Modeling and Control of Arterial Oxygen Saturation in Premature Infants

---

A Thesis presented to the Faculty of the Graduate School

University of Missouri – Columbia

In Partial Fulfillment

Of the Requirements of the Degree

Masters in Science

By

BRADLEY KRONE

Dr. Roger Fales

JULY 2011

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled

Modeling and Control of Arterial

Oxygen Saturation in Premature Infants

Presented by Bradley Krone

A candidate for the degree of Master of Science

And hereby certify that in their opinion it is worthy of acceptance.

_____

Professor Roger Fales

_____

Professor Craig Kluever

_____

Professor Stephen Montgomery-Smith

ACKOWLEDGMENTS

I would like to thank my supervisor, Dr. Roger Fales, whose expertise, understanding, and patience, helped me greatly during my graduate experience. I appreciated his knowledge and skill in the controls field and his assistance in editing my thesis. I would also like to thank the other members of my committee, Dr. Stephen Montgomery-Smith and Dr. Craig Kluever for their assistance they provided.

Table of Contents

Table of Figures

Table of Tables

ABSTRACT

In this work modeling and control of neonatal infant's blood oxygen saturation (SpO_2) is investigated. Through analyzing the biological system, it was shown that the blood is oxygenated through the mechanisms of ventilation, perfusion, and diffusion. It was also shown that the fraction of inspired oxygen (FiO_2), heart rate, and respiratory rate had an effect on the oxygenation of the blood. Four different models were used to model the SpO_2. The models investigated are a neural network, fuzzy logic, updating transfer function, and dynamic transfer function model. The best performing model was the dynamic transfer function model. This model was able to adjust to changes in the infant's biological system and accurately estimate the SpO_2 for a prolonged time. Three different controllers were designed. The controllers are a linear quadratic regulator proportional integral controller, an adaptive controller with feed-forward disturbance rejection, and robust controller. All controllers were designed using the dynamic transfer function model. The controllers were designed to regulate the SpO_2 by adjusting the FiO_2. Each controller attempted to reject the disturbances caused by the heart rate and respiratory rate. The controllers were tested on simulated data. The best controller was found to be the robust controller with an average SpO_2 of 6.6423e-004% and a maximum SpO_2 value of 0.0725%. The SpO_2 is normalized at 90% and the FiO_2 is normalized at 21%. The SpO_2 and FiO_2 values presented here are the difference between the actual values and the nominal values.

Chapter 1: Introduction

1.1. Background, Motivation, and Objectives

A common problem in neonatal infants is Respiratory Distress Syndrome (RDS), caused by a lack of a protective substance surfactant, which helps the lungs inflate and prevents the air sacs from collapsing [1]. A treatment for RDS is to place the infant on a respiratory support device to allow the lungs to further develop. While on the respiratory support device the infant is supplied with a controlled fraction of inspired oxygen (FiO_2). Currently the FiO_2 is regulated by a nurse. The regulation is done by monitoring the SpO_2 and adjusting the FiO_2 as needed. The regulation of FiO2 is based on accepted medical practices and the nurses' judgment. By adjusting the FiO_2, the SpO_2 can be regulated. The desired SpO_2 range is from 85% to 92% [2]. If the SpO_2 level is maintained below 85% a state of hypoxia could occur which can result in tissue damage, brain injury, and even death. On the other hand, if the SpO_2 level is above 92% the infant is at risk for retinopathy of prematurity which is associated with visual impairments and even blindness [3]. Research has shown that neonatal infants spend only 50% of the time within the acceptable ranges under manual control of the FiO_2. The remaining 20% is spent below the acceptable SpO_2 range and 30% above the acceptable SpO_2 range [4, 5]. Safety limits are set to notify the nurses when the infant is outside of the acceptable range. However it has been shown that safety limits are often set outside the recommended ranges [6]. The incorrect setting of safety limits can cause situations where the infant is outside acceptable limits for prolonged time. It is thought that by

automatically regulating the FiO_2 that the SpO_2 can be better regulated and prevent injury to the infant.

The goal of this thesis is to investigate automatic control of the FiO_2 to regulate the measured SpO_2. It is thought that by automatically regulating the SpO_2 the work load of the nurses can be reduced and the SpO_2 can more accurately be regulated. To better understand how the SpO_2 can be controlled and kept within the specified bounds, several different modeling techniques were used. These models investigated the relationship between the SpO_2 and the FiO_2, heart rate (HR), and respiratory rate (RR). By using the HR and RR in the model it is thought that the model will be able to take into account apnea events. Apnea is defined as an absence of breathing for 20 seconds or longer, or for shorter times if bradycardia occurs [7]. Bradycardia is when the heart is beating abnormally slow. Apnea is observed in more than half of surviving premature infants mass less than 1.5 kg at birth [8]. Modeling these events will lead to a better understand of how apnea events effect the infant and thus better treatment.

The models investigated are models that can be trained on previous data and produce SpO_2. This was needed due to the continuously changing nature of the infant caused by the surroundings, drug administration, and the biological changes in the infant. Since the system is constantly changing a fixed model was not able to adequately model the data. To take into account the uncertainty in the infant dynamics, a model that can be generated by applying a model training algorithm to the measured signals was created. A controller was then able to be designed based on the model that was able to best model the SpO_2. The benefit of a controller is that it can monitor the infant at all time whereas a nurse has a variable workload and multiple infants to monitor at one time. By

continuously monitoring the infant it is thought that the controller will be able to keep the infant well within the acceptable bound of 85 to 92%. By continuously monitoring the infant the system will react faster than a nurse could, to the onset of apnea and mitigate the affect. Several different controllers were designed and tested to see which performed the best. The controller could only use noninvasive measurements such as HR, RR, and SpO_2. The controllers would select an optimal FiO_2 input to keep the infant at a safe range of SpO_2. The controller also attempted to reject the effect the HR and RR has on the infants SpO_2.

1.2. Literature Review

Respiratory system of humans was first investigated by Gray in 1945 [9]. His model was only able to take into account steady-state responses of the system to CO_2 inhalation, arterial anoxemia, and metabolic disturbances. Grodins et al added dynamics and transport delay into the respiratory system model in 1967 [10]. This was done through a system of nonlinear equations that can be used to predict changes in the blood gases during conditions of hypocapnie and hypercapnia. Grevisse developed a predictive global model of the pulmonary functions using only physically accessible variables. His system consisted of sub-systems that modeled different parts of the pulmonary system [11]. Yu developed a model that mapped the relationship between the arterial oxygen partial pressure (P_aO_2) and FiO_2 [12]. This system allowed the model to adapt to different infants and thus better model the infants response to increases in FiO_2. In this work Grevisse's idea of sub-systems was combined with Yu's idea of an adaptive model to create a robust model of the infant's biological system.

Tehrani et al designed a proportional, integral, derivative controller For the control of the ventilation of premature infants in 1991 [13]. The controller adjusts the oxygen concentration of the inspired gas to allow for sufficient oxygenation of the blood. The oxygenation was monitored to prevent damaging effects of oxygen toxicity. Yu used a multiple model adaptive controller (MMAC) to regulate the arterial oxygen saturation (S_aO_2) by adjusting the FiO_2 [12]. The MMAC assumes that the system can be represented by a finite number of models. A controller was then designed for each model to give an acceptable closed-loop response with zero steady-state error. Keim developed a single robust controller based on a linear model [2]. The robust controller was designed based on an error model and performance specifications. Keim also developed an adaptive controller based on estimated parameters and disturbances [2]. The controller attempted to regulate the FiO_2 while mitigating the affect of the disturbances.

In this work, Yu's idea of multiple controllers and Kiem's ideas of robust controller were combined and advanced. This is done by creating multiple robust controllers and an adaptive controller that takes into account the affect of the HR and RR on the SpO_2 and attempts to reject there affect on the SpO_2. The robust controller was designed based on collected estimated parameters. Multiple robust controllers were needed to allow the controller to achieve robust stability and performance over the broad range of parameters. Keim's idea of adaptive controllers is also improved by replacing the estimated disturbance with disturbances caused by the HR and RR. The controller estimates the affect the HR and RR has on the SpO_2 and chose to either reject or accept

the disturbances allowing the controller to use its own disturbances to drive the system to the desired set point.

1.3. Thesis Overview

This thesis consists of three parts. In Chapter 2, the biological system of the infant was investigated to determine what parameters gave insight into the SpO_2 levels. The values of FiO_2, HR, RR, and SpO_2 are recorded from infants at the Neonatal Intensive Care Unit at the Columbia Regional Hospital. The data is then analyzed to see how each parameter affect the SpO_2 levels.

In Chapter 3, four different modeling techniques were used. The modeling techniques used are neural network, fuzzy logic, transfer function model, and dynamic transfer function model. These models were designed to be trained on a window of data and produce a relationship between the inputs and output. The inputs to the neural network and fuzzy logic models are the FiO_2, HR, RR, and the first derivate of each. The inputs to both transfer function model are the FiO_2, HR, and RR. All of the models have an output of SpO_2. The models were tuned using a design of experiments (DOE) to obtain optimum performance. The models were tested to see how well each model could adequately represent the SpO_2.

In Chapter 4, three different controllers were designed based on the dynamic transfer function model. The controllers designed are a linear quadratic regulator proportional integral controller (LQR-PI), robust controller, and an adaptive controller with feed forward disturbance rejection (AC-FFDR). The three controllers were tested to see how well they would reject disturbances caused by the HR and RR. The disturbance

simulation was designed to simulate an apnea event. The best model was select based on three criteria, the maximum SpO_2, maximum FiO_2, average SpO_2, and average FiO_2 value.

Chapter 2: Data Analysis

In this Chapter the infant's biological system is investigated by analyzing measured signals to see how the blood is oxygenated and what measurable parameters will give insight into the SpO_2 levels. Analyzing the measured parameters will lead to a better understanding of what parameters should be recorded when monitoring an infant. The recorded data will then be analyzed in an attempt to quantify how each parameter affects the SpO_2.

2.1 Biological System and data collection

In the infants biological system the blood is oxygenated through the mechanisms of ventilation, perfusion, and diffusion [14]. The ventilation process is a component of the respiratory system and responsible for moving gases in and out of the lungs. The perfusion process is a component of the cardiovascular system and responsible for transporting respiratory gases via blood to the lungs and tissues. The diffusion process is the component that links the respiratory and cardiovascular system and involves the exchange of gases between the lungs and blood vessels.

To analyze how much each mechanism affects the SpO_2 levels, data was collected on the premature infants with frequent desaturation periods. The data collected had to be done through noninvasive measures. The parameters measured were the heart rate (HR), respiratory rate (RR), the fraction of inspired oxygen (FiO_2), and blood oxygen saturation levels (SpO_2). The heart rate was collected to give insight into the cardiovascular system. The respiratory rate and fraction of inspired oxygen were measured to take into account the respiratory system. The data was collected at the Neonatal Intensive Care Unit at the

Columbia Regional Hospital. Keim developed a data acquisition program using LabView that recorded the infant's data every 5 seconds [14]. The setup of the data acquisition system can be seen in Figure 1. For more information on data collection please refer to [14].

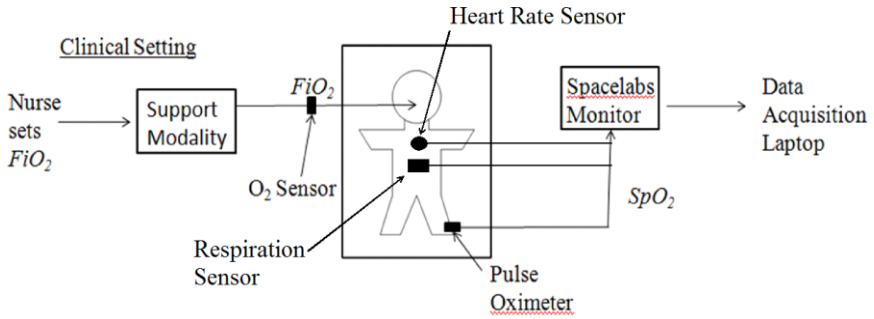

Figure 1. Data acquisition setup

2.2 FiO_2 and SpO_2 relationship

To better understand the relationship between the FiO_2 and SpO_2, events were identified when the nurse increased the FiO_2. A first-order transfer function was then fit to the SpO_2 response to the FiO_2 inputs. The gains and time constants were then analyzed for any overall trends.

The first step in analyzing the relationship is to find FiO_2 events. The events were found by detecting when the FiO_2 was increased by 1.5% or greater. To reduce false identification the FiO_2 data was filtered using a moving average with an averaging window of 15 data points. This window size was chosen to minimize the lag while maximizing the accuracy of FiO_2 event detection. Only increases in FiO_2 were investigated in this Section. Once the events were detected a relationship between the increase in FiO_2 and the SpO_2 response was needed. Through visual analysis it was

decided that a first-order transfer function would result in a reasonable model of the relationship. As seen in Figure 2 there is a lag between the FiO_2 input and the SpO_2 response. This lag is due to the pulse oximeter sensor's dynamics and the time it takes oxygenated blood to reach the sensor location. This lag was not constant throughout the data and will need to be taken into account.

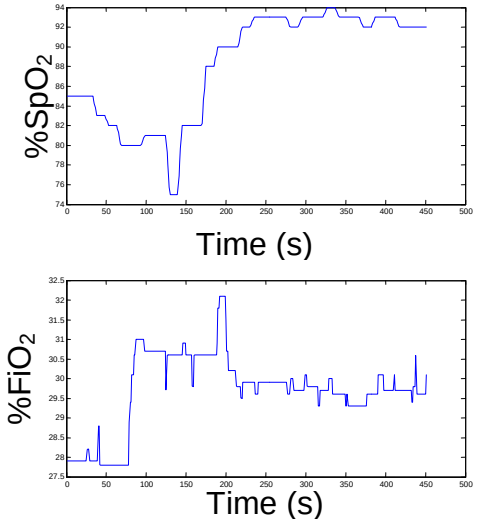

Figure 2. Example of first-order response of the SpO_2 to a FiO_2 input

To quantify the relationship, a first-order transfer function was fit to the SpO_2 response to an increase of FiO_2. The fitting process was done by first finding the starting and steady state point of the SpO_2 and FiO_2 event. Due to the lag in the system the starting point of the SpO_2 is not at the same time as the FiO_2 increase. To find the starting point the minimum SpO_2 value within a 200 second window, past the starting point of the FiO_2 event, was found. This point was to be considered the start of the SpO_2 response. The steady-state value of the SpO_2 ($SPO2_E$) was found by finding the maximum value of SpO_2 with in a window of 500 seconds past the start time of the FiO_2 event. The maximum value of the SpO_2 can be considered the steady-state response since the system has no overshoot. The FiO_2 starting value ($FIO2_s$) was found by taking

the initial value at the beginning of the identified event. The FiO_2 end value ($FIO2_E$) was found by finding the maximum value the FiO_2 reached from the start of the event to the corresponding time of the end of the SpO_2 event. Figure 3 shows an example of a step input of FiO_2 and its corresponding SpO_2. In Figure 3 the circles and squares represent the start and end of the event. The triangle represents the 63% of the SpO_2 value. The green line on Figure 3 indicates the filtered data and the blue line indicate the measured data. The SpO_2 data was filtered using a moving average with a window size of 15 seconds were the data is sampled every 5 seconds. All calculations were done using the filtered SpO_2 data. Filtering was done to reduce the noise in the system and all for better approximation of the gains and time constants. The gain for the first order transfer function can be found using the start and steady state points found above. The dc gain is given as,

$$G_g = \frac{SPO2_E - SPO2_S}{FIO2_E - FIO2_S} \tag{2.1}$$

where the initial time constant (τ) was found by finding the time it takes the SpO_2 to get to 65% of the overall value. The gain and time constant can be combined to create the initial guess of the first-order transfer function. The first order transfer function is

$$T_F(s) = \frac{G_g}{\tau s + 1} \tag{2.2}$$

The initial fit of the transfer function did well in finding the gain of the system but poorly estimated the time constant. To better fit the transfer function a bisection minimization method was used. The minimization was done by finding the point where the SpO_2 and the step response to the transfer function first reached steady state. The

distance between the two points was then found. The bisection minimization method was used to minimize the distance between the two points. This method greatly increased the accuracy of the fitted transfer function. An example of the fitted transfer function and be seen in Figure 4. The corresponding FiO_2 step input for Figure 4 can be seen in Figure 3. The dc gain and time constant for the fitted transfer function was found to be 4.727 and 67.25 seconds.

