Probing Control of a Fedbatch Fermentation.
Simulation and Implementation in ABB Industrial IT

Tory Li

Title and subtitle
Probing Control of a Fedbatch Fermentation. Simulation and Implementation in ABB Industrial IT (Pulsregulatorn för fedbatchfermentering. Simulering och implementering i ABB Industrial IT).

Abstract
During the production of pharmaceutical products, a central stage is so called fermentation process i.e. cultivation of bacteria in a biological reactor. In the scope of this Master Thesis the product is recombinant protein. The goal for the fermentation process is to keep a high cell growth and a high production of recombinant proteins. For these purposes, an automatic control algorithm has been presented in (Åkesson, 1999), with some modifications described well in (Velut et al, 2002). The algorithm is designed to avoid acetate accumulation while maintaining a high glucose feed by monitoring the dissolved oxygen tension in the biological reactor. The acetate is an undesired byproduct that inhibits both cell replication and production of recombinant proteins.

The automatic control algorithm and a process model has first been implemented in Matlab and later in Control Builder,(a developing environment for IEC 61131 Control Languages. It is part of the Industrial IT 800xA System developed by ABB and is a comprehensive process automation system. The Control Builder is mainly used to implement the controller programs while the Industrial IT 800xA System has been used to control/monitor/log different variables during simulation and later in the field test. After verifying the implementation in Control Builder by comparing the simulation in Soft controller (a simulation tool for real time simulation) with simulation in Matlab, the controller programs are ready for the final test. The final test was performed in a physical biological reactor located at division of Biotechnology at the Chemical Center of Lund Institute of Technology where the controller programs were downloaded into an ABB AC800M controller.

The final test of the controller programs in Lund showed to be very successful at the first attempt. It managed to keep a high cell growth and at the same time it has prevented production of acetate.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden. Fax +46 46 222 44 22

CONTENTS

1   Abstract

During the production of pharmaceutical products, a central stage is so called fermentation process i.e. cultivation of bacteria in a biological reactor. In the scope of this Master Thesis the product is recombinant protein. The goal for the fermentation process is to keep a high cell growth and a high production of recombinant proteins. For these purposes, an automatic control algorithm has been presented in (Åkesson, 1999), with some modifications described well in (Velut et al, 2002). The algorithm is designed to avoid acetate accumulation while maintaining a high glucose feed by monitoring the dissolved oxygen tension in the biological reactor. The acetate is an undesired byproduct that inhibits both cell replication and production of recombinant proteins.

The automatic control algorithm and a process model has first been implemented in Matlab and later in Control Builder,(a developing environment for IEC 61131 Control Languages. It is part of the Industrial IT 800xA System developed by ABB and is a comprehensive process automation system. The Control Builder is mainly used to implement the controller programs while the Industrial IT 800xA System has been used to control/monitor/log different variables during simulation and later in the field test. After verifying the implementation in Control Builder by comparing the simulation in Soft controller (a simulation tool for real time simulation) with simulation in Matlab, the controller programs are ready for the final test. The final test was performed in a physical biological reactor located at division of Biotechnology at the Chemical Center of Lund Institute of Technology where the controller programs were downloaded into an ABB AC800M controller.

The final test of the controller programs in Lund showed to be very successful at the first attempt. It managed to keep a high cell growth and at the same time it has prevented production of acetate.
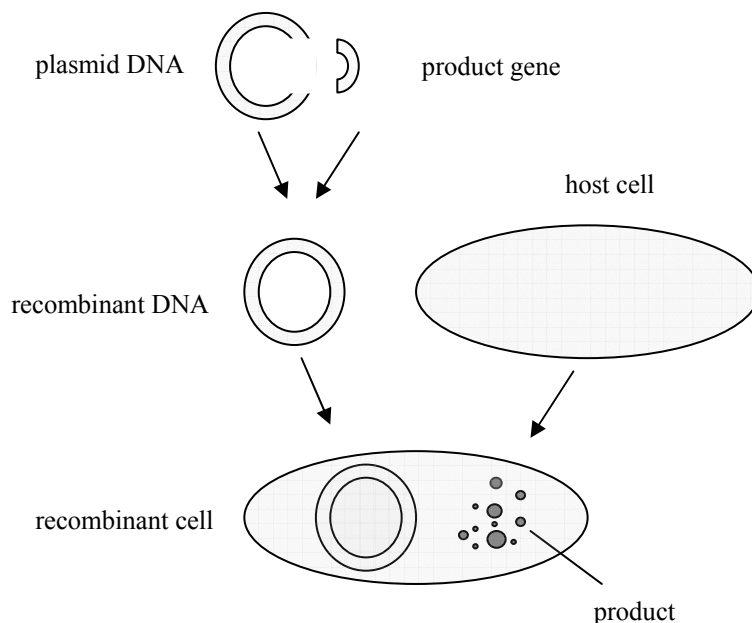
2   Introduction

ABB Corporate Research is one of the participants in a research project, Bio-Man (Biotechnology Manufacturing). The aim of the project is to study biotechnical production of pharmaceutical products. A central stage in the production is the so called fermentation, i.e. cultivation of bacteria in a biological reactor. By gene manipulation, some bacteria strains will be able to produce the desired product that could potentially be used as pharmaceutics in the case of Bio-Man project. This desired product is a recombinant protein.

The goal of this Master Thesis is to implement an automatic glucose feeding strategy to keep high cell growth at the same time preventing production of acetate (an unwanted byproduct in the fermentation process) by using ABB's AC800M controller.

2.1 Background

DNA (deoxyribonucleic acid) contains the genetic information about living cells. By introducing a foreign DNA sequence for recombinant protein into a plasmid DNA and then inserting the resulting DNA sequence into a host cell, the host cell will be able to produce the desired product, i.e. recombinant protein, see Figure 1. The host cell that is used in this Master Thesis is Escherichia coli and it is also used during industrial-scale production of medical products such as human growth hormone and human insulin.

**Figure 1**   Forming of recombinant cell

Fermentation, i.e. cultivation of bacteria in a biological reactor, is a central stage during the production of pharmaceutical products. In the scope of this Master Thesis the fermentation is a combination of a batch phase followed by a fed-batch phase. During the batch phase, all substrate components are added initially and the cells will grow exponentially until some

substrates are consumed or the concentration of some byproducts becomes inhibitory. When the substrate is consumed, in this case the glucose, the batch phase is over and the fed-batch phase starts. During the fed-batch phase, the derived feed rate of glucose varies depending on the growth of the cells and the concentration of the byproduct that is produced. This Master Thesis will concentrate on the automatic control of the feed rate during the fed-batch phase. The purpose of the automatic control is to avoid production of byproduct (often acetate) and at the same time retain a high cell growth in the biological reactor.

The cells in the fermentation process consume glucose and oxygen and at the same time they replicate themselves. When there exists a substantial amount of cells in the biological reactor, induction is the next step i.e. chemical agents are injected into the reactor to start production of recombinant proteins. If the feed rate is too high during the cultivation process, the byproduct acetate is produced. Then the cell growth and the production of recombinant protein are inhibited. If on the other hand the feed rate is too low, the cells starve. A natural idea is then to avoid starvation and production of byproduct by feedback control.

A major difficulty, however, is that the byproduct, acetate, cannot be reliably measured. Therefore an indirect approach was developed at the Automatic Control Department of Lund Institute of Technology together with Pharmacia (now Pfizer), and presented in (Åkesson, 1999). The main idea behind the algorithm is that when acetate is produced an extra feed of glucose shows no response in dissolved oxygen tension (Dot). Details about the feedback algorithm used in this Master Thesis refer to Appendix A.

By varying the stirrer speed (N) in a biological reactor, the oxygen transfer to cells can be controlled. With different feed rate (F) the cells will consume different amount of oxygen. If a pulse of feed is given to the reactor at the same time as the stirrer speed is kept constant, there will be a certain decrease of oxygen level. If the oxygen deviation is not noticeable, it indicates that the byproduct acetate has been produced and the feed rate is decreased. Otherwise the feed rate is increased unless the maximum stirrer speed has been reached. The oxygen level is measured by a dissolved oxygen tension sensor (Dot sensor) and sent to both the controller for stirrer speed and the controller for feed rate, see figure below. Since the stirrer speed has to be kept constant during the feed pulse, the mode of the Dot controller is controlled by the Pulse controller.
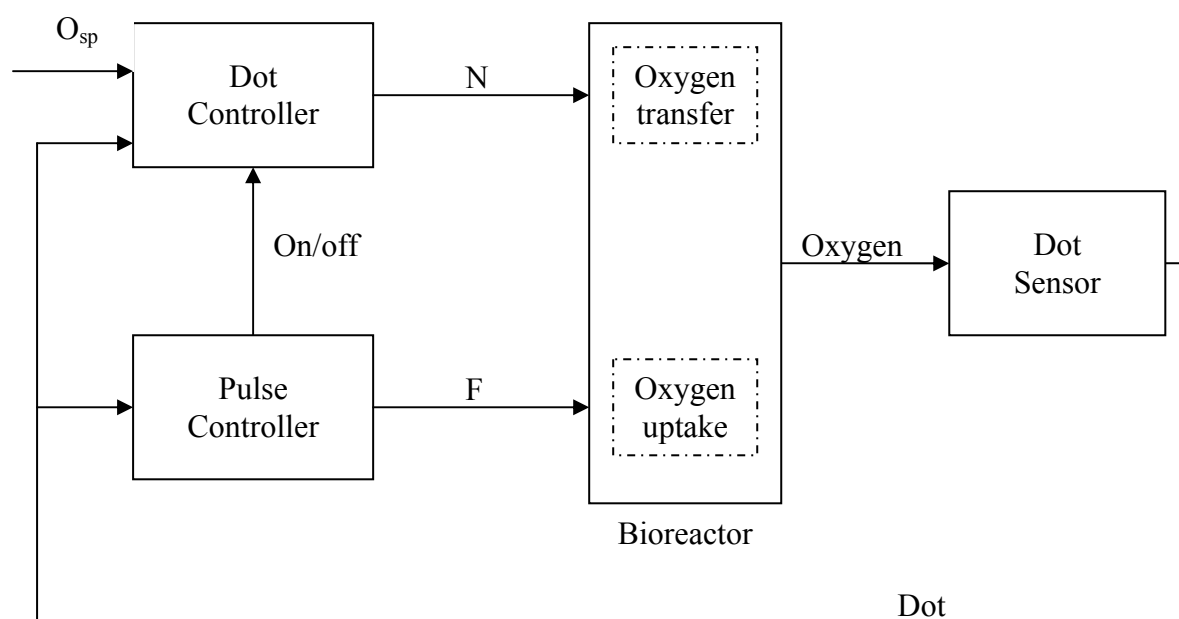
Figure 2 Block diagram of the closed-loop system

2.2 Ideas behind the Master Thesis

The first task is to simulate the cultivation process in Matlab by using the toolbox Simulink. ABB has developed a basic simulation program in Simulink for this purpose. By making some improvements and modifications, the cultivation process is simulated.

After verifying that the simulations in Matlab are correct, the next step is to implement the process model and the controllers (both Dot controller and Pulse controller) in Control Builder, the ABB development environment for IEC 61131 Control Languages. It is part of the Industrial IT 800xA System that is a comprehensive process automation system. With Matlab simulations as the reference simulations, simulations in Control Builder are compared and verified. In Simulink, the model and the controllers are simulated in the continuous time domain, but in a physical controller the entire process can only be simulated in the discrete time domain. This requires discretization of the model and the controllers.

After verifying the controller programs and the model implemented in Control Builder, the final field test is the next step. Here a biological reactor at Chemical Center in Lund is used. The implemented program in Control Builder that is verified with the Matlab simulation is downloaded into an AC800M controller with physical I/Os units connected. The AC800M controller regulates both the feed rate and the stirrer speed. The dissolved oxygen tension is monitored during the entire cultivation process.

