to the user's demand.

The MFCS OPC server acts like a bridge between the bioreactor system and the main computer. The OPC server contains a list of different tags that represent different variables on the DCU controller. A complete list of the tags on the MFCS OPC server is given in Table 3. The Matlab Simulink model is a OPC client for the OPC server. The OPC client can subscribe to the tags on the server, read and write to server. It can be observed that each variable contains six properties, among which Value, Ctrl_status, Ctrl_mode and Setpoint are the most important properties. The Simulink fermentation model can read the actual value and send setpoint online through the OPC server.

Variable	Variable description
exp{i}.ton	time step of online data(s)
exp{i}.toff	time step of offline data(h)
exp{i}.Acetate	offline acetate measurement(g/L)
exp{i}.Glucose	offline glucose measurement(g/L)
exp{i}.OD	offline optical density measurement
exp{i}.Data	collection of all the online fermentation data
exp{i}.Data(:,1)	online substrate setpoint(%)
exp{i}.Data(:,2)	online substrate value(%)
exp{i}.Data(:,3)	online stir speed setpoint
exp{i}.Data(:,4)	online stir speed value
exp{i}.Data(:,5)	online pump 2 setpoint
exp{i}.Data(:,6)	online pump 2 value
exp{i}.Data(:,7)	online temperature setpoint
exp{i}.Data(:,8)	online temperature value
exp{i}.Data(:,9)	online dissolved oxygen setpoint
exp{i}.Data(:,10)	online dissolved oxygen value
exp{i}.Data(:,11)	online pH setpoint
exp{i}.Data(:,12)	online pH value
exp{i}.Data(:,13)	online mass flow rate setpoint
exp{i}.Data(:,14)	online mass flow rate value
exp{i}.Data(:,15)	online balance A value
exp{i}.Data(:,16)	online balance B value
exp{i}.Data(:,17)	online off-gas oxygen value
exp{i}.Data(:,18)	online off-gas relative humidity
exp{i}.Data(:,19)	online off-gas temperature
exp{i}.Data(:,20)	online off-gas pressure
exp{i}.Data(:,21)	online base totalizer value
exp{i}.Data(:,22)	online off-gas CO2 value
exp{i}.Data(:,23)	online pseudo real-time violation
exp{i}.Data(:,24)	online Gasmx setpoint
exp{i}.Data(:,25)	online Gasmx value

Table 2: Variable description for the fermentation experiment database, i is the number of the experiment

Variable	Properties	Specifications
AIRFL	Var_status	Word, read only
	Value	Real8, read and write, 0.0-3.0, slpm
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0.0-3.0, slpm
	Ctrl_output	Real8, read only, offVal %
GASMX_ctrl	Var_status	Word, read only
	Value	Real8, read and write, 0-100, %
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0-100, %
	Ctrl_output	Real8, read only, offVal %
PH	Var_status	Word, read only
	Value	Real8, read and write, 2.0-12.0, pH
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 2.0-12.0, pH
	Ctrl_output	Real8, read only, offVal %
PO2	Var_status	Word, read only
	Value	Real8, read and write, 0-100, % saturation
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0-100, %saturation
	Ctrl_output	Real8, read only, offVal %
STIRR	Var_status	Word, read only
	Value	Real8, read and write, 0-1200, rpm
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0-1200, rpm
	Ctrl_output	Real8, read only, offVal %
SUBS1	Var_status	Word, read only
	Value	Real8, read and write, 0-100, % motor operating time
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0-100, % motor operating time
	Ctrl_output	Real8, read only, offVal %
SUBS2	Var_status	Word, read only
	Value	Real8, read and write, 0-100, % motor operating time
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0-100, % motor operating time
	Ctrl_output	Real8, read only, offVal %
TEMP	Var_status	Word, read only
	Value	Real8, read and write, 0-150, Degree C
	Ctrl_status	Word, read and write, range: local(1), remote(3), rem/cal(7), rem/prf(10)
	Ctrl_mode	Word, read and write, range: off(0), auto(1), manual
	Setpoint	Real8, read and write, 0-150, Degree C
	Ctrl_output	Real8, read only, offVal %

Table 3: MFCS OPC server(BBI.MFCSSOPCS.1) variables

Appendix C Matlab Multi-axis Display With Java Extension

The display graph program is a multi-axis data visualization tool. There are more than twenty different variables for the fermentation experiments. Some of the variables have strong correlation between each other. It is beneficial to have a program that can specify arbitrary combinations of the variable and display on the same graph with desired scale. The display graph program fulfills all these needs.