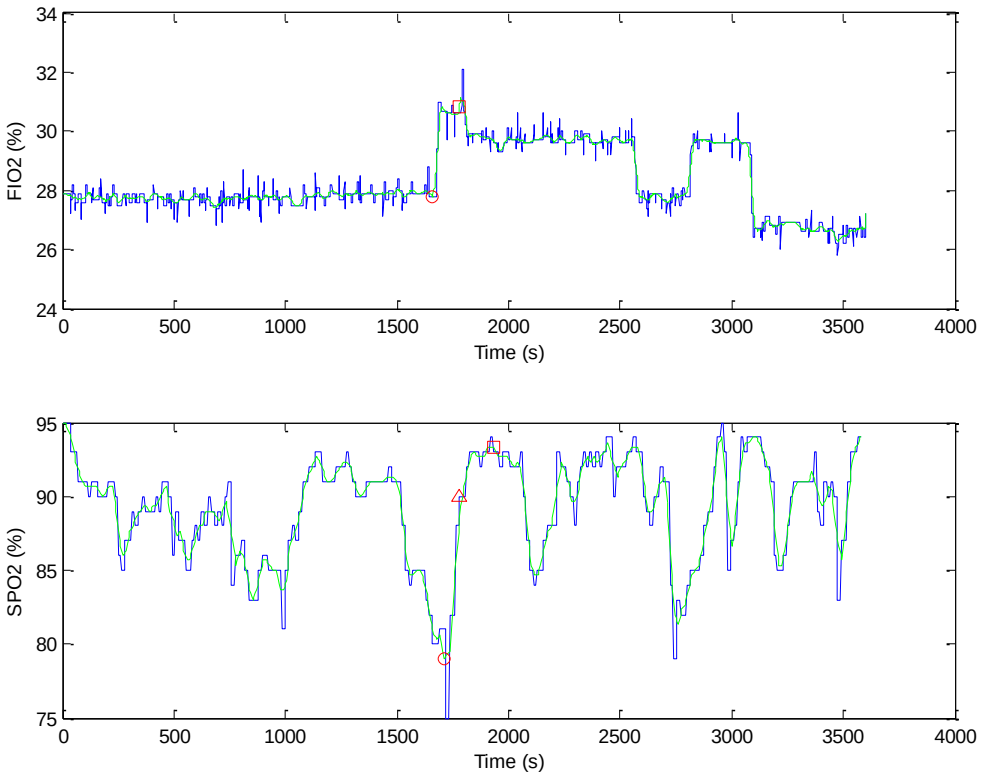

Figure 3. Plot of event identification

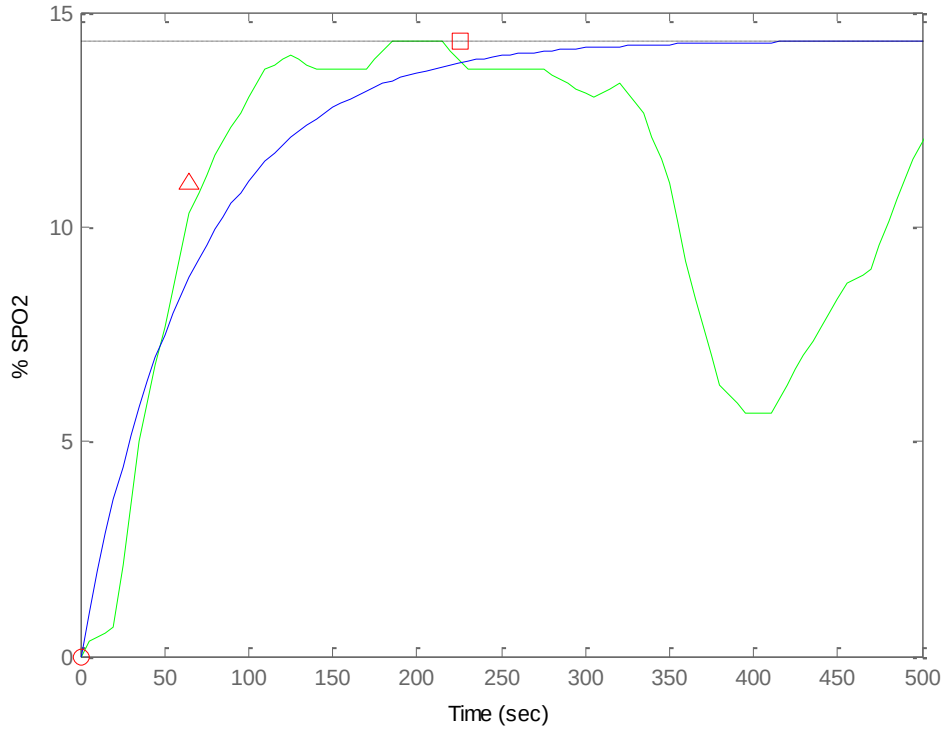

Figure 4. Fitted transfer function to a FiO_2 step input

The data collected at Neonatal Intensive Care Unit was analyzed and first-order transfer function was fit to identified FiO_2 step inputs. To assure that the program was only recording well fitted transfer functions, the program only recorded transfer function values that had an error less than 10%. The error was found by finding the percent error between the transfer function and the SpO_2. The error was calculated starting at $SPO2_s$ and going to $SPO2_E$. The data was analyzed for any possible trends between the gains and time constants and other recorded parameters. A plot of all the time constants and corresponding gains can be seen in Figure 5. The recorded data was analyzed for trends using statistical method, plots, and neural networks and no direct trends could be detected. Since there was no trend between the gains and time constants, one generalized model will not be able to model the SpO_2 responses. However if the gains and time

constants can be estimated correctly, the SpO_2 response to a FiO_2 input can be modeled with a first order transfer function.

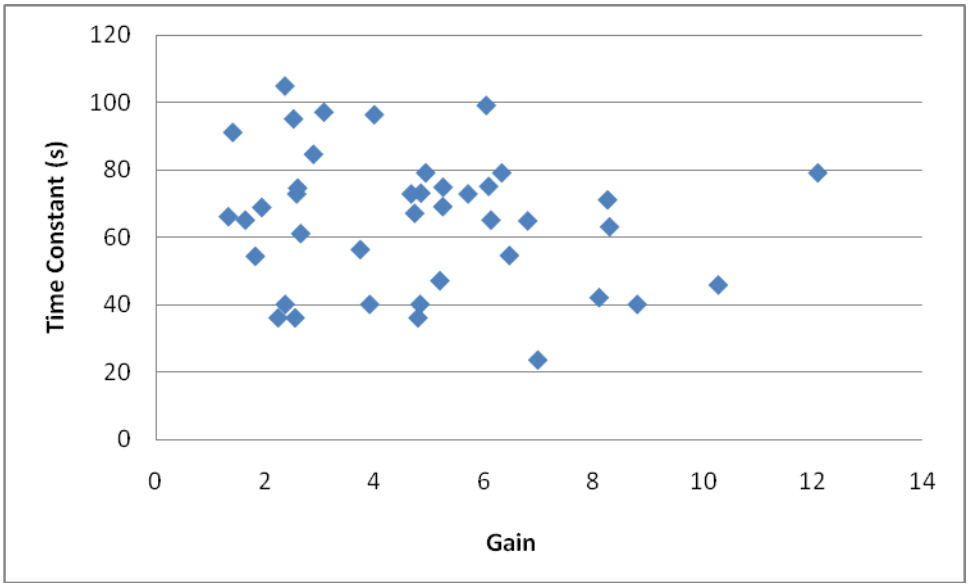

Figure 5. Plot of the dc gain vs. time constant for all fitted transfer function

2.3 Heart Rate and Respiratory rate effect on SpO_2

To quantify the affects the HR and RR have on the SpO_2 is inherently difficult to do and is usually done through subjective observation. The effect of the HR and RR on SpO_2 though can be seen in some situations. An example of how the RR affects the SpO_2 can be seen in Figure 6. The affect of the RR on SpO_2 can be seen at time 1170 seconds where the RR goes up a significant amount. The increased in the RR is mirrored by the SpO_2 starting at 1300 seconds. Since there is relatively little change in the FiO_2 and HR the change in the SpO_2 can be attributed to the RR. The relationship between the HR and the SpO_2 is only visible when there is a drastic change in the infants HR. This relationship can be seen in Figure 7. In Figure 7, the HR and SpO_2 was normalized around 0 so that the data could be compared easier. As seen in Figure 7 when there is a large decrease in HR, there is an immediate decrease in the SpO_2. However for both the

HR and RR the higher frequency changes have no visible affect on the SpO_2. This does not mean it does not have an effect on the SpO_2 but rather they cannot be visually observed.

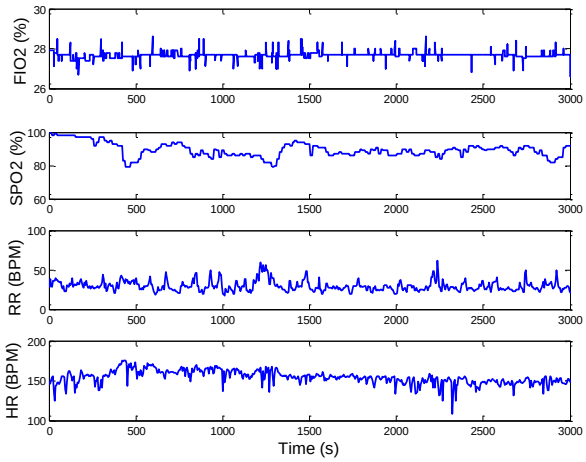

Figure 6. Respiratory Rate effect on SpO_2

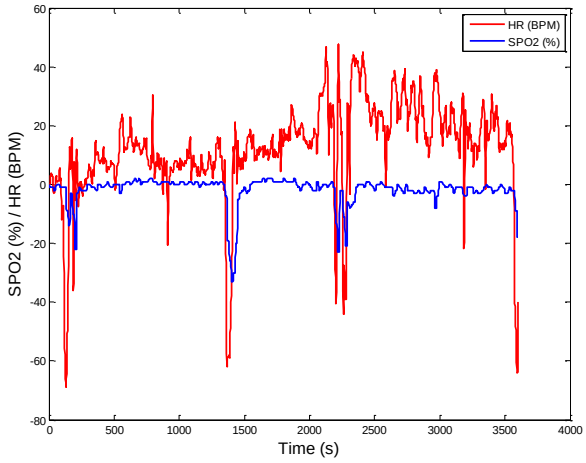

Figure 7. Heart rate effect on SpO_2

2.4 Conclusion and Analysis

Through analyzing the biological system, it was shown that the blood is oxygenated through the mechanisms of ventilation, perfusion, and diffusion [14]. To

help identify how each of these parameters affect the infants SpO_2, the heart rate, respiratory rate, FiO_2, and SpO_2 were recorded from infants with frequent desaturation periods. It was found that the relationship between the FiO_2 and SpO_2 can be modeled using a first order transfer function. The range for the gains and time constants was found to be [1.3, 12] and [20, 110]. For the HR and RR there was no method to quantify the relationship with the SpO_2. However the effect of the HR and RR can be visually observed in specific cases. Through visual analysis it was found that any drastic increase or decrease for a prolonged time in the HR or RR had a direct affect on the SpO_2. There was no conclusive evidence that the higher frequency signal components of the HR and RR had an effect on the SpO_2. Through the analysis of the biological system and recorded data it is though that by considering the relationship between the SpO_2 and the FiO_2, HR, and RR that a model will be able to better simulate the SpO_2 response to inputs.

Chapter 3: Modeling

In this Chapter several different modeling techniques were investigated. The models were chosen to be models that could be trained on previous FiO_2, HR, and RR data and simulate SpO_2 responce. This type of model not only allows the system to be customized for each infant but also allows the model to evolve in case the infant biological system further develops or degrades. The models map the relationship between the inputs and output of the infant. In doing so the model should be able to take into account the changes in the infant's dynamics. There are two types of changes that can take place measuredand non-measured changes. An example of a measured change is if the HR goes up there is a faster transfer of blood which can cause a change in the relationship between the HR and SpO_2. A non-measured change would be a change in the infant that does not show up on any measured data. These type of changes can be measured but usually require invasive measure and are not common under standard clinical care. One of the more common non-measured changes is shunting. Shunting is when venous blood enters into the bloodstream without passing through the lungs [15]. This can produce dangerously low SpO_2 levels. To quantify the effect of non-measured changes is inherently difficult to do and is generally accounted for by subjective observation [16]. However the effect will show up in the relationship between the inputs and outputs as an increased or decreased response. It is thought that by mapping the relationship between the three inputs and the output, the model should be able to take into account for a wide variety of changes in the cardiovascular and respiratory system. This will allow an increase in model accurse and a better understand of the SpO_2.

3.1 Updating Models

The concept of a model that updates as new measurements are received (updating model) was chosen for its ability to have a generalized model of the SpO_2 while allowing the model to be updated when biological changes occur. The models investigated are a neural network, fuzzy logic, and transfer function model. The basic structure of the models is to have FiO_2, HR, and RR supplied to the model and SpO_2 produced. The models were designed by creating a window of data and training on the corresponding data. The training window consists of the previous set of data points starting at time 0 and going back a set amount of time. The training data consisted of FiO_2, HR, and RR. An example of a training window can be seen in Figure 8. The model was run starting at time 0 and run until the model no longer adequately modeled the SpO_2. The window of data is then moved up to the current time and the model is retrained. In the next subsections three different updating models will be investigated: neural network, fuzzy logic, and transfer function models.

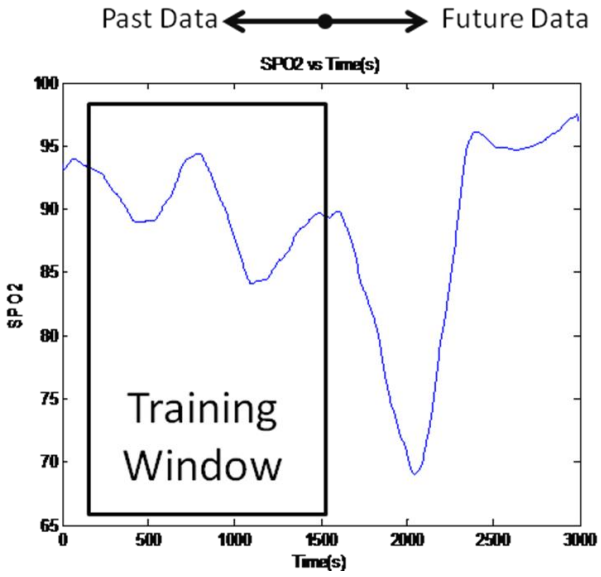

Figure 8. Example of SPO2 training window

3.1.1 Neural Network Model Structure

The neural network model was chosen for its ability to analyze the input and output of a system and produce a highly nonlinear model with no previous information. This allows for detection of complex nonlinear relationships between the dependent and independent variables [17]. The structure of the model consists of a two-layer feed-forward network with a sigmoid hidden layer and a linear output neuron layer. The sigmoid function was chosen for the first hidden layer for the ability to create nonlinearity between the neuron's input and output. The sigmoid function though does prevent the system from extrapolating beyond any trained data. The inability to extrapolation is caused by the saturation affect of the sigmoid function which can be seen in Figure 9 were any value above 4 produce a value of 1 and any value below -4 produces a value of 0. The second hidden layer was chosen to be a linear scaling function which rescale the values to realistic ranges of SpO_2. Since all values need to be rescaled with the same value only one neuron was used in the second hidden layer, thus scaling the values the same amount. The combination of the two hidden layers allows the neural network to approximate any function given a sufficient number of neurons in the first hidden layer [17]. The inputs into the neural network are FiO_2, HR, RR, and the first derivative of each. The first time derivative of each was chosen to allow the neural network to be more dynamic. The model's structure can be seen in Figure 10.

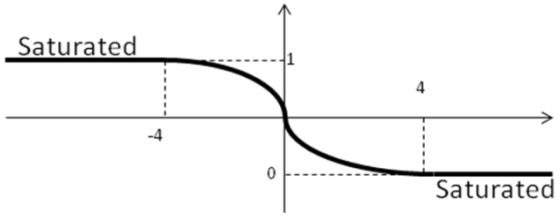

Figure 9. Sigmoid function saturation example

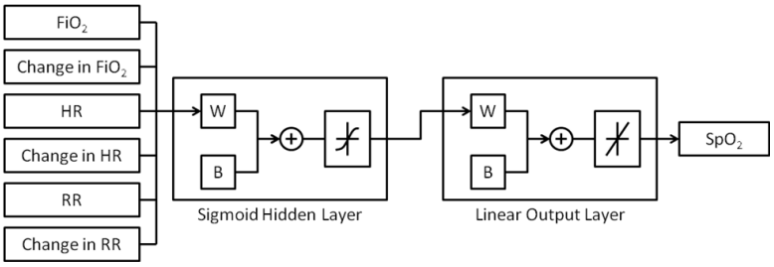

Figure 10. Neural network Model

The model was trained using Levenberg – Marquardt backpropagation [18]. This training method was chosen for its ability to approach second-order training speeds without having to compute the Hessian matrix. Second-order training speeds is important since it can be time consuming and difficult to calculate the second-order derivative. Since the network is in the sum of squares format the Hessian matrix can be approximated as

$$H = J^T J \tag{3.1}$$

Where J is the Jacobian matrix that contains the first derivatives of the network errors with respect to the weights and biases. The Jacobian can be computed through a backpropagation technique. The gradient of the network was computed as,

$$g = J^T e \tag{3.2}$$

Where e is the vector of network errors. Equation (3.1) and Eq. (3.2) can be combined and given as,

$$x_{k+1} = x_k - \frac{J^T e}{[J^T J + \mu I]} \tag{3.3}$$

Where μ is the step size and I is the identity matrix. To adjust between gradient descent and Newtons's method the step size is adjusted. For example when μ is large, Eq. (3.3) turns into a gradient descent method. On the other hand when μ is small, Eq. (3.3) turns into a Newton's method. Since the Newton's method works faster the goal is to achieve a small μ value. A small μ value is acheived by decreasing the value of μ every time the error is successfully decreased. The fit between the actual SpO_2 and the simulated SpO_2 was found using a mean squared error method. The model was trained until generalization stopped improving. This was done to prevent overtraining of the neural network. Generalization was identified by detecting when the mean squared error of the validation samples increases.