3   Simulink Implementation

By using the Matlab toolbox, Simulink, the cultivation process including the process model, the Pulse controller, the Dot controller and the Dot sensor (the sensor to measure the dissolved oxygen tension) are built together. Both the Pulse controller and the fermentation model are implemented by using so called S-functions. The Dot controller and the Dot-sensor are built by using reusable blocks provided by Simulink.
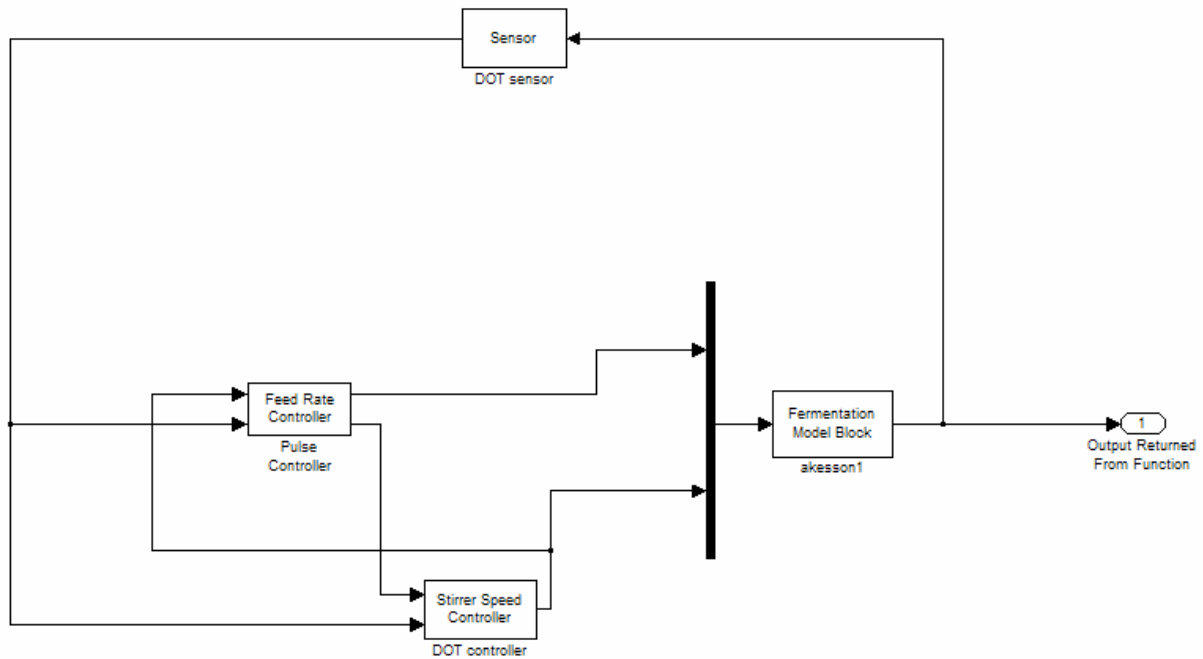
Figure 3   The Simulink implementation of the cultivation process

The fermentation model starts with user-defined initial conditions and calculates the dissolved oxygen concentration in the reactor. This concentration is measured by the Dot sensor and the dissolved oxygen tension (Dot) is determined. The Dot is then sent to both the Pulse controller and the Dot controller. In order for the Pulse controller to determine the feed rate, the present stirrer speed is also sent to the Pulse controller from the Dot controller. During a feed rate pulse the Dot controller has a constant stirrer speed. Since feed pulses are generated by the Pulse controller, the Pulse controller also controls the Dot controller. With the dissolved oxygen tension and the present stirrer speed, the Pulse controller determines an appropriate amount of glucose that is pumped into the reactor (the fermentation model). Depending on the control signal from the Pulse controller and the dissolved oxygen tension measured by the Dot sensor, the Dot controller gives a suitable stirrer speed and this stirrer speed is passed to both the fermentation model and the Pulse controller. When the fermentation model has received the new feed rate and the new stirrer speed, it will update the derivatives of for example, X (Biomass concentration), V (Liquid volume), G (Glucose concentration), A (Acetate concentration) and O (dissolved oxygen tension). The newly calculated dissolved oxygen tension is then measured by the Dot sensor and sent to both the Pulse controller and the Dot controller. The complete algorithm for the Pulse controller (the so called the feedback algorithm) is shown in Appendix A and the equations for the process model are shown in Appendix B.

During the early stage of the implementation in Matlab, the Pulse controller and the model are running in continuous time domain while the Dot controller is implemented in discrete time domain. Since the implementation in Control Builder will be in the discrete time domain, a discretization of both the Pulse controller and the model is necessary. This discretization is then implemented in Matlab and verified by the simulation in continues time domain. Then it is implemented in Control Builder. Further, in order to speed up the simulation of the cultivation process, the entire system is scaled. Both the discretization method and the scaling method is presented in separate chapters.

3.1 Process Model

In the biological reactor, there are several variables to be computed in order to model the cultivation process. These variables are V (Liquid volume), X (biomass concentration), G (glucose concentration), A (acetate concentration) and O (oxygen concentration). In Appendix B there are expressions to calculate the derivatives of these variables and the results are then used to update these variables.

The implementation of the process model is achieved by using an S-function in Matlab. At the beginning, the model is simulated in continuous time domain that is later discretized to be simulated in discrete time domain. The discretization is described in Chapter 3.2.

3.2 Pulse Controller

The Pulse controller is implemented according to the feedback algorithm in Mats Åkesson's PhD thesis (Åkesson, 1999), with some modifications described well in (Velut et al 2002). At Biotechnology department in the Chemical Center in Lund there is a SattCon system (an automatic control system developed by ABB and it is a predecessor to Industrial IT). The implementation of the Pulse controller in this Master Thesis is almost the same in functionality as the implementation in the SattCon system.

The feedback algorithm addresses the problem of acetate accumulation, which inhibits both cell growth and production of recombinant proteins. The key idea is to exploit the characteristic saturation in the respiratory system that occurs at acetate formation, when the glucose uptake rate q_g exceeds q_g^{crit} (Critical specific glucose uptake rate). By superimposing pulses of the feed rate and measuring the response of the dissolved oxygen tension, an appropriate feed rate can be computed and used.

After a feed pulse, the latest dissolved oxygen response at the end of the pulse is measured. This measured Dot is believed to be very close to the lowest dissolved oxygen response during the feed pulse. If the dissolved oxygen response is less than Oreac (a user-defined constant to determine the appropriate dissolved oxygen reaction from a pulse), then the q_g^{crit} has been reached and the feed rate is reduced to prevent production of acetate. Otherwise the feed rate is increased unless the stirrer speed has reached its maximum. The amount of the feed increment (ΔF) is determined by a PI-controller. An anti-windup method is used to prevent integrator windup. In the Pulse controller, the Integral part has been implemented. But during simulations and the field test in Lund, it has been bypassed.

Between the pulses, the Pulse controller checks the stirrer speed frequently. If the stirrer speed exceeds its maximum limit, then the feed rate is lowered gradually. In this way, the cell growth is slowed down and the cells consume less oxygen.

The feedback algorithm that controls both the feed rate and the Dot controller using the feed pulse combined with the dissolved oxygen tension response is given in Appendix A.

3.3 Dot controller

It is hard to find a constant PI parameter setting which gives acceptable performance during entire cultivation process. The usual solution is to use so called gain scheduling method. Here a slightly different approach is used.

The Dot Controller controls the dissolved oxygen tension in the reactor by varying the stirrer speed. It is basically a PI-controller with gain and Ti determined by using the λ-tuning method.

The Dot model
The transfer function from stirrer speed to oxygen measurement (Åkesson, 1999) can be approximated

$$G_{ON}(s) = \frac{b}{s + k_L A} * \frac{1}{s * T_p + 1} = \frac{\frac{b}{k_L A}}{\frac{s}{k_L A} + 1} * \frac{1}{s * T_p + 1}$$ (1)

T$_P$ is a time constant for the dissolved oxygen sensor and is here approximated to 20 s. $k_L A$ is volumetric oxygen transfer coefficient and is approximated by the following expression
$$k_L A(N) = \alpha * (N - N0), N \in (300,1100)rpm$$ (2)
where α is a constant which in the simulations is set to 1.4, and N0 is another constant for which the value 150 has been used.
The b parameter in (1) is given by the following equation
$$b = (O^* - O) * \frac{\partial k_L A}{\partial N} = (O^* - O) * \alpha$$ (3)
where O* is dissolved oxygen tension in equilibrium with air bubbles, which has been set to 100. O is the measured dissolved oxygen tension and can be approximated by its setpoint 30. Thus, b is approximately constant, (100-30)*1.4=98.

Studying the two time constants of G$_{ON}$, it is noticed that Tp is constant over the entire fermentation process (Tp ≈ 20s = 0.0056 h), while the other time constant varies with N:
$$\frac{1}{k_L A} \in (\frac{1}{1.4 * 150} = 0.00476, \frac{1}{1.4 * 950} = 0.000752)h$$

Consequently, early in the fermentation process the time constants are similar, but for the major part Tp is the dominant one. For the controller tuning the following approximated transfer function has therefore been chosen
$$G_{ON}(s) = \frac{b/k_L A}{1 + s * T_p}$$ (4)

The λ-tuning
The closed-loop system from setpoint to output can be written as
$$Y = F * R, \text{ where } F = \frac{C * G}{1 + C * G}$$ (5)
Now define the desired F to be
$$F = \frac{1}{\lambda * s + 1}$$ (6)
From (5) the controller C can be solved for, which yields
$$C = \frac{1}{G} * \frac{F}{1 - F}$$ (7)

9

Inserting (4) and (6) then gives

$$C = \frac{T_p}{b/k_L A * \lambda} * (1 + \frac{1}{T_p * s})$$ (8)

By comparing (8) with the equation for an ordinary PI-controller

$$C = K * (1 + \frac{1}{T_i * s})$$ (9)

the following conclusions can be achieved about the constants K and Ti (if λ=Tp):

$$K = \frac{1}{b/k_L A}, Ti = Tp$$ (10)

Inserting (2) and (3) into (10), the following results are achieved.

$$K = \frac{N - N0}{70}, Ti = Tp$$ (11)

However, after several simulations, the Ti has been instead chosen to be 2*Tp. The gain is updated all the time as the stirrer speed varies.

The results from a 10 hours simulation are shown below.
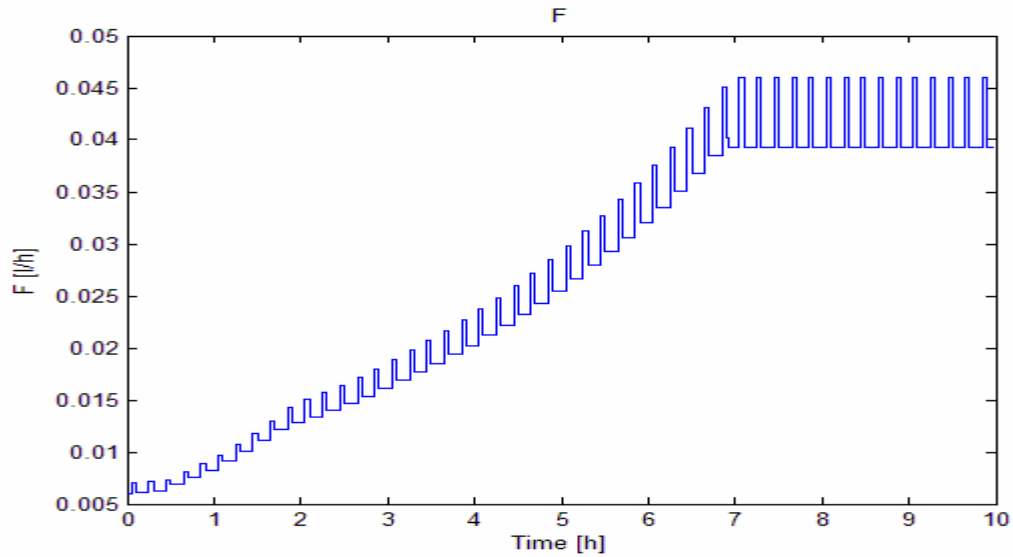
Figure 4   The glucose feed rate
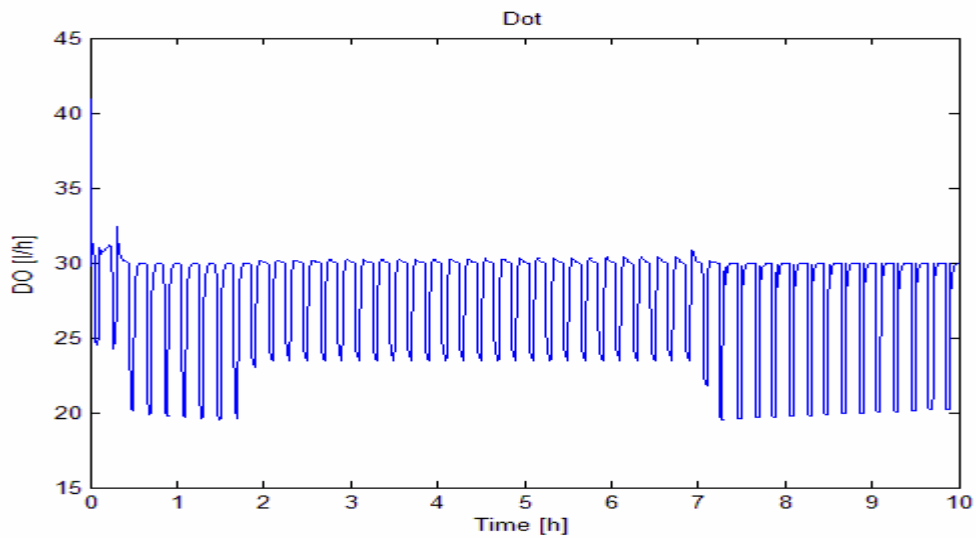
Figure 5   The dissolved oxygen tension

10
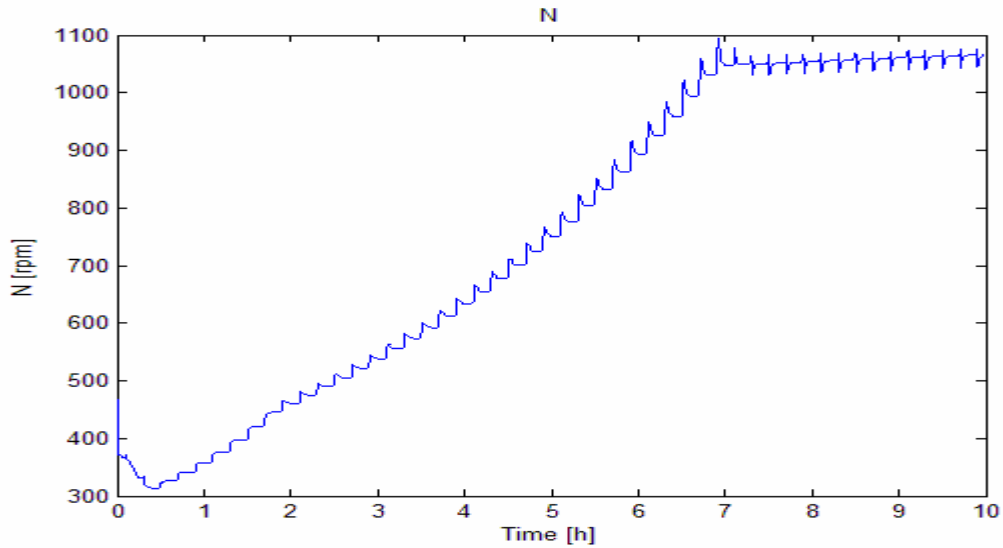
Figure 6   The stirrer speed

The plots are satisfying, since the dissolved oxygen tension remained around its setpoint, 30 percent. The parameters for the Probing controller, the Dot controller and the process model during the simulation with the λ-tuning method are shown in Appendix D.

