A Comparative Design Study for Multivariable Feedback Control System Regulation for pneumatic feed mixing system

دراسة مقارنة في تصميم نظم التحكم ذات المتغيرات المتعددة لنظام الخلط الهوائي متعدد التغذية

by

MOSAB KHAIR ABDELRADY KHALIL

Dissertation submitted in fulfilment
of the requirements for the degree of
MSc SYSTEMS ENGINEERING

at

The British University in Dubai

November 2018

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

____

Signature of the student

Abstract

In this research, mixing system regulation was investigated. Two control methodologies were studied. Least effort method from the Modern British School, and H-infinity controller from Modern American School. The closed-loop system transients and steady-state response were validated. Controller disturbance rejection and energy consumption were verified, and the results were compared between the two design methodologies.

This research is started with a brief history of control systems, major effective millstones of this science and examples of main implementations. After which the main idea about multivariable systems is presented.

The system under study is a pneumatic feed mixing system used to simulate an industrial mixing process. It consists of two inputs and two outputs. Research objectives are to evaluate the two controllers under study and compare their performance and energy consumption used to achieve this performance.

This system was raised as a control problem used by Dutton (1997) to compare several controllers, and re-used by Whalley and Ebrahimi (2006). Revalidation work is done for least effort controller, MATLAB® code and Simulink® model were designed, and the result is compared with another controller designed based on optimal H-infinity method.

After a comparison between the two control techniques, this research concludes that the least effort control method is capable of giving superb solution to the control problem, all the objectives achieved with simple model and perfect energy consumption. H-infinity controller offered perfect response in terms of speed, disturbance rejection, and steady state de-coupling, but it has shown high energy cost with complicated model. This shall promote the use of least effort controller for general industrial multivariable mixing process and similar processes.

خلاصة البحث

في هذا البحث، تمت دراسة عملية التحكم في نظام الخلط ذو التغذية الهوائية. تمت تسليط الضوء على منهجيتين للتحكم. طريقة التحكم بالجهد الأدنى، من المدرسة البريطانية الحديثة، وطريقة اتش اللانهائية من المدرسة الأمريكية الحديثة. تم التحقق من أداء النظام الناتج من حيث سرعة الأداء ورفض الاضطرابات الطارئة من الخارج واستهلاك الطاقة في دراسة مقارنة بين منهجيتي التصميم.

وقد بدأ هذا البحث بسرد تاريخ موجز لأنظمة التحكم، والمرور بشكل سريع على نقاط التحول الرئيسية في تاريخ هذا العلم مرورا بسرد امثلة رئيسية لتطبيقات حتى الوقت الحاضر.

النظام الذي طبقت عليه الدراسة هو نظام خلط هوائي متعدد التغذية وهو نموذج مخبري استخدم لمحاكاة عملية خلط صناعية.

بعد المقارنة بين كلا تقنيات التحكم، يخلص هذا البحث إلى أن طريقة التحكم بالجهد الأدنى قادرة على إعطاء حل ممتاز لمشكلة التحكم، جميع الأهداف تم تحقيقها مع نموذج بسيط واستهلاك مثالي للطاقة. قدمت وحدة التحكم المصممة بطريقة اتش اللانهائية استجابة مثالية من حيث السرعة، ورفض التغيرات الطارئة من الخارج، ولكنها أظهرت تكلفة طاقة عالية نسبيا مقارنة بطريقة التحكم بالجهد الأدنى كمان ان وحده التحكم كانت معقدة التصميم. هذا البحث يفضل استخدام وحدة تحكم المصممة بنظام التحكم بالجهد الأدنى في عملية الخلط الصناعية وغيرها من العمليات المماثلة.

Acknowledgments

I am sincerely thankful to the soul of Professor Robert Whalley, who taught me how to climb the first step in the long stair of control engineering. It has been a life time chance to learn one of the theories applied in this research from the theorist himself. It was a great honor to be Professor Whalley's student.

My special thanks and gratitude to Dr. Alaa A. Ameer for the kind supportive words and all the time open office. Dr. Alaa patience shown to me during the period of this research was the most important motive for this research to see the light. Thank you, Dr. Alaa.

My thanks are extended to all management personnel of the British University in Dubai for their professional support. I am gratefully indebted to their help shown through out this reaserch.

Dedication

To my father, Mother, and brothers, sister, and friends,

To whom who tough me the value of hard work,

To whom who continuously believed in me,

To whom who always stod by my side,

To the people of my country, to the people of the Arab world.

I dedicate this work to all of you

"I'm a great believer in luck, and I find the harder I work, the more I have of it."

Thomas Jefferson

Table of Contents

List of Notations

$A(s)$	$G(s)$ matrix numerator
$\Gamma(s)$	Finite time delay matrix
$y(s)$	Transformed output vector
$u(s)$	Transformed input
$r(s)$	Transformed reference input
$\bar{r}(s)$	Transformed inner loop reference input
$\delta(s)$	Transformed disturbance vector
$h(s)$	inner loop feedback
h	Feedback gain
$\mathbf{K}(s)$	Feed forward controller model
$k(s)$	Feed forward function
k	Feed forward gain
P	Pre-compensator matrix
F	Outer loop feedback matrix
f_j	Outer loop gain
$\mathbf{I}_m$	Identity array
$S(s)$	Sensitivity matrix
S_s	Steady-state matrix
$\mathbf{G}(s)$	Transfer function array
$G(0)$	Steady-state transfer function value
$k >< h$	Outer product of k and h
$<k,h>$	Inner product of k and h
$L(s)$	$G(s)$ left side factors

$R(s)$	$G(s)$ right side factors
Q	Coefficient array
J	Performance index
$\mathbf{H}(s)$	Feedback path compensator
$n, n_1, ., n_{m-1}$	Gain ratios
$\delta(s)$	disturbance
$\Vert \quad \Vert_\infty$	H-infinity norm
W_1, W_2, W_3	Weighting matrices
$\mathbf{S}(s)$	Sensitivity transfer function
$\mathbf{T}(s)$	Complementary Sensitivity transfer function

List of Figures

Chapter I: Introduction

1.1 Research Background

Industrial automation growth in current manufacturing plants system calls for well-defined rules between equipment manufacturers and their customers. Production methods with automatic machines integrated on them or used in their operation are required to fulfill the growth of the market and the high demand for supply. Automated machines are demanded not only because they speed the production lines, but also, they make manufacturing processes easier, and more efficient. Moreover, well-regulated machines reduce the human intervention to the minimum which results in accurate processes gained by rolling-off the human error factor. One of the most commonly used machines in the modern industrial plants are mixers (figure 1.1.1).

Figure 1.1.1: Industrial mixers (EKATO Holdings GmbH)

Mixers are used since modern industrial processes mostly involve some form of mixing. Mixing is defined in industrial process systems engineering as a process that involves changing a heterogeneous material targeting to make it less heterogeneous or more homogeneous. If suitable machine is sellected, mixing of liquid, solid, or gas into another liquid, solid, or gas is possible.

Mixing process is not simple to achieve especially in a larg (industrial) scale, efficient mixing can be difficult to achieve. Considerable engineering effort is spent on designing and enhancing industrial mixing processes.

As a matter of fact, , static mixers are not the only type of mixers that's used in industrial scale. Dynamic mixing, in which the mixing is performed in batches, is also used. In this type of mixing, the mixers use motors which move in very high speeds (thousands of RPM). Motor speeds are reduced by gearboxes which increases the torque. Sometimes, multi-shaft mixers are used, where both mixer types are being used for complete blending of the mixture. (figure 1.1.2). Additional to batch mixing processes, mixing at industrial scale can be performed using a continuous processor in a continues manner. Using a Continuous Processor, one or more dry or liquid ingredients can be fed to the machine in an accurate continues way, and in the same continuous pattern, a homogeneous mixture will come out of the machine. Due to many advantages, like lower energy consumption, ease of maintenance and cleaning, and ability to control, many industries shifted to continues mixing using continues mixers like the twin screw continues processor.

Figure 1.1.2: Continues mixing system (GEA Group, 2018)

The system to be studied is described by Dutton (1997), it is an industrial process in which two streams of liquid feedstock, one is hot, and the other is cold, are poured and mixed into a vessel. The mixture is continuously drawn off from the vessel in a variable flow rate into the next part of the process. The aim is to control both the temperature of the mixture and its head in the mixing vessel. This is required so that the next part of the process will be fed under constant (regulated) temperature and head conditions. The control is done using flow control valves on two feeding lines. To obtain the system model a pneumatic prototype was created from which the system transfer function is interpolated.

1.2 Problem Statement

Pneumatic feed mixing system regulation will be studied in this research. The system is a 2-input 2-output multivariable system with a transfer function extracted by exiting the system by applying an input and then the output is measured, Dutton (1997). The inputs to the system are referred to by u_1, u_2 which simulate hot and cold feed lines. And outputs are y_1, y_2 which simulates fluids mix head and temperature.

Since multivariable systems had the difficulty of output interaction, they are difficult to control because of cross couplings in the process. If every input or set point biased several process output variables instead of one, and if the multivariable controller did not counteract such issue, then the system performance will be affected in two ways. Any change in one setpoint will cause a response in every output variable of the process corresponding to the set point. Also, the system which is complicated dynamic system and consisting of many control loops will be "observed" by the controller, resulting in a narrow system stability margin, and the robustness of the system will be affected.

Considering the above, the main requirement of the regulation problem is to regulate both the outputs confirming faster system response assuring near zero or low interaction at the output variables. The system disturbance recovery capability to be confirmed. Finally, all of the above is achieved with low energy, that's why the system energy expenditure will be evaluated.

1.3 Aims and Objectives

A pneumatic feed mixing system will be controlled with two multivariable controllers, A least effort controller, which is introduced by Whalley R. and Ebrahimi M. (2004), and an H-infinity controller which was introduced at the late 1980' and widely used as it is applicable to control design problems involving multivariate systems with coupling between channels. Least effort controller is promoted by offering a good solution to multivariable control problems with least energy cost. In this research, a comparative study between H-infinity controller and least effort controller will be established to evaluate each controller strengths and weaknesses. The main fields of comparison are based on the ability of controllers to maintain the system stability for the closed-loop systems and to enhance the performance of the system in both the steady state and the transient periods. Also, to improve input-output interaction by limiting it to 10%, and to study the controller capability of performing disturbance rejection. Controller energy consumption will be studied and monitored. Also, the simplicity of the controller model and ease of implementation will be on point to aim.

1.4 Research Organization

This research is organized in seven chapters as the following. The first chapter is an introduction that highlights the research background, were a summary about mixing systems is given, after which research problem is stated and then aims and objectives are presented. Chapter two is basically a literature review, general historical background about the development of control theory, and main types of controllers; the light will be focused on the history of the implemented controllers and their main applications. Chapter three is presenting the open-loop system model and its response to step a change in both the inputs for later comparison with the controlled system. Then the mathematical derivation and analysis theory will be reviewed for least effort controller as per Whalley R. and Ebrahimi M. (2004). The same task is done for H-infinity controller in what is considered as a theoretical review before the implementation of the design in chapter four. In chapter four, the concepts presented in chapter three is applied to the mixing system model. Least effort controller design work done by Whalley R. and Ebrahimi M. (2004) was validated. And with the assistance of MATLAB® control toolbox, H-infinity controller was designed. Chapter five mainly represent the simulation results for the control models from chapter four, system response for two inputs was simulated and plotted. In chapter six, a comparison study was conducted between the two controller models regarding closed-loop response, disturbance rejection, output interaction, control energy consumption and ease of implementation. In chapter seven the research is concluded, the positives and the drawbacks for every method is highlighted. Also, some recommendations for future work development were mentioned.

Chapter II: A Literature review

2.1 Introduction

Control theory dated from the nineteenth century, when the operation of centrifugal governor of James Watt's steam engine was first mathematically described by differential equations by Maxwell (J Maxwell, 1868).