3.1.2 Fuzzy Logic model Structure

The fuzzy logic model was chosen for its linguistic properties. The linguistic properties allows the model to have easily understood rules that can give insight into how the infant will react to similar situation in the future. An example of the linguistic properties is the FiO_2 is constant and the HR is going down and RR is going down then the SpO_2 will go down. This process is very intuitive and works in a similar way that the nurses do when determining how the baby will respond to inputs of FiO_2 [19].

The first step in setting up a fuzzy logic model is to generate a set of rules and membership functions. The membership function were set up using the genfis2

command in MATLAB. Genfis2 command analyzes the input and output data, and uses subtractive clustering to set up a hypothesized number of membership functions and corresponding rules. An example of the initial membership functions can be seen in Figure 11.

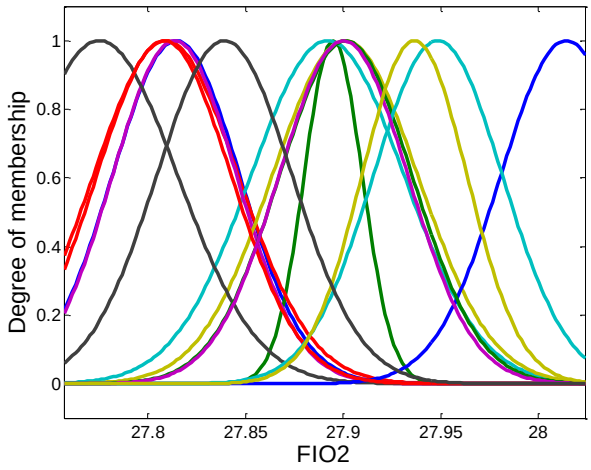

Figure 11. Initial membership functions for FiO_2 input.

The adaptive neural-fuzzy inference system (anfis), command in MATLAB, was then used to adjust the membership function to best model the data [20]. The anfis command modifies the membership function parameters such that the error between the SpO_2 and modeled SpO_2 is minimized. The computation of the parameters is done by a gradient vector method. The gradient vectore provides a measure of how well the fuzzy model is modeling the input/output data for a given set of parameters. Once the gradient vector is obtained the parameters are adjusted to reduce the error in the output by backpropagation and a least-squares estimation hybrid. The error is quantified by the sum of squared difference between the actual and desired outputs. The anfis command trains until either a set number of epochs have been achieved or the error drops below a specified value. Once the model is created with the anfis command the evalfis command was used to

evaluate the fuzzy model. The centroid defuzzification method was used for the defuzzification process. The equation for the centriod defuzzzification can be seen in Eq. (3.4).

$$x^* = \frac{\int \mu_i(x) x dx}{\int \mu_i(x) dx} \tag{3.4}$$

In Eq. (3.4), x and $\mu_i(x)$ are the output variable and the aggregated membership function. The inputs into the fuzzy logic model are FiO_2, HR, RR, and the first derivative of each. The first derivative of each was chosen to allow the fuzzy logic model to be more dynamic. A diagram of the model can be seen in Figure 12.

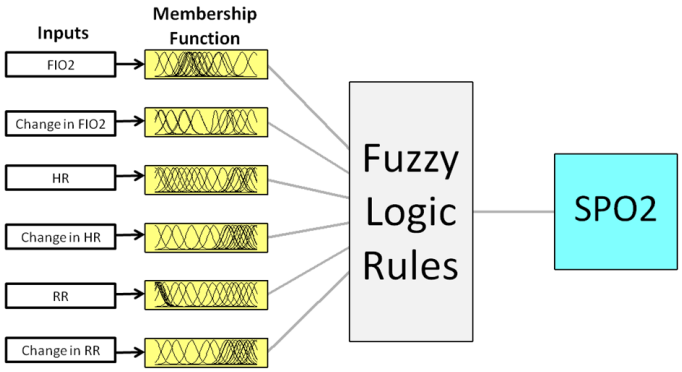

Figure 12. Fuzzy logic Model

3.1.3 Transfer Function Model Structure

The transfer function model was chosen for its compatibility with well established control design methodologies. This model also gives insight into how the SpO_2 will be affected by changes in each input, allowing for a control of the system that is further investigated in Chapter 4. The combination of these traits gives a good base for design of a controller to be discussed in Chapter 4.

The transfer function model consists of three first order transfer functions. The FiO_2 transfer function is given as,

$$\frac{G_{FIO2}}{T_{FIO2}s+1} \tag{3.5}$$

where G_{FIO2} and T_{FIO2} are the dc gain and time constant for the FiO_2. Transfer functions for HR and RR are the same as in Eq. (3.5) except for the subscripts. Each input is entered into its corresponding transfer function. The outputs of the transfer functions are then summed to produce the final SpO_2 value. Figure 13 shows a block diagram of the model.

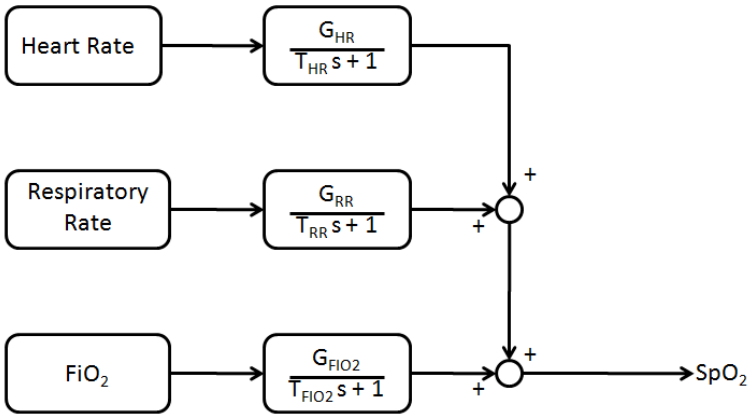

Figure 13. Transfer Function Model

The genetic algorithm was used to select the gain and time constants for each transfer function. Genetic algorithms are search methods that mimic the process of natural evolution. This search method was chosen for its ability to resist local minimums and find a global minimum in a large search space without calculating the derivative [19]. The genetic algorithm is set up by first generating an initial population of individuals. The individuals were set up using floating point representation with 6 genes in each individual, where the first three genes are the gain and the last three are the time

constants. The bounds for the gain and time constants were found by running the transfer function model over a large data set with bound of (1.3, 20) for the FiO_2 gain, (-20, 20) for the HR and RR gain, and (0.1, 1000) for all time constant. The lower limit of the FiO_2 was selected to be the smallest gain value found in Chapter 2. This was done to keep the gain from taking on a negative value which can cause the system to become unstable when applying control. Limiting the gain also keeps the FiO_2 gain at a reasonable value when the system is not fully excited. When the system is not fully excited the gains and time constants tend to wonder and not correctly estimate the parameter. The bounds for the time constants and the HR and RR gains were selected to be much larger than any previously seen gain and time constant seen in preliminary tests allowing the genetic algorithm to search for the global minimum while being unhindered by the bounds. The data was analyzed and a range of gains and time constants were found that minimized the search space while still allowing for all reasonable gains and time constants. By minimizing the search space the genetic algorithm was able to converge faster on the optimum solution. Table 1 presents the ranges for the gain and time constant.

Table 1. Range of gains and time constants

Inputs	**Gains**	**Time Constants (sec)**
FiO_2	[1.6 - 6]	[0.1 – 200]
Heart Rate	[-6 - 6]	[0.1 – 200]
Respiratory Rate	[-6 - 6]	[0.1 – 200]

The genes for each individual were generated using the rand() command in MATLAB ®. Each gene was scaled to be within the bounds specified in Table 1. The initial population size was set to a specified amount. Each subsequent generation would

maintain the same number of individuals as the first generation. All genes within a generation would stay within the bounds specified in Table 1.

The fitness of each individual was found by running a simulation using each individual's gains and time constants and placing them into the transfer function model. The simulation was run on the window of data and produced a simulated SpO_2. The fitness of the individual was found by comparing the simulated SpO_2 to the actual SpO_2. The error between the two was quantified using a mean squared error method given as,

$$E_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(P_i - T_i\right)^2 \tag{3.6}$$

Where n is the number of data point, P_i is the model SpO_2 at location i, and T_i is the actual SpO_2 at position i. A lower error indicates better modeling of the data and thus a more fit parent. The fitness value of the parent was set to be the value found by Eq. (3.6). Since this value is always positive the genetic algorithm was set to find the individual with the lowest fitness value.

The individuals are then reproduced to create a population of offspring. The reproductive process is done through a combination of crossover and mutation. The crossover process is done by first selecting two parents to be crossed over. The selection of the parents was done using the roulette wheel selection method. The roulette wheel selection was chosen due to the property that the fittest individuals will have a greater chance of surviving to the next generation than the weaker individuals. Weak individuals are not without a chance though. Even though the individual may be weaker they might have a good trait. In allowing the weak to have a chance to move on the

genetic coding might prove useful in the future. This means the genetic algorithm is less likely to be susceptible to local minimums. The equation used to find the relative fitness of an individual is given as,

$$P_R = \frac{1}{1 + F_{\max} - F} \tag{2.7}$$

Where F_{max} is the maximum fitness value of each generation and F is the fitness of the individual. The percent fitness is given as,

$$F_P = \frac{P_R}{\sum P_R} \tag{2.8}$$

where each value of F_p represents a bin on the roulette wheel as shown in Figure 14. In using the roulette wheel method the strongest individuals have the largest share of the roulette wheel and the weakest have the smallest share. A random number was then generated using the rand() command in matlab. This number corresponded to a position on the roulette wheel. That position selected a bin which thus selected the individual that was linked to the bin. Each individual was allowed to be selected once. Once the individual was selected they were removed from the roulette wheel. The process was then run until each individual was selected. The individuals were then matched up to perform a crossover. The matching process was done by taking the first two selected and matching them. The next two were then selected and matched. This process was repeated until all individuals were matched.

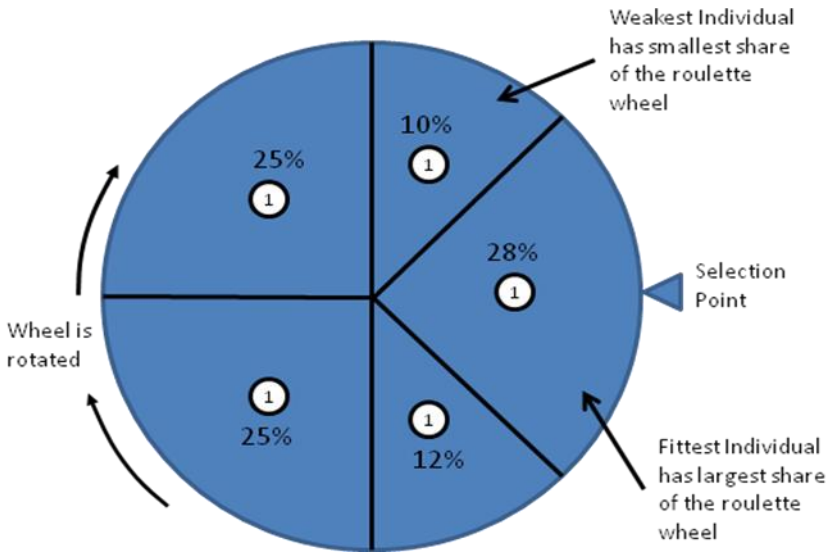

Figure 14 Roulette wheel

To create the offspring a floating point crossover method was used. The crossover was done by taking the two matched individuals and performing the floating point crossover which produced two offspring. The offspring were created using Eq. (3.9) and Eq. (3.10).

$$O_1 = \begin{bmatrix} (1-\xi_1)*C_{11} + \xi_1 * C_{21} \\ (1-\xi_1)*C_{12} + \xi_1 * C_{22} \\ \vdots \\ (1-\xi_1)*C_{ij} + \xi_1 * C_{ij} \end{bmatrix} \tag{3.9}$$

$$O_2 = \begin{bmatrix} (1-\xi_2)*C_{11} + \xi_2 * C_{21} \\ (1-\xi_2)*C_{12} + \xi_2 * C_{22} \\ \vdots \\ (1-\xi_2)*C_{ij} + \xi_2 * C_{ij} \end{bmatrix} \tag{3.10}$$

In Eq. (3.9) and Eq. (3.10), ξ is the crossover value, C is the corresponding generations from the selected chromosome, and O is a vector of model parameters . Mutation was then performed on some offspring to allow for genetic diversity. The selection of the

parameter, to be mutated, was done by generating a random number. If the random number was less than the mutation rate the parameter would then be mutated. The parameter was then replaced by a randomly generated number that was within the given parameter bounds. A decaying mutation rate was used were the initially mutation rate is 50% for the first 25% of the total generations. The mutation rate was then decreased to 20% for the next 50% of the total generations. The last step dropped the mutation rate to 10% for the remanding generations. By decreasing the mutation rate over time the genetic algorithm was able to fully search the feature space in the beginning generation and hone in on an exact solution in the later generations. The decaying mutation rate also prevented premature convergence onto local minimums.

The offspring and the parents then need to be culled to set up the next generation. This was done to maintain a constant population. The culling process was done by selecting 20% of the parents to live on to the next generation. This selection was done using the roulette wheel method as discussed above. The fittest individual was always moved to the next generation. The offspring was culled by selection the first 80% of the offspring to be moved onto the next generation. The fitness process was then repeated until the stop criterion was reached, which corresponds to a specified number of generations.

3.1.4 Validation of Updating Models

To validate that the updating models are able to adequately represent the data each model was tested on two different sets of simulated data. One set of data was generated with fixed parameters and the other with changing parameters. These two data sets were used to see how well the models did under fixed parameters and changing

parameters. The simulated data was designed to mimic a desaturation event in an infant. The data was generated using the same structure as seen in Figure 13. To simulate a desaturation event, inputs were applied to the HR and RR transfer functions. The HR data simulated the fluctuation in the HR that causes a high frequency oscillation in the SpO_2 data. The HR input was simulated with a sin wave with amplitude of 1 bpm, a frequency of 0.1Hz, and with a mean value of -2bpm. The RR data was constructed to simulate a desaturation period. A desaturation period is a period where the infant stops breathing which causes a dramatic drop in the SpO_2. The desaturation was simulated by applying a box wave to the RR transfer function. The box wave was decreased from -2 to -12 every 800 seconds for a period of 160 seconds. A sine wave was superimposed onto the box wave. The sine wave simulated the constantly changing nature of the infant's respiratory rate. The sine wave had an amplitude of 1 and a frequency of 0.05 rad/s. Both the HR and RR initial values were set to be zero. The disturbances were then ramped down to -2 with a slope of -0.2, allow the model training system to work properly. The HR and RR inputs where put through a low-pass Bessel eigth-order filter to remove any sharp edges in the data. The passband edge frequency for the HR data was 1 rad/s and 0.5 rad/s for the RR data. The simulated inputs to the HR and RR transfer function can be seen in Figure 15. The HR and RR were then added together and summed with the FiO_2 transfer function output. The data was fed back and subtracted from the reference point of zero. In doing so the system completed a closed loop. A closed loop was used to allow the FiO_2 transfer function to be excited by the feedback of the system. A block diagram of how simulated data was created can be seen in Figure 16. For the fixed parameter model the gains and time constants kept at the same values

throughout the simulation. For the variable parameter data the gains and time constants were constant for the first 150 seconds then changed for the remaining time. The values for the gains and time constants for the fixed and variable-parameter data can be seen in Table 2.