In these plots, it took almost 7 hours for the stirrer speed to reach its maximum, while later in the field test in Lund it only took 2 hours. This is mainly because of different start values of X (biomass concentration), A (acetate concentration), G (glucose concentration) and F0 (the initial feed at the beginning of the fed-batch phase). For comparison of these parameters refer to Appendix D, Appendix H and Chapter 5.1.

11

4 Implementation in Control Builder

After verifying the cultivation process in Simulink, the next step is to implement the process model and the controllers in Control Builder (part of Industrial IT) by using IEC 61131 Control Languages. Before the implementation in Control Builder, the model has to be discretized in order to run in discrete time domain. This is described in chapter 3.2. The discretization of the model is first performed in Matlab to verify with the implementation in the continuous time domain. Afterwards, the discretized model is rewritten in Control Builder in the same way as it is implemented in Matlab.

Since the Pulse controller is written with variable sample time in Matlab, no discretization is needed. The Pulse controller can be implemented directly with fewer modifications in Control Builder.

The implementation of the Dot controller is achieved by reusing one of the predefined modules, a PID controller. Here only the P-part and the I-part are used. Gain and Ti are updated by the Pulse controller using the λ-tuning method. When an up pulse of feed is sent to the model, the Dot controller is controlled to track/follow the latest output signal from the previous scan. This means that the Dot controller will generate the same stirrer speed during pulse time. Between two pulses, the Dot controller controls the stirrer speed using a set-point (Osp, set-point for the dissolved oxygen tension, often 30%).

The Dot sensor is implemented as a first-order filter.

The process model and the Dot sensor are written in two separate modules. The controllers and the model are implemented by using a new concept, called Control module that is supported by the development environment Control Builder. The Control module promotes object-oriented programming and is specially suited for implementation of large systems. The Control module concept is described in detail in Chapter 3.1.

An application containing both the Probing controller and the process model is built. Since the cultivation simulation runs in the discrete time domain, different sampling intervals are given to the Probing controller and the process model. Then the application is downloaded into a Soft controller. The Soft controller is used to simulate applications without using the real I/O units, but these simulations are as close to the real simulation as possible.

By using developing environment in the Industrial IT 800xA System, the results from the simulations are drawn during the simulation and can be used for analysis after the simulation (this is described in details in the Chapter 3.3). The simulation results can then be extracted from the Industrial IT 800xA System to Matlab to make comparisons between the simulations in Matlab and the simulations in Soft Controller. By using the Industrial IT 800xA System the operator panel including faceplate (online graphical controls to monitor/modify different variables during simulation) for the controllers is built. In this Master Thesis several faceplates have been implemented, but due to the lack of time the operator panel was not completed.

4.1 The Control Builder programming environment

The Control Builder is an environment to implement programs that are used in programmable controllers. The programming program is IEC (International Electrotechnical Commission) 61131 Control Languages. The IEC 61131-3 is the first and so far only, global standard for programmable controllers.

The IEC 61131-3 standard specifies five programming languages: Ladder diagrams (LD), Instruction list (IL), Structured Text (ST), Function Block Diagram (FBD) and Sequential Function Charts (SFC). IL and ST are textual languages while LD, FDB and SFC are based on graphical metaphors. The implemented program in this Master Thesis has mainly used ST and SFC. Besides the IEC 61131 Control Languages, the Control Builder also supports so called Control modules. A Control module is a container for both application program and graphical objects. It is an extension of IEC 61131 Control Languages developed by ABB to provide extra functionality to give more support for object-oriented programs. An example of a single Control module, the Pulse controller that is mainly implemented by using Structured Text, is shown in figure 7. In the Control module, parameters and variables are defined separately. In the Structured Text field, some code is also shown. Another example shows how a Control module can be built by using other Control modules. The new Control module is in this case the Probing controller and it is shown in figure 8.
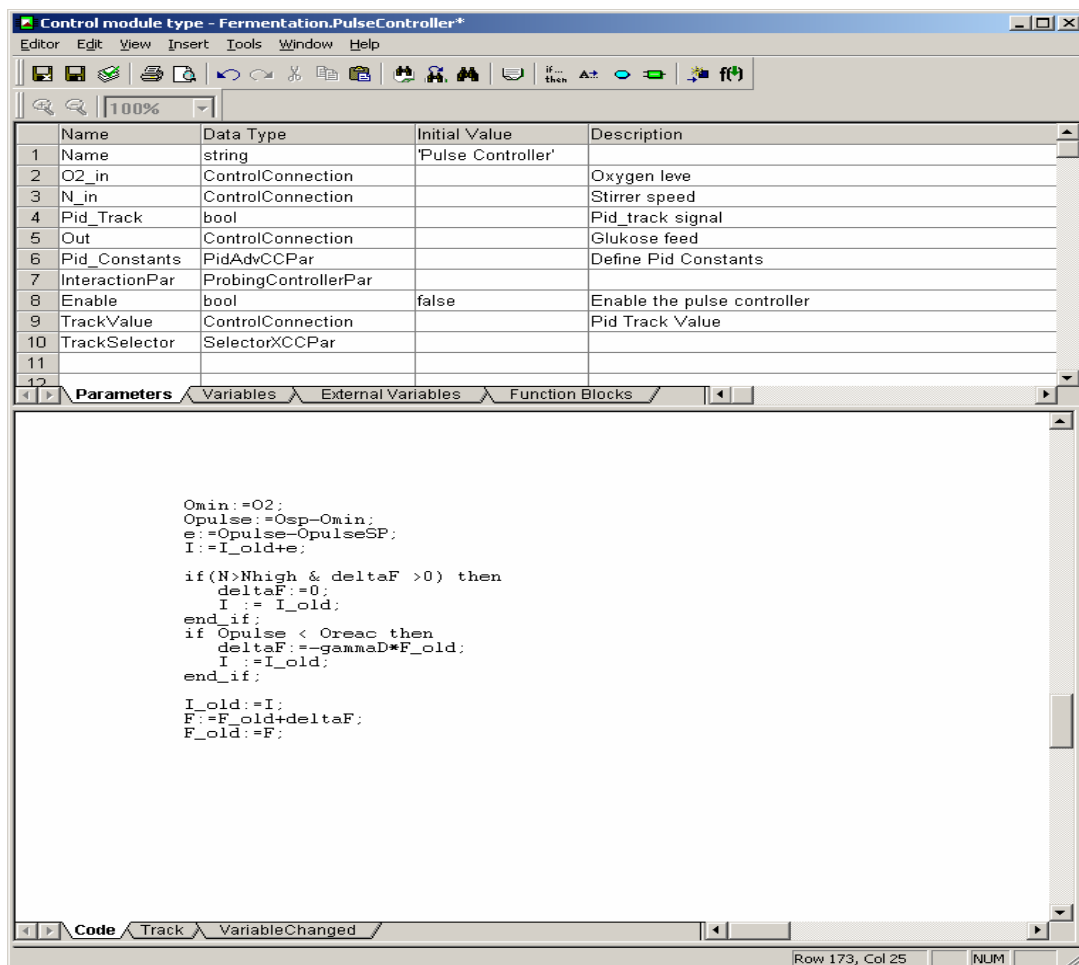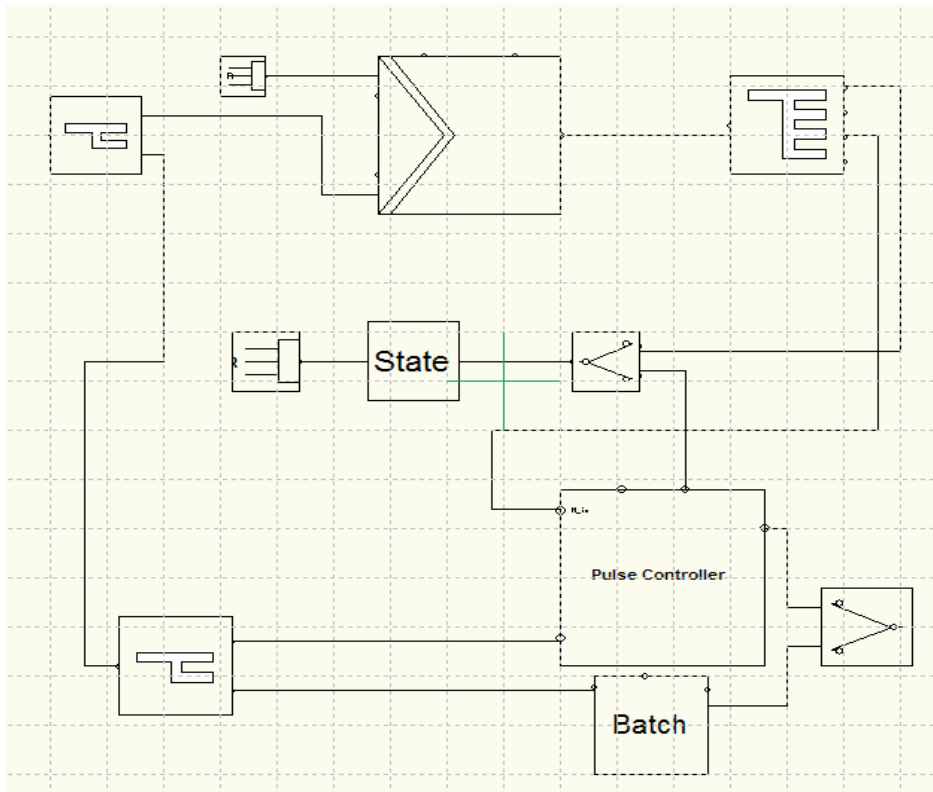

Figure 7 Pulse controller as a Control module and some of the implemented program using Structured Text

13

Figure 8 The implementation of the Probing controller using the concept of Control module.

In the Control Builder, there exist several libraries. In these libraries, there are many objects with standard functions to be reused for different purposes. The user can also create his/her own libraries. During this Master Thesis, two libraries have been created, the Fermentation library and the Model library. The Fermentation library includes the Pulse controller and the Dot controller. Since the mode of the Dot controller is controlled by the Pulse controller, the Pulse controller, the Dot controller and a BatchWatch module together form a new module, Probing controller. The Model library includes the process model and the Dot sensor. The process model and the Dot sensor together form a new module, MainModel. The Fermentation library is described in detail in Chapter 3.3.

After having built libraries and their contents, the next step is to connect several modules from different libraries together to form a cultivation process. This is achieved by using Application Window. The Application Window contains graphics of an application. An example of Application Window containing the cultivation process is shown in figure 9.
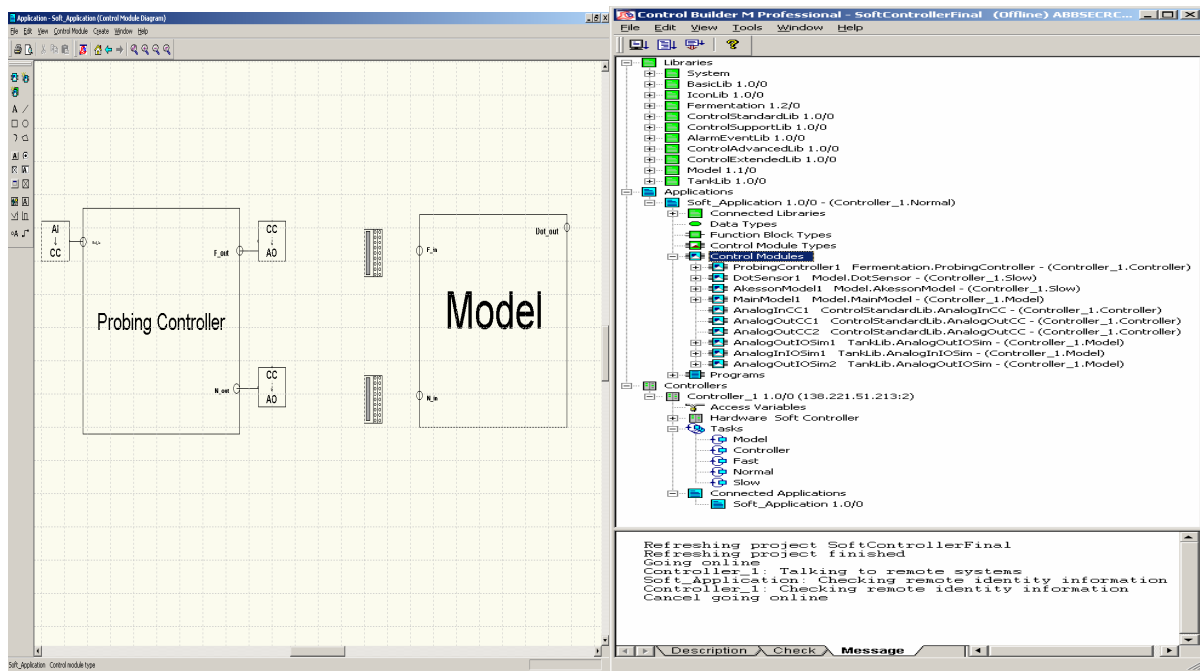
**Figure 9** Application Window containing the cultivation process, the Control Builder enviroment

In the Application Window there are several Control modules. Each Control Module is given a specific Task time i.e. the sampling interval for the object, for example the Dot controller has a sampling interval of 1 s while the model has a sampling interval of 0.5 s. The method to determine sampling interval is described in Chapter 3.2. Finally the application is downloaded into the Soft Controller for real time simulation.