Centrifugal governors are governors with a feedback system which regulate an engine speed by managing the amout of flow of the fuel fed to that engine. These governors were meant to maintain a constant speed, regardless of any change in load or fuel-supply conditions. The controller used was a proportional controller.

Figure 2.1.1: Centrifugal governor in Watt engine
(Science Museum Group Collection, 1788)

Maxwell concept was to linearize the motion differential equation to find the characteristic equation of the system. And then he proved the system stability when the characteristic equation roots have negative real parts.

Routh–Hurwitz theory

In 1874, Edward Routh, an English mathematician who was a classmate of Maxwell and working at the University of Cambridge, invinted a numerical methode for checking if a polynomial has negative poles, and contributed to the development of control theory, by establishing the concept of control stability (Routh E., 1877). At the same time and independently, a German mathematician called Adolf Hurwitz was specialized in algebra, and number theory. Adolf Hurwitz studied system stability and analyzed it using differential equations. In 1895, Routh–Hurwitz theory, named after Routh, and Hurwitz, was proved. This theory was and still used to determine the stability of polynomials by checking if all roots of the polynomial are located in the LHS plane.

Lyapunov stability

Independently of Maxwell, using differential equations, Alexander Lyapunov worked on the nonlinear differential equations stability, he used a generalized notion of energy (Lyapunov, 1893). Alexander Lyapunov invented methods for stability in the time domain that helped in the development of control theory and are still in use till date.

In 1913, Henry Ford introduced a mechanized assembly line used in automobile production, this assembly line is considered a significant step forward in the application of control theory, in 1922, ford said about the completion of his 1913 assembly line: “I believe that this was the first moving line ever installed. The idea came in a general way from the overhead trolley that the Chicago packers use in dressing beef". (Ford & Crowther 1922, p. 81). Later at 1922 and the years after, PID control theory came into the picture, Minorsky was the first to introduce it (Minorsky, Nicolas,1922).

Figure 2.1.2: Ford's 1913 assembly line (Swan, Tony, April 2013)

Frequency domain method

In 1927, the frequency domain method was introduced while engineers where trying to solve the problem of distorsion that took place on long distance telecommunication systems. After six years of intensive research on amplifiers, the American engineer Harold S. Black revolutionized telecom systems when he proposed the negative feedback in 1927 (Black, 1934). This invention had a high impact in many control applications and opened the door for frequency method to be used in control systems.

In 1932, at Bell Laboratories H. Nyquist came into the picture by developing a frequency domain based method of analyzing the stability of systems (amplifiers). He worked reacearch department of AT&T from 1917 to 1934. Additional to his significant contribution to the telecom field that developed information theory afterwords, he contributed significantly in the area of control theory. His published classic paper on the stability of feedback amplifiers is a milestone in control theory (H. Nyquist, 1932). The Nyquist criterion of stability was oriented about plotting a polar plot of a system's transfer function, it is found in all textbooks on feedback control theory.

Later, also at Bell Laboratories, H.W. Bode came into the picture by analyzing feedback amplifiers. He is considered as a pioneer of control theory by revolutionizing the control reaserch with new meethods. Later in the second world war, Bode introduced essential improvement to the control and design of anti-aircraft wepons, and his work was continued after the war, to the design of missile flight control systems. H.W. Bode also contributed to control theory and introduced his own way of analyzing the closed-loop stability of linear systems mathematically and graphically by introducing the transfer function frequency response magnitude and phase plots, which was named after him (Bode plots). Bode also introduced the concept of phase margin and gain margin (Bode, 1940).

The hysteresis controller

Control theory became an important area of research by the second world war. The method of discontinuous automatic control systems was developed in the time of second world war by German -American control theorist Irmgard Flugge-Lotz, the bang-bang principle (hysteresis controller) was applied to enhance an aircraft flight control device (Flugge-Lotz, Irmgard, Titus, Harold A., 1962). Mechanical approaches were implemented to enhance the stability of some systems like in ship fin stabilizers which used a gyroscopic control system.

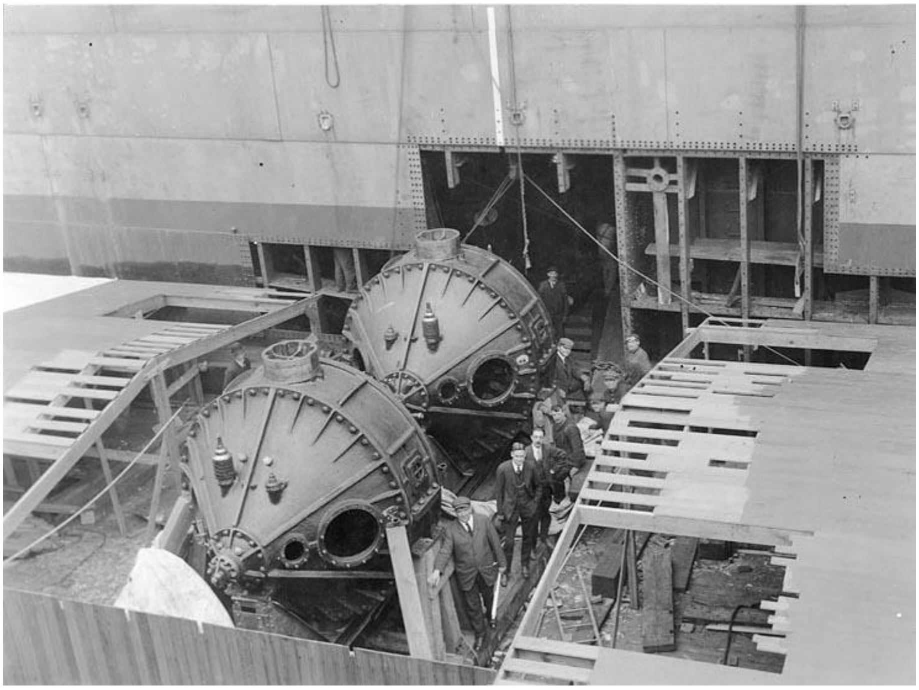

Figure 2.1.3: Two gyroscopes being installed in the first large ship to use gyroscopic stabilization (Gleaves, Albert 1921)

Nichols chart

In 1947, at USA, Massachusetts Institute of Technology, Nathaniel B. Nichols invinted what is called “Nichols chart”. He used it as method of design of feedback control systems based on the frequency domain. By the same time, he established the “theory of servomechanism” and his research was of a great value (James, Nichols and Phillips, 1947).

First industrial robot

In the year of 1954, George Devol developed the “programmed article transfer”, which is typically the first industrial robot design, and in 1960 this design saw the light by introducing the first Unimate robot. And by 1961 this robot was installed in a die-casting machine. (Dorf and Bishop, 2008).

Figure 2.1.4: Unimate, The First Industrial Robot
(Robotic Industries Association, 2018)

Root locus method

Then, in 1965, a major step in control systems design was introduced by Walter R. Evans, that is the root locus method. This method was a frequency domain method based on the use of open-loop system information (poles and zeros) to analyze the the closed-loop when changing one of the system parameters - mainly the gain within a feedback system-. (Evans, Walter R. 1965).

Modern control & Kalman optimal controller

In 1960, a significant achievement came to light, and the modern era of control theory started by Rudolf Kalman writings. Kalman presented the main problems of nonlinear system theory, and he was able to solve them smartly. Kalman considered the Lyapunov nonlinear systems stability in (Kalman and Bertram, 1960). Then in (Kalman, 1960a) he introduced his optimal control theory and a new design of what is know now as the linear quadratic requlator (lqr). Kalman also wrote about “optimal filtering and estimation theory”, and in his book (Kalman, 1960b), he presented the design for the discrete Kalman filter. A year later, in (Kalman and Bucy, 1961), Kalman introduced his continues time filter.

One of the major achievements of Kalman is introducing the “state” which is a “a mathematical entity that mediates between inputs and outputs” (Gopal M., 1993). Kalman used this concept to solve the difficulties faced with time domain higher order differential equations solution, without being obligated to solve the control probblems in the frequency domain avoiding its limitations.

Later in 1980, Robust control system design was widely studied. And in the 1990’s, feedback control systems started to be commonly used in mega industries like car manufacturing.

2.2 Control Fundamentals Review

Control theory is basically about dealing with any type of system miss behavior like a delay or unwonted dynamics. A mathematical model of the physical system is formulated, a control method is applied to the open-loop system model, and then, the complet closed-loop system model is tested on a software or hardware prototype before applyied to the physical system. Although the mathematical formulation of control problems is sometimes complex, the basic concept upon which control theory is formulated in not as complicated as mathematical formulation. Control theory is based on three basic fundamental concepts.

The first concept is “feedback”. Although the feedback concept is older in the history of science, but the term “feedback” that means feeding back an output to regulate an input came in use by engineers from AT&T Bell Lab. And it was first used in stabilization of op-amps (Mayr, 1970). In a feedback process, one or more of the system states or outputs determines the control action by the controller .

Another main concept in control theory is the concept of obligatory transients. Any system need to be in transient state before settling. It is important to mention that there is no requirement to force the system with high energy to drive it to steady state in no time. It is more practical and physically realizable while controlling a system to allow the system fluctuation in the transient period and in the same time to sellect "well behaived" transients which will allow the system to settle down to its steady state without driving it unstable.

Another major control theory concept in that of "optimization". In brief words, optimization in control theory sets a target in control problems that is giving the required control action with acceptable transients and steady state performance with maximized profit or minimized costs.

Open-loop systems

Control loops can be categorized into two categories, open-loop control or (feed-forward), and closed-loop control or (feedback). In open-loop, the output of the system is not sensed or measured and this leads to a system actuating device with no information about the system output.

Figure 2.2.1: Typical open-loop system representation (Dorf and Bishop, 2008)

Closed-loop control

In closed-loop systems, the controller "control action" depends on the reference input and also on the process output. A feedback signal is introduced to ensure that the controller maintains the pre-set (reference) input. In closed-loop, the control system monitors the feedback, and then the difference between the reference and the feedback is set to be the new input at which the control action will react to trying to reduce this deviation to zero.

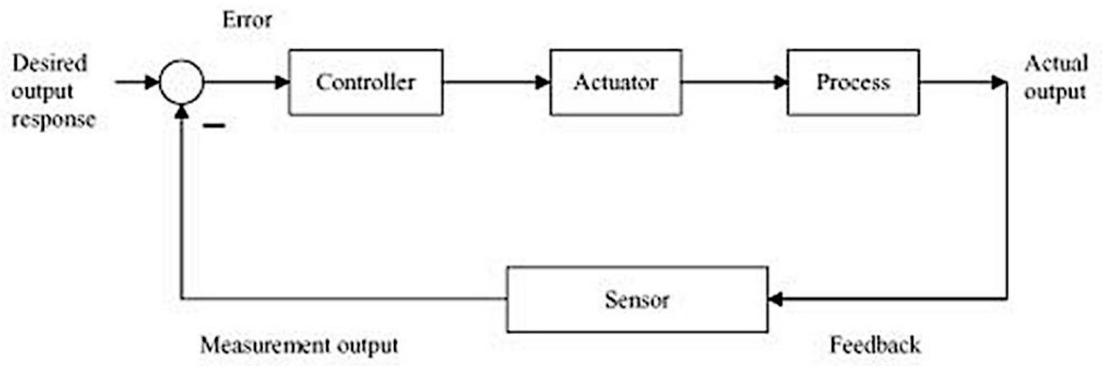

Figure 2.2.2: Typical closed-loop system representation (Dorf and Bishop, 2008)

While open-loop control is simple, and easy to construct and maintain, closed-loop controllers supersede open-loop control because it offers the following

- Disturbance rejection.
- Improved reference tracking performance
- Possibility to stabilize unstable processes.

Some application used both open-loop and closed-loop. The open-loop controller is used in such applications to set the reference tracking performance to a higher level. A well-known closed-loop controller architecture is the proportional, integral, derivative (PID) controller.