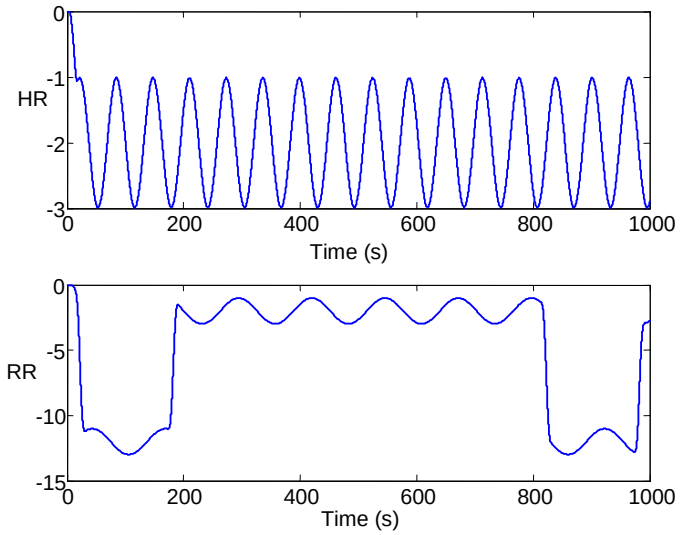

Figure 15. Plot of simulated HR and RR inputs

Table 2. Simulated Time constants and gains

	Gain		Time Constants (sec)	
Inputs	Fixed	Variable	Fixed	Variable
FIO2	4	4 to 2	120	120 to 40
HR	3	3 to 1	100	100 to 150
RR	1.5	1.5 to 3.5	110	110 to 160

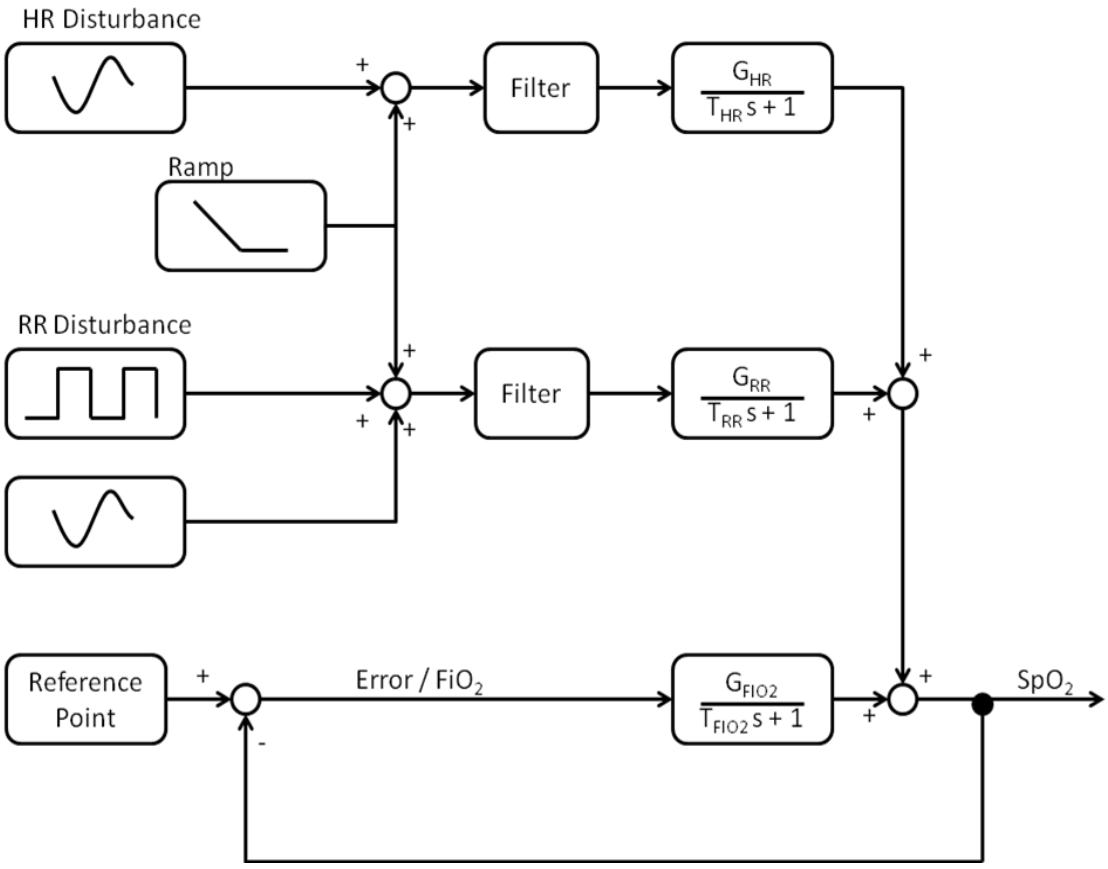

Figure 16. Block Diagram of simulated disturbance rejection Data

Each model was tested on both data sets. The models were trained on 1000 seconds of data. The models were then run for 1000 seconds to see how well the models represent the data. The neural network model had an over training value of 0.1 and 16 neurons in the first hidden layer. The fuzzy logic had an over training value of 0.1 and a cluster size of 0.2. The transfer function model was training for 50 generations. The results from the linear and nonlinear tests can be seen in Figure 17 and Figure 18 where the vertical dashed line represents the transition between training and estimating.

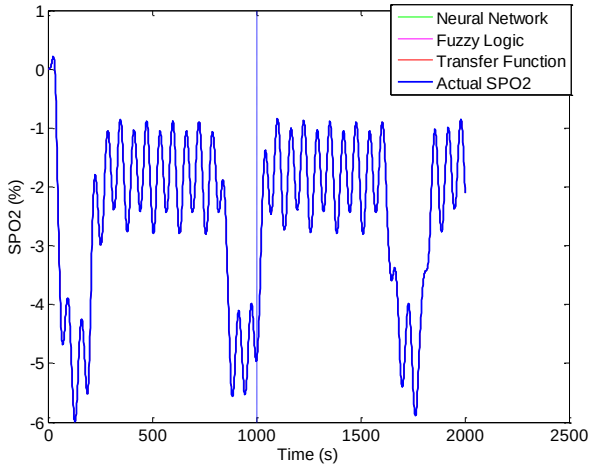

Figure 17. Static validation test for updating models were the vertical line indicate the end of the training window

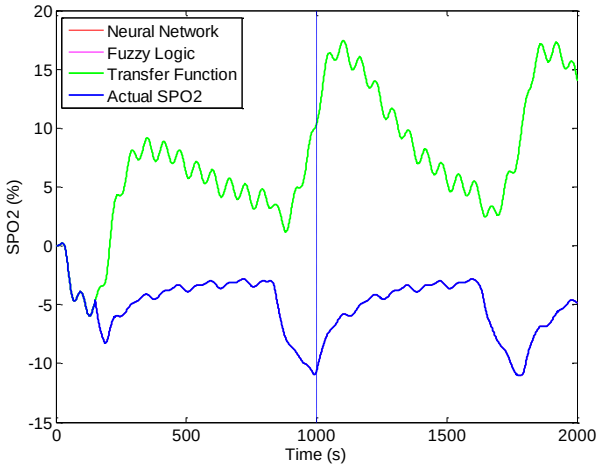

Figure 18. Nonlinear validation test for updating models were the vertical line indicate the end of the training window

The validation tests show that all of the updating models are able to adequately estimate future values if the system has static parameters. However for the variable parameters the transfer function model was not able to adjust to the changing parameters causing the model to poorly estimate the future SpO_2. The inability to model variable parameter data is to be expected since the model is linear. The neural network and fuzzy logic model though were able to adjust to the change in parameter and estimate the future values accurately.

3.1.5 Model Tuning

To tune the updating models to perform at their optimum, a method to quantify how long each model adequately represents the future SpO_2 values is needed. This method needed to identify the point where the model's SpO_2 finally diverges from actual SpO_2. The point where the model diverges will be called the holding time. The two parameters selected to identify the holding time are the overall error and the derivative error. The total error shows how well the model tracks the low frequency trends in the SpO_2. Derivative error shows how well the model tracked the changes in the SpO_2. To seamlessly mesh these two characteristics, a fuzzy error model was used not to be confused with the fuzzy logic model presented in Section 3.1.2. The fuzzy error model was created by setting up three membership functions for each input and output. The inputs to the error analysis are total error and derivative error. The output is the fitness value. A low fitness value indicates a close correlation between the simulated SpO_2 and the actual SpO_2. Each membership function was specified with three levels low, medium, and high. The membership function was chosen to be a triangle shape for ease of calculation and adjustability. Table 3 presents the values for each membership function. A rule base was set up for each membership function. An example of the rules is if the overall error is medium and the slope error is low, then fitness is high. The membership functions setup for the inputs and output can be seen in Figure 19. The overall error was found by calculating the average percent error for each interval. Where an interval starts from time 0 and goes to the current data point. The slope error was found using percent error of the slope at each point. The fitness value was found for each data point in the simulated data. The holding length of the model is found by finding the point where the

fitness value exceeds 0.2 and stays above 0.2 for 10 seconds or if the fitness value exceeds 0.5. One of the downfalls of the error analysis system is that it is not able to detect when the model becomes saturated. Saturation is when the model no longer respond to the inputs. To counteract this flaw the standard deviation of the past 5 points of the estimated SpO_2 was taken into account. If the standard deviation value dropped below 0.0001 the fitness value was adjusted to be 1. By changing the fitness value to 1 the fuzzy error model indicate that the model is no longer adequately modeling the SpO_2.

Table 3. Error analysis membership function values

	Low	Medium	High
Total Error	[0 – 0.08]	[0.075 – 0.1 - 0.163]	[0.12 – ∞]
Derivative Error	[0 – 0.02]	[0.018 – 0.04 – 0.07]	[0.06 - ∞]
Fitness	[0.597 - ∞]	[0.087 – 0.49 – 0.89]	[0 – 0.4]

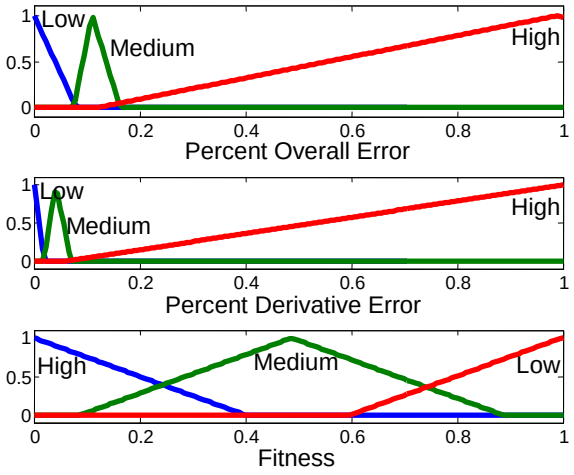

Figure 19. Fuzzy logic error membership functions. The y-axis is unitless

The tuning process, for each model, was done using a design of experiments (DOE) to test for the best design parameters that would produce the longest holding time. By using a design of experiments multiple variables were able to be changed for each

trial. This allowed for a reduction in the number of experiments run while still being able to find the best configuration of parameters. The parameters being tuned can be seen in Table 4. The DOE tested to see how each parameter affected the holding length of the model. The length of time the model held for was found by the analysis of model error as discussed above. The fuzzy logic and neural network models used an L_{18} DOE matrix. The transfer function model used an L_{16} DOE matrix [21]. The reason for the differences in the DOE matrix is due to the different number of parameters for each type of model. Each model was run two times, on the same starting time, with the parameters shifted. By shifting the parameters, the DOE was able to search more of the feature space. Each experiment was run on 10 randomly selected starting points and the results averaged thus minimized the randomness that is inherently built into each model. The results were analyzed and the best combinations of parameters were found. The results of the best combination of parameters can be seen Table 4. In Table 4, a dash line indicates the tuning parameter is not applicable to that model.

Table 4. Results from tuning DOE

Parameters	Fuzzy Logic	Neural Network	Transfer Function
FIO2 *MA* window	10	40	5
Change in FIO2 *MA* window	25	25	-
HR *MA* Window	50	50	35
Change in HR *MA* window	25	5	-
RR *MA* window	50	20	35
Change in RR *MA* window	25	5	-
Cluster Size	0.35	-	-
Number of neurons	-	25	-
Over training	0.2	0.2	-
Population	-	-	32
Generations	-	-	350

An optimum training window size was then needed for each model. The window size was found by running each model, with the optimized design parameters, over a large sample of starting points with different training window sizes. The model was tested using the analysis of model error as discussed above. The results were analyzed by looking at the average holding length and the standard deviation of the holding length for each training window length. The standard deviation was used to analyze the stability of the model. A model with a low standard deviation would produce a more stable model that would work over a broad range of situations. This trait was needed since the modeling system would be training online and not monitored. The best model was to have the highest average length and the lowest standard deviation. This assures that the model was stable and performed well over a large range of data. The best window size results for each model can be seen in Table 5. As shown in Table 5, the transfer function model performs the best with an average holding length of 43.2 time steps which corresponds to 51.8 seconds.

Table 5. Best Results from training window test optimization

Model	Best Training Window Size	Standard Deviation (s)	Average Holding Length (s)
Fuzzy Logic	100	16.3	43.4
Neural Network	100	12.5	36
Transfer Function	300	17.67	51.8

3.2: Dynamic Modeling

The motivation to investigating the parameter estimating transfer function model is to allow the transfer function model to better model nonlinear systems and reduce the downfalls in the transfer function model. It was shown in Section 3.1.4 that the transfer function model does well when trying to model the data when the system has fixed

parameters. However the system does very poorly when modeling nonlinear systems. The transfer function model also has problems with the training window caused by the system having no information on pervious data. The training window can cause situation where a step input can occur immediately before training window. The training process will either incorrectly assign the changes in the SpO_2 to another parameter or produces a poorly trained model. This problem is enhanced by the model normalizing the inputs and output data around 0. The normalization is done to allow the transfer function model to better model the data. In doing so all information outside the training window is erased. Methods of adjusting the training window size each time the model was trained was investigated but produced poor results. Training the model on all pervious data was also investigated. This however was not able to account for changes in the dynamic system and produce poor results. It is though that by creating a dynamic transfer function model the model will be able to model nonlinear systems and reduced if not eliminated the downfall discussed above. By eliminating the downfalls of the transfer function model, and allowing it to model nonlinear systems, the controller design based on the dynamic transfer function model will be based on a better estimate of the parameters and therefore be able to control the SpO_2 better.

3.2.1 Dynamic Transfer Function Model Procedure

The dynamic transfer function model was chosen for its ability to break the problem up into smaller sub problems allowing the program to be solved faster and more accurately. The basic idea is to have the model trained over the past 10 second of data. The model will then be moved up 5 seconds and retrained. In doing so a 5 second overlap

between the previous and the current data set is created. The overlap of 5 seconds was chosen through trial and error. This was found to be the overlap that allows the system to adapt to nonlinear systems while allowing the system to still converge.

The training process is done using a genetic algorithm. During the training process the error between the measured SpO_2 and the simulated SpO_2 is minimized. The error will be minimized by adjusting the gains and time constant of the FiO_2, HR, and RR transfer functions. The found parameters will be applied to first 5 seconds in the previous window of data. The structure of the model is the same as seen in Figure 13. The training process uses a similar genetic algorithm to find the gains and time constants as used in the Section 3.3. One of the changes is that the genetic algorithm uses elites selection rather than roulette wheel selection. Elites selection only selects the fittest individuals. The culling process also used elites selection. This type of selection was chosen to increase the convergence rate. However elites selection is more likely to prematurely convergence due to it decrease in diversity in the selection process. Another difference is how the fitness was found. The fitness first looked at the mean squared error between the SpO_2 and the simulated SpO_2 using Eq. (3.6). If the error is below 0.001 the fitness value is found by finding the error between the slope of the measured SpO_2 and simulated SpO_2 slope using equation Eq. (3.6). If the error is above 0.001the fitness value is the error between the measured SpO_2 and the simulated SpO_2. This system of fitness was chosen to help mitigate the overshoot problem which will be discussed later in this Section. The last major change in the genetic algorithm is the parameter boundaries. The parameter boundaries are kept the same as in Section 3.3 until time exceeds 100 seconds. After 100 seconds the boundaries are then reduced to be centered

on the pervious fittest individual. The gains range is ±0.5 and the time constants are ±10 sec from the fittest individuals parameters. This reduces the feature space allowing the genetic algorithm to converge faster.

To allow the model to have information about the past and to learn over time, three different methods were used. The first method used was to have the system run continuously causing the first data point in each window to have the initial conditions of the 5^{th} point in the previous window of data. Running the system continuously was done to counteract the training window problem as seen in Section 3.3. By doing this, the system is able to better track the SpO_2. This does cause some problems with over correction. Overcorrection is caused by the system never being able to be perfectly correct. Since the model always has some error, the model will try and correct itself to reduce the error in the model which can cause over correction. Over correction is where the model will continuously overshot the actual SpO_2 causing the simulated SpO_2 to continuously vary around the actual SpO_2 causing incorrect identification of the parameters. To reduce the effect of this, the fitness values attempted to minimize the error in the slope between the SpO_2 and the simulated SpO_2 when the error is less than 0.001. In doing so, the genetic algorithm is able focuses on modeling the dynamics of the system, rather than driving the error to zero, when the error is less than 0.001. The next method used was to pass the pervious population of individuals to the next optimization process. This not only increases the knowledge of the past but also allows the system to evolve and learn over time while speeding up the convergence process. The last method was to seed future populations with the predicated best individuals. Seeding the population was done by taking the last 10 best individuals from each parameter and

fitting a first-order polynomial to the data. Once the line was fit the next best parameter can be found through extrapolation. This individual then replaces the worst performing individual. The genetic algorithm can either choose to use the individual or not. This it will help lead to faster convergence and a better guided solution.

3.2.2 Tuning and validation of dynamic transfer function model

The genetic algorithm was tuned to allow the system to achieve the optimum performances. This was done using a design of experiments. The DOE used an L_9 matrix [21]. The parameters were the number of generations run and mutation rate. The range of generations was chosen to be 1, 2, and 3 generations. These values were chosen to allow the system to run in under 3 seconds. Three seconds was chosen to give the program a 2 second buffer before it would have to make any other calculations. The mutation rate was chosen to be 0.05, 0.2, and 0.4. These values were chosen to span the range or reasonable mutations rates. The DOE tested to see how well the time constants and gains could be estimated. The nonlinear data generated in Section 3.1.4 was used to for the DOE tests. To quantify how well the dynamic transfer function tracked the gains and time constants, the error between estimated parameter and the actual parameter was found. The error was normalized and the overall average error was found. The results of the DOE can be seen in Figure 20. The DOE shows that the best number of generations is 2 with a mutation rate of 0.2.