4.2 Discretization

In order to implement the process model in Control Builder, it is necessary to use numerical methods to solve the differential functions in the process model. The choice is between the forward Euler method and the implicit Euler method. The easiest way is to use the forward Euler method, but if the sampling interval is not chosen carefully enough, instability can result. On the other hand, the implicit Euler method always gives a stable system, but an iterative method has to be used. Since the iterative methods involve both complicated mathematical calculations and potential convergence problems, the forward Euler method is chosen. Finding an appropriate time interval for the forward Euler method will probably take less time than to implement a complicated implicit method.

The starting point is a 20 hours simulation in Matlab with limited stirrer speed. The sample time is chosen to 2 seconds. The following figures are obtained:
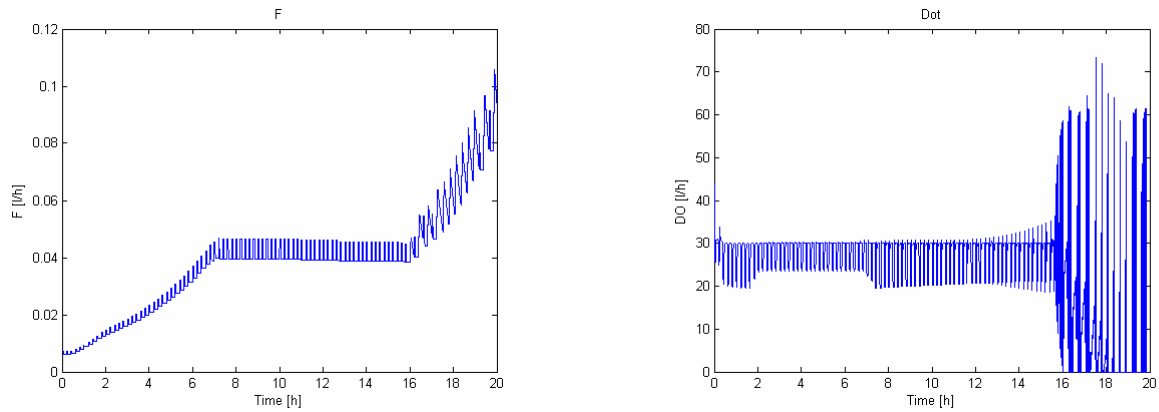
Figure 10   The glucose feed rate and the dissolved oxygen tension

From the above figures the conclusion is that sample time 2 seconds gives unstable system.

The next step is to lower the sample time to 1 second with limited stirrer speed. The following figures are obtained:
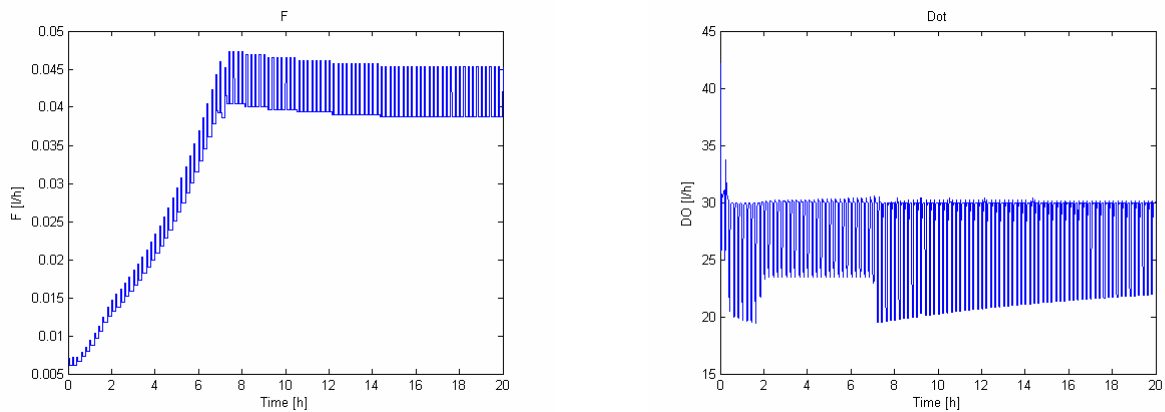
Figure 11   The glucose feed rate and the dissolved oxygen tension

The above figures show that the model is stable if the interval time is chosen to be 1 second. The stability limit is between 1 second and 2 seconds. In the later simulations a sample time of 0.5 s is chosen for the model. The Dot controller is also implemented in discrete time interval with a sample time of 1 s. The parameters for the simulations are shown in Appendix F.

4.3 Fermentation Library

The Fermentation library contains the following Control modules: the Probing controller, the Pulse controller, the BatchWatch and several modules to make it possible for the user to interact with the implemented program during online mode.

The Pulse controller is the module that contains the feedback algorithm that is described in Appendix A. Combining the Pulse controller, a pre-defined PID controller that will be used as the Dot controller and the BatchWatch module the Probing controller is formed.

The main difference between the Matlab implementation and the Control Builder implementation is the existence of the BatchWatch module in the Control Builder implementation to monitor the batch phase. This means that initially there is a small amount

of cells in the reactor and the start feed is manually injected into the reactor. Then the BatchWatch module watches the growth of the cells in the reactor by monitoring the dissolved oxygen tension. At the beginning, the cells consume glucose and replicate themselves. When consuming glucose, the cells also consume oxygen and the Dot controller compensates it by varying the stirrer speed to keep a constant oxygen concentration in the reactor. After a while the cells have consumed almost all glucose and the oxygen consumption reduces. When the oxygen consumption reduces, the stirrer speed also decreases. Since there exists a lower limit for the stirrer speed, the stirrer speed remains above this level although the cells consume less and less oxygen. After a while the dissolved oxygen tension starts to increase. When the Dot goes above 50%, the initial feed is pumped automatically by the BatchWatch module into the reactor. The BatchWatch module will continue to watch the dissolved oxygen tension until it stabilizes around 30% (the setpoint for Dot). Then the Pulse controller is activated.

During online mode, the user often wants to watch and control, for example, the feed, the stirrer speed and parameters in the feedback algorithm and so on. This is made possible by building several Interaction Windows. These windows will allow the user to both watch and change important parameters/constants in both the Dot controller and the Pulse controller.
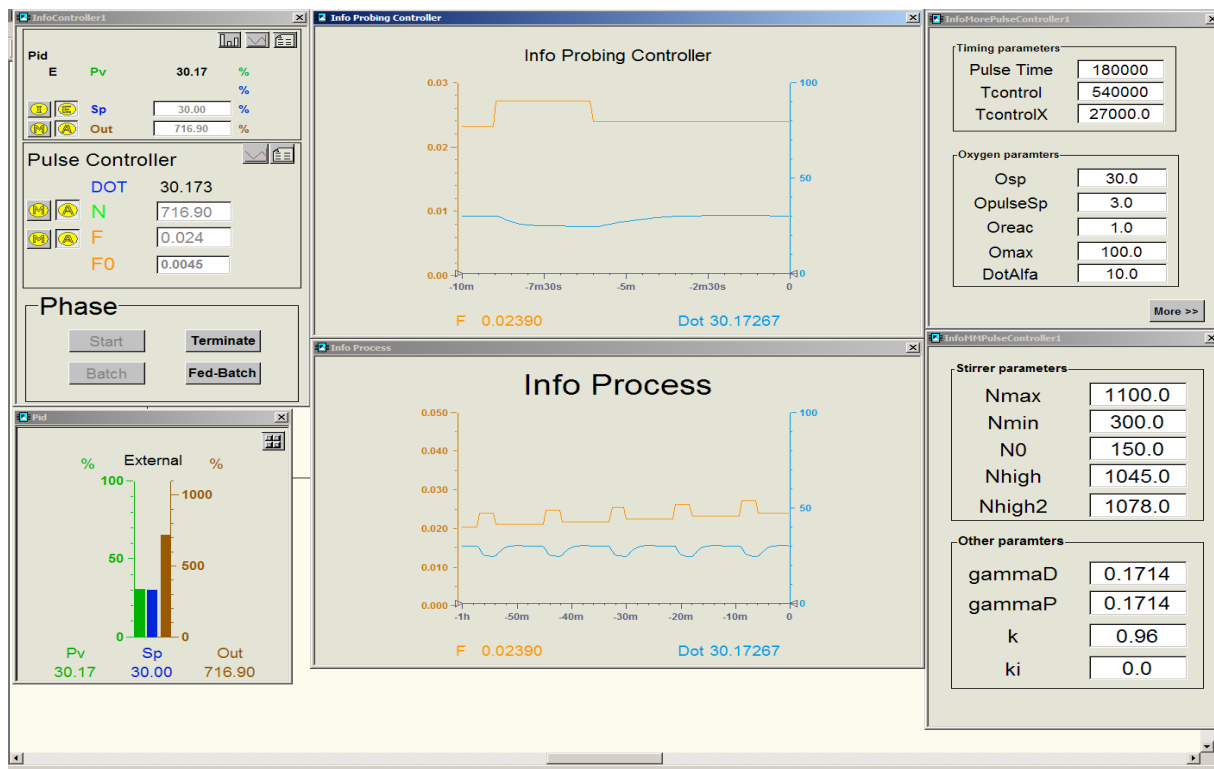
Figure 12   Several Interaction windows during online mode

To have better views of the simulation data, Plant Explorer Workspace in the Industrial IT 800xA System is used. The Plant Explorer Workspace is a developing environment to build operator panels. In this Master Thesis it is mainly used to log data, evaluate data during/after simulation and control different parameters/constants during online mode.
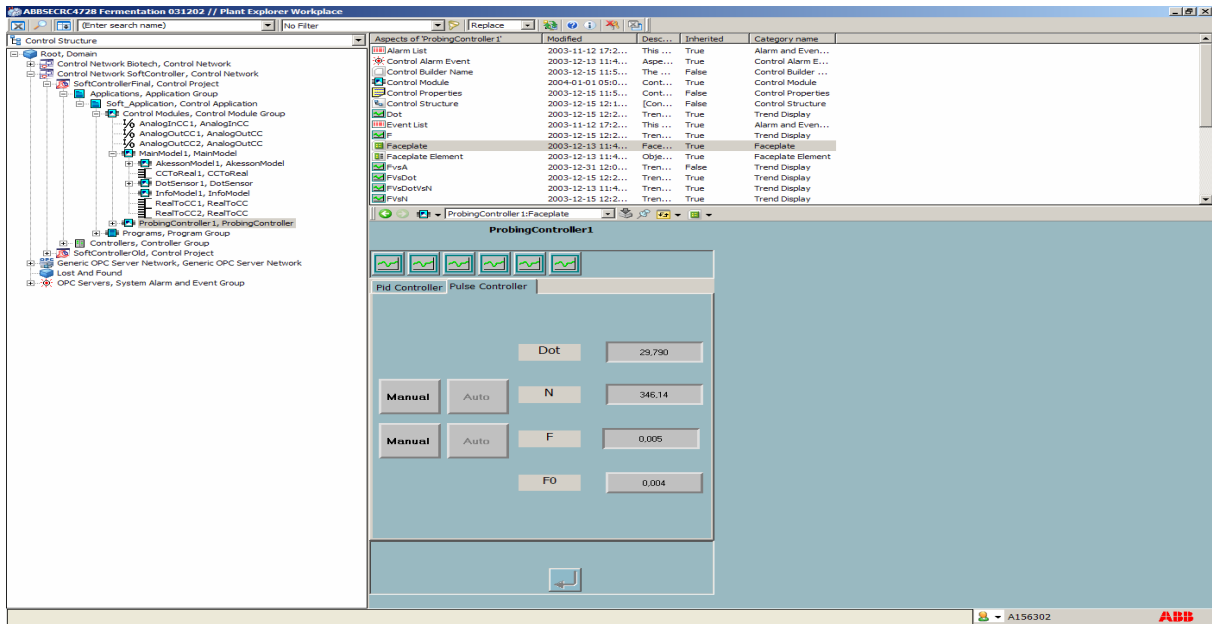
Figure 13 Plant Explorer Workspace with the faceplate for Probing Controller

4.4 Soft controller

Before the field test, the implemented controller programs are simulated with the implemented process model. This is achieved by connecting the Probing controller and the process model in the same application. Then the application is downloaded into a Soft controller. The Soft controller is a simulation tool for real time simulation. It means that the result from the simulation in the Soft controller is very close to the result from the field test with AC800M controller and its connected I/Os.
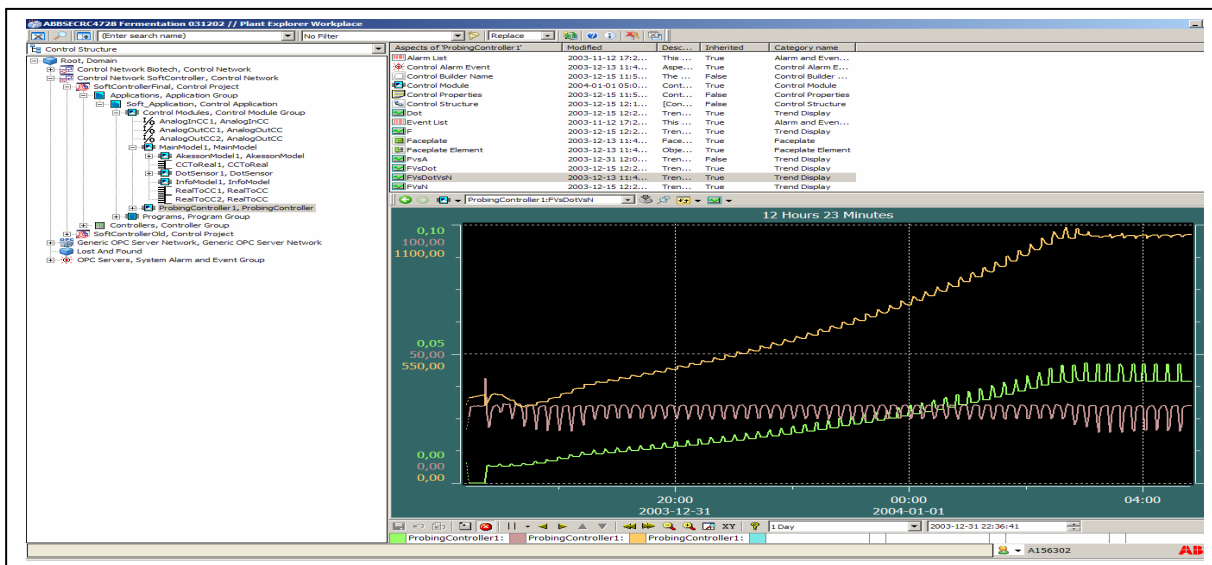
Figure 14 Plant Explorer Workspace with trend plot for stirrer speed (the orange curve), the feed rate (the green curve) and the dissolved oxygen tension (the purple curve). This is typically results after a simulation in Soft controller.