Figure 5 illustrates the menu of the display graph function. The user can specify the online data file location and offline data file location. Arbitrary combinations of the online and offline variables can be selected to plot together. The range for each of the variables can also be specified. After the "Display Graph" button is clicked, the main program will check to see if all the necessary conditions are satisfied, otherwise the program will stop and display warning. If every necessary parameters are specified properly, the program will start plotting with the JFreeChart Java extension.

Figure 6 shows an example run of the display graph function. Each variable is assigned with a distinct axis with distinct color. This program makes the analysis of the fermentation variables very straight forward. The program can also be extended to display processed or calculated variables.

The drawback of the display graph program is that it takes so much memory and can't be used for online display. The Simulink model running with pseudo real-time simulation requires that enough computing power is assigned to the program to finish the task in each sample time. The display graph program needs to convert the large fermentation data set into Java variables and takes over lots of computing resources. If pseudo real-time violation happens, the simulation can run at the correct rate which will lead to false readings. A more computational efficient method needs to found to satisfy the online display requirement.

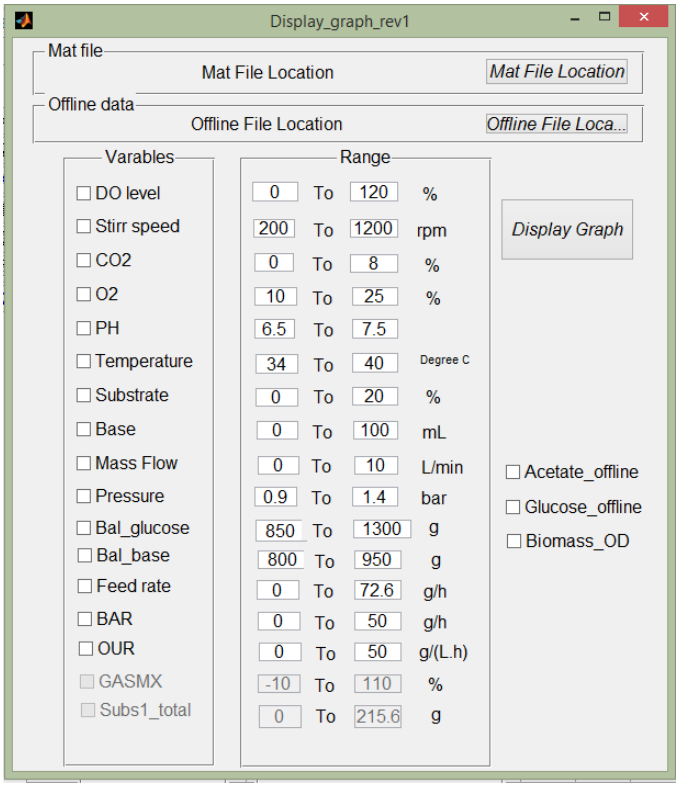


Figure 5: Display graph user interface

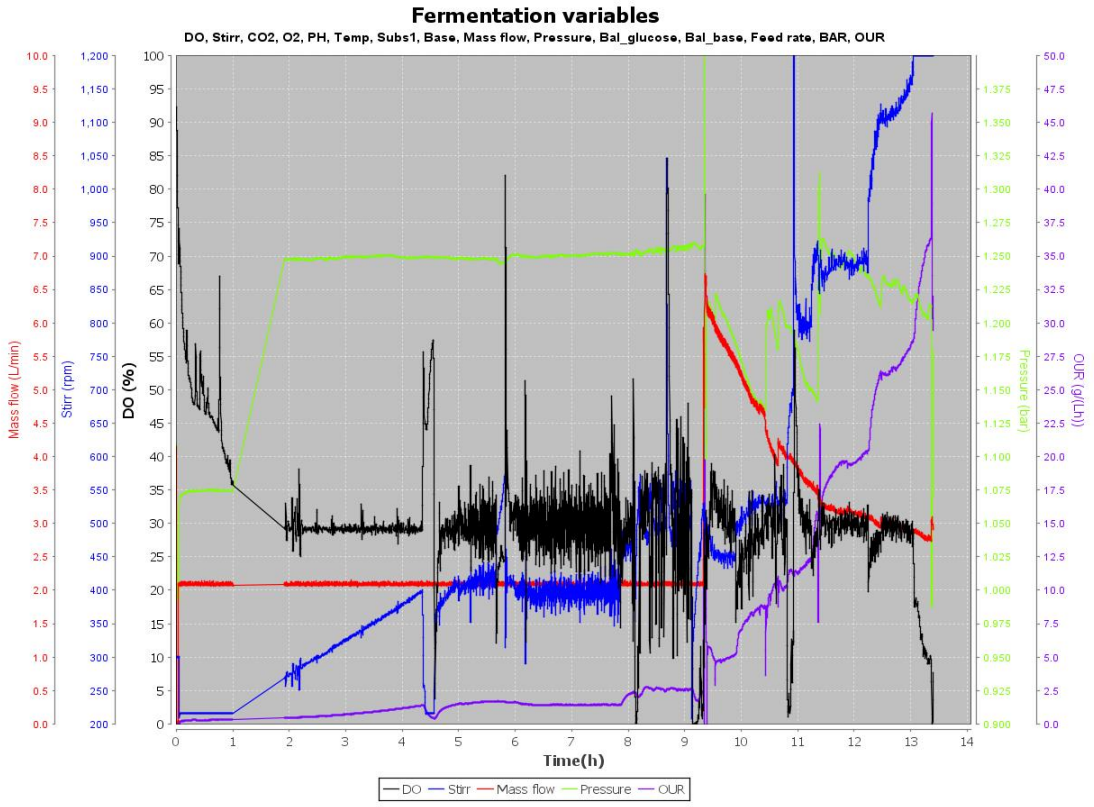


Figure 6: Multi-axis fermentation parameters display

Bibliography

[1] M. Aehle. Application of bluesens gas analyzers in a cell culture process. In *BlueSens Report No. 1*, pages 22–28, 2010.