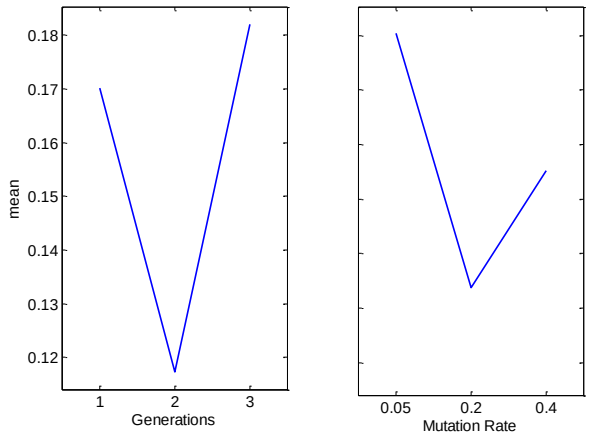

Figure 20. Dynamic Transfer function DOE

To validate that the program can model fixed and variable parameters systems and estimate the gains and time constants correctly the model was run on the generated data in Section 3.1.4. The results from the validation tests can be seen in Figure 21 - 23. Where the red lines in Figure 21 and Figure 22 are the real gain and time constants, the green circle represent the actual estimate, and the blue x's represent the average estimate. The validation tests showed that the dynamic transfer function does well with tracking the gains and modeling the SpO_2 but has trouble with tracking the time constants. The reason for this is that the system is not fully excited. When the system is not fully excited the first order transfer functions act as a gain. When not fully excited, the time constants have very little effect on the output causing the estimated time constant to wander. The model also has some trouble when the gains and time constants are changed. The model takes about 200 seconds to start tracking the SpO_2 again. However the average gain and time constant during this time is estimated within acceptable error. This shows that by making transfer function model dynamic that performance is drastically increased in modeling nonlinear systems.

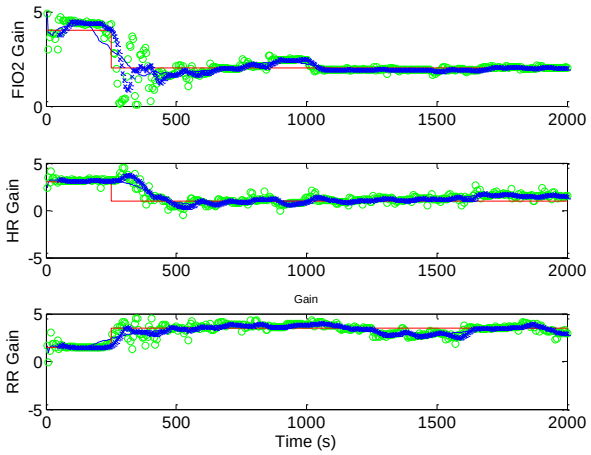

Figure 21. Dynamic Transfer Function Estimated Gains

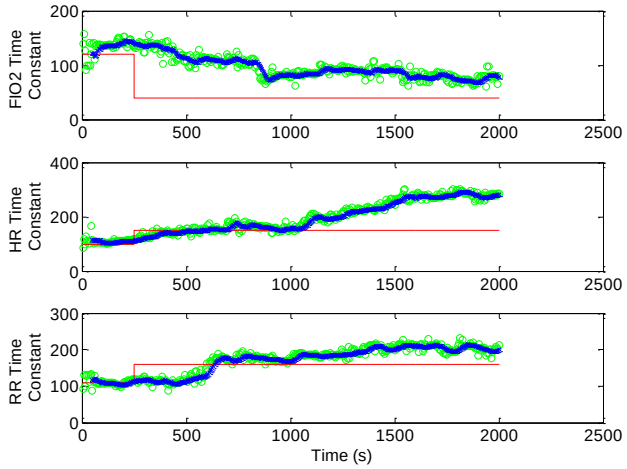

Figure 22. Dynamic Transfer function estimated time constants

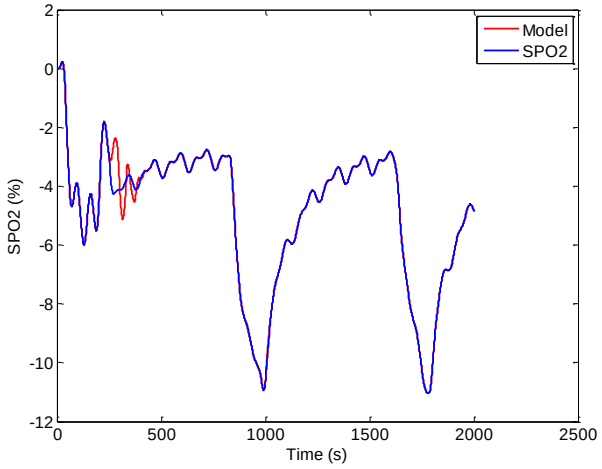

Figure 23. Dynamic Transfer function Estimated SpO_2

3.3 Analysis and Conclusion

To select which model performs the best on real data each model was tested on a large set of data and analyzed to see which model performances the best. One randomly selected data set was selected to be plotted for visual analysis. The models were trained on the appropriate data window and then tested on the next 1000 seconds. The best model will be selected to be the model that adequately represents the future SpO_2 for the longest period of time. The vertical line shown in Figure 24 through Figure 29 represents the end of the training window.

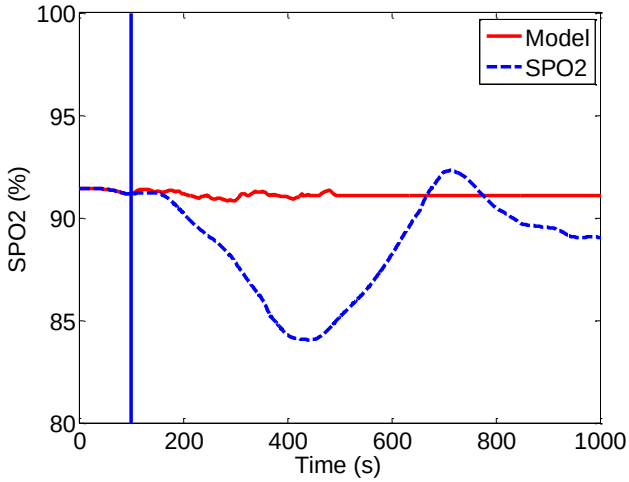

Figure 24. Neural network model example

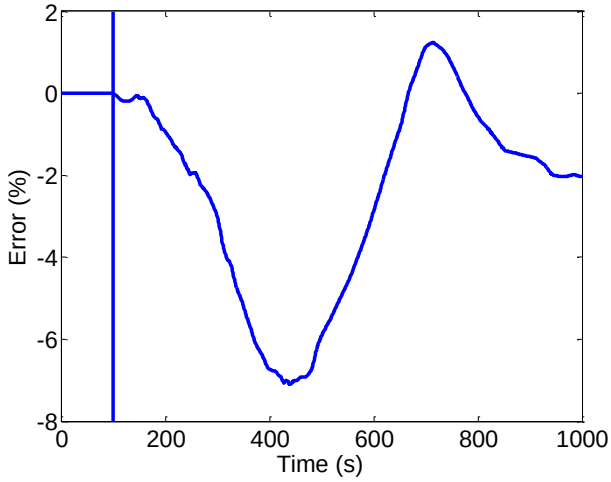

Figure 25. Neural network model error

The results from the neural network model can be seen in Figure 24. The neural network model on average holds for 36 seconds. Through visual analysis of the error shown in Figure 25 the model holds through data point 162 which corresponds to 74.4 seconds. Figure 24 shows how the neural network tends to stay within the training bound of the training windows SpO_2 and cannot extrapolate. The inability to extrapolate is caused by sigmoid saturation problem discussed in Section 3.1.1 results in poor tracking of the SpO_2 when modeling data outside the training bounds.

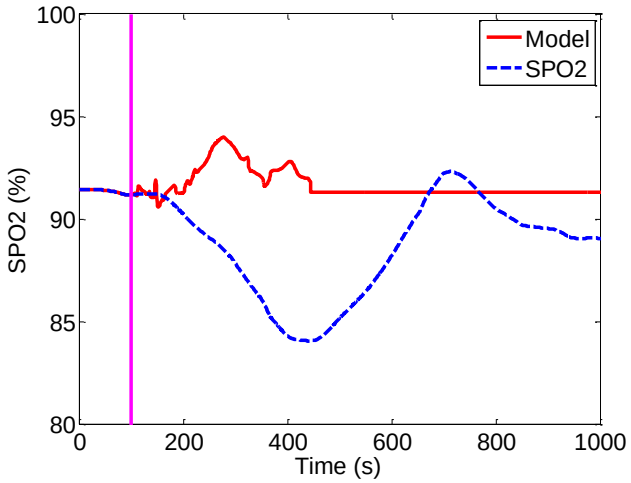

Figure 26. Fuzzy logic model example

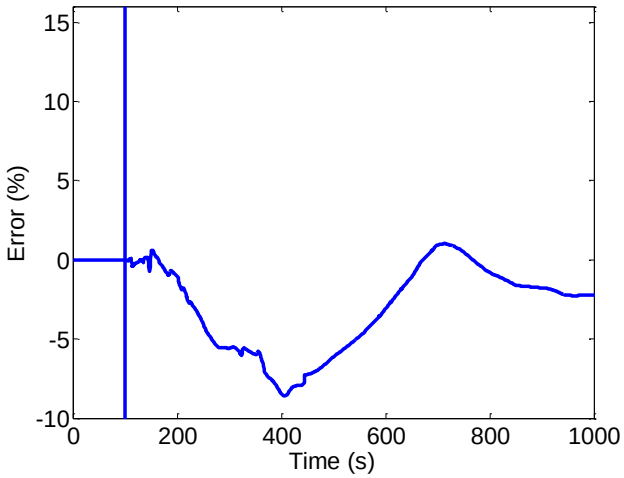

Figure 27. Fuzzy logic model error

The results for the fuzzy logic model can be seen in Figure 26. The analysis of model error reported that the fuzzy logic model holds on average for 43.4 seconds. Through visual analysis of Figure 27 the model holds for 40.8 seconds. The fuzzy logic model generally does well simulating the SpO_2 while the inputs stay within the models bounds. Once the model exceeds the bounds the model extrapolates in an attempt to model the data. This can cause an instability problem in the modeling process, which results in extremely poor modeling.

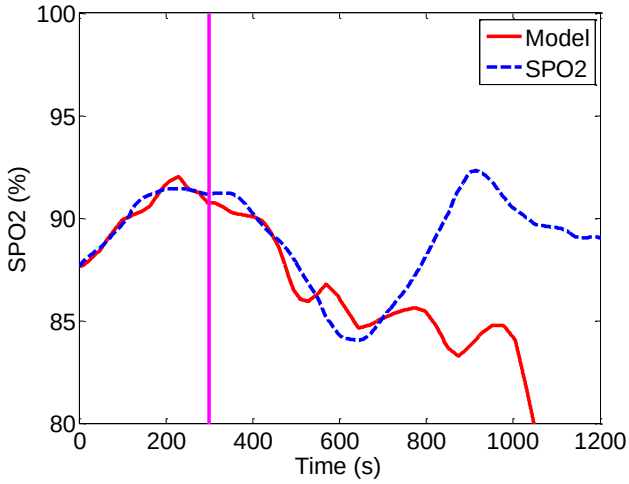

Figure 28. Transfer Function Model Example

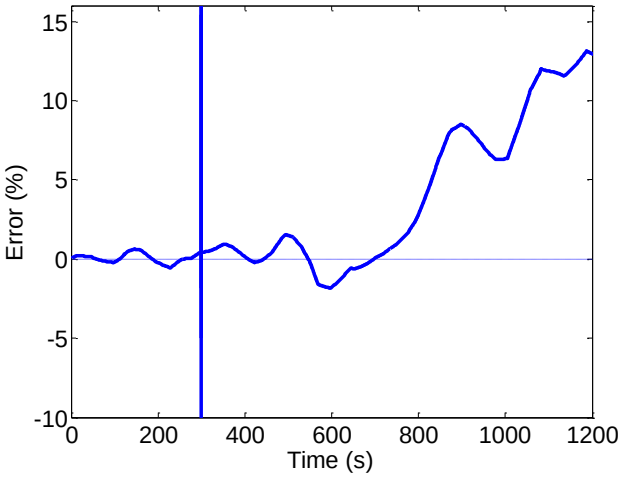

Figure 29. Transfer Function Model Error

The results from the transfer function example can be seen in Figure 28 were the models parameters are fixed after the training period. On average the transfer function model held for 51.8 seconds. However through visual analysis of the error shown in Figure 29, the model actually appears to hold until data point 700 which corresponds to 480 seconds. The model does not have problems with saturations and has demonstrated the ability to accurately deal with data outside of the training window bounds. The model however is not able to take into account the nonlinearities in the system.

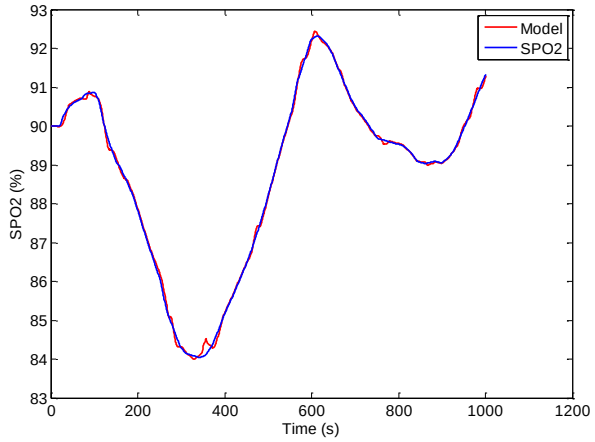

Figure 30. Dynamic Transfer function SpO_2

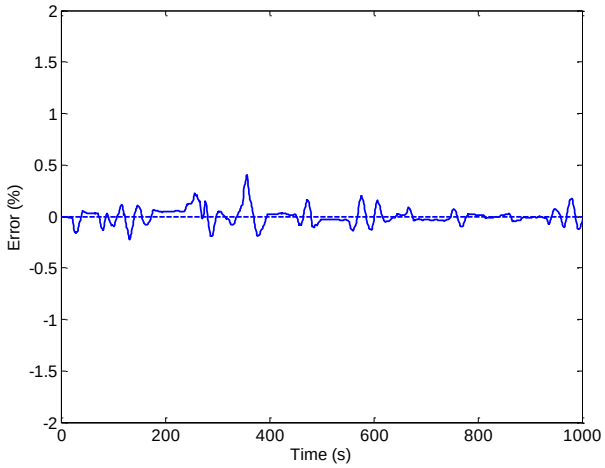

Figure 31. Dynamic Transfer function SpO_2 error

The results from the dynamic transfer function test can be seen in Figure 30. It was reported that the model on average holds for 1000 seconds which corresponds to the maximum holding time tested for. It is though that the model would be able to hold for longer periods of time. This is validated by Figure 31 where the maximum error is 0.4 and holds for the full 1000 seconds. The dynamic transfer function model is able to model the SpO_2 accurately for a prolonged period of time. The model is also shows that the model can correct itself if it started to get off track. This can be seen at time 350 in Figure 30 were the model start to deviate from the SpO_2. The model quickly responds and corrects itself by time 400 showing that the model is able to take into account the nonlinearities of them system and model them adequately.

Through the testing process the best model was found to be the dynamic transfer function model. The model on average is able to adequately model the SpO_2 for 1000 seconds or greater due to the model being continually updated. It was shown that the model is reliable over many situations and can adequately estimate the gains and time constants in simulated data. The downfall of the model is that when the system is not fully excited the model cannot accurately estimate the time constants. Due to the drastic increase in performance the dynamic transfer function model will be the only model a controller is designed for in the next chapter.

Chapter 4: Controller Design and Analysis

Automatic control of the FiO_2 administered to a neonatal infant to regulate the SpO_2 is investigated in this paper. It is thought that automatically regulating the FiO_2 will lead to better control of the infants SpO_2 and reduction in the work load of the nurses. Some research has been done on controllers for automatic oxygen saturation control systems but none have included the effect of heart rate and respiratory rate on the SpO_2. As shown in Chapter 2, some fluctuations in the SpO_2 can be attributed to changes in the heart rate and respiratory rate. Since these two parameters can be measured but not controlled they will be considered disturbances in the system. By understanding how these disturbances will affect the SpO_2 a controller can be designed that will mitigate the affect of the disturbances on the infants SpO_2.

Three design characteristics were taken into account when selecting the controller. The controller will need to be automatically updated every time the model is updated, insure stability when updated, and reject disturbances. The controllers that fulfilled these design requirements are a robust controllers and an adaptive controller with feed forward disturbance rejection. A linear quadratic regulator PI controller was also designed for comparison purposes. All controllers will be designed using the dynamic transfer function model.