4.5 Time Scaling

When the model and the controller programs are downloaded into the Soft controller, the execution runs in real-time. In order to obtain a 20 hours simulation, it will take 20 hours. In Matlab, it only takes less than 15 minutes for a 20 hours simulation if the computer has appropriate hardware. For demonstration purpose, it is more preferable if a 20 hours simulation can be reduced to run in for example 2 hours or even faster. This requires that both the model and the controller programs to be time scaled.

Basic Theory

The theory behind the time scaling is to simply insert a scaling factor called α into the derivatives in the process model. Assume that the derivative of a variable x is $\frac{dx}{dt} = f(x,u)$.

Then introduce a new time variable s defined by s= α*t and hence $\frac{ds}{dt} = \alpha$. A variable substitution in the derivative of x then gives $\frac{dx}{dt} = \frac{dx}{ds} * \frac{ds}{dt} = \frac{dx}{ds} * \alpha = f(x,u)$ and forms $\frac{dx}{ds} = \frac{1}{\alpha} * f(x,u)$. Consequently inserting the factor $\frac{1}{\alpha}$ into all derivatives of the process model, the model is hereby time scaled. The next step is to choose the sample time for the discrete cultivation process. The easiest way is to scale the sample time with the factor $\frac{1}{\alpha}$.

The Pulse controller and the Dot controller also contain some time variables and they are also scaled with the factor $\frac{1}{\alpha}$.

Simulation verification
In Matlab, the α-factor is chosen to be 0.5 which means that the system runs 2 times faster than its normal performance. A comparison of a 20 hours ordinary simulation with a 10 hours scaled simulation gives results presented in Figs 15-17.

Figure 15   The glucose feed rate
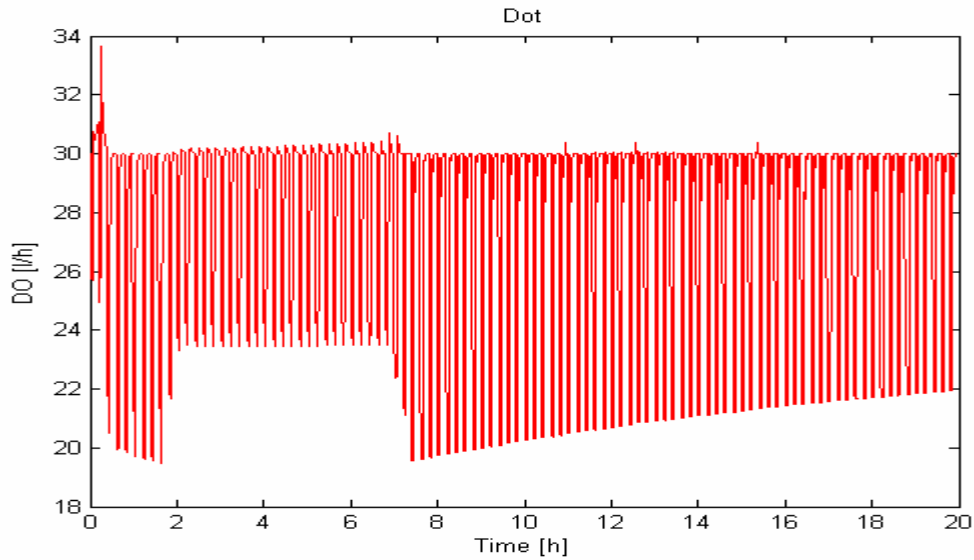
Figure 16   The dissolved oxygen tension
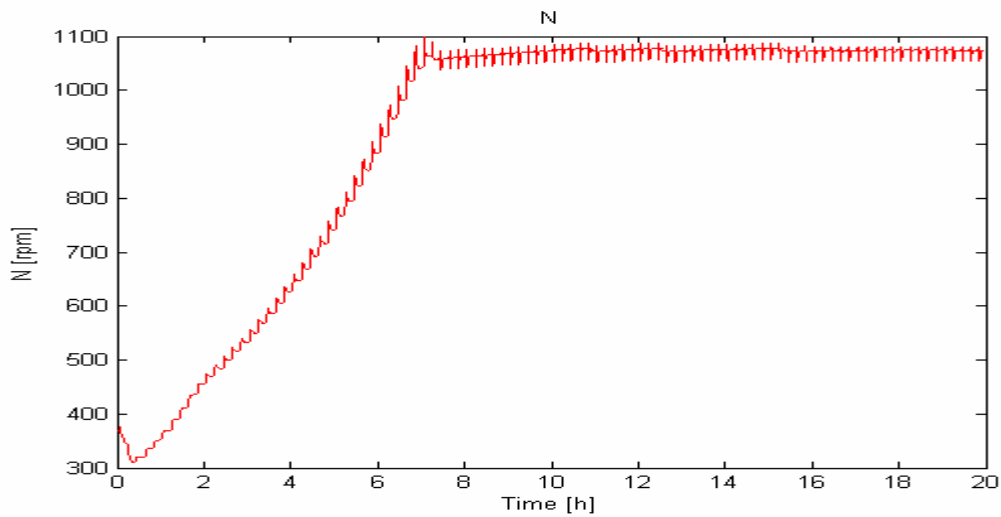
Figure 17   The stirrer speed

In the above figures, the 10 hours scaled simulation has been expanded to be able to compare with the 20 hours ordinary simulation. The result is as expected, i.e. the two simulations are identical. The parameters for the simulations are shown in Appendix E.

The time scaling is also implemented in Control Builder. A 20 hours simulation has been compared with a 10 hours simulation that has the scaling factor of 0.5 i.e. the Soft Controller simulation runs twice as fast as the ordinary simulation. The following figures show the result from the scaled Soft controller simulation:
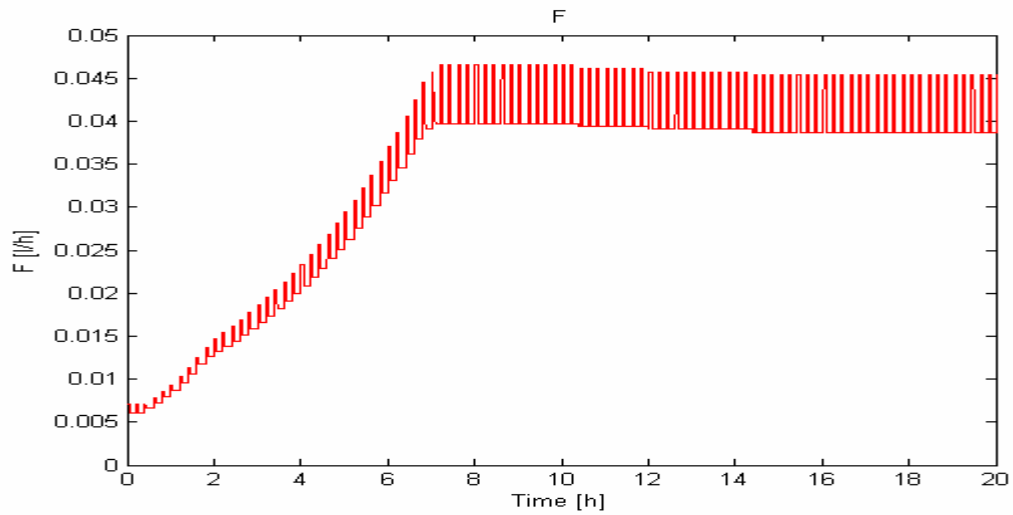
Figure 18　The glucose feed rate

Figure 19　The Dissolved oxygen tension
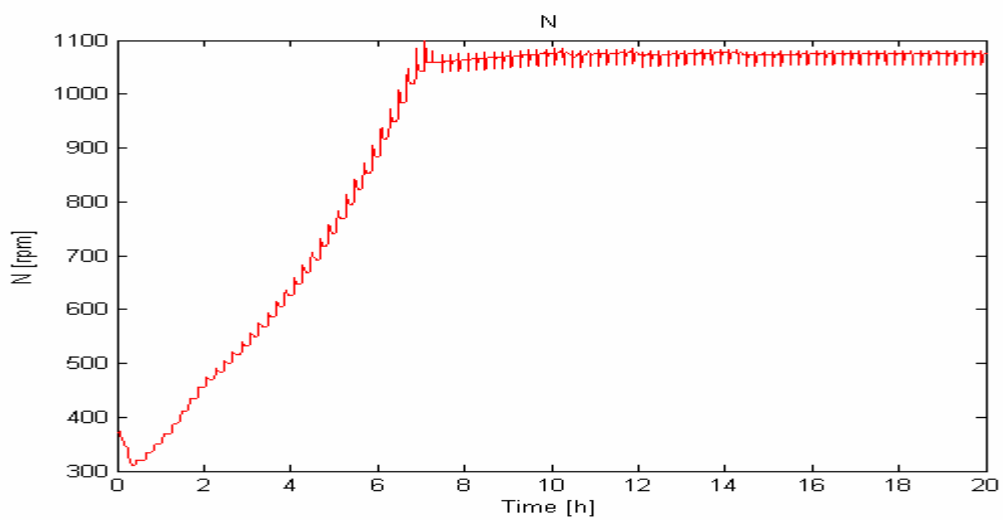
Figure 20　The stirrer speed

21

The simulation results have been expanded from 10 hours to 20 hours in order to compare with 20 hours ordinary simulation. Overall the two simulations are the same, but there exists some minor differences. The following figures show comparison between the two simulations when the maximum stirrer speed has been reached.
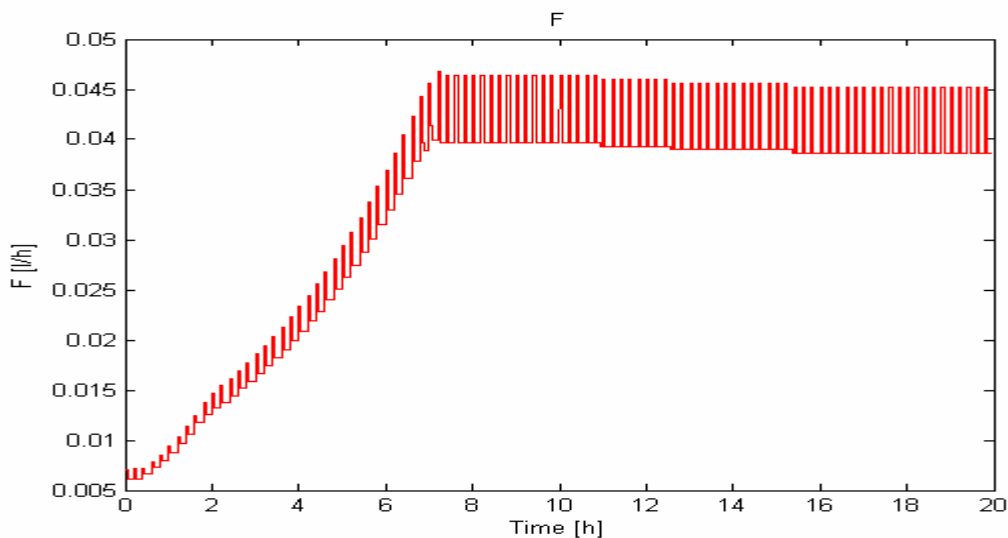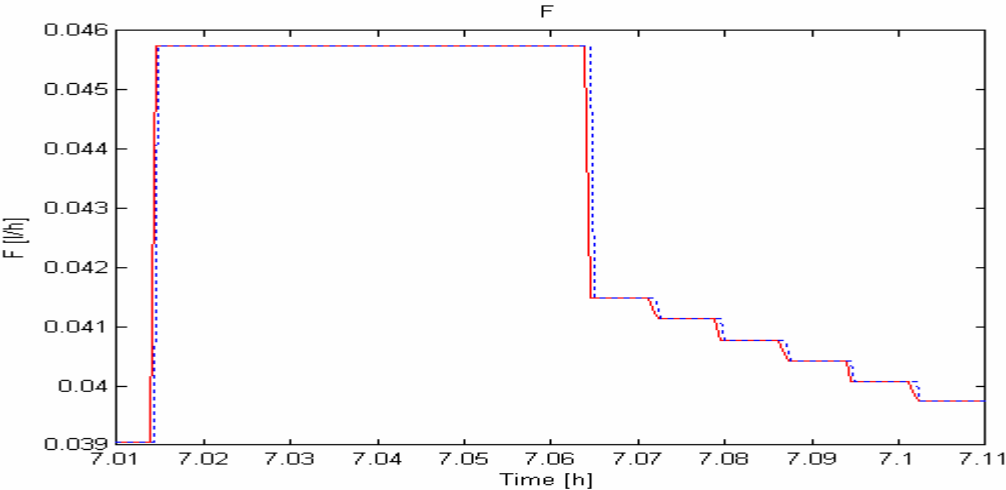