[2] M. Åkesson and P. Hagander. A gain-scheduling approach for control of dissolved oxygen in stirred bioreactors. In *Preprints 14th World Congress of IFAC*, pages 505–510, 1999.

[3] T. Anderlei and J. Büchs. Device for sterile online measurement of the oxygen transfer rate in shaking flasks. *Biochemical Engineering Journal*, 7(2):157–162, 2001.

[4] Karl-Johan Åström and Torsten Bohlin. Numerical identification of linear dynamic systems from normal operating records. In *Theory of self-adaptive control systems*, pages 96–111. Springer, 1966.

[5] M. Boon, K. C. A. M. Luyben, and J. J. Heijnen. The use of on-line off-gas analyses and stoichiometry in the bio-oxidation kinetics of sulphide minerals. *Hydrometallurgy*, 48(1):1–26, 1998.

[6] H. Djelal, F. Larher, G. Martin, and A. Amrane. Effect of the dissolved oxygen on the bio-production of glycerol and ethanol by hansenula anomala growing under salt stress conditions. *Journal of biotechnology*, 125(1):95–103, 2006.

[7] R. C. Dorresteijn, C. D. De Gooijer, J. Tramper, and E. C. Beuvery. A method for simultaneous determination of solubility and transfer coefficient of oxygen in aqueous media using off-gas mass spectrometry. *Biotechnology and bioengineering*, 43(2):149–154, 1994.

[8] R. C. Dorresteijn, C. D. De Gooijer, J. Tramper, and E. C. Beuvery. A simple dynamic method for on-line and off-line determination of kla during cultivation of animal cells. *Biotechnology techniques*, 8(9):675–680, 1994.

[9] D. Filippou, T. C. M. Cheng, and G. P. Demopoulos. Gas-liquid oxygen mass-transfer; from fundamentals to applications in hydrometallurgical systems. 20:447–502, 2000.

[10] F. Garcia-Ochoa and E. Gmez. Mass transfer coefficient in stirred tank reactors for xanthan gum solutions. *Biochemical Engineering Journal*, 1(1):1–10, 1998.