4.1 Data Collection

To design a static controller, the recorded infant data was analyzed to find the expected ranges of gains and time constants for the FiO_2, HR, and RR transfer functions. This was done by running the dynamic transfer function algorithm over 108 hours of

data. A moving average was applied to the all inputs and outputs. This was done to smooth the data and remove some noise in the system. The filter window size for the FiO_2, HR, and RR data was set to 10 and the SpO_2 window was set to 20. These values were selected through visual analysis to minimize the noise while still adequately representing the data. A plot of the actual and filtered data can be seen in Figure 32 where the blue lines indicate the actual data and the red lines the filtered data. While the model was running the gains and time constants were estimated and recorded every 5 seconds. An example of the estimated SpO_2, gains, and time constants can be seen in Figure 33 through Figure 35. In Figure 34 and Figure 35 the blue circle represent the average estimated value and the green circles represent the actual estimated value. To allow the analysis to only consider estimations that are good all estimates that have an error of 0.1 or greater were removed. A histogram of the estimated gains and time constants can be seen in Figure 36 and Figure 37. In Figure 36 and Figure 37 the red lines indicate the mean value, green lines one standard deviation from the mean, and purple line two standard deviations from the mean.

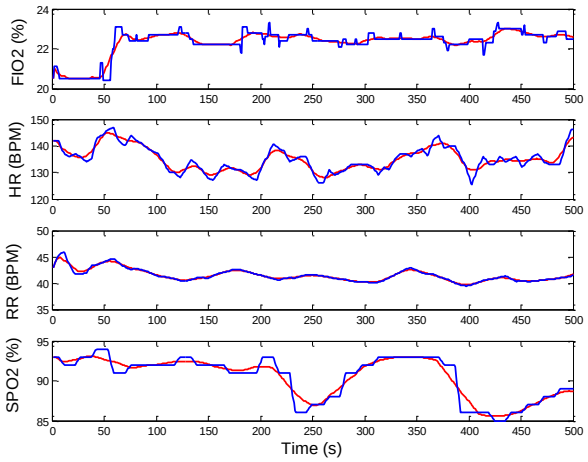

Figure 32 Filters FiO_2, HR, RR, and SpO_2. Where the blue line indicates the actual data the red line indicates the filtered data

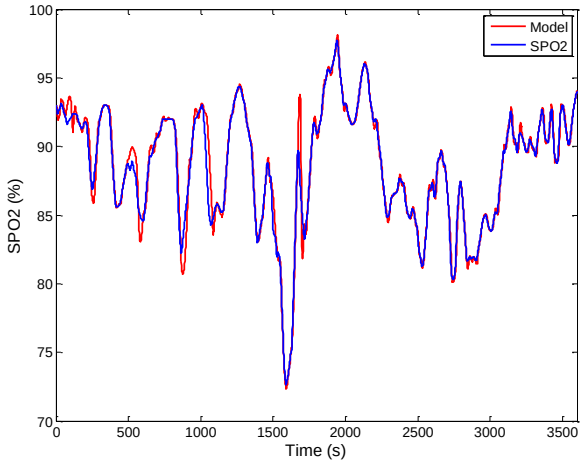

Figure 33. Hour long segment of estimated SpO_2

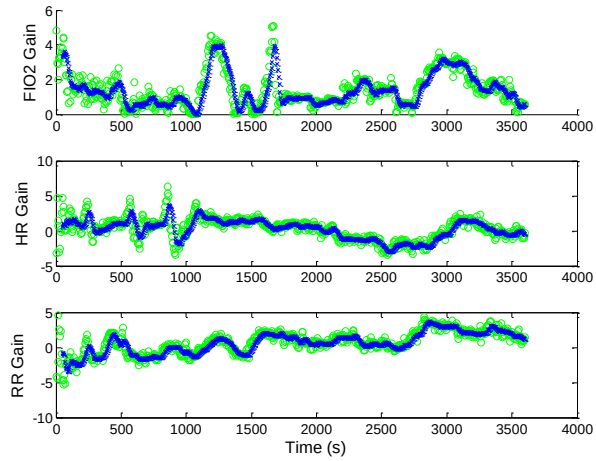

Figure 34. Estimated gain for hour long test segment. Green circles indicate the actual estimate and blue circles indicate the average estimate

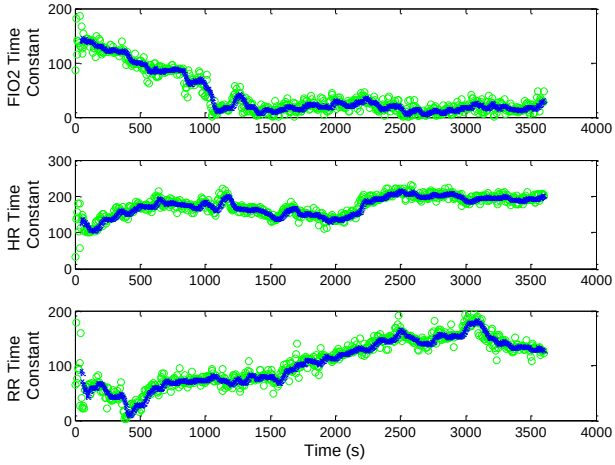

Figure 35. Estimated time constants for hour long test segment

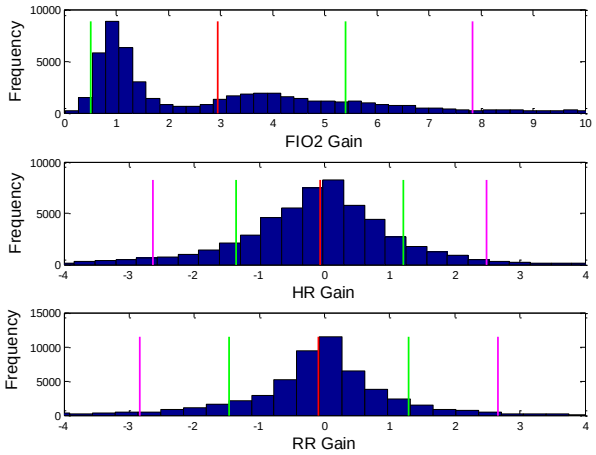

Figure 36 Estimated Gains

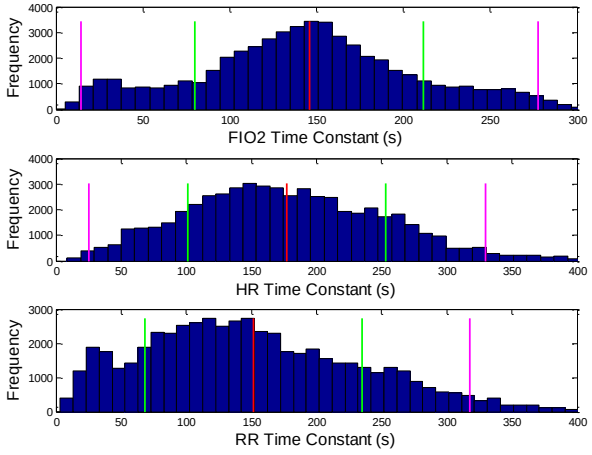

Figure 37 Estimated Time Constants

Through analyzing the gain and time constant histogram some vital insight into the infant's system can be seen. The gain for the FiO_2 was shown to have two distinct peaks at 0.927 and 3.77. The HR and RR gains however were centered around zero causing the gains to be negative and positive. The zero mean is most likely caused by the lag in the biological system. Since there is no lag built into the model the controllers compensate for this by assigning the negative gains. For the time constant histograms it is shown that there is only one peak for each parameter, showing that there is a most

common time constant for each of the parameters across a broad range of infants. The mean value and the standard deviation, found from the histogram, can be seen in Table 6.

Table 6. Gains, Time Constant, and standard deviation of estimated parameters from histogram

	Gain		Time Constant (s)	
	Mean	Standard Deviation	Mean	Standard Deviation
FiO_2	2.957	2.444	145.9	65.9
HR	-0.07	1.282	177.2	76.1
RR	-0.8929	1.3767	151.3	83.3

4.2 Linear Quadratic Regulator PI Controller

Linear Quadratic Regulator controller was chosen for the ability to balance the desired closed – loop response characteristics and control effort. Since the controller can only control the FiO_2, the HR and RR will be considered disturbances into the system. The controller was designed by selecting gains that minimizes a specified performance criterion, which is given as

$$V_{\infty} = \frac{1}{2}\int_{t}\left[x^{'}Qz + u^{'}Ru \right]dt \tag{4.1}$$

Where Q is a state weighting matrix which penalizes the states deviation from the origin, and R is the controller weighting matrix which represents the cost of control [22]. The Q matrix can also be thought of as the weighting matrix that specifies the importance of each state compared to the other. Currently the LQR can only produce a proportional controller. The proportional will therefore always produce a steady state offset. To drive the system to have zero steady state error an integral control is needed. The integral controller is added by appending an integrator to the plant before computing the feedback

gain. To do this the integral error between the reference point and the reference input is given as,

$$\dot{e}_I = r - y(t) = r - C_y x(t) \tag{4.2}$$

Where r is the reference input, $y(t)$ is reference output, C_y is the output matrix for the output y, and $x(t)$ is the state. The plant state equation is then augmented with Eq. (4.2), where the plant state equation is give as,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y &= Cx(t) \\ A &= -0.0069 \\ B &= 1 \\ C &= 0.0203 \end{aligned} \tag{4.3}$$

Where A and B are the state and input parameters. The augmented state equation is then

$$\begin{bmatrix} \dot{x} \\ \cdots \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & \vdots & 0 \\ \cdots & \cdots & \cdots \\ C_y & \vdots & 0 \end{bmatrix} \begin{bmatrix} B \\ \cdots \\ 0 \end{bmatrix} + \begin{bmatrix} B \\ \cdots \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \cdots \\ I \end{bmatrix} r \tag{4.4}$$

Where C_y is the output matrix. LQR theory can be used to generate the state feedback for the augmented plant [23]. To do this the reference input is ignored and a cost function that penalizes the integral of error is used instead, which can be seen in Eq. (4.5).

$$J = \int_0^{t_f} e_I^T(t) e_I(t) + u^T R u(t) dt \tag{4.5}$$

The optimal control written in terms of tracking error is given as,

$$u(t) = -K_p(t)x(t) - K_I(t)\int_0^t \{r - y(t)\}\, dt \tag{4.6}$$

Where K_p is the proportional gain and K_I is the integral gain. To solve for the system gains, the Riccati solution (P) needs to be solved. This can be done by solving the Algebraic Riccati Equation (ARE) equation for P. The ARE equation can be seen in Eq. (4.7).

$$PA + A^T B_U R^{-1} B_U^T P + Q = 0 \tag{4.7}$$

The steady-state feedback gain matrix can then be found using Eq. (4.8).

$$K = R^{-1} B^T P \tag{4.8}$$

In Eq. (4.8) K is a matrix consisting of K_I and K_P. The values for Q and R were selected through trial and error to find the values that minimized the control effort while maximized the disturbance rejection. The final selection of Q and R can be seen in Eq. (4.9) and Eq. (4.10).

$$Q = \begin{bmatrix} 20 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{4.9}$$

$$R = 1 \tag{4.10}$$

A controller was then designed using the FiO_2 parameters shown in Table 6. The proportional gain was found to be 4.464 and the integral gains as 0.01.

The closed-loop system is tested for nominal stability and robust stability. Nominal stability is found by finding the poles and zeros of the closed-loop system. If all the poles and zeros lie on the right half of the complex plane the system is considered to

be nominally stable. Nominal stability was checked for the system and all poles were on the left-half plane thus achieving nominal stability. Robust stability is needed to assure stability over a range of gain uncertainty. This is important due to the continually changing nature of the system. To check for robust stability the stability range in the gain was determined from the Nyquist plot of the nominal loop transfer function, which is given as,

$$G_l(s) = P_{FIO2}(s) * K(s) \tag{4.11}$$

Robust stability is check by finding the gain margin. The gain margin is found by finding the points where the Nyquist plot cross the real axis. These points can be seen in Figure 38 as -*b* and –a. The gain margin (GM^+) and downside gain margin (GM^-) are then found using Eq. (4.12) and Eq. (4.13).

$$GM^+ = \frac{1}{a} \tag{4.12}$$

$$GM^- = \frac{1}{b} \tag{4.13}$$

The range of gains that guarantees stability are then

$$\left(GM^- \quad , \quad GM^+\right). \tag{4.14}$$

The stability range for the closed loop system was found to be (0,∞). This shows that the system is stable as long as the gains are positive. Since the dynamic transfer function model is not allowed to estimated negative gains for the FiO_2 the system will be stable

across all ranges of uncertainty. Figure 40 shows the structure for the (LQR-PI) controller.

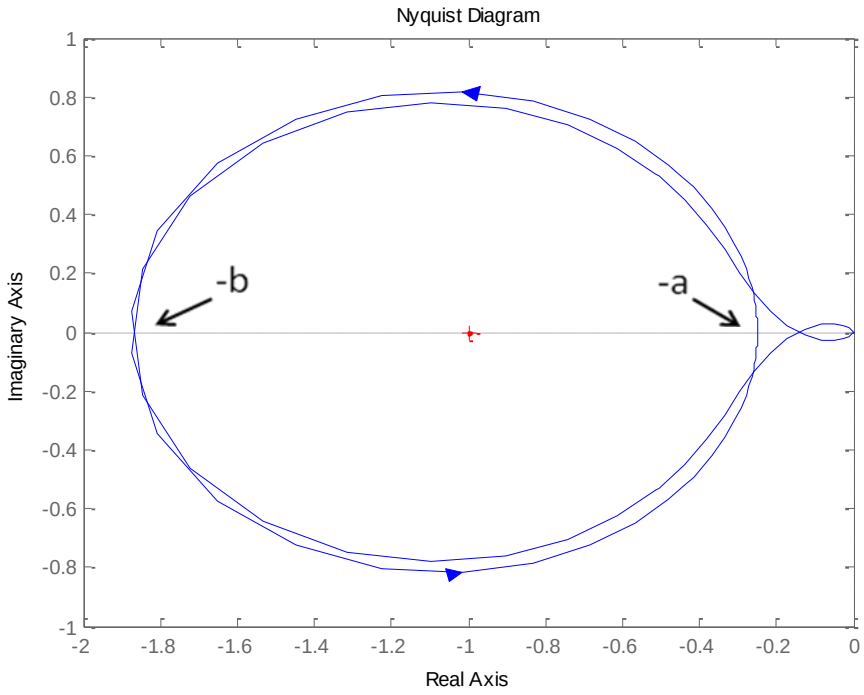

Figure 38. Example diagram of a Nyquist plot

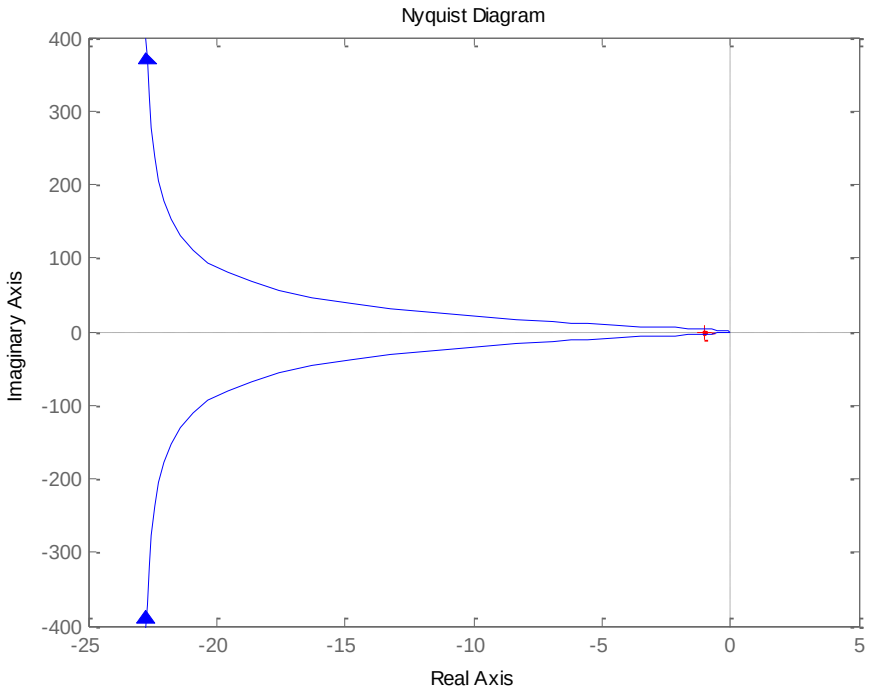

Figure 39. Nyquist plot of the closed-loop system.