PID controller

PID notation is taken of proportional, integral, derivative. It is a closed-loop controller widely used in nowadays industry due to its simplicity and variety of many applications. PID controller continuously observe the output and measure it, by minusing the measured feedback output from the reference input (the required set point). The controller afterwhich tries to enhance the output by setting a proportional, integral, and derivative correction terms.

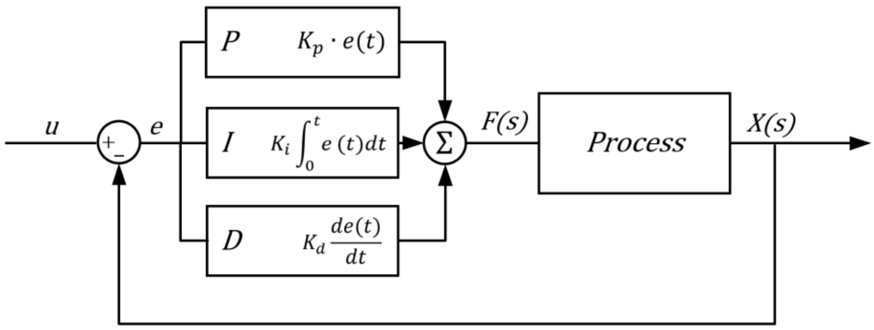

Figure 2.2.3: Typical representation of a PID controller (Dorf and Bishop, 2008)

From figure 2.2.3, the PID controller equation in time domain is:

$$f(t) = K_p.e(t) + K_i \int_0^t e(t).dt + K_d \frac{d(e(t))}{dt}$$

and in frequency domain:

$$F(s) = K_p.E(s) + \frac{K_i}{s}.E(s) + K_d.S.E(s)$$

Proportional component

The proportional gain determines the ratio of output response to the error signal. It can be mentioned that if the proportional gain increased, the speed of the control system response will increase. But, if the proportional gain is larger than some limit, the system response will start oscillating, and for larger gains, it may go out of control.

Integral component

The integral part of the PID controller calculates the summation of the error over time, which means that small error measurements will keep adding up with time till they become higher and the integral part effect will keep increasing in slow manner targeting the steady-state error that feeds if until the error goes to zero. Integral windup may come to picture if integral action saturated the controller without driving the error to zero.

Derivative component

The derivative component of the PID controller is directly proportional to how fast the system changes its response. The higher the rate of change of the system response, the higher the effect of the derivative component. The derivative component role is to force the output to decrease if it is increasing fast. Giving more weight to the derivative component will have a proportional effect on the system speed of response. However, sellecting a high derivative component is not a practice in real practical systems, because of the higher this component, the more sensitive the system to external noise or unwanted disturbances.

System transient response

System transient response is the system response to changing from a steady state or equilibrium. Step response can be considered as a transient response to a step input.

From figure 2.2.4 below, the following can be defined:

Rise Time: The time required by the system output $y(t)$ to rise from 10 percent to 90 percent of the system steady-state response y_{final}.

Settling Time: The time required by the error between the current $y(t)$ and the final steady-state y_{final} to be bounded within 2 percent of y_{final}.

Overshoot: The maximum value of $y(t)$ minus y_{final} , relative to (devided by) y_{final}.

Peak: Absolute value of max $y(t)$. **Peak Time**: Time when peak value take place.

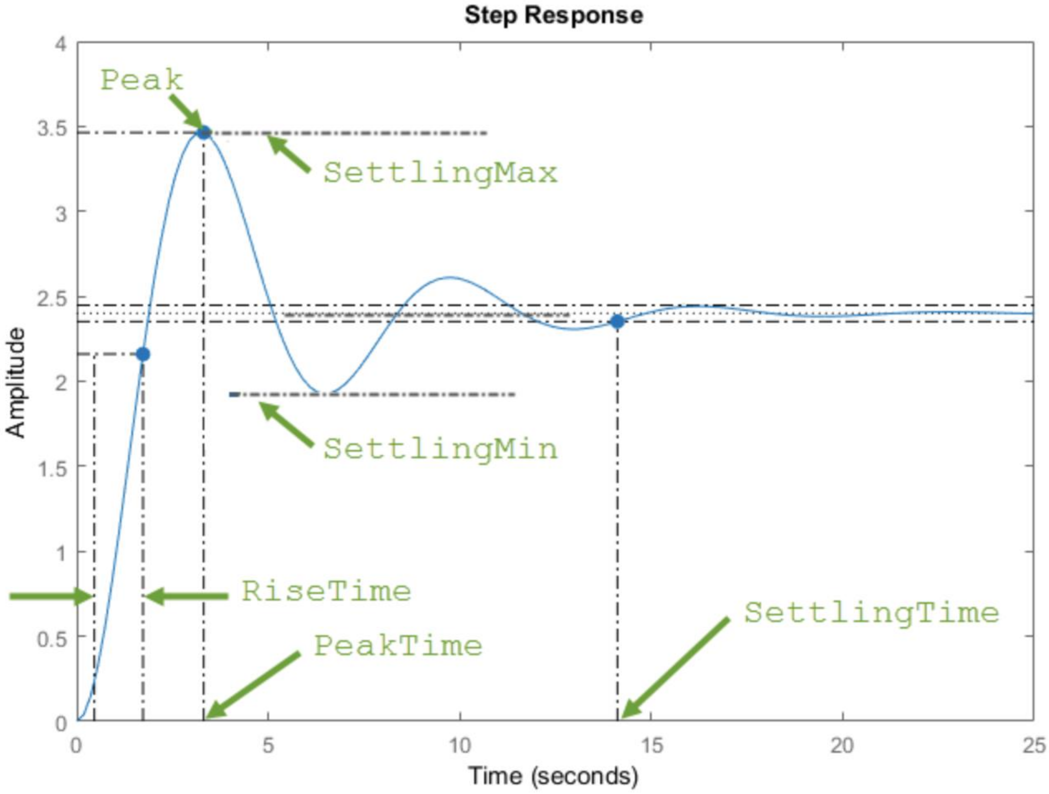

Figure 2.2.4: Block diagram of PID controller (The MathWorks, Inc, 2018)

2.3 Frequency domain approach

Frequency domain method of control is oriented on transforming the differential equations used to represent the system from the time domain to the frequency domain, Laplace Transform is used to achieve this requirement:

$$F(S) = \int_{0}^{\infty} e^{-st} f(t)\, dt$$

Many control design methods from classic control used the Laplace transform, the most famous among them are those who were developed by Bode, Nyquist, Nichols, and Evans. In these methods, after the system is transferred to the frequency domain, its response is analyzed depending on the “magnitude” and “phase plots of the frequency response, or in some methods, the open-loop system transfer function poles and zeros give an indication of the closed-loop system behavior. These methods mentioned above are applicable and effective for “single input-single output" (SISO) systems since the system frequency-response, and the system transfer function poles and zeros can be easily determined. Moreover, robust control can be implemented using gain and phase margin concept. For time-domain approach, if the system is complex, the block diagram algebra is commonly used, internal description of the system dynamics is not important if the overall input/output relationship is obtained.

On the other hand, for "multi input-multi output" (MIMO) systems, graphical methods are not practical for employment. That is because the interaction introduced between the control loops in multi-variable systems. The "quantitative-feedback-theory" (QFT) proposed by Horowitz solved many issues like the highlighted above and provided a powerful method to control multi-variable systems (Horowitz, 1982).

QFT is a method that used "Nichols chart" to obtain an acceptable controller performance in terms of robustness over a pre-defined region of system uncertainty. The transfer function is bounded by translating the required time domain response into frequency domain tolerances. Also, for non-linear systems, classic control theory can apply the control method on a linearized version in a region where the system behavior is considered liner. Even though Horowitz overcame many frequency domain limitations by the (QFT), his method didn't look after important factors which are the internal dynamic changing inside the system that produces the output from a given.

2.4 Time-domain design method

This design method is oriented on differential equations. Since frequency domain techniques have difficulties in solving non-linear problems as they are limited to linear systems, the time domain is widely used to analyze real-world nonlinear systems. Even time domain differential equations are not easy to solve, modern computer simulation techniques have made their analysis much easier.

System stability was one of the most important issues studied in time-domain methods. As previously highlighted, Maxwell studied the Watt's governor stability in time-domain, and Routh provided a numerical technique used till the moment to determine when a system's mathematical representation is not stable with (negative-roots) polynomial. As a basic concept and vital requirement, stability issue in time domain was studied repeatedly, and many solutions were introduced, but the most effective and general solution was introduced by Alexander Lyapunov who introduced methods to study the stability of nonlinear differential equations. Additional to stability, optimal control was a persisting requirement in time-domain control methods. In 1950's two principles for optimality were introduced, they used means of a feedback law to characterize the optimal control.

The first principle was introduced in 1957 by Richard Bellman who pioneered in dynamic programming and used it to solve the optimality problem. Bellman was able to introduce his principle that solves the optimization problem using a nonlinear first order partial differential equation. By introducing a value function (the Bellman function) which solves the Hamilton-Jacobi equation. Majority of the control theory problems that can be solved using optimal control theory can also be optimized using Bellman's equation.

The work of Rudolf Kalman on state space models of the system came at 1960's to open the door for the modern control by revolutionizing the control theory in general and time domain techniques in particular. Kalman's state space realization of the system consists of a set of first-order differential equations the relate the input and output as follow:

$$\dot{x} = Ax + Bu \qquad \dots (2.4.1)$$

$$y = Cx + Du \qquad \dots (2.4.2)$$

where: A, B, C, D are the state, input, output, feedforward matrices, respectively.

And x, y, u are the state, output, and input vectors respectively.

Taking Laplace for equations (2.4.1), (2.4.1) consequently yields in:

$$Sx(s) - x(0) = Ax(s) + Bu(s) \quad \text{... (2.4.3)}$$

$$y(s) = Cx(s) + Du(s) \quad \text{... (2.4.4)}$$

Assuming zero initial conditions, and strictly proper system equations (2.4.3) and (2.4.4) will become:

$$\mathrm{Sx(s) = Ax(s) + Bu(s)} \quad \text{… (2.4.5)}$$

$$\mathrm{y(s) = Cx(s)} \quad \text{… (2.4.6)}$$

Rearranging (2.4.5):

$$\mathrm{(SI - A)x(s) = Bu(s)}$$

And:

$$\mathrm{x(s) = (SI - A)^{-1}Bu(s)}$$

$$\mathrm{y(s) = C(SI - A)^{-1}Bu(s)}$$

$$\mathrm{\frac{y(s)}{u(s)} = G(s) = C(SI - A)^{-1}Bu(s)} \quad \text{… (2.4.7)}$$

Where the equation in (2.4.7) is the syatem transfer function.

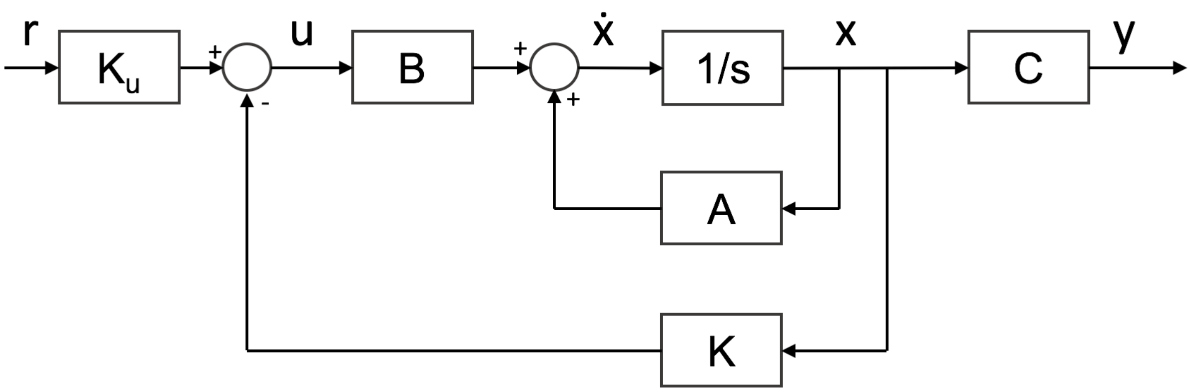

Figure 2.4.1: Block diagram of state space realization (The MathWorks, Inc, 2018)

While frequency domain approach managed to convert the difficult to solve differential equations to simple algebraic equations in the frequency domain, Kalman was able to convert complicated higher order differential equations to simple first-order differential equations. According to Kalman, system states, inputs, and outputs can be represented in a form of vectors, so that the equations will be written in a matrix form. That will make it possible for the state space representation of a system to be compact and easier for analyzing, which will ease the job of analyzing multiple inputs-multiple outputs systems.