**Figure 21**   The glucose feed rate
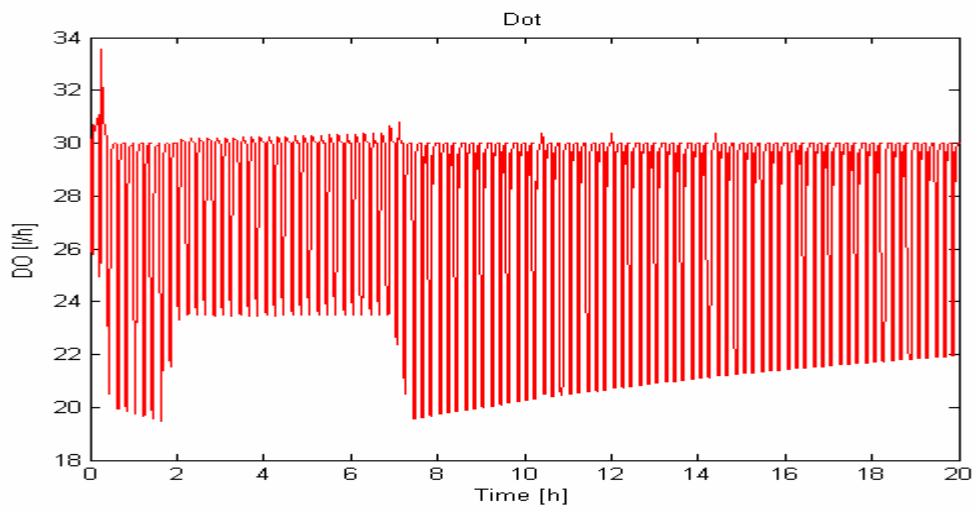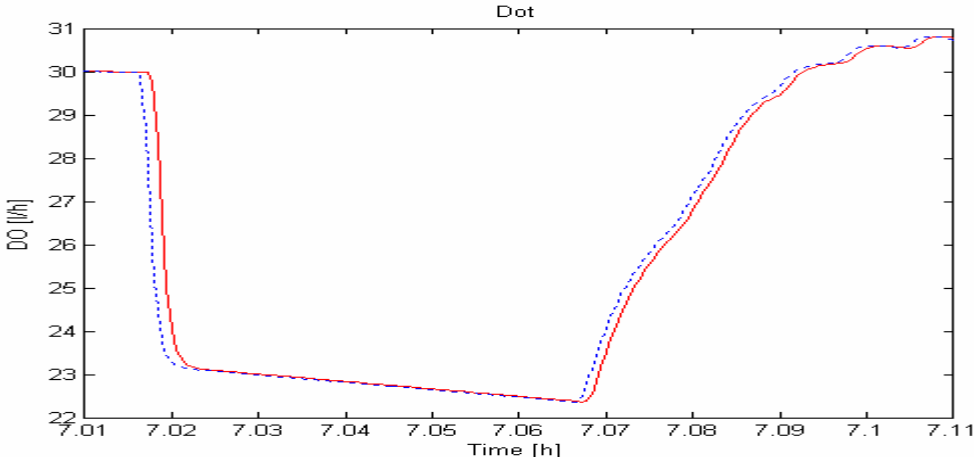

Figure 22   The dissolved oxygen tension
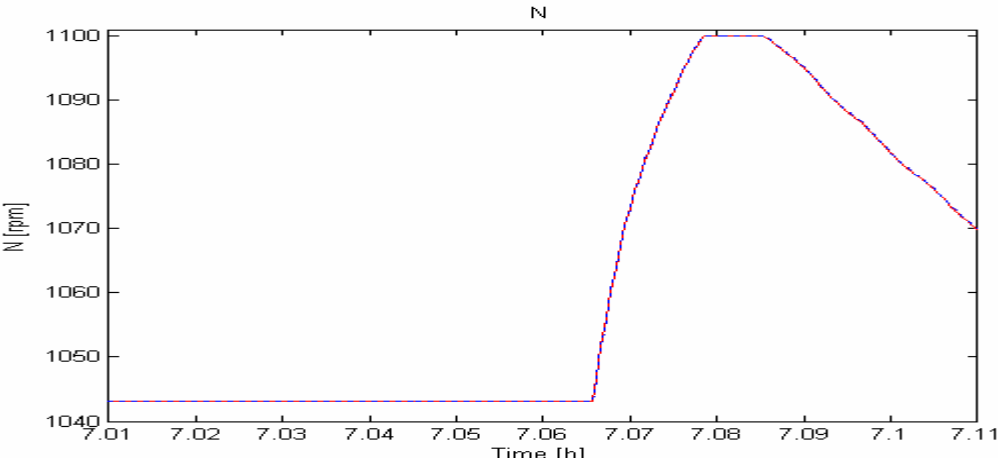
Figure 23   The stirrer speed

In the above figures, the Dotted lines are the unscaled simulation while the solid lines are the scaled simulation. The two simulations are almost the same. However, there exists a 1.8 seconds time difference between them when maximum stirrer speed has been reached. The reason for this difference is due to different real time delays. Since the implemented program is simulated in real time, there exists different delay time for different simulations. Despite the delays, the two simulations have almost identical behavior during the entire simulations.

The simulation result has verified that the scaling implementation is correct. In this case, the system is only scaled with the factor 0.5. Actually it can be scaled with a factor up to 0.1 (10 times faster than the ordinary simulation) or even more. A 20 hours simulation can then run in 2 hours during a demonstration. The parameters for the Soft controller simulations are shown in Appendix E.

This Master Thesis has had an approach to first implement the desired program in Matlab. After verifying the Matlab implementation, the next step was to implement the program in Control Builder. If the Control Builder simulations can run fast enough, this approach can be replaced by directly implementing the desired program in Control Builder. A 20 hours simulation in Matlab takes around 10-15 minutes. For a 20 hours simulation in Control Builder to run in 12 minutes, it requires a scaling factor of 0.01 i.e. 100 times faster. This means that the model needs to have a sample time of 5 milliseconds while the Probing controller needs to have a sampling interval of 10 milliseconds. The faster the simulations are executed, the delays will have bigger impact on the simulations. Therefore it is not recommended to run the program 100 times faster. The safest way to implement a program that has long simulation time is to first verify in Matlab, and then implement the program in Control Builder.

5   Comparison

During this stage the simulation in Simulink (the forward Euler model) is compared with the simulation in Soft controller. In this way, it is easier to verify the Control Builder implementation.

The main difference between the two implementations is the existence of both a batch phase and a fed-batch phase in the Control Builder implementation while the Simulink implementation it only contains a fed-batch phase. During the batch phase in the cultivation process, there are significant amount of new cells produced. This will effect the calculations of V (Liquid volume), X (biomass concentration), G (glucose concentration), A (acetate concentration) and O (oxygen concentration). Due to this reason, the comparison will be useless if the Simulink implementation and the Control Builder simulation start at different states. The easiest way to solve this problem is to use the values of variables: V, X, G, A, X, N (the stirrer speed) and F (the feed rate) after the batch phase in Soft controller simulation as the starting values in the Simulink simulation. Then only the fed-batch phase is compared. The batch phase is easily implemented in Control Builder and therefore is easier to verify.

The following results are achieved
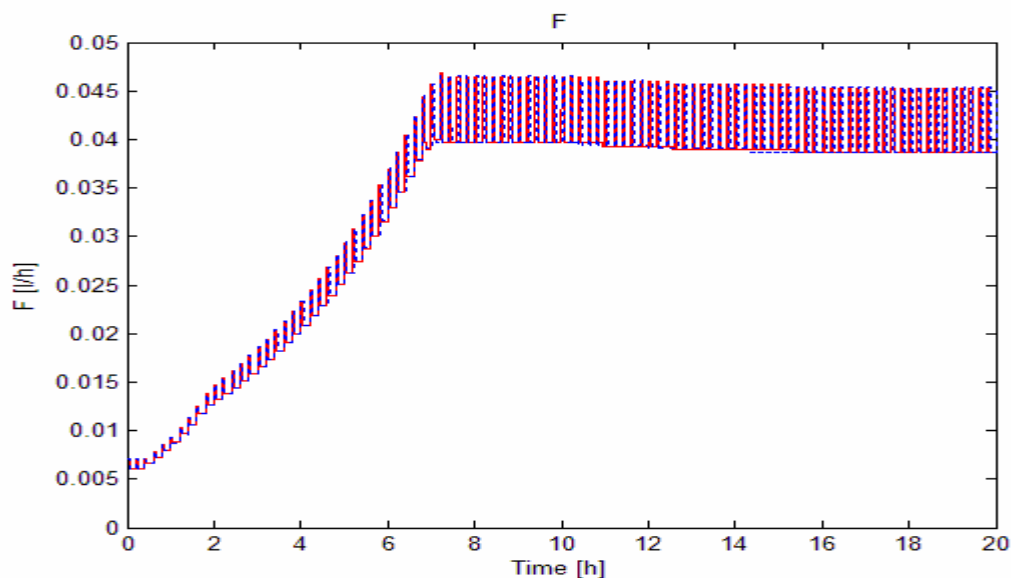
Figure 24   The feed rate, the solid line is the Simulink simulation and the Dotted is the Soft controller simulation
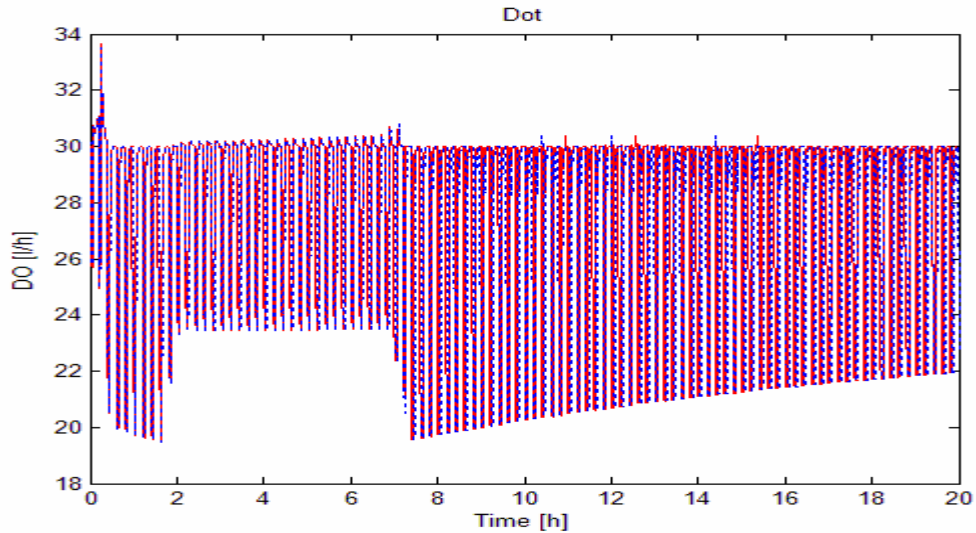
Figure 25   The dissolved oxygen tension, the solid line is the Simulink simulation and the Dotted is the Soft controller simulation
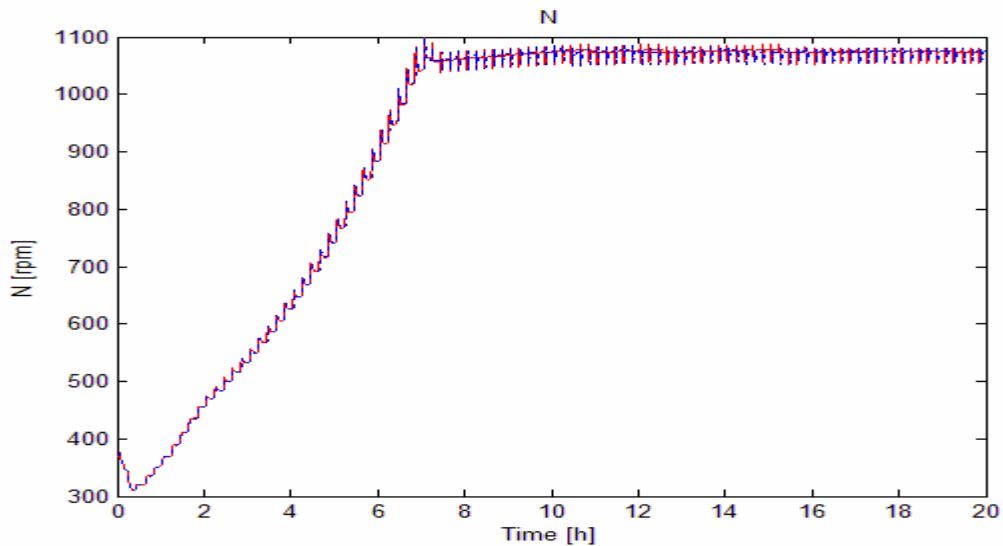
Figure 26   The stirrer speed, the solid line is the Simulink simulation and the Dotted is the Soft controller simulation

The simulation in Control Builder is almost the same as the simulation in Simulink. The main difference is a delay of 54 seconds between the simulations when the stirrer speed has reached its maximum limit. This is mainly caused by the fact that the Matlab simulation is executed on the Windows platform while the Soft controller simulation is executed outside the Windows platform. The Soft controller shares the CPU time with the Windows Platform and this can introduce real time delays. In a 20 hours simulation, this delay is therefore noticeable. This delay is also noticed while comparing simulation results during time scaling, see Chapter 4.5. Otherwise, the good simulation results indicate that the implementation in Control Builder is correct.

The parameters for the Probing controller, the Dot controller and the model during the Comparison simulations are shown in Appendix G.

After verification of the Control Builder implementation, the field test in Lund is the next step.