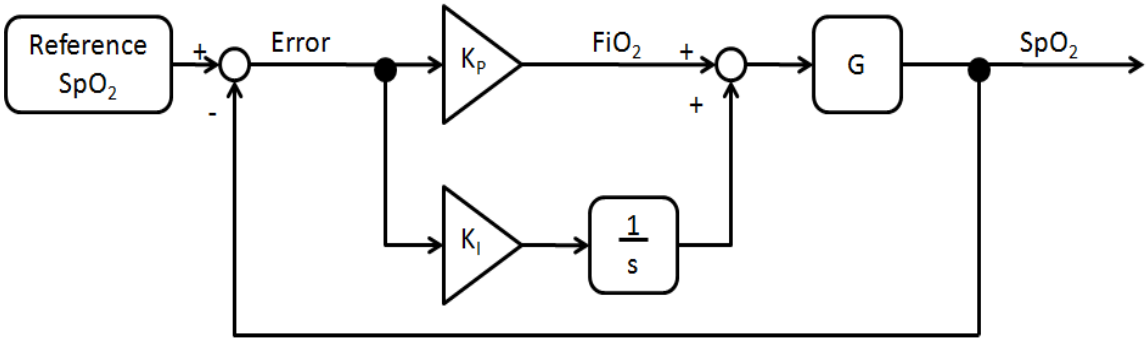

Figure 40. Block diagram of pi controller

4.3 Robust Controller

A robust controller was designed to take the uncertainties in the gain and time constant into consideration. Taking the uncertainties into consideration is needed to ensure stability and performance over the range of uncertainties that are inherently in the modeling process and the constantly changing parameters of the infant. A μ-synthesis

controller design procedure was chosen to obtain robust performance with uncertainties in the model. The uncertainties in the model are the gains and time constant of the FiO_2, HR, and RR transfer functions. The controllers will be designed to be disturbance rejection controllers. The controller will try to reject the HR and RR effect on the SpO_2. To allow the controller to be designed properly, the input to the HR and RR were scaled to have all disturbances into the system be less than 1.0. The scaling factor (*d*) for the HR and RR was 1/30. Figure 41 shows a block diagram of the model with uncertainty included.

The first step to designing a controller is to define the performance weights. Two performance weights were placed on the controller. The first one is the performance weight, $w_p(s)$, which corresponds to the maximum allowed error at high and low frequencies. The bandwidth, w_b, was adjusted to achieve the desired performance for each controller. The maximum error at lower frequencies was set to be 5% and at higher frequencies was set to 300%. The performance weight is

$$w_p(s) = \frac{\frac{1}{3}s + w_b}{s + 0.05w_b} \tag{4.15}$$

The second performance weight, $w_u(s)$, is placed on the control signal. This is done to limit the maximum controller effort value during the design process. The $w_u(s)$ is set to 1/0.79 so the maximum control signal value does not exceed 100% oxygen where ambient oxygen is 21%.

Now that the performance is defined, the transfer function gains and time constants for the FiO_2 are split up into groups to allow the controller to perform

adequately based on the specified design requirements via a controller scheduling scheme (i.e. multiple controllers will be designed for use depending on which gains and time constants are estimated). The range of gains and time constants for the HR and RR is the same for any FiO_2 level. Keeping the range the same was done to reduce the amount of controllers that needed to be designed. The range for the HR and RR took into account 68% (one standard deviation) of the gain and time constants show in Figure 36 and Figure 37. The range for the HR and RR can be seen in Table 7. For the FiO_2, the gains and time constants were broken up into 6 different groups. Different groups were need to allow the controllers to achieve the specified performance while still covering the range of all reasonable gains and time constants. The range of the gains, time constants, and mean value for each group can be seen in Table 8.

Table 7. Range of HR and RR gains and time constants for H_∞ controller

HR				RR			
Gain		Time Constant (s)		Gain		Time Constant (s)	
Range	Nominal Value	Range	Nominal Value	Range	Nominal Value	Range	Nominal Value
-1.35 – 1.2	-0.073	101.17 – 253.3	177.2	-1.466 – 1.29	-0.09	68 – 234.7	151.3

Table 8. Range of FiO_2 transfer function gains and time constants for H_∞ controller

	Gain		Time Constant (s)	
	Range	Nominal Value	Range	Nominal Value
1	0.25 - 3	2	70.9 - 120	95
2	0.25 - 3	2	120 - 170	145
3	0.25 - 3	2	170 - 220	195
4	3 - 8	5.5	70.9 - 120	95
5	3 - 8	5.5	120 - 170	145
6	3 - 8	5.5	170 - 220	195

The uncertainty in FiO_2, HR, and RR plant transfer functions was then found using multiplicative uncertainty. The multiplicative uncertainty error is defined as,

$$E(j\omega) = \left| \frac{G_{per}(j\omega) - G_{nom}(j\omega)}{G_{nom}(j\omega)} \right| \tag{4.16}$$

where $G_{per}(j\omega)$ are the perturbations from the nominal plant and $G_{nom}(j\omega)$ is the nominal plant. The nominal plant is a first order transfer function with the gains and time constants being the mean value of the selected plant. Multiplicative uncertainty modeling is performed on the FiO_2, HR, and RR plant. A weighting uncertainty, W_{xx}, was found using an automated transfer function fitting process for each transfer function, with the subscript (xx) as FiO_2, HR, and RR. This is a transfer function that bounds the maximum error, $|E(j\omega)|$, for all frequencies and all possible $G_{per}(j\omega)$.

Once all values have been found, the block diagram shown in Figure 41 can be reduced into the generalized plant *P*. Where the generalized plant and the four partitioned plants are given as,

$$\begin{bmatrix} u_{\Delta 1} \\ u_{\Delta 2} \\ u_{\Delta 3} \\ Z_1 \\ Z_2 \\ u \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & W_{FiO_2} \\ 0 & 0 & 0 & 0 & \frac{W_{RR}}{d} & 0 \\ 0 & 0 & 0 & \frac{W_{HR}}{d} & 0 & 0 \\ -W_P G_{FiO_2} & -W_P G_{RR} & -W_P G_{HR} & \frac{-W_P G_{HR}}{d} & \frac{-W_P G_{RR}}{d} & -W_p G_{FiO_2} \\ 0 & 0 & 0 & 0 & 0 & W_U \\ -G_{FiO_2} & -G_{RR} & -G_{HR} & \frac{-G_{HR}}{d} & \frac{-G_{RR}}{d} & -G_{FiO_2} \end{bmatrix} \begin{bmatrix} y_{\Delta 1} \\ y_{\Delta 2} \\ y_{\Delta 3} \\ d_1 \\ d_2 \\ v \end{bmatrix}$$

$$P_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{W_{RR}}{d} \\ 0 & 0 & 0 & \frac{W_{HR}}{d} & 0 \\ -W_P G_{FiO_2} & -W_P G_{RR} & -W_P G_{HR} & \frac{-W_P G_{HR}}{d} & \frac{-W_P G_{RR}}{d} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P_{12} = \begin{bmatrix} w_{FiO_2} \\ 0 \\ 0 \\ -W_p G_{FiO_2} \\ W_U \end{bmatrix}$$

$$P_{21} = \begin{bmatrix} -G_{FiO_2} & -G_{RR} & -G_{HR} & \frac{-G_{HR}}{d} & \frac{-G_{RR}}{d} \end{bmatrix}$$

$$P_{22} = \begin{bmatrix} -G_{FiO_2} \end{bmatrix} \qquad (4.17)$$

Where *G* and *W* are the nominal plant and error for the HR, RR, and FiO_2. A μ-synthesis controller was designed for each range of FiO_2 using the generalized plant *P*. The design of the controller was done by the program dksny.m in matlab. This program used D-K iteration to create a robust controller such that the system is stable and achieves the performance requirements for all perturbations of the plant. The D-K procedure is an approximation of the μ-synthesis controller design [24].

Each controller is checked for nominal stability, nominal performance, robust stability, and robust performance. This is needed to make sure the controller is stable and performs well over the range of uncertainties. Nominal stability tests to see if the controller is stable under nominal conditions, i.e. no plant perturbations. This is done by checking if there are any right half plane poles of the nominal close loop system. Nominal performance checks to see if the controller performs according to the performance criteria under nominal conditions. The test for nominal performance is given as,

$$NP = \left| W_P(j\omega)\left(\frac{1}{1+KG_{FIO2}} \right) \right|_\infty \tag{4.18}$$

Where K is the controller being tested. To achieve nominal performance the maximum value of Eq. (4.18) must be less than 1. Robust stability checks to see if the controller is stable over the range of uncertainties. The equation for robust stability is

$$RS = \left| W_i(j\omega)\left(\frac{KG_{FIO2}}{1+KG_{FIO2}} \right) \right|_\infty \tag{4.19}$$

To achieve robust stability the maximum value from Eq. (4.19) must be less than 1. Robust performance checks to see if the controller performs according to the performance criteria over a range of inputs. Robust performance can be checked using Eq. (4.20).

$$RP = \left| W_P(j\omega)\left(\frac{1}{1+KG_{FIO2}} \right) \right|_\infty + \left| W_i(j\omega)\left(\frac{KG_{FIO2}}{1+KG_{FIO2}} \right) \right|_\infty \tag{4.20}$$

To achieve robust performance the maximum value of Eq. (4.20) must be less than 1. The values found for nominal stability, nominal performance, robust stability, robust performance, and the bandwidth (ω_b) used to design each of the six controllers can be seen in Table 9. In Table 9, the values in the nominal stability column represent the number of right half plane poles. Table 9 shows that all controllers achieve nominal stability, nominal performance, robust stability, and robust performance.

Table 9. Stability and performance values for all μ synthesis controllers

Controller	nominal stability	nominal performance	robust stability	robust performance	Bandwidth W_b (Hz)
1	0	0.335	0.882	0.969	0.0125
2	0	0.335	0.875	0.946	0.006
3	0	0.335	0.874	0.944	0.0051
4	0	0.34	0.837	0.882	0.0105
5	0	0.336	0.698	0.741	0.0069
6	0	0.336	0.629	0.667	0.0051

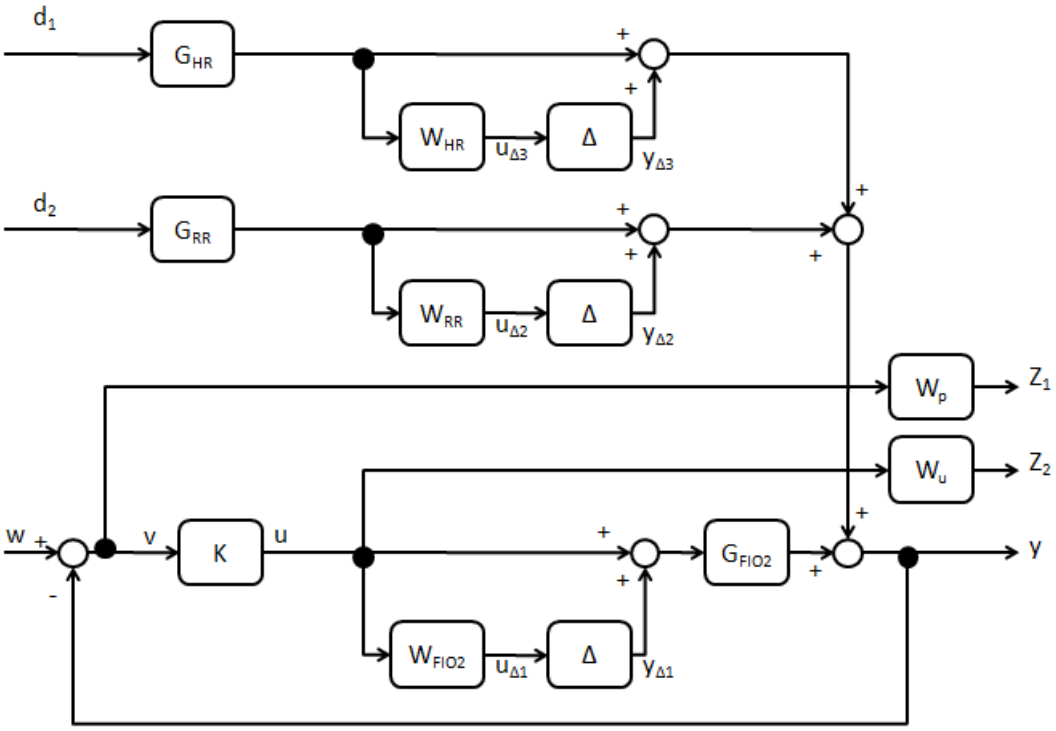

Figure 41. Block diagram of controller with uncertainty analysis

To allow the system to know which of the six controllers is needed to be selected, the dynamic transfer function model is used to estimate the gains and time constants. The estimation is done by having the dynamic transfer function model estimate the gains and time constants of the FiO_2 every 5 seconds. These values are analyzed and the appropriate controller is selected. To prevent a bump in the control effort when the controllers are changed, a bumpless transfer system is used. This system is designed to reduce if not eliminate a spike in the control effort when switching controllers. A block diagram of the system can be seen in Figure 42. This system works by using an integrator to add up all the changes in the system. The system will give a smooth transition between the controllers provided the switch is made when the feedback error is zero [25].

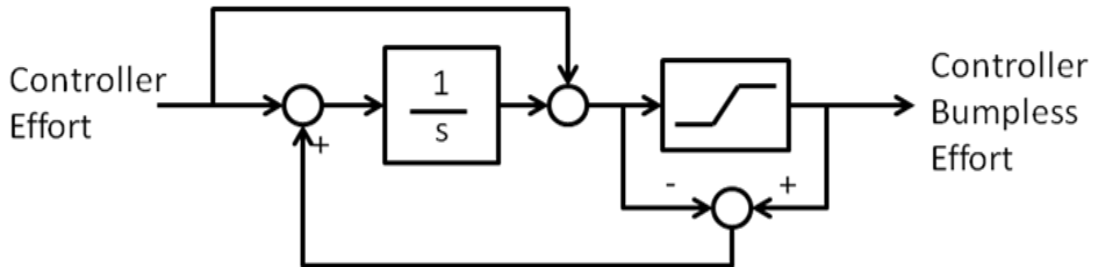

Figure 42. Block diagram of bumpless transfer system

4.4 Adaptive Controller with Feed Forward Disturbance Rejection

This type of controller was chosen for its ability to analyze the disturbance and choose to either reject the disturbance or allow the disturbance to drive the system to the set point. The disturbances in the system are considered to be the HR and RR. The system consisted of three controllers; a feedback controller for the FiO_2 and a feed forward controller on the HR and RR. The FiO_2 controller is used to minimize the error between the set point and the actual SpO_2. This is done through regulating the supplied FiO_2. The set point for the system was set to be 90% SpO_2, which is a safe level for neonatal infants. To adjust this to work best with a transfer function model, 90% was subtracted from the set point making the new set point 0%. To find the error the SpO_2 is subtracted from the set point creating the error signal. The value is then feed into the FiO_2 controller. The FiO_2 controller uses proportional gain controllers. The proportional gain value is given as,

$$K_{FIO2} = \frac{\left(\frac{1}{e_{ss}}\right) - 1}{G_{FIO2}} \tag{4.21}$$

where e_{ss} is the maximum allowable steady state error and G_{FIO2} is the estimated FiO_2 plant gain [2]. The acceptable steady state error was chosen to be 5%.

For the disturbances, a feed-forward disturbance controller was used to reject the disturbances caused by the heart rate and respiratory rate. The disturbance controllers are designed to only reject the disturbance if the disturbance is increasing the error. This allows the system to use its own disturbances to drive the error of the system to zero thus reducing the control effort needed. The decision to either reject the disturbances or use the disturbance was done by comparing the error signal and the derivative of the estimated SpO_2 disturbance. The estimated SpO_2 disturbance was found by taking the HR and RR values and inputting them into the estimated first order transfer function for the HR and RR found by the dynamic transfer function model. The derivative of the estimated SpO_2 disturbance was chosen to allow the system to know if disturbance is driving the system away or toward the set point. The decision to accept or reject the disturbance was given as,

$$D_s = \begin{cases} \dot{D} & \text{if} \quad \left|Error + \dot{D}\right| < \left|Error\right| \\ 0 & \text{if} \quad \left|Error + \dot{D}\right| \geq \left|Error\right| \end{cases} \tag{4.22}$$

where $\dot{D}$ is the derivative of the estimated disturbance SpO_2. The disturbance rejection signal, $\dot{D}$, is then multiplied by the disturbance controller gain K_D. There are two disturbance controller gains K_{Dhr} and K_{Drr} for the *HR* and *RR*. The value for K_{Dhr} and K_{Drr} are set to a constant value of 100. This value was found through trial and error balancing the need for stability and performance. The two disturbance controllers are added together to create the disturbance control effort. This value is feed through a mean value block with a window size of 5 seconds and then into a rate limiter with slew rate of

±0.7 FiO_2. The rate limiter reduces the high frequency oscillations that can occur in the disturbance effort. The disturbance control effort is then subtracted from the FiO_2 creating the total control effort. The block diagram for the controller can be seen in Figure 43. All controllers and estimated parameters are updated every 5 seconds. The estimation parameters are found by the dynamic transfer function model.