2.5Robust Control

A controller is considered robust if its properties do not change in a considerable manner when applied to a system which is not identical to the mathematical system used for its analysis. Taking into consideration that most of the real-life physical systems don't behave exactly like the set of differential equations that are used for the mathematical representation of a system. In other words, robust control is a design method for controllers which concentrate on the issue of systems un-certainty. Controllers with high robustness capabilities are meant to work in an acceptable stable way provided that the system has uncertain parameters and subjected to external disturbances.

The methods developed by Bode and other researchers who belong to the classic control school and used frequency domain approach were robust enough, but the robustness rank was found to be not in a sufficient level for other methods of modern control school which came into the picture by 1960s and were state-space oriented. Having such lacks lead to the start of robust control studies, by the late 1970s. later, many methods were introduced to solve the uncertainty issues in pre-defined and known boundaries.

H-infinity loop-shaping is considered to be a very common and famous example of a robust-control technique in modern control. This method was developed at Cambridge University by Duncan McFarlane and Keith Glover. Loop-shaping will guarantee that the system behavior will not change in a considerable manner once the system is subjected to un-wanted disturbances. This will be achieved by modifying the sensitivity of the system over frequency. Loop-shaping affects the response of the system at lower or higher frequencies, but it is increasing the stability margin if the response around unity gain of the system is adjusted.

Other examples of robust control design methods include Loop Transfer Recovery (LQG/LTR), Lyapunov based control, quantitative feedback theory (QFT), and passivity-based control.

To be highlighted also is that Least effort controller is considered among controllers that preserve system response while subjected to disturbance and hence, it has a good robustness quality.

2.6 Least Effort regulation

Whalley, R and M Ebrahimi offered the least effort control method (Whalley, R and M Ebrahimi, 2006) to solve a control problem of mixing system, the same system was previously regulated by Characteristic loci (CL), and inverse Nyquist array (INA) methods. Whalley, R and M Ebrahimi proved in that paper that the least effort controller could perform the same control task with less energy than the other mentioned techniques.

The least effort control strategy is based on a closed-loop analysis which targets minimum control energy cost with acceptable control action for multivariable systems. The control action of the least effort controller will be based on simple gain feedback, structured in two loops, an inner loop with feedforward and feedback gain vectors k(s) and h(s), respectively. The inner loop is adjusted to produce the required system transients like rise-time, settling-time, overshoot, etc.

After adjusting the inner loop feedforward and feedback vectors, the system outer loop parameters are designed. First, the outer loop feed-forward gain is adjusted to give the required de-coupling between the outputs, after which the feedback gain f is calibrated to produce the required dynamics and disturbance rejection properties.

2.7 H-infinity Control Method.

In this research, the automatic control of feed mixing system is also achieved by using H_∞ control methodology.

H_∞ loop-shaping control method is classified among the high-rank methods used to produce robust results in the modern control school. This method was spread not only because it results in robust controller but also because it wont affect the required system transients. This methodology practicality is one more advantage as it is capable to solve complex control problems with high practical association.

This methodology was introduced in control theory by George Zames (Feedback and optimal sensitivity, 1981), J. William Helton (broadband matching, 1978), and Allen Tannenbaum (gain margin optimization, 1980).

H_∞ controllers can be considered as optimal controllers which minimize H_∞ norm rather than the usual L_2 quadratic norm. (Grimble, 1988). The result of this is the robust stability criteria of H_∞ controllers, which is the main reason for the development of H_∞ techniques (Zhou et al., 1996).

H_∞ controller is highly dependant on the selection of the weighting functions, which is a very critical decision to make since a non-proper selection of weighting functions will produce a poor controller. The problem of choosing proper weighing functions is solved by using optimization techniques, such as Genetic Algorithms which optimize the weighting functions automatically (Goldberg, 1989). Genetic Algorithms (GAs) are numerical optimization techniques used to solve nonlinear or non-differentiable optimization problems by mimicking natural biological evolution.

It is worthy to mention that referring to GAs usage in selecting proper weighting functions is not common in control theory references. Most designers use Loop-shaping method with GA method. If the structure of the loop shaping compensators is pre-defined, GA can be used in the optimization task of the parameters. Examples of such researches are (Dohna et al., 1997), and (Dakev et al., 1997). The range of practical applications of H_∞ controller is vast, maybe flight control by (Sveriduk et al. 1998) was one to be highlighted.

Chapter III: Research Methodology

3.1 Pneumatic mixing system model

The pneumatic system in figure (3.1.1) with the system process mentioned represents the system under study. This system is detailed by Dutton (1997) and studied for a comparison between least effort controller and other controllers by Whalley (2006). The system is a pneumatic laboratory prototype machine used to simulate an industrial mixing process.

The mixing process consists of two feeds of liquid, one hot and the other is cold. These two feeds are poured and mixed in a vessel. The mixture is taken from the vessel in a variable flow rate to the next step of the process. The requirement is to control the temperature and the level of the mixture in the vessel. And to maintain constant temperature and level, the system can change the flow rate of the hot and cold feeds.

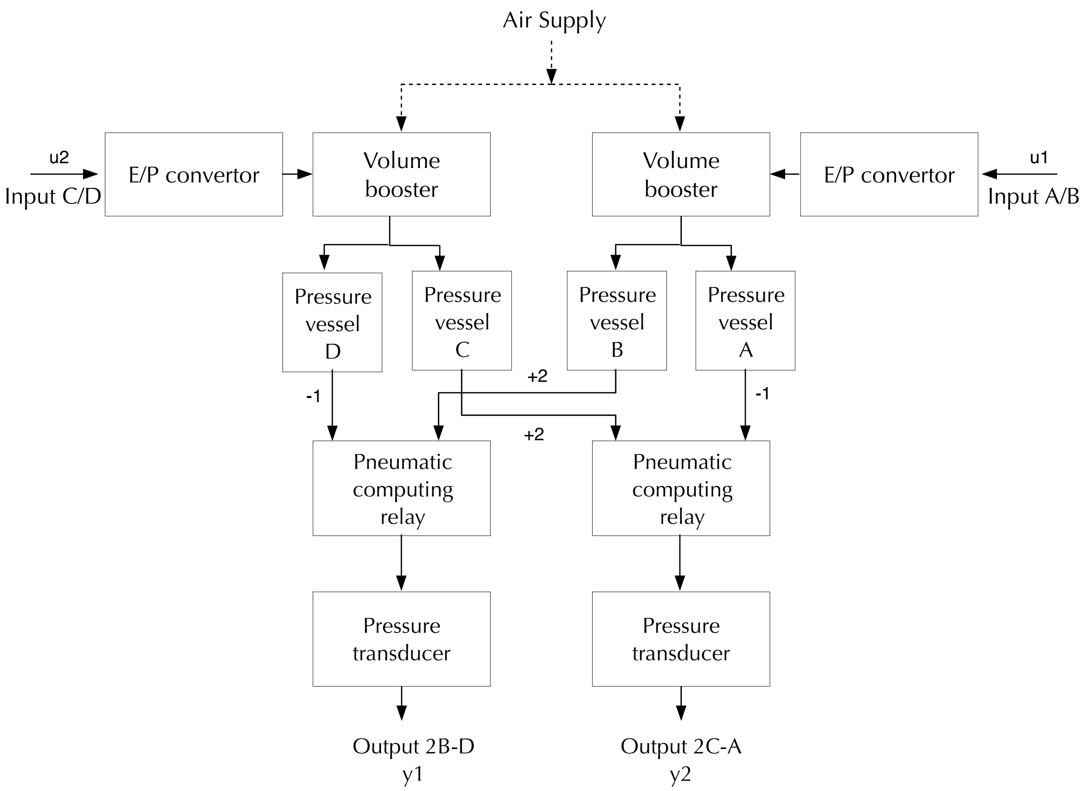

Figure 3.1.1: Multi-variable pneumatic system, (Dutton, 1997)

The lab machine to simulate this process consists of four pressure vessels with different dynamic behaviors. The four pressure vessels are fed in pairs from voltage input through voltage to pressure (E/P) converters. The machine has two outputs which depend on the pressure in two pressure vessels converted to a voltage signal by pressure transducer. (Dutton, 1997).

Interaction is present in this system since the outputs depend on combined pressure gauge readings of two pressure vessels that are not fed from the same input. Which means, the feeding the system from each input separately will influence both the outputs at the same time.

The other system components comprise of two volume boosters that amplifies the weak output of the voltage to pressure converters. And two pneumatic computing relays that produce an output equal to twice the gauge pressure in one vessel minus the gauge pressure in the other vessel. The model of this system was obtained by applying a step signal to each input and measuring the system responses on both the outputs. It can be noted that the four pressure vessels are simple cylinders fitted with a flow restriction, four simple first order time delays can represent this.

The E/P converters, and the volume boosters, shall have their own dynamics. However, they can be considered very fast compared to the slow dynamics of the pressure vessels. And they can be neglected accordingly.

Fitting four 1st order transfer function models to the lab machine measured response suggested the following approximate results:

$$\frac{Y_1}{U_1}(s) = \frac{1.02}{11.76s+1} = g_{11}(s) \qquad \text{... (3.1.1)}$$

$$\frac{Y_1}{U_2}(s) = \frac{-0.52}{10.25s+1} = g_{12}(s) \qquad \text{... (3.1.2)}$$

$$\frac{Y_2}{U_1}(s) = \frac{-0.54}{10.25s+1} = g_{21}(s) \qquad \text{... (3.1.3)}$$

$$\frac{Y_1}{U_1}(s) = \frac{1.04}{2.6s+1} = g_{22}(s) \qquad \text{... (3.1.4)}$$

Considering all the above, the open-loop system model can be in the form:

$$\mathbf{y}(\mathrm{s}) = \mathbf{G}(\mathrm{s})\mathbf{u}(\mathrm{s}) + \boldsymbol{\delta}(\mathrm{s})$$

Where

$$\mathbf{y}(\mathrm{s}) = \begin{pmatrix} \mathrm{y}_1(\mathrm{s}) \\ \mathrm{y}_2(\mathrm{s}) \end{pmatrix}$$

And, $\mathrm{y}_1(\mathrm{s})$ and $\mathrm{y}_2(\mathrm{s})$are the output signals related to the feed mixture temperature and the head level, respectively. And:

$$\mathbf{u}(\mathrm{s}) = \begin{pmatrix} \mathrm{u}_1(\mathrm{s}) \\ \mathrm{u}_2(\mathrm{s}) \end{pmatrix}$$

are the voltage input signals related to the feeding valves.

And the pneumatic open-loop system model mentioned above can be represented by:

$$\begin{bmatrix} \mathbf{g_{11}(s)} & \mathbf{g_{12}(s)} \\ \mathbf{g_{21}(s)} & \mathbf{g_{22}(s)} \end{bmatrix} = \mathbf{G(s)} \approx \begin{bmatrix} \frac{\mathbf{1.02}}{\mathbf{11.76s+1}} & \frac{\mathbf{-0.52}}{\mathbf{10.25s+1}} \\ \frac{\mathbf{-0.54}}{\mathbf{10.25s+1}} & \frac{\mathbf{1.04}}{\mathbf{2.6s+1}} \end{bmatrix} \quad \text{... (3.1.5)}$$

3.2 Open-loop Response and Control Objectives

In the last section, the open-loop transfer function matrix is developed as in equation (3.1.5), the open loop transfer function is represented in the simulation model shown in figure (3.2.1).