25

6 Final Testing

6.1 Introduction

The final testing of the feedback algorithm implemented by using IEC 61131 Control Languages was performed at the division of Biotechnology at the Chemical Center of Lund Institute of Technology. A Belach fermentor with SattCon system is located there. For the purpose of this test, the dissolved oxygen tension (Dot) was monitored by a AC800M controller. The feedback algorithm and the Dot controller that has been downloaded into the AC800M controller calculate appropriate glucose feed and stirrer speed. The new values are then sent to the fermentor. By using the Plant Explorer, signals such as feed rate, dissolved oxygen tension and stirrer speed were stored. These three signals were also sent to the SattCon system for logging.

The feedback algorithm was tested using a 3 *l* fermentor with an initial medium volume of 2 *l*. The production of the recombinant product was induced by injecting a chemical agent into the reactor. The initial glucose concentration was 10 *g/l*. The initial feed of glucose from batch phase to fed-batch phase was 0.2 *g/l*. During the fed-batch phase the glucose was added according to the feedback algorithm. The pH was kept at 7.0 by controlled occasional addition of ammonia. The temperature was regulated at 37º *C* during the cultivation. The dissolved oxygen tension was kept around 30*%* and controlled by varying the stirrer speed in the range 350-1100 *rpm*. The air flow during the cultivation process was 2 *l/min*. For more detailed description of the test material, see (Åkesson, 1999).

6.2 Testing

The Probing controller containing both the Pulse controller and the Dot controller received the Dot value and sent out feed rate and stirrer speed. The feedback algorithm used a P-controller to determine a new feed rate after each feed pulse. There existed no limit for the feed rate.

The criteria for the transition from the batch phase to the fed-batch phase were that 4 *h* had elapsed and that the dissolved oxygen tension exceeded 50%. At the beginning of the fed-batch phase, the initial feed was pumped into the reactor. The criteria for the first pulse were that 10 *min* had passed after the initial feed and that the dissolved oxygen tension had reached around 30% (the setpoint for the Dot).

6.3 Results and discussion

Results

The test lasted for almost 16 hours. The batch phase ended after more than 7 hours and then the initial feed was injected automatically into the reactor. It took almost one and half hours to reach the feed limit due to the limit of the stirrer speed was reached. The induction was made after the feed limit was reached. During the first three hours after the feed-limit was reached (t = 9 *h* to 12 *h*), the feed remained almost constant. Then the feed decreased slowly during the remaining time.
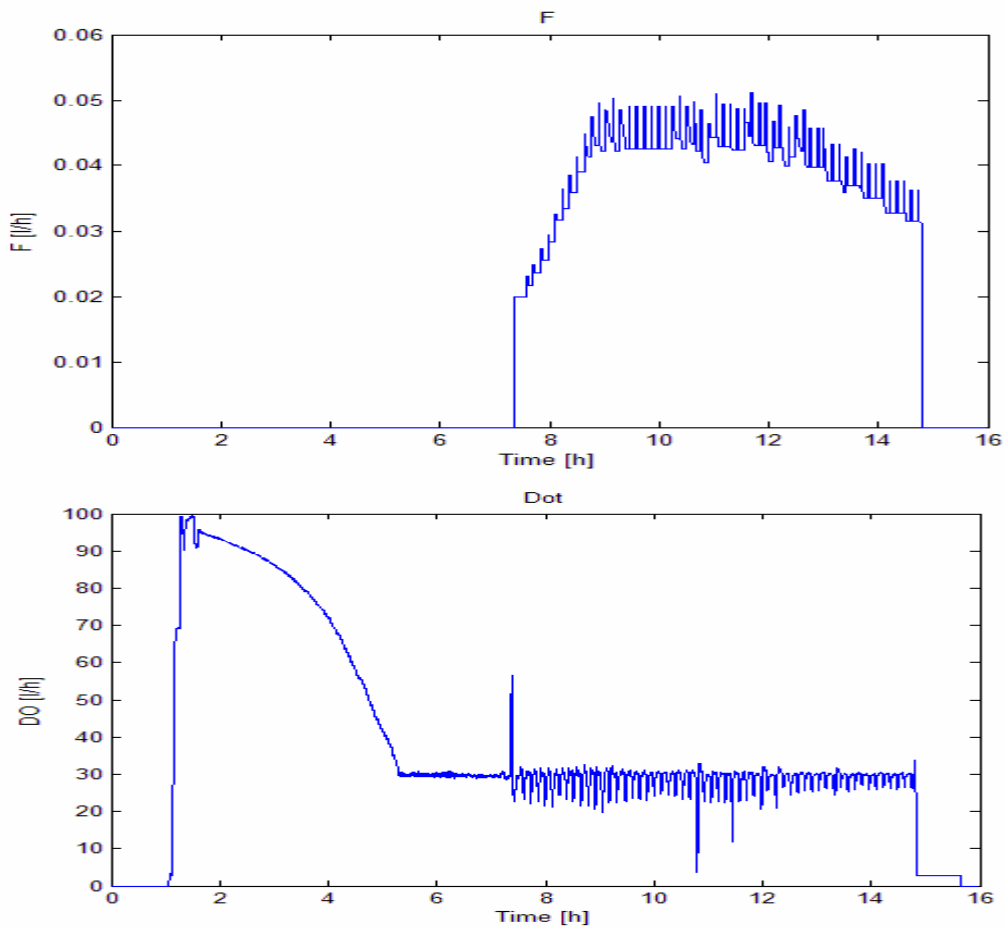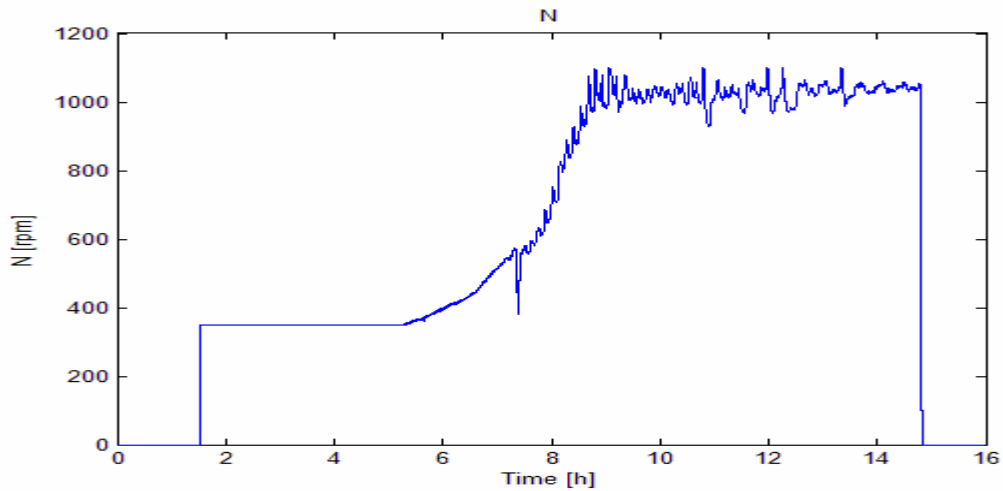
Figure 27   Cultivation 2003-12-10

Discussion

Satisfactory results have been obtained at the first attempt. The results are very similar to the results normally obtained at the Biotechnology laboratory, see (Åkesson, 1999).

The occasional decreases of the feed rate between t = 9 *h* to 12 *h* were caused by injections of anti-foam medium. After the feed limit was reached, there often built foam at the top of the cells in the fermentor. To reduce this foam, the anti-foam medium was injected into the fermentor manually. The foam was built because the cells in the fermentor felt "stressed" and they broke down. When the anti-foam medium was injected, it caused smaller response from feed pulse than usual and this resulted in lower feed rate.

There was a severe decrease of dissolved oxygen tension at t =10.5 *h*. The dissolved oxygen tension was below 10%. This was caused by changing the filter for the outcoming air. Normally there was a certain amount of air going to the reactor. The outcoming air passed through a filter. When changing this filter the air pump was stopped and there was no fresh air coming into the reactor, which resulted in the sudden drop of the dissolved oxygen tension.

This test has also proved the use of the λ-tuning method to determine gain and integral time to control the stirrer speed. But further tests have to be performed before coming to the conclusion that the λ-tuning method is suited for all sizes of fermentors and every kind of Escherichia coli strain.

28

7 Conclusions

The feedback algorithm has been implemented first in Matlab and later in Control Builder. Because of the careful verifications of both the Matlab implementation and the Control Builder implementation, a successful field test has been achieved in Lund.

Using the forward Euler method in discretization stage proved to be enough. Appropriate sampling interval has also been determined. The forward Euler method simplified the implementation compared with other implicit methods, such as backward Euler.

The use of the λ-tuning method is another successful method combined with the feedback algorithm. The use of the λ-tuning method is an adaptive approach of controlling Dot and with further improvements it will be independent of the biological reactors.

An ordinary simulation often takes around 10 to 20 hours which is a rather long time to wait. For demonstration purpose, the scaling of both the process model and its controllers are made. This shortens the simulation time up to 10 times or even more.

The success of the final test in Lund means also that ABB's controller and their developing environment, Industrial IT is suitable to use in controlling fermentation processes.

The developing environment Control Builder has proved to be both easy to learn and comprehensive as a platform to develop programs for programmable controllers using IEC 61131 Control Languages. The Plant Explorer that has much more functionalities than it is used in this Master Thesis had been more time consuming to learn. During this Master Thesis, different versions from version 2.1 and version 3.1 (the latest version that is release candidate) have been used. From using the eldest version (2.1) and update through several intermediate versions to finally use the latest version (3.1), significant improvements of usability has been experienced.

8   Discussion

An automated feeding strategy for E.coli cultivations that is designed to avoid acetate accumulation while maintaining a high glucose feed has been implemented and tested using ABB's Industrial IT. By using a standard dissolved oxygen sensor combined with the feedback algorithm, the overflow metabolism can be detected and then avoided. This Master thesis has showed a way of using Industrial IT to implement programs that are then downloaded into a physical controller in order to control cultivation process for pharmaceutical purpose.

The controller programs have been tested in a small-scale biological reactor under normal conditions. This means that the robustness of the controller programs have not yet been tested. The feedback algorithm is strictly dependent on the Dot response after a feed pulse. This response is measured by a Dot sensor. The sensor can in some cases generate noise. The feedback algorithm wants to register the lowest Dot value after a feed pulse. It is therefore important to make sure that the registered value is not noise generated from Dot sensor. This can be accomplished by using a low-pass filter with proper filter time for the Dot values before sending them to the Pulse controller.

In the Lund test, it has been observed that the limitation of the feed rate is dependent on the stirrer speed. If the maximum limit for the stirrer speed is reached, the feed rate can not be increased further since the maximum oxygen transfer capacity is also reached. After a while with maximum stirrer speed and an almost constant feed rate, the feed rate slowly started to decrease. One way to maintain the constant feed rate is to lower the temperature before the feed rate starts to decrease. In this way, the cell activity is reduced and thereby the constant feed rate can be remained. This approach is patented at Silversparre, G.,S.-O. Enfors, L.Han, L. Häggström and H.Skogman (2002) :" Method for growth of bacteria, minimizing the release of enDotoxins from the bacteria into the surrounding media." Patent: International publication number WO 02/36746.

9   Acknowledgements

I wish to express my sincere gratitude to all who have assisted me in making my Master Thesis possible.

First of all, my deep gratitude goes to Professor Alf Isaksson, my supervisor at ABB and Professor Per Hagander, my supervisor at the Lund Institute of Technology. Without their guidance and timely instructions, none of my progress would be achievable. Their scientific ideas and valuable comments were abundant sources for improvement of my work and their sustained encouragements, patience and support helped me to overcome one obstacle after another.

Mikael Petersson, ABB Malmö, deserves my sincere gratitude for his persistent and cautious guidance, help and endless patience, especially in advising me during implementation in Control Builder and helping me during the final test in Lund.

I would like to give my thanks to Drs Lars Pernebo (ABB, Malmö) and Stefan Israelsson (ABB, Västerås) for their generous and valuable help.

My gratitude goes also to Stéphane Velut and Lena de Maré, PhD students at the Lund Institute of Technology, for their patient help and to everyone in the AT department at ABB Corporate Research in Västerås, for their continuous help whenever I knock on their doors.

Above all, I hold all my heartfelt appreciations to my mother, Cairu, and my father, Wenxi, for their profound love and understanding, consistent encouragements and support along with me during the past half year.

10 References

Åkesson, M (1999), "Probing Control of Glucose Feeding in Escherichia coli Cultivations". PhD Thesis, Wallin & Dalholm Boktryckeri AB, Lund Sweden.

P.F.Stanbury, A. Whitaker and S.J.Hall (2000), "Principles of Fermentation Technology", MPG Books Ltd, Great Britain.

Velut, S, L. De Maré, J. P. Axelsson and P. Hagander (2002), "Evaluation of a probing feeding strategy in large scale cultivations", Department of Automatic Control, Lund Sweden.

Enfors S.O and L.Häggström (2000), "Bioprocess Technology Fundamentals and Applications", Högskoletryckeriet, Royal Institute of Technology, Stockholm Sweden

ABB (2001), IEC 61131 Control Languages, Sweden, Document number: 3BSE 021 358 R201 Rev B.

ABB (2003), Addendum, Product Data and Performance, Control Software for AC 800M Version 3.4, Sweden, Document number: 3BSE 032 560 R201.

ABB (2003), Basic Control Software, Introduction and Configuration, Control Software for AC 800M Version 3.4, Sweden, Document number: 3BSE 030 816 R201.

ABB (2003), Analog Process Control, Objects and design, Control Software for AC 800M Version 3.4, Sweden, Document number: 3BSE 030 820 R101 Rev A.

ABB (2003), Communication, Protocols and Design, Control Software for AC 800M Version 3.4, Sweden, Document number: 3BSE 030 821 R101.

ABB (2003), OPC Server for AC 800M, Installation and Configuration, Control Software for AC 800M Version 3.4, Sweden, Document number: 3BSE 030 826 R301.