[11] F. Garcia-Ochoa and E. Gomez. Bioreactor scale-up and oxygen transfer rate in microbial processes: an overview. *Biotechnology advances*, 27(2):153–176, 2009.

[12] B. Gourich, C. Vial, N. El Azher, M. Belhaj Soulami, and M. Ziyad. Influence of hydrodynamics and probe response on oxygen mass transfer measurements in a high aspect ratio bubble column reactor: Effect of the coalescence behaviour of the liquid phase. *Biochemical Engineering Journal*, 39(1):1–14, 2008.

[13] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[14] V. Linek, J. Mayrhoferova, and J. Mošnerová. The influence of diffusivity on liquid phase mass transfer in solutions of electrolytes. *Chemical Engineering Science*, 25(6):1033–1045, 1970.

[15] John R McWhirter, Jia-Ming Chern, and Joesph C Hutter. Oxygen mass transfer fundamentals of surface aerators. *Industrial & engineering chemistry research*, 34(8):2644–2654, 1995.

[16] K. S. Narendra and A. M. Annaswamy. *Stable adaptive systems*. Courier Dover Publications, 2012.

[17] B. Özbek and S. Gayik. The studies on the oxygen mass transfer coefficient in a bioreactor. *Process Biochemistry*, 36(8):729–741, 2001.

[18] N. Patel and J. Thibault. Enhanced in situ dynamic method for measuring $k_L a$ in fermentation media. *Biochemical Engineering Journal*, 47(1):48–54, 2009.

[19] S. Pratt, Z. Yuan, D. Gapes, M. Dorigo, R. J. Zeng, and J. Keller. Development of a novel titration and off-gas analysis (toga) sensor for study of biological processes in wastewater treatment systems. *Biotechnology and bioengineering*, 81(4):482–495, 2003.

[20] M. S. Puthli, V. K. Rathod, and A. B. Pandit. Gas–liquid mass transfer studies with triple impeller system on a laboratory scale bioreactor. *Biochemical engineering journal*, 23(1):25–30, 2005.

[21] David Redmon, William C Boyle, and Lloyd Ewing. Oxygen transfer efficiency measurements in mixed liquor using off-gas techniques. *Journal (Water Pollution Control Federation)*, pages 1338–1347, 1983.

[22] K. Van't Riet. Review of measuring methods and results in nonviscous gas-liquid mass transfer in stirred vessels. *Industrial & Engineering Chemistry Process Design and Development*, 18(3):357–364, 1979.

[23] M. Scheidle, J. Klinger, and J. Büchs. Combination of on-line ph and oxygen transfer rate measurement in shake flasks by fiber optical technique and respiration activity monitoring system (ramos). *Sensors*, 7(12):3472–3480, 2007.

[24] V. Singh. Disposable bioreactor for cell culture using wave-induced agitation. *Cytotechnology*, 30(1-3):149–158, 1999.

[25] S. Suresh, V. C. Srivastava, and I. M. Mishra. Techniques for oxygen transfer measurement in bioreactors: a review. *Journal of chemical technology and biotechnology*, 84(8):1091–1103, 2009.

[26] Mustafa Türker. Development of biocalorimetry as a technique for process monitoring and control in technical scale fermentations. *Thermochimica Acta*, 419(1):73–81, 2004.

[27] S. Valentinotti, B. Srinivasan, U. Holmberg, D. Bonvin, C. Cannizzaro, M. Rhiel, and U. Von Stockar. Optimal operation of fed-batch fermentations via adaptive control of overflow metabolite. *Control Engineering Practice*, 11(6):665–674, 2003.

[28] F. Veglio, F. Beolchini, and S. Ubaldini. Empirical models for oxygen mass transfer: a comparison between shake flask and lab-scale fermentor and application to manganiferous ore bioleaching. *Process biochemistry*, 33(4):367–376, 1998.

[29] B. Xu, M. Jahic, and S. O. Enfors. Modeling of overflow metabolism in batch and fed-batch cultures of escherichia coli. *Biotechnology Progress*, 15(1):81–90, 1999.

[30] J. Yuan and W. M. Wonham. Probing signals for model reference identification. *Automatic Control, IEEE Transactions on*, 22(4):530–538, 1977.