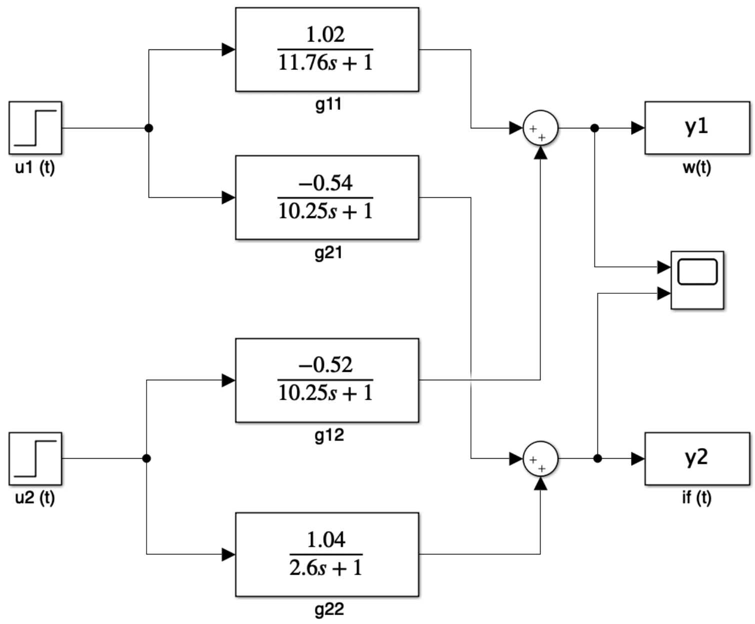

Figure 3.2.1: SIMULINK model for open-loop system

The response of the open-loop system for a unit step change on first and second inputs is simulated, and shown in figures 3.2.2, 3.2.3 respectively. These plots show that it takes the system about 50 seconds to reach steady state following a change in either $u_1(t)$, or $u_2(t)$.

Both outputs are noticed to be coupled in a considerable manner, it is noticed to reach about 50% for every input variation which is very high.

In this study, the controller will be designed to drive the system to steady state conditions in a time frame less than 10 seconds that is increasing the speed of reaction with no noticeable overshot problems and with less than 10% of steady state interaction. It is also required to minimize the steady-state output variations to unit step changes in $\boldsymbol{\delta}(t)$, which indicates a good disturbance rejection capability. Moreover, the design illustrated next chapter will aim to accomplish the above-mentioned improvements with minimum energy consumption and simple design controller for practical implementation purposes.

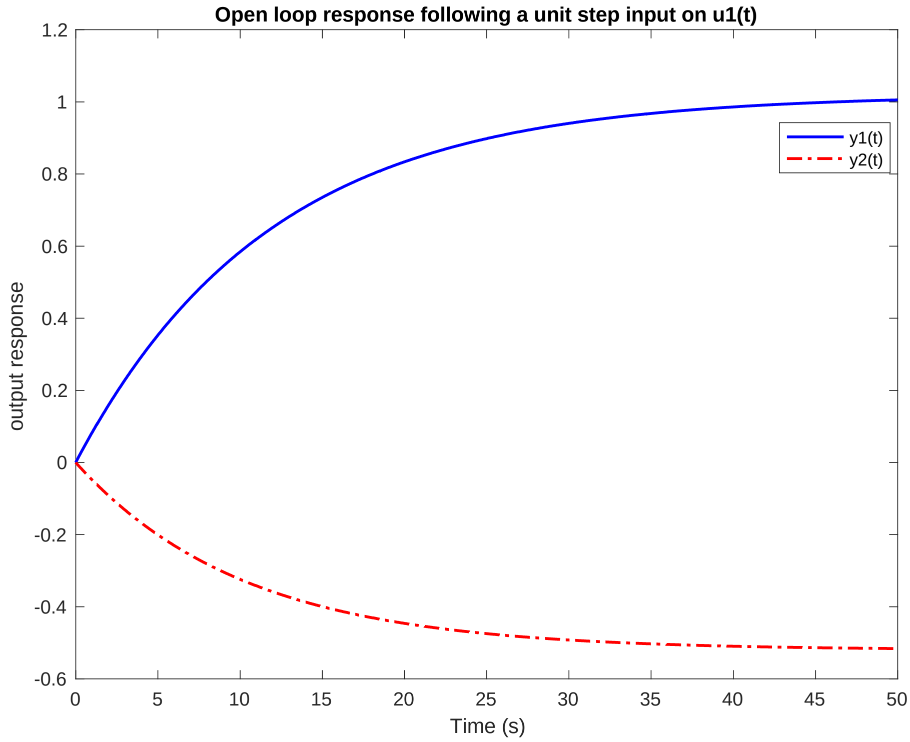

Figure 3.2.2: Response for a unit step change in u_1(open-loop)

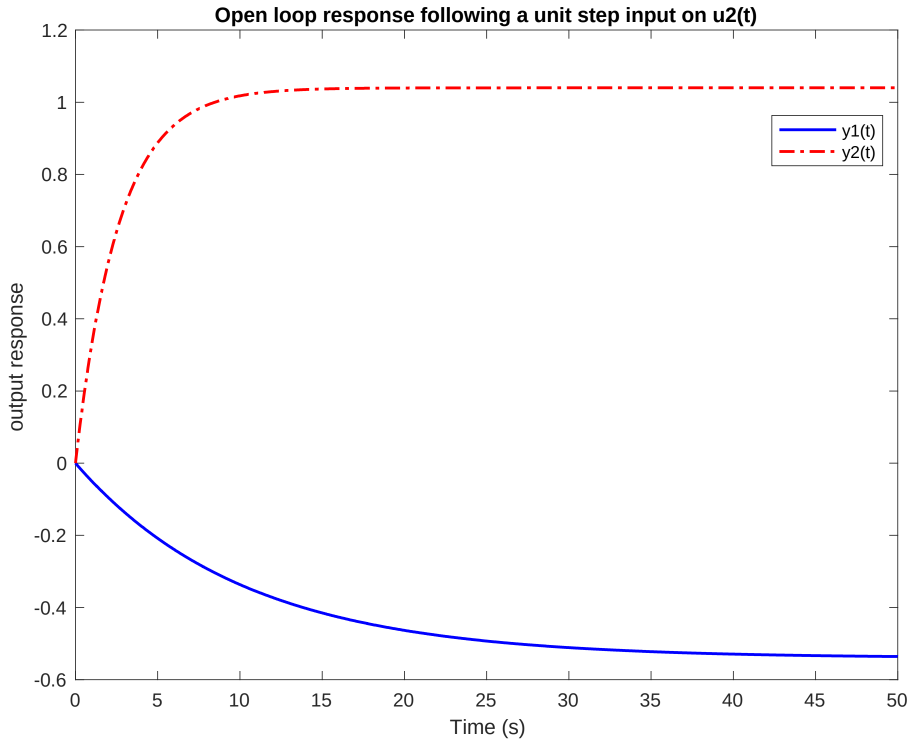

Figure 3.2.3: Response for a unit step change in u_2(open-loop)

3.3 Least Effort Control Method

In the design method outlined in this section, two loops will be used to archive the required system performance. An inner loop will work on the regulation task to maintain system stability and regulate the system dynamics, and an outer loop which will develop the system de-coupling and disturbance rejection criteria. The derivation outlined below is with accordance to least effort control method by Whalley and Ebrahimi (2006).

Design strategy

The open-loop system Laplace representation in the frequency domain is represented by:

$$\boldsymbol{y}(s) = \boldsymbol{G}(s).\boldsymbol{u}(s) + \boldsymbol{\delta}(s) \quad \text{... (3.3.1)}$$

And the control low of the proposed feedback (closed-loop):

$$u(s) = k(s)\big(\bar{r}(s) - h(s)y(s)\big) + \boldsymbol{P}(r(s) - \boldsymbol{F}y(s)) \quad \text{... (3.3.2)}$$

In equations 3.3.1 and 3.3.2 there are **m** independent inputs, disturbances, and outputs

Where $\mathbf{F} = \mathrm{Diag}(\mathrm{f}_1, \mathrm{f}_2, \cdots \mathrm{f}_\mathrm{m}),\ 0 < \mathrm{f}_\mathrm{j} < 1,\ 1 \leq \mathrm{j} \leq \mathrm{m}$ ... (3.3.3)

The inner loop control law is

$$\boldsymbol{k}(s)[\bar{r}(s) - \boldsymbol{h}(s)y(s)] \quad \text{... (3.3.4)}$$

If $\mathbf{k}(\mathrm{s})$ and $\mathbf{h}(\mathrm{s})$ were carefully designed. The inner loop will satisfy the desired system dynamics.

The control low of the outer loop is

$$\boldsymbol{P}(r(s) - \boldsymbol{F}y(s)) \quad \text{... (3.3.5)}$$

P and **F** are designed to achieve the required system steady state interaction and disturbance suppression.

Considering $\bar{r}(s) = 0$, inner and outer loops control law can be written as:

$$y(s) = \left(\boldsymbol{I}_m + \boldsymbol{G}(s)(\boldsymbol{k}(s) >< \boldsymbol{h}(s) + \boldsymbol{PF})\right)^{-1} \times (\boldsymbol{G}(s)\,\boldsymbol{P}r(s) + \boldsymbol{\delta}(s)) \qquad \text{... (3.3.6)}$$

In equation (3.3.6) $\|\mathbf{G}(s)(\mathbf{k}(s) >< \mathbf{h}(s) + \mathbf{PF})\|_\infty$is finite for all s on the D contour.

Now let

$y(0) = \mathbf{S}_s r(0),$

where $\mathbf{S}_s$ is the steady state matrix

substituting $S = 0$ in (3.3.6) yields

$$\boldsymbol{P} = (\boldsymbol{G}(0)^{-1} + \boldsymbol{k}(0) >< \boldsymbol{h}(0))\,\boldsymbol{S}_s(\boldsymbol{I} - \boldsymbol{F}\boldsymbol{S}_s)^{-1} \qquad \text{... (3.3.7)}$$

To achieve steady state de-coupling, the steady state matrix shall be ideally identity matrix $\mathbf{S}_s = \mathbf{I}_m$. On the other hand, to obtain lower SS interaction, consider unity diagonal elements with off-diagonal elements less than 1, $|s_{i,j}| \ll 1$.