ABB (2003), AC 800M Controller Hardware, Hardware and Operation, AC 800M Version 3.0, Sweden, Document number: 3BSE030 827 R201.

ABB (2003), Control IT S800 I/O, General Information and Installation, User's Guide, Sweden, Document number: 3BSE 020 923R501.

ABB (2003), Graphics, Sweden, Document number: 3BSE 030 335R101.

ABB (2003), Operator Workplace Configuration, Sweden, Document number: 3BSE 030 322R101.

ABB (2003), Operator Workplace Operation, Sweden, Document number: 3BSE 030 359R101.

ABB (2003), Workplace & Plant Explorer Reference, Sweden, Document number: 3BSE 032 788R101.

ABB (2003), Data Access Reference, Sweden, Document number: 3BSE 031 934R101.

ABB (2003), System Configuration, Sweden, Document number: 3BDS 011 222R101_d5_work.

ABB (2003), Introduction and Installation, Sweden, Document number: 3BUF 001 nnn R1 Beta1.

ABB (2003), Operation, Sweden, Document number: 3BSE 034 464R101.

ABB (2003), Trends & Historian Reference, Sweden, Document number: 3BSE 032 786R101.

Appendices

A. Feedback algorithm

1. Freeze the stirrer speed for Dot control

2. Make an up pulse in the feed: Fpulse(t) = gammaP * F(t-1)

3. Measure the pulse response O2

4. O2pulse = O2sp – O2

5. Update the control error e(t) = O2pulse – O2pulseSP

6. Compute delta F(t):

    ```
    IF O2pulse < O2reac
       THEN delta F(t) = -gammaD*F(t-1)
    ELSE
       IF N(t) > Nhigh
          THEN delta F(t) =0
       ELSE
          I(t) = I(t-1) +e(t)
          IF (k*e(t)+ki*I(t)) > O2pulse
             THEN I(t) = I(t) -  (k*e(t) + ki*I(t) – O2pulse)/ki
          END
          delta F(t) = (k*e(t)+ki*I(t))*F(t-1)/(O2max - O2sp)
          IF delta F(t) > Fpulse (t)
             THEN delta F(t) = Fpulse (t)
          END
       END
    END
    ```

7. Update F(t) = F(t-1) + delta F(t)

8. Set the set-point for the feed control to F(t)

9. Activate the Dot control

    a) Wait a time TcontrolX
        ```
        IF ActualTime for Dot-control >= Tcontrol
           Then Goto 1
        END
        ```
    b) Decrease F(t) when the stirrer speed is too high:
        ```
        IF N(t) > Nhigh2
           THEN F(t) = F(t-1) – TcontrolX * Fadj/Tcontrol
        END
        ```
    c) Goto a)

B. Model equations

$$\frac{dV}{dt} = F$$

$$\frac{d(VX)}{dt} = \mu * V * X$$

$$\frac{d(VA)}{dt} = (q_a^P - q_a^C) * V * X$$

$$\frac{d(VO)}{dt} = K_{La}(N) * V * (O^* - O) - q_0 * H * V * X$$

$$K_{La}(N) = \alpha * (N - N_0), N > N_0$$

$$q_g(G) = q_g^{maz} \frac{G}{k_s + G}$$

$$q_m = \min(q_g, q_{mc})$$

if $q_g < q_g^{crit}$ **then**

$$q_a^P = 0$$

$$q_a^C = \min(\frac{q_a^{C,\max} A}{k_a + A}, \frac{q_a^{\max} - (q_g - q_m)Y_{og} - q_m Y_{om}}{Y_{oa}})$$

$$q_0 = (q_g - q_m)Y_{og} + q_m Y_{om} + q_m Y_{om} + q_a^c Y_{oa}$$

$$\mu = (q_g - q_m)Y_{xg}^{ox} + q_a^c Y_{xa}$$

if $q_g \geq q_g^{crit}$ **then**

$$q_a^P = (q_g - q_g^{crit})Y_{ag}$$

$$q_a^c = 0$$

$$q_o = q_o^{\max} = (q_g^{crit} - q_m)Y_{og} + q_m Y_{om}$$

$$\mu = (q_g^{crit} - q_m)Y_{xg}^{ox} + (q_q - q_q^{crit})Y_{xg}^{fe}$$

For details about the above equations refer to (Åkesson, 1999).

C. List of parameters

The parameters/constants for the Probing controller:

Symbol	Value/Unit	Description
Osp	30 %	Setpoint for the dissolved oxygen tension
Nmax	Rpm	Maximum value for the stirrer speed
Nmin	Rpm	Minimum value for the stirrer speed
Nhigh	Rpm	The maximum stirrer speed used to determine if the feed rate after a feed pulse should change or not, see Appendix A
Nhigh2	Rpm	The maximum stirrer speed used to determine if the feed rate should decrease or not during the time between two feed pulses, see Appendix A
OpulseSP	%	Setpoint for the dissolved oxygen tension response after each feed pulse
k		Gain used in the Pulse controller
ki	0	Integral gain in Pulse controller (not used)
Oreac	%	Dot constant used after feed pulse to determine if there is acetate generation
gammaP		Percentage of feed that should increase after a feed pulse
gammaD		Percentage of feed that should decrease after a feed pulse
Tpulse	Min	Time for the feed pulse
Tcontrol	Min	Time between two pulses
TcontrolX	Sec	Divides Tcontrol into smaller intervals to supervise that the stirrer speed is not above its upper limit (here the Tcontrol is divided into 20 smaller intervals)
alfaScale		Scaling factor
DotAlfa		Another feed pulse will not be given before the dissolved oxygen tension has reached this threshold i.e. Dot+DotAlfa and Dot-DotAlfa
F0	l/h	Initial feed rate at beginning of the fed-batch phase

The parameters/constants/initial values/state for the Dot controller

Symbol	Value/Unit	Description
T	Sec	Sampling interval for the Dot-controller
Ti	Sec	Integral time
N0_CB	Rpm	Initial stirrer speed in Control Builder simulation
N0_Sim	Rpm	Initial stirrer speed in Simulink simulation

The constants/initial values for the model

Symbol	Value/Unit	Description
Gin	600 g/l	Glucose concentration in feed
H	14000 (l%)/g	Henry's constant

Ks	0.01 g/l	Saturation constant for glucose uptake
Ka	0.05 g/l	Saturation constant for acetate uptake
N0	150 rpm	Minimum value for the stirrer speed
O*	100 %	Equilibrium dissolved oxygen tension
$Q_a^{c,max}$	0.2 g/(gh)	Maximum specific acetate consumption rate
Q_g^{max}	1.0 g/(gh)	Maximum specific glucose uptake rate
Q_g^{crit}	0.53 g/(gh)	Critical specific glucose uptake rate
Q_{mc}	0.06 g/(gh)	Maintenance coefficient
Q_o^{max}	0.30 g/(gh)	Maximum specific oxygen uptake rate
Tp	20 s	Time constant for dissolved oxygen sensor
Y_{ag}	0.55 g/g	Acetate / glucose yield
Y_{oa}	0.55 g/g	Oxygen / acetate yield
Y_{og}	0.50 g/g	Oxygen / glucose yield for growth
Y_{om}	1.07 g/g	Oxygen / glucose yield for maintenance
Y_{xa}	0.40 g/g	Biomass / acetate yield
Y_{xg}^{ox}	0.51 g/g	Oxidative biomass / glucose yield
Y_{xg}^{fe}	0.15 g/g	Fermentative biomass / glucose yield
α	1.4 l/(h*rpm)	Constant
Dot0_CB	100 %	Initial dissolved oxygen tension in Control Builder simulation
V0_CB	2 l	Initial volume in Control Builder simulation
A0_CB	0 g/l	Initial concentration of acetate in Control Builder simulation
G0_CB	g/l	Initial concentration of glucose in Control Builder simulation
X0_CB	g/l	Initial biomass concentration in Control Builder simulation
Dot0_CB	100 %	Initial dissolved oxygen tension in Control Builder simulation
V0_Sim	l	Initial volume in Simulink simulation
X0_Sim	g/l	Initial glucose concentration in Simulink simulation
A0_Sim	g/l	Initial concentration of acetate in Simulink simulation
G0_Sim	g/l	Initial concentration of glucose in Simulink simulation
Dot0_Sim	%	Initial dissolved oxygen tension in Simulink simulation

D. Simulation parameters for test of λ-tuning

The parameters for the Probing controller:

Parameter	Value	Comment
Nmax	1100 rpm	
Nmin	300 rpm	
Nhigh	1045 rpm	95% of Nmax
Nhigh2	1078 rpm	98% of Nmax
OpulseSP	3 %	
k	0.96	
Oreac	1 %	
gammaP	0.1714	
gammaD	0.1714	
Tpulse	3 min	
Tcontrol	9 min	
TcontrolX	27 sec	
alfaScale	1	
DotAlfa	10	In this simulation the DotAlfa value is very high so it will not give a noticeable effect during the simulation
F0	0.006 l/h	

The parameters/state for the Dot controller:

Parameter	Value	Comment
T	1 sec	
Ti	40 sec	
N0_Sim	370 rpm	

The initial values for the model:

Initial value	Value	Comment
V0_Sim	2 l	
X0_Sim	5.41 g/l	
A0_Sim	0.16 g/l	
G0_Sim	0.005 g/l	
Dot0_Sim	30 %	

The constants for the Probing controller and the parameters for the model can be found in Appendix C.

E. Simulation parameters for test of scaling

The parameters for the Probing controller:

Parameter	Value	Comment
Nmax	1100 rpm	
Nmin	300 rpm	
Nhigh	1045 rpm	95% of Nmax
Nhigh2	1078 rpm	98% of Nmax
OpulseSP	3 %	
k	0.96	
Oreac	1 %	
gammaP	0.1714	
gammaD	0.1714	
Tpulse	3 min	
Tcontrol	9 min	
TcontrolX	27 sec	
alfaScale	1 and 0.5	1 if it is unscaled and 0.5 if the simulation runs twice as fast as the unscaled simulation
DotAlfa	10	In this simulation the DotAlfa value is very high so it will not give a noticeable effect during the simulation
F0	0.006 l/h	

The parameters/state for the Dot controller:

Parameter/Initial value	Value	Comment
T	1 sec	
Ti	40 sec	
N0_Sim	370 rpm	
N0_CB	300 rpm	

The parameters for the model

Initial value	Value	Comment
G0_CB	1 g/l	
X0_CB	5 g/l	
V0_Sim	2 l	
X0_Sim	5.41 g/l	
A0_Sim	0.16 g/l	
G0_Sim	0.005 g/l	
Dot0_Sim	30 %	

The constants for the Probing controller and the parameters for the model can be found in Appendix C.

F. Simulation parameters used during discretization

The parameters for the Probing controller:

Parameter	Value	Comment
Nmax	1100 rpm	
Nmin	300 rpm	
Nhigh	1045 rpm	95% of Nmax
Nhigh2	1078 rpm	98% of Nmax
OpulseSP	3 %	
K	0.96	
Oreac	1 %	
gammaP	0.1714	
gammaD	0.1714	
Tpulse	3 min	
Tcontrol	9 min	
TcontrolX	27 sec	
alfaScale	1	
DotAlfa	10	In this simulation the DotAlfa value is very high so it will not give a noticeable effect during the simulation
F0	0.006 l/h	

The parameters/state for the Dot controller:

Parameter/Initial value	Value	Comment
T	1 sec	
Ti	40 sec	
N0_Sim	370 rpm	
N0_CB	300 rpm	

The initial values for the model:

Initial value	Value	Comment
G0_CB	1 g/l	
X0_CB	5 g/l	
N0_Sim	370 rpm	
V0_Sim	2 l	
X0_Sim	5.41 g/l	
A0_Sim	0.16 g/l	
G0_Sim	0 g/l	
Dot0_Sim	30 %	

The constants for the Probing controller and the parameters for the model can be found in Appendix C.

G. Simulations parameters used during comparison

The parameters for the Probing controller:

Parameter	Value	Comment
Nmax	1100 rpm	
Nmin	300 rpm	
Nhigh	1045 rpm	95% of Nmax
Nhigh2	1078 rpm	98% of Nmax
OpulseSP	3 %	
k	0.96	
Oreac	1 %	
gammaP	0.1714	
gammaD	0.1714	
Tpulse	3 min	
Tcontrol	9 min	
TcontrolX	27 sec	
alfaScale	1	
DotAlfa	1	
F0	0.006 l/h	

The parameters/state for the Dot controller:

Parameter/Initial value	Value	Comment
T	1 sec	
Ti	40 sec	
N0_Sim	370 rpm	
N0_CB	300 rpm	

The initial values for the model:

Initial value	Value	Comment
G0_CB	1 g/l	
X0_CB	5 g/l	
N0_Sim	370 rpm	
V0_Sim	2 l	
X0_Sim	5.41 g/l	
A0_Sim	0.16 g/l	
G0_Sim	0.005 g/l	
Dot0_Sim	30 %	

The constants for the Probing controller and the parameters for the model can be found in Appendix C.

H. Parameters used during the final test

The parameters for the Probing-Controller:

Parameter	Value	Comment
Nmax	1100 rpm	
Nmin	350 rpm	
Nhigh	1000 rpm	
Nhigh2	1050 rpm	
OpulseSP	1.5 %	
K	1.2	
Oreac	1 %	
gammaP	0.15	
gammaD	0.075	
Tpulse	1.5 min	
Tcontrol	6 min	
TcontrolX	27 sec	
alfaScale	1	
DotAlfa	1	
F0	0.02 l/h	
G0_CB	10 g/l	

The parameters for the Dot controller:

Parameter	Value	Description
T	1 sec	Sample time for the Dot-controller
Ti	40 sec	Ti value before the limit for the stirrer speed is reached

The constants for the Probing controller can be found in Appendix C.