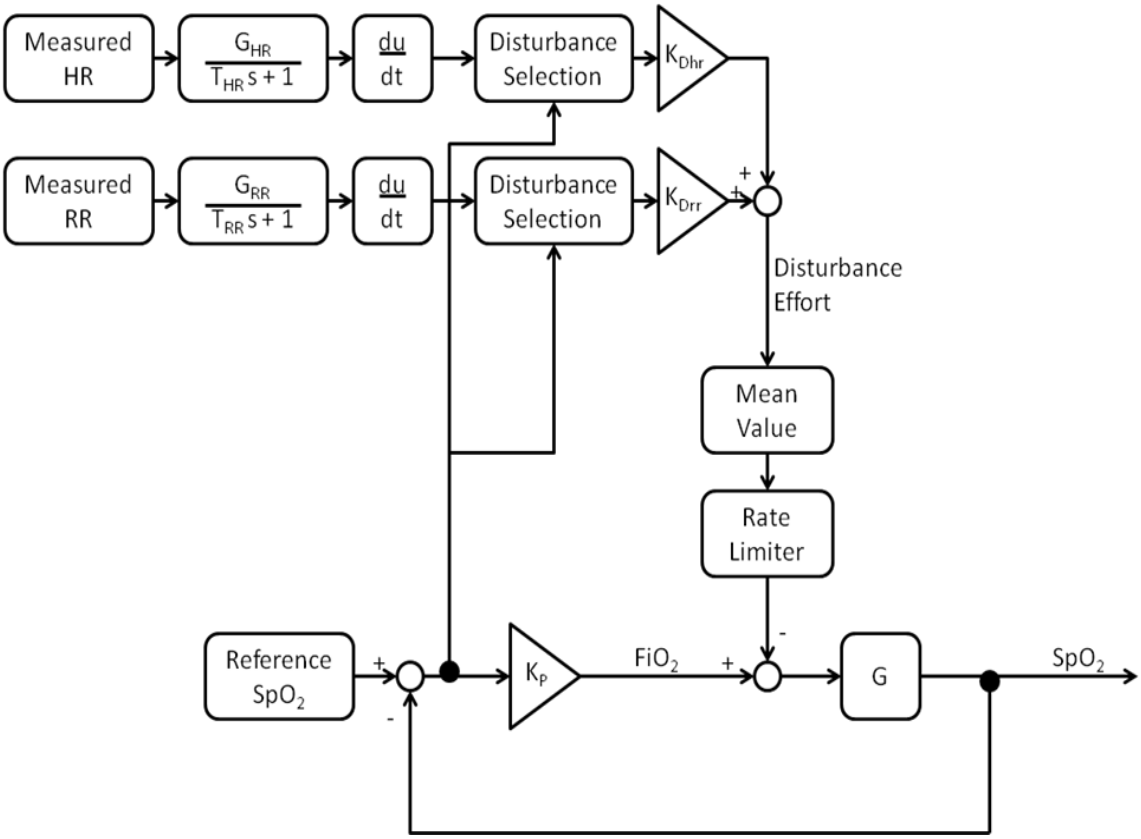

Figure 43. Block diagram of adaptive control with feed forward disturbance rejection

4.4 Controller Test

To analyze the performance of the controllers a disturbance rejection simulation was performed. This allows for comparison between each of the controllers. The parameters tested were the maximum SpO_2, maximum FiO_2, average SpO_2, and average FiO_2. By testing for each of these parameters the best controller can be found. To

simulate the infant, the block diagram structure shown in Figure 13 was used. To allow the system to have constantly changing parameters the gains and time constants for the FiO_2, HR, and RR were set to be a sin wave. In doing so, the test will better resemble the constantly changing parameter of the actual infants. A plot of the gains and time constants can be seen in Figure 44 and Figure 45. The inputs into each of the transfer functions are the same as used in Section 3.1.4. The controllers were then applied to the system starting at time 0 and run for 1000 seconds. The control signal for each controller has a saturation limit such that the signal does not go below 0% which corresponds to room air oxygen levels. The controlled signal (% SpO_2) and the control effort (% FiO_2) can be seen in Figure 46. The results from the parameters tested can be seen in Table 10.

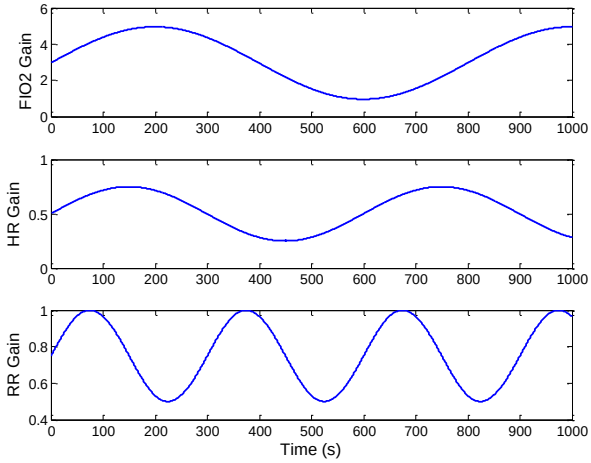

Figure 44. Gains used in plant in controller test

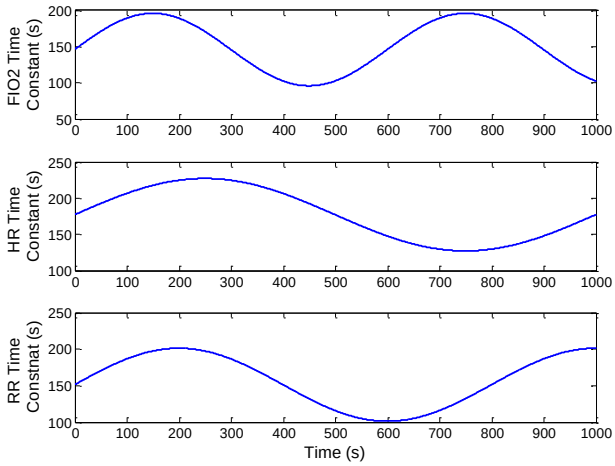

Figure 45. Time Constants used in plant in controller test

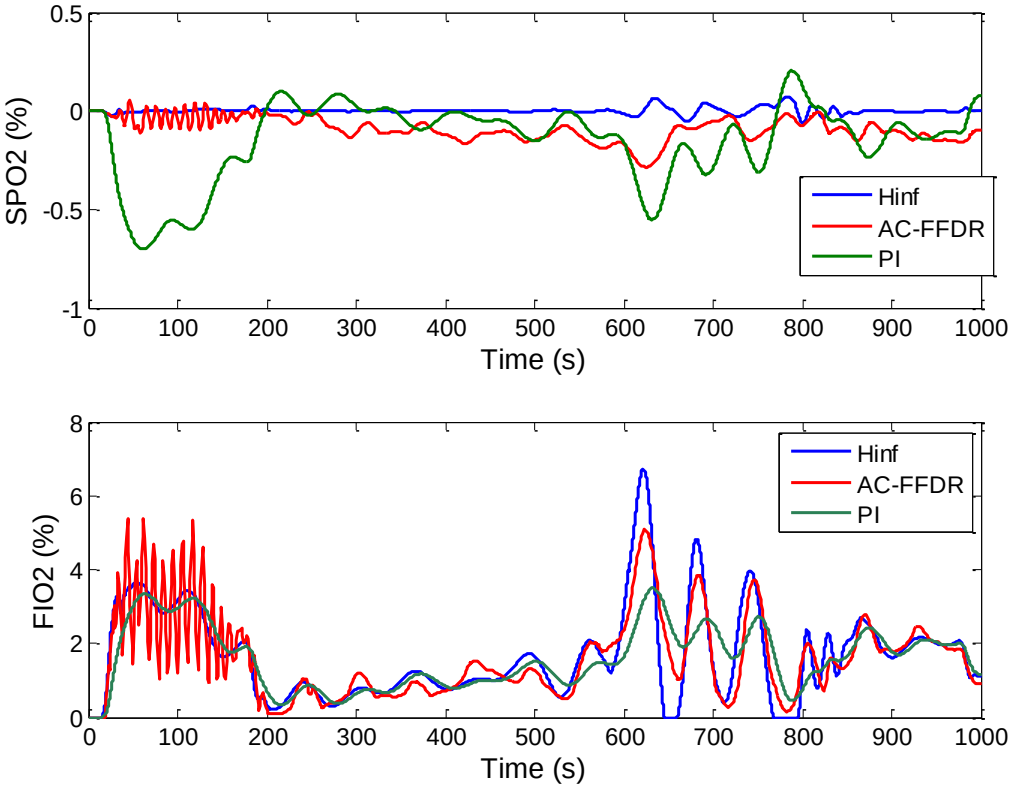

Figure 46. Plot of controller test results for the SpO_2 and FiO_2

Table 10. Results from controller tests

	Max SpO_2 (%)	Max FiO_2 (%)	Average SpO_2 (%)	Average FiO_2 (%)
LQR-PI	0.7029	3.5085	-0.1508	1.5976
Robust Controller	0.0725	6.7368	6.6423e-004	1.6778
AC-FFDR	0.2849	5.3801	-0.0919	1.6287

As shown in Figure 46 all of the controllers are able to reject the disturbance caused from the HR and RR. However the best performing controller is the robust controller. This controller has the lowest maximum SpO_2 and average SpO_2. The controller however does have the largest maximum FiO_2 and average FiO_2 values. The values of the FiO_2 are well within the limitations of the FiO_2

Chapter 5: Summary and Conclusions

Currently when infants are placed on a respiratory support device the nurses manually adjust the blend valve to regulate the amount of FiO_2 supplied to the infants based on the infants SpO_2. Due to the nurse's work load, the infant may not receive immediate attention when the desaturation event occurs. This can cause dangerously low SpO_2 levels which can lead to brain damage and even death. On the other hand if the SpO_2 level is too high to infant is at risk for Respiratory Distress Syndrome. The goal of this paper is to design a controller that can regulate the infants SpO_2 and reduce the time the infant is outside of the acceptable safe range of SpO_2. The main topics in this paper are the investigation of the biological system, modeling, and control of the infant's SpO_2.

The infant's biological system was investigated to see how the blood is oxygenated and what measurable parameters will give insight into the SpO_2 levels. It was found that the blood is oxygenated through the mechanisms of ventilation, perfusion, and diffusion. To find out how each variable affects the infant's SpO_2 data was collected at Neonatal Intensive Care Unit at the Columbia Regional Hospital. The data collected was the FiO_2, HR, RR, and SpO_2. To find out how each recorded signal affected the SpO_2 a relationship between the SpO_2 and the FiO_2, HR, and RR was found. It was found that a first order transfer function could approximate the relationship between the FiO_2 and SpO_2. Through analysis of the recorded data it was found that the range for the gains and time constants was [1.3, 12] and [20, 110]. The HR and RR were found to only have an effect on the SpO_2 when there is a drastic increase or decrease in the HR or RR for a prolonged time.

Two different modeling methods were used to model the infants SpO_2. The first type was the updating models. This modeling type was chosen for its ability to have a generalized model of the SpO_2 while allowing the model to be updated when needed. The models investigated are a neural network, fuzzy logic, and transfer function model. The models were tested on two types of simulated data sets, fixed and variable parameters. All of the models were able to adequately estimate future values if the system had fixed parameters. However for the variable parameters, the transfer function model was not able to adjust to the nonlinear parameters. This is to be expected since the model is linear. The neural network and fuzzy logic model though were able to adjust to the change in parameter and estimate the future values accurately. The other modeling type was a dynamic model. The model investigated was the dynamic transfer function model. The dynamic transfer function model was chosen to allow the transfer function model to better model a system with variable parameters. The model was tested on the same simulated data sets and was able to adequately model both data sets. All of the models were then tested on collected infant data and the results compared. The neural network and fuzzy logic models had trouble with extrapolating data which caused the model to perform poorly. It was shown that the neural network and fuzzy logic model on average could model the data for 36 and 43.4 seconds respectively. The transfer function model did not have trouble with extrapolation and was shown to model the data adequately for 51.8 seconds. The dynamic transfer function model performed the best. This model was able to model the data adequately for 1000 seconds. This model also gave the most insight into the infant's biological system. This is due to the model being able to estimate how the FiO_2, HR, and RR are affecting the SpO_2.The dynamic transfer

function model was run over 108 hours of data and the gains and time constants for the FiO_2, HR, and RR were recorded every 5 seconds. It was found that the gains for the FiO_2 histogram had two distinct peaks at 0.927 and 3.77. The histogram for the HR and RR gains however were centered at 0 causing the gains to be negative and positive. The histogram for the FiO_2, HR, and RR was 145.9, 177.2, and 151.3.

Three different controllers were designed based on the dynamic transfer function model and the estimated range of gains and time constants. The controllers designed are a linear quadratic regulator PI controller (LQR-PI), robust controller, and adaptive controller with feed forward disturbance rejection (AC-FFDR). The controllers were tested using simulated data. All controllers attempted to reject the disturbances caused by variations in HR and RR and keep the SpO_2 at a given set point. The LQR-PI controller was the least affective at controlling the SpO_2 with an average SpO_2 value of -0.1508 and a maximum SpO_2 value of 0.7029. The AC-FFDR performed the next best with an average SpO_2 value of -0.0919 and maximum SpO_2 value of 0.2849. This controller however had problems with excessive controller effort and thus caused an oscillation around the set point. The best performing controller was the robust controller with an average SpO_2 value of $6.6423*10^{-4}$ and maximum SpO_2 value of 0.0725. The SpO_2 is normalized at 90% and the FiO_2 is normalized at 21%. The SpO_2 and FiO_2 values presented here are the difference between the actual values and the nominal values.

In future work, the performance of the dynamic transfer function could be increased. The performance could be increased by introducing transport delay into the model. By introducing the transport delay into the system, the model would be less likely to produce negative gains for the HR and RR. Another improvement to the dynamic transfer function model would be to decrease the computation time. The computational time could be decreased by trying different global optimization methods.

References

[1] Merenstein, G., and Gardner, S. L., 2006, Handbook of Neonatal Intensice Care, Mosby.

[2] Timothy, K., Ramak, A., and Roger, F., 2010, "MODELING AND CONTROL OF THE OXYGEN SATURATION IN NEONATAL INFANTS," Dynamic Systems and Control.

[3] Keim, T., Amjad, R., and Roger, F., 2009, "Modeling and control of the oxygen saturation in neonatal infants," Dynamic System and Control ConferenceArlington, VA, pp. 1-2.

[4] Laptook, A., Salhab, W., Allen, J., Saha, S., and Walsh, M., 2006, "Pulse Oximetry in Very Low Birth Weight Infants: Can Oxygen Saturation by Maintained in the Desired Range?," Perinatol, pp. 337-341.

[5] Hagadorn, J., Furey, A., Nghiem, T., Schmid, C., Phelps, D., Pillers, D., and Cole, C., 2006, "Achieved Versus Intended Pulse Oximeter Saturation in Infants Born Less than 28 Weeks Gestation: the AVIOx study," Pediatrics, 118, pp. 1574-1582.

[6] Clucas, L., Doyle, L. w., Dawson, J., Donath, S., and Davis, P. G., 2007, "Compliance With Alarm Limits for Pulse Oximetry in Very Preterm Infants," Pediatrics, 119, pp. 1056 - 1060.

[7] Poblano, A., Marquez, A., and Hernandez, G., 2006, "Apnea in Infants," Indian Journal of Pediatrics, 73, p. 1085.

[8] Kenner, C., Lott, J. W., and Flandermeyer, A. A., 1998, Comprehensive neonatal nursing : a physiologic perspective, W.B. Saunders.

[9] Gray, J. S., 1945, "The Multiple Factor Theory of Respiratory Regulation," Army Air Force School of Aviation Medicine Proj, 1945.

[10] Grodins, F. S., Ruell, J., and Bart, A., 1967, "Mathematical Analysis and Digital Simulation of the Respiratory Control System," Journal of Applied Physiology, 22, pp. 260 - 276.

[11] Grevisse, P., Demeester, M., and LECOCQ, H., 1975, "A Pulmonary Model for the Automatic Control of a Ventilator."

[12] YU, C., 1986, "Improvement in Arterial Oxygen Control Using Multiple - Model Adaptive Control Procedures," IEEE Transcations on Biomedical Engineering, BME-34(8).

[13] Tehrani, F., and Bazar, A., 1991, "An Automatic Control System for Oxygen Therapy of Newborn Infants," Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2180-2182.

[14] Potter, P. A., and Perry, A. G., 2009, Fundamentals of Nursing 7th edition, Elsevier, St. Louis.

[15] 2011, "Shunting," http://www.britannica.com/EBchecked/topic/542345/shunting.

[16] Morozoff, E., & Saif, M, 2008, "OXYGEN THERAPY CONTROL OF NEONATES - PART I: A MODEL OF NEONATAL OXYGEN TRANSPORT," Control and Intelligent Systems, 36(3), pp. 227-237.

[17] Dayhof, J. E., and DeLeo, J. M., 2001, "Artificial neural networks openining the black box," CancerArlington, Virginia, pp. 1615 - 1635.

[18] "Levenberg-Marquardt," http://matlab.izmiran.ru/help/toolbox/nnet/backpr12.html.

[19] Engelbrecht, A. P., 2007, Computational Intelligence, John Wiley & Sons, Ltd, Weset Sussex.

[20] Jang, J.-S. R., ""ANFIS: Adaptive-Network-based Fuzzy Inference Systems," IEEE Transactions on Systems, 23(3), pp. 665-685.

[21] Taguchi, G., Chowdhury, S., and Wu, Y., 2005, Taguchi's Quality Engineering handbook, Wiley, Livonia, Machigan.

[22] Peterson, J., 2003, "Directional Control of a Track - Type Vehical," Masters of Science, University of Missouri, Columbia.

[23] Burl, J. B., 1999, Linear Optimal Control, Addison Wesley Longman.

[24] Skogestad, S., 2005, Multivariable Feedback Control, John Wiley and Sons Ltd.

[25] Astrom, K. J., and Murray, R. M., 2008, Feedback System: An Introduction for Scientists and Engineers, Princeton University Press.