Substituting $\mathbf{S}_s = \mathbf{I}_m$ for **P** in (3.3.7) yields in (3.3.6) to be

$$\mathbf{y}(s) = \{(\mathbf{I}_m + \mathbf{G}(s)[\mathbf{k}(s) >< \mathbf{h}(s) + (\mathbf{G}(0)^{-1} + \mathbf{k}(s) >< \mathbf{h}(s))(\mathbf{I_m} - \mathbf{F})^{-1}\mathbf{F}]\}^{-1} \times$$
$$\{(\mathbf{G}(s)\,\mathbf{P}r(s) + \boldsymbol{\delta}(s))\} \qquad \text{... (3.3.8)}$$

At low frequencies

$\mathbf{G}(s) \cong \mathbf{G}(0)$ and $\mathbf{G}(s)\mathbf{G}(0)^{-1} \cong \mathbf{I}_m$

Substituting in (3.3.8), $\mathbf{y}(s)$ will be in the following form:

$$y(s) \cong (\mathbf{I}_m + \mathbf{G}(s)\mathbf{k}(s) >< \mathbf{h}(s) + (\mathbf{I}_m + (\mathbf{I}_m - \mathbf{F})^{-1}\mathbf{F}))^{-1} \times (\mathbf{G}(s)\,\mathbf{P}r(s) + \boldsymbol{\delta}(s)) \text{... (3.3.9)}$$

Choosing **F** matrix such that $0 < f_m < 1$, the output will be:

$$y(s) \cong (1-\boldsymbol{f})(\boldsymbol{I}_m + \boldsymbol{G}(s)\boldsymbol{k} >< \boldsymbol{h}(s))^{-1} \times (\boldsymbol{G}(s)\,\boldsymbol{P}r(s) + \boldsymbol{\delta}(s)) \qquad \text{... (3.3.10)}$$

From equations (3.3.7) and (3.3.9)

$$\boldsymbol{G}(s)\boldsymbol{P} = \boldsymbol{G}(s)(\boldsymbol{G}(0)^{-1} + \boldsymbol{k}(0) >< \boldsymbol{h}(0))(\boldsymbol{I}_m - \boldsymbol{F})^{-1} \qquad \text{... (3.3.11)}$$

At low frequencies, this can be reduced to

$$\boldsymbol{G}(s)\boldsymbol{P} \cong \frac{1}{1-f}(\boldsymbol{I}_m + \boldsymbol{G}(s)\boldsymbol{k}(0) >< \boldsymbol{h}(0)) \qquad \text{... (3.3.12)}$$

Equation (3.3.12) under steady state condition becomes

$$\boldsymbol{y}(s) = \boldsymbol{I}_m r(s) + \boldsymbol{S}(s)\boldsymbol{\delta}(s) \qquad \text{... (3.3.13)}$$

And $\mathbf{S}(s)$ is the sensitivity matrix at low frequency

$$\boldsymbol{S}(s) = (1-f)\big(\boldsymbol{I}_m + \boldsymbol{G}(s)\boldsymbol{k}(s) >< \boldsymbol{h}(s)\big)^{-1} \quad 0 < f < 1 \qquad \text{... (3.3.14)}$$

It can be noted from (3.3.13) that $\mathbf{S}_\mathrm{s} = \mathbf{I}_\mathrm{m}$, which means steady state de-coupling requirement will be granted despite any changes in $\mathrm{r(s)}$. Moreover, from (3.3.14), increasing f without exceeding unity the system response shall be enhanced in terms of disturbance rejection criteria.

For easy implementation purposes, a conventional multivariable regulator can be assumed with $\mathbf{K}(\mathrm{s})$, and $\mathbf{H}(\mathrm{s})$ forward path and feedback gains, respectively. Then the closed-loop system will be:

$$\boldsymbol{y}(s) = \big(\boldsymbol{I}_m + \boldsymbol{G}(s)\boldsymbol{K}(s)\boldsymbol{H}(s)\big)^{-1}(\boldsymbol{G}\,\boldsymbol{K}(\boldsymbol{s})\boldsymbol{r}(s) + \boldsymbol{\delta}(s)) \qquad \text{... (3.3.15)}$$

Comparing equations (3.3.6) and (3.3.15) yields in

$$\mathbf{K}(s) = \mathbf{P} \qquad \text{... (3.3.16)}$$

$$\mathbf{K(s)H(s)} = \mathbf{k(s)} >< \mathbf{h(s)} + \mathbf{PF}$$

and

$$\mathbf{H(s)} = \mathbf{P^{-1}k(s)} >< \mathbf{h(s)} + \mathbf{F} \qquad \text{... (3.3.17)}$$

Where $\mathbf{K(s)}$ and $\mathbf{H(s)}$ are constant full rank $\mathrm{m} \times \mathrm{m}$ matrices, also, the feedback matrix $\mathbf{H(s)}$ is a stable proper easy to construct matrix.

"In view of the above theory, the design methodology implemented here is to adjust the inner loop vectors $\mathbf{k}(\mathrm{s})$ and $\mathbf{h}(\mathrm{s})$ to achieve the desired system dynamics. After that, to achieve acceptable steady state coupling condition a pre-compensator **P** to be configured. Finally, outer loop feedback gain **f** to be selected to achieve final systems dynamics and acceptable disturbance rejection". (Whalley and Ebrahimi, 2006)

Inner loop design:

In this section, the design strategy for configuring the pre-compensator **P** and adjusting inner loop vectors $\mathbf{k}(\mathrm{s})$ and $\mathbf{h}(\mathrm{s})$ will be illustrated.

The Laplace transfer open-loop system given in (3.3.1) is assumed to be m × m square linear, proper or strictly proper that can be factorized as:

$$\boldsymbol{G}(s) = \boldsymbol{L}(s)\frac{A(s)}{d(s)}\boldsymbol{R}(s)\boldsymbol{\Gamma}(\mathrm{s}) \qquad \text{... (3.3.18)}$$

Where $\mathbf{L}(\mathrm{s})$, $\mathbf{A}(\mathrm{s})$, $\mathbf{R}(\mathrm{s})$, $\mathbf{\Gamma}(\mathrm{s})$, and the elements of $\frac{\mathrm{A(s)}}{\mathrm{d(s)}} \in \mathrm{H}_\infty$, $\mathrm{s} \in \mathbb{C}$

In equation (3.3.18), $\mathbf{L}(\mathrm{s})$ consists of $\mathbf{G}(\mathrm{s})$ left row factors

$$\mathbf{L}(\mathrm{s}) = \mathrm{Diag}\left(\frac{\gamma_\mathrm{j}(\mathrm{s})}{\mathrm{p_j(s)}}\right)$$

And $\mathbf{R}(\mathrm{s})$ consists of $\mathbf{G}(\mathrm{s})$ right column factors

$$\mathbf{R}(s) = \mathrm{Diag}(\frac{\rho_j(s)}{q_j(s)})$$

And $\mathbf{\Gamma}$(s) is the transformed finite time delay elemnt

$$\mathbf{\Gamma}(\boldsymbol{s}) = \mathrm{Diag}(e^{-sT_j}), \quad 1 \le j \le m$$

And $\mathbf{A}(s)$ is a non-singular matrix of rational functions where $\det \mathbf{A}(s) \neq 0$ with elements

$$a_{ij}(s) = a_{ij}s^{m-1} + b_{ij}s^{m-1} + \cdots + \gamma_{ij} \quad 1 \le i, j \le m$$

Since the output response is:

$$y(s) = \mathbf{G}(s)u(s) + \boldsymbol{\delta}(s) \qquad \text{... (3.3.19)}$$

Considering the inner loop control low to be

$$\mathbf{u}(s) = \mathbf{k}(s)\ [\bar{r}(s) - \mathbf{h}(s)\ y(s)] \qquad \text{... (3.3.20)}$$

Combining (3.3.19) and (3.3.20)

$$y(s) = \left(\boldsymbol{I}_m + \boldsymbol{G}(s)\boldsymbol{k}(s) >< \boldsymbol{h}(s)\right)^{-1}(\boldsymbol{G}(s)\boldsymbol{k}(s)\bar{r}(s) + \boldsymbol{\delta}(s)) \qquad \text{... (3.3.21)}$$

$\mathbf{\Gamma}(\boldsymbol{s})$ can be ordered with $T_i \ge T_j$ with $1 \le j \le m$, and $i \neq j$ then the forward path vector will become

$$\boldsymbol{k}(s) = \left(k_1(s)e^{-s(T_i-T_j)}, k_2(s)e^{-s(T_i-T_j)}, \dots \dots, k_1, \dots \dots ., k_m(s)e^{-s(T_i-T_j)}\right)^T \qquad \text{... (3.3.22)}$$

$$\text{Since } \boldsymbol{h}(\boldsymbol{s}) = \left(h_1(s), h_2(s), \dots \dots h_m(s)\right) \qquad \text{... (3.3.23)}$$

Let $k_j(s) = k_j\emptyset_j(s)$ and $h_j(s) = h_jx_j(s)$, $1 \le j \le m$

Where $\emptyset_j(s)$ and $x_j(s)$ are proper or strictly proper, stable and minimum phase realization, then they may be chosen such that equation (3.3.21) becomes

$$y(s) = \left(I_m + e^{-sT_i} n(s)L(s)\frac{A(s)}{d(s)}k(s) >< h(s)\right)^{-1} \times \left(n(s)L(s)\frac{A(s)}{d(s)}k(s)e^{-sT_i}r(s) + \delta(s)\right)$$

... (3.3.24)

Where $\mathbf{k}(\boldsymbol{s}) = (\mathrm{k}_{\mathbf{1}}, \mathrm{k}_{\mathbf{1}}, \dots \dots \mathrm{k}_{\mathrm{m}})^{\mathrm{T}}$ ... (3.3.25)

$$\boldsymbol{h}(\boldsymbol{s}) = (h_1, h_2, \dots \dots, h_m) \qquad \text{... (3.3.26)}$$

$\mathrm{d}(\boldsymbol{s}) = \mathrm{s}^{\mathrm{k}} + \mathrm{a}_1 \mathrm{s}^{\mathrm{k}-1} + \cdots + \mathrm{a}_0$

$\deg\big(\mathrm{n(s)a_{ij}(s)} < \mathrm{k}\big), 1 \leq \mathrm{i,j} \leq \mathrm{m}$

Det. Of equation (3.3.24) is

$$det\,[\boldsymbol{I}_m + e^{-sT_i} n(s)\boldsymbol{L}(s)\frac{\boldsymbol{A}(s)}{d(s)}\boldsymbol{k}(s) >< \boldsymbol{h}(s)] = 1 + e^{-sT_i} n(s) < \boldsymbol{h}\frac{\boldsymbol{A}(s)}{d(s)}\boldsymbol{k} > \qquad \text{... (3.3.27)}$$

Where the inner production in (3.3.27) equals

$$< \boldsymbol{h}\frac{\boldsymbol{A}(s)}{d(s)}\boldsymbol{k} > = [1, s, \dots, s^{m-1}] \times \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{mm} \\ \vdots & \ddots & \vdots \\ b_{11} & \cdots & b_{mm} \\ a_{11} & \dots & a_{mm} \end{bmatrix} \times \begin{bmatrix} k_1 h_1 \\ k_2 h_1 \\ \cdot \\ \cdot \\ k_m h_m \end{bmatrix} \qquad \text{... (3.3.28)}$$

If the gain ratios in equation (3.3.28) satisfy

$$\mathrm{k}_2 = \mathrm{n}_1 \mathrm{k}_1, \dots \dots \dots \dots \dots \ \mathrm{k}_{\mathrm{m}} = \mathrm{n}_{\mathrm{m}-1} \mathrm{k}_1 \qquad \text{... (3.3.29)}$$

And

$$< \boldsymbol{hA}(s)\boldsymbol{k} > = b(s) \qquad \text{... (3.3.30)}$$

Then, equation (3.3.30) implies that

$$k_1[\boldsymbol{Q}]\boldsymbol{h} = (b_{m-1}, b_{m-2}, \dots, b_0)^T \qquad \text{... (3.3.31)}$$

Where

$$\mathbf{Q} = \begin{bmatrix} \gamma_{11} + \gamma_{12}n_1 + \gamma_{1m}n_{m-1} & \vdots & \gamma_{21} + \gamma_{22}n_1 + \gamma_{2m}n_{m-1} & \vdots & \cdots & \gamma_{m1} + \gamma_{m2}n_1 + \gamma_{mm}n_{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{11} + b_{12}n_1 + b_{1m}n_{m-1} & \vdots & b_{21} + b_{22}n_1 + b_{2m}n_{m-1} & \vdots & \cdots & b_{m1} + b_{m2}n_1 + b_{mm}n_{m-1} \\ a_{11} + a_{12}n_1 + a_{1m}n_{m-1} & \vdots & a_{21} + a_{22}n_1 + a_{2m}n_{m-1} & \vdots & \cdots & a_{m1} + a_{m2}n_1 + a_{mm}n_{m-1} \end{bmatrix}$$

And b_j, $0 \le j \le m-1$ are the coefficients of $b(s)$given in equation (3.3.30), considering that $n_1, n_2, \dots, n_{m-1}$ is selectable in equation (3.3.30) so that the unique solution for $(h_1, h_2, \dots, h_m)k_1$ exists and the matrix is invertible.

So, if suitable $b(s)$ function selected, and gain ratios, the closed-loop dynamics defined in equation (3.3.24) will be defined. And if equation (3.3.30) is solved, then “h” can be calculated based on the selection of k_1.

Optimization of least effort (minimum energy)

In view of the above, the closed-loop model has been established. Free choice of $n_1, n_2, \dots, n_{m-1}$ as gain ratios indicates the freedom to optimize this process.

A benchmark to be considered initially is achieving the disturbance rejection and in the same time maintaining the inner loop required dynamics, with absolute minimum (least) control effort provided the constraint that the controller model generates a particular polynomial.

The controller effort at time t is proportional to:

$$(|k_1h_1| + |k_2h_1| + \cdots |k_mh_1|)|y_1(t)| + (|k_1h_2| + |k_2h_2| + \cdots |k_mh_2|)|y_2(t)| + \cdots$$
$$+ (|k_1h_m| + |k_2h_m| + \cdots |k_mh_m|)|y_m(t)|$$

So, the control energy cost is proportional to:

$$E(t) = \int_{t=0}^{t=T_f} \left(\sum_{i=1}^{m} k_i^2 \sum_{j=1}^{m} h_j^2 \, y_j^2(t) \right) dt \qquad \dots (3.3.32)$$

Then the changes in the transformed output y following arbitrary disturbances:

$$J = \sum_{i=1}^{m} k_i^2 \sum_{j=1}^{m} h_i^2 \qquad \dots (3.3.33)$$

By minimizing the performance index in equation 3.3.33, the required control energy in 3.3.32 would be minimized.

If

$k_2 = n_1 k_1, k_3 = n_2 k_1 \dots k_m = n_{m-1} k_1$

Then the performance index J in (3.3.33) can be re-written as

$$J = (k_1)^2 (1 + n_1^2 + n_2^2 + \dots + n_{m-1}^2) \times (h_1^2 + h_2^2 + \dots + h_m^2) \qquad \dots (3.3.34)$$

And $(h_1^2 + h_2^2 + \dots + h_m^2) < h, h >$

The closed-loop determinant of equation (3.3.27) and inner product in equation (3.3.30) and from equation (3.3.31)

$$\mathbf{h} = \mathbf{k}_1^{-1} \mathbf{Q}^{-1} \mathbf{b} \qquad \dots (3.3.35)$$

Substituting for $\mathbf{h}$ in equation (3.3.35), equation (3.3.34), becomes:

$$J = (1 + n_1^2 + n_2^2 + \dots + n_{m-1}^2) \mathbf{b}^T (\mathbf{Q}^{-1})^T \mathbf{Q}^{-1} \mathbf{b} \qquad \dots (3.3.36)$$

For 2×2 system, $m = 2$, and

$$J = (1 + n_1^2) \mathbf{b}^T (\mathbf{Q}^{-1})^T \mathbf{Q}^{-1} \mathbf{b} \qquad \dots (3.3.37)$$

With J is minimized when

$$\frac{\partial J}{\partial n_1} = 0, \qquad \frac{\partial J}{\partial n_2} = 0$$

$$\frac{\partial^2 J}{\partial n_1^2} \frac{\partial^2 J}{\partial n_2^2} - \left(\frac{\partial^2 J}{\partial n_1 \partial n_2} \right)^2 > 0, \text{if } \frac{\partial^2 J}{\partial n_1^2} > 0$$

Disturbance Rejection

Designing for minimum control energy is not generally enough to achieve the disturbance recovery conditions. To achieve maximum disturbance suppression, the outer-loop feedback gain can be tuned between 0 and 1 as in equation (3.3.14).

With f being adjusted, the system transient response will be changing, the target in this stage is to select feedback that gives the best transient response.

Stability of the combined system

The stability of the system depends on the denominator of the input-output relationship given in equation (3.3.8).

For simple analysis purpose, if $f_{1,}f_{2,\dots\dots\dots,}f_m = f$,

Then the denominator will be

$$\det\left\{\mathbf{I}_m + \mathbf{G}(s)\left[\frac{k(s)><h(s)}{(1-f)} + \frac{G(0)^{-1}f}{(1-f)}\right]\right\} \quad \dots (3.3.38)$$

from equation 3.3.37, it is noticable that, when $f \rightarrow 1$, the feedback compensator matrix elements $\left[\frac{k(s)><h(s)}{(1-f)} + \frac{G(0)^{-1}f}{(1-f)}\right]$ approaches infinity resulting in instability.

Selecting f to be $0 < f \leq 0.5$, increases the impact of the inner loop feedback gain, and reduces the impact of the outer loop. However, selecting f to be $0.5 < f \leq 1$, leads to amplifying the effect of both loops' gain.

3.4 H-infinity control approach

H-infinity control technique is widely used as it applies to control design problems involving multivariate systems with coupling between channels. "It is a design technique with a state-space computational solution that utilizes frequency-dependent weighting functions to tune the controller's performance and robustness characteristics." (John Bibel, 1992).

H-infinity method is based on weighted control arrangement as in figure (3.4.1). Three weighting functions are used, $W_1(s)$, $W_2(s)$, and $W_3(s)$. $W_1(s)$ is the weighting function on the sensitivity transfer function. It is choosen to obtain a required performance characteristics (time response).

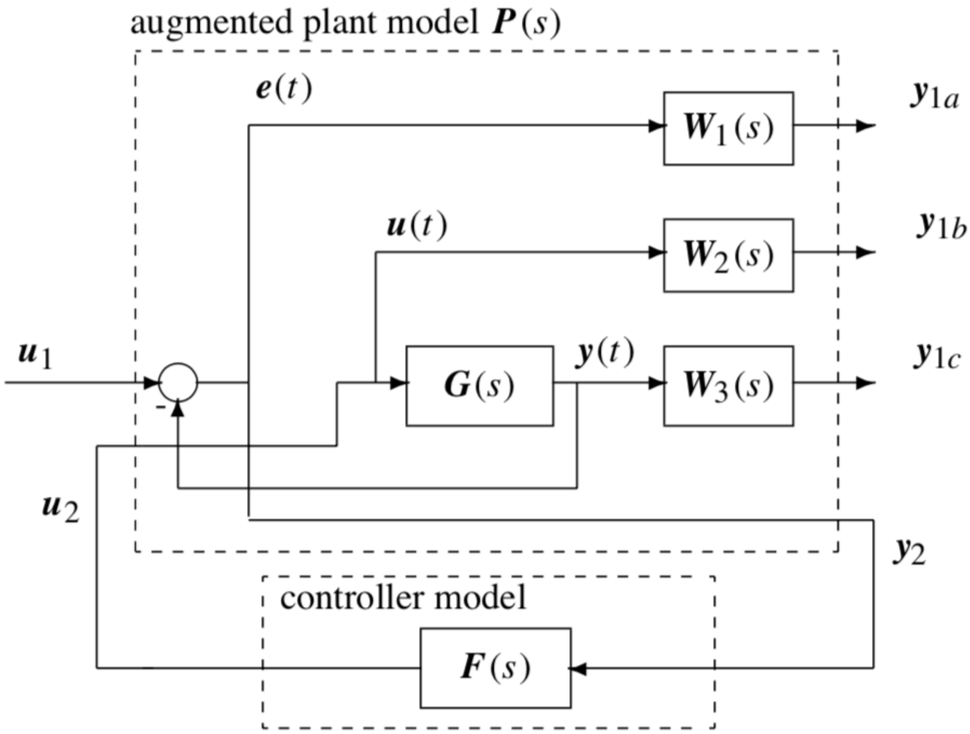

Figure 3.4.1: Block diagram of standard H-infinity feedback control.

(Dingyu X, YangQuan C and Derek P, 2007)

At low frequencies, $W_1(s)$ is designed with low gain a for good tracking and with high gain at high frequencies for limiting the overshoot.

$W_1(s)$ is designed with high gain at high frequencies for limiting the overshoot. On the other hand, limiting the overshot mean increasing the system damping which will affect the system speed of response in a negative manner. Mainly, a tradeoff to be set between limiting overshot and system speed of response. Usually at low frequencies, $W_1(s)$ is designed with low gain for good tracking and with high gain at high frequencies for limiting the overshoot.

The complementary sensitivity function T gives the response of the system to reference and noise inputs. T need to be ≈ 1 to achieve propper tracking and near zero noise. High pass weight $W_2(s)$is usually used on T to achieve insensitivity to noise since most of the noise energy is located at high frequency, where input signal usually comes at lower frequencies. The design purpose is to keep $T \approx 1$ at low frequencies and low at high frequencies.

Assuming that $G(s)$, $W_1(s)$, and $W_3(s)G(s)$ are all proper functions. To be mentioned that $W_3(s)$ itself is not required to be proper. "The output vector $y_1 = [y_{1a}, y_{1b}, y_{1c}]$ is not used directly to construct the control signal vector $u(t)$. We should understand that y_1 is actually for the control system performance measurement, So, it is not strange to include the filtered "input signal" $u(t)$ in the "output signal" y_1 because one may need to measure the control energy to assess whether the designed controller is good or not " (Dingyu X, YangQuan C, and Derek P, 2007).

Considering a general robust system's augmented plant model represented as

$$P(s) = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \quad \text{... (3.4.1)}$$

With the following augmented state space

$$\dot{x} = Ax + [B_1\ B_2]\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \text{ and}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \text{... (3.4.2)}$$

The system closed-loop transfer function will be

$$T_{y_1u1}(s) = \mathbf{P}_{11}(s) + \mathbf{P}_{12}(s)[\mathbf{I} - \mathbf{F}(s)\mathbf{P}_{22}(S)]^{-1}\mathbf{F}(s)\mathbf{P}_{21}(s) \quad \text{... (3.4.3)}$$

The form in equation (3.4.3) is called the LFT (linear fractional transformation of the interconnected system), and the goal of control in this section is to find a controller $u_2(s) = \mathbf{F}y_2(s)$ such that $\|T_{y_1u1}\| < 1$ and $\min_{\mathbf{F}(s)}\|T_{y_1u1}\|_\infty$.

Given the system with state space model as $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$.

Let state space representation for $\mathbf{W_1}$:

$\mathbf{W_1} = (\mathbf{A_{w1}}, \mathbf{B_{w1}}, \mathbf{C_{w1}}, \mathbf{D_{w1}})$

and $\mathbf{W_2}$:

$\mathbf{W_2} = (\mathbf{A_{w2}}, \mathbf{B_{w2}}, \mathbf{C_{w2}}, \mathbf{D_{w2}})$

and $\mathbf{W_3}$**:**

$\mathbf{W_3} = \mathbf{C_{w3}}(s\mathbf{I} - \mathbf{A_{w3}})^{-\mathbf{1}}\mathbf{B_{w3}} + \mathbf{P}_m s^m + \cdots + \mathbf{P}_1 s + \mathbf{P_0}$

Also, considering a general mixed sensitivity problem where all three weighting functions present, equation (3.4.1) can be re-written as:

$$P(s) = \begin{bmatrix} W_1 & -W_1 G \\ 0 & W_2 \\ 0 & W_3 G \\ I & -G \end{bmatrix} \qquad \text{... (3.4.4)}$$

And the LFT for this problem can be represented by

$$T_{y_1 u1} = \begin{bmatrix} W_1 S \\ W_2 FS \\ W_3 T \end{bmatrix} \qquad \text{... (3.4.5)}$$

In optimal $\mathcal{H}_\infty$controller design, the optimal criterion is defined as

$$\max_{\boldsymbol{\gamma}} \left\| T_{y_1 u1} \right\| < \frac{1}{\boldsymbol{\gamma}}$$

And in general

$$\max_{\boldsymbol{\gamma}} \begin{bmatrix} W_1 S \\ W_2 FS \\ W_3 T \end{bmatrix} \leq \frac{1}{\boldsymbol{\gamma}} \qquad \text{... (3.4.6)}$$

Where $\mathbf{F}(s)$is the controller to be designed, $\mathbf{S}(s)$ is the sensitivity transfer function

$$\mathbf{S}(s) = [\mathbf{I} + \mathbf{F}(s)\mathbf{G}(s)]^{-1} \qquad \text{... (3.4.7)}$$

And $\mathbf{T}(s)$is the complementary sensitivity transfer function

$$\mathbf{T}(s) = \mathbf{I} - \mathbf{S}(s) = \mathbf{F}(s)\mathbf{G}(s)[\mathbf{I} + \mathbf{F}(s)\mathbf{G}(s)]^{-1} \qquad \text{... (3.4.8)}$$

All three terms in the matrix of (3.4.6) can be individually weighted by γ. An iteration method, known as the γ -iteration method, can be used in finding the optimal γ.

Chapter IV: Controller Design

4.1 Design of least effort controller

Based on the methodology presented in section 3.3 in the previous chapter, direct implementation on the mixing system model will be discussed here. The transfer function representing the model as found in the last chapter is

$$G(s) = \begin{bmatrix} \frac{1.02}{11.76s+1} & \frac{-0.52}{10.25s+1} \\ \frac{-0.54}{10.25s+1} & \frac{1.04}{2.6s+1} \end{bmatrix} \qquad \text{... (4.1.1)}$$

To apply least effort design, the open-loop transfer function $G(s)$ can be re-written as

$$G(s) = L(s)\frac{A(s)}{d(s)}R(s)\Gamma(s) \qquad \text{... (4.1.2)}$$

For simplicity, the structure in equation (4.1.3) can be achieved with $L(s)$ matrix with “null” off-diagonal elements and first row comprises of the multiplication of first row denominators and second row which is a multiplication of second-row denominators as below

$$L(s) = \begin{bmatrix} \frac{1}{(11.76s+1)(10.25s+1)} & 0 \\ 0 & \frac{1}{(2.6s+1)(10.25s+1)} \end{bmatrix} \qquad \text{... (4.1.3)}$$

$$\text{With } R(s) = \Gamma(s) = I \qquad \text{... (4.1.4)}$$

Consequently,

$$A(s) = \begin{bmatrix} 1.02(10.25s+1) & -0.52(11.76s+1) \\ -0.54(2.6s+1) & 1.04(10.25s+1) \end{bmatrix} \qquad \text{... (4.1.5)}$$

If the inner loop control low mentioned as in equation (3.3.20) applied

$$\mathbf{u}(s) = \mathbf{k}(s)[\bar{r}(s) - \mathbf{h}(s)y(s)]$$

Considering no time delay, the closed-loop equation will become

$$y(s) = \left(\boldsymbol{I}_m + \boldsymbol{G}(s)\boldsymbol{k}(s) >< \boldsymbol{h}(s)\right)^{-1}\boldsymbol{G}(s)\bar{r}(s) \quad \text{... (4.1.6)}$$

Similar to equation (3.3.27) considering $\mathrm{R(s) = \Gamma(s) = I}$

$$det\,[\boldsymbol{I}_m + \boldsymbol{L}(s)\frac{\boldsymbol{A}(s)}{d(s)}\boldsymbol{k}(s) >< \boldsymbol{h}(s)] = 1 + < \boldsymbol{h}\frac{\boldsymbol{A}(s)}{d(s)}\boldsymbol{k} > \quad \text{... (4.1.7)}$$

In equation (4.1.7), Similar to equation (3.3.23)

$$\mathbf{h}(\boldsymbol{s}) = \left(\mathrm{h_1(s), h_2(s), \dots \dots h_m(s)}\right)$$

Let $\boldsymbol{h}(\boldsymbol{s}) = \left(\frac{11.76s+1}{2.6s+1}h_1, h_2\right)$ ... (4.1.8)

Then, with phase advance compensator suggested in equation (4.1.8)

$$< \mathbf{h}(\mathrm{s})\frac{\mathbf{A}(\mathrm{s})}{\mathrm{d(s)}}\mathbf{k} > = [\mathrm{h_1, h_2}]$$

And so

$$\frac{\mathrm{A(s)}}{\mathrm{d(s)}} = \frac{\begin{bmatrix} 1.02(10.25\mathrm{s}+1) & -0.52(11.76\mathrm{s}+1) \\ -0.54(2.6\mathrm{s}+1) & 1.04(10.25\mathrm{s}+1) \end{bmatrix}}{(10.25\mathrm{s}+1)(2.6\mathrm{s}+1)}$$

With $\mathbf{h}(\boldsymbol{s}) = [\mathrm{h_1(s), h_2}]$ ... (4.1.9)

$\boldsymbol{k} = [k_1, k_2]$ ... (4.1.10)

And $\mathrm{d(s) = (10.25s + 1)(2.6s + 1)}$ ... (4.1.11)

After changing the system to the required format of equation (4.1.2), the inner loop regulator design can be considered to improve the system transients, the inner-loop will be implemented as in equation (3.3.30) after including $\mathrm{d(s)}$ as follows

$$< h\frac{A(s)}{d(s)}k > = \frac{b(s)}{d(s)} \quad \text{... (4.1.12)}$$

Then, equation 4.1.12 can be considered as the system characteristic equation

$$-1 = \frac{b(s)}{d(s)} = \frac{(s+0.09)b_0}{(10.25s+1)(2.6s+1)} \quad \text{... (4.1.13)}$$

Where, $d(s) = (10.25s + 1)(2.6s + 1)$, the poles of the system reside at $s_1 \approx 0.385$and $s_2 \approx 0.098$. It is clear that s_2 is slower and by choosing

$$b(s) = (s + 0.09)b_0 \qquad \text{... (4.1.14)}$$

The slowest system pole effect will be canceled moreover, the other pole at $s_1 \approx 0.385$ will be allowed to migrate along the negative real axis with increased values of b_0 which leads to the enhancement of the system speed and stability refer to root locus plot at figure (4.1.1). b_0 was selected to be 0.33. In the coming steps of design, selection of $f > 0$, will result in larger b_0 as $b_0 = \frac{b_0}{(1-f)}$. which will result in higher performance index J maximum, giving improved disturbance rejection properties.

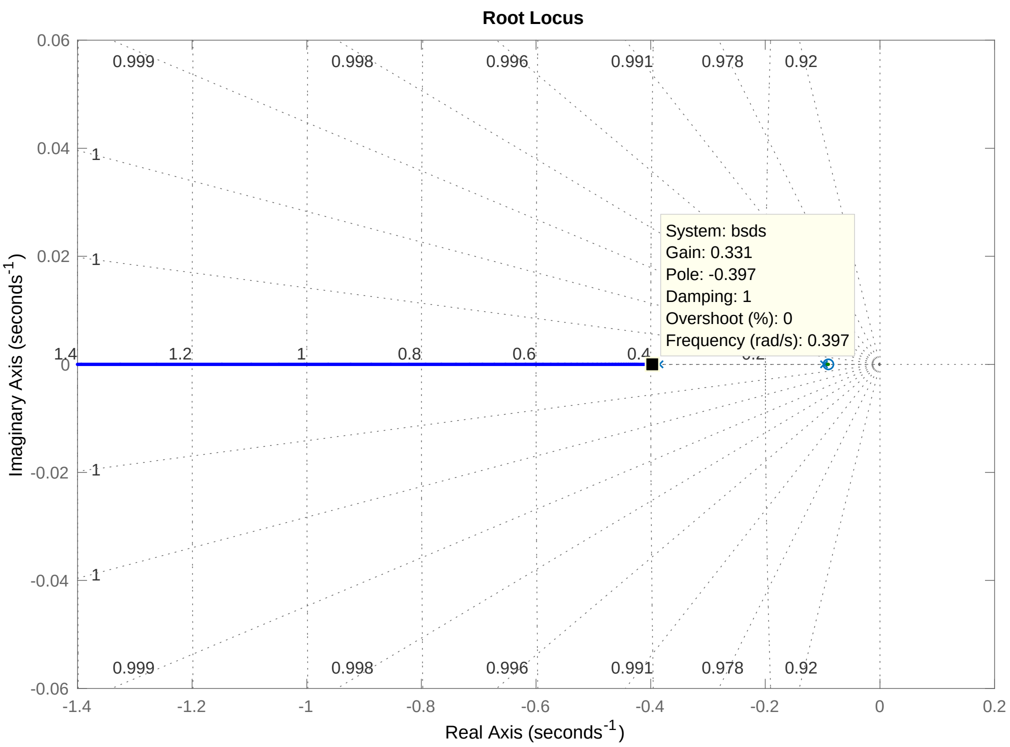

Figure 4.1.1: Root Locus for equation (4.1.13)

From equation (4.1.14) and after selecting $b_0 = 0.33$

$$\mathbf{b}(s) = \begin{bmatrix} 0.09 \\ 1 \end{bmatrix} 0.33 = \begin{bmatrix} 0.0297 \\ 0.33 \end{bmatrix} \qquad \text{... (4.1.15)}$$

Now let $k_1 = 1$

From A(s) given in equation (4.1.5), the Q matrix of this system can be formulated in accordance with equation (3.3.28) as

$$< h.A(s).k > = [h_1, h_2] \times \begin{bmatrix} 1.02(10.25s + 1) & -0.52(11.76s + 1) \\ -0.54(2.6s + 1) & 1.04(10.25s + 1) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

$$= k_1\big((10.455s + 1.02)h_1 - (1.404s + 0.54)h_2\big) + k_2((10.66s + 1.04)h_2 - (6.1152s + 0.52)h_1)$$

$$= [1 \quad s] \begin{bmatrix} 1.02 & -0.052 & -0.54 & 1.04 \\ 10.455 & -6.1152 & -1.404 & 10.66 \end{bmatrix} \begin{bmatrix} k_1h_1 \\ k_2h_1 \\ k_1h_2 \\ k_2h_2 \end{bmatrix}$$

With

$k_1 = 1$, $k_2 = nk_1$, (n is the gain ratio)

$$= [1 \quad s] \begin{bmatrix} 1.02 & -0.052 & -0.54 & 1.04 \\ 10.455 & -6.1152 & -1.404 & 10.66 \end{bmatrix} \begin{bmatrix} h_1 \\ nh_1 \\ h_2 \\ nh_2 \end{bmatrix}$$

Since $< h.A(s).k > = \mathbf{Q}[h_1, h_2]^T$

Then

$$Q = \begin{bmatrix} 1.02 - 0.52n & -0.54 + 1.04n \\ 10.455 - 6.1152n & -1.404 + 10.66n \end{bmatrix} \qquad \text{... (4.1.16)}$$

As m = 2, the performance index to be minimized is

$$J = (1 + n^2)\mathbf{b}^T(\mathbf{Q}^{-1})^T\mathbf{Q}^{-1}\mathbf{b} \qquad \text{... (4.1.17)}$$

And after substituting with

Q from equation (4.16), and $\mathbf{b}(s)$ from (4.1.15), with $k_1 = 1$

$$J = 741.2 \frac{(1 + n^2)(4.4329 - 1.5467n + 0.1854n^2) \times 10^{-3}}{(5.265 - 3.215n + 1.021n^2)^2}$$

To find J_{min} deriving J with respect to n

$$\frac{dJ}{dn} = 0.1482 \times 10^4 \times \frac{(10.18 + 12.78n - 9.847n^2 + 0.1934n^4)}{(5.265 - 3.215n + 1.021n^2)^3}$$

Leting $\frac{dJ}{dn} = 0$ and solving the equation

Results in the extremum values of J as n = -7.026, -0.5598, 1.8797, 7.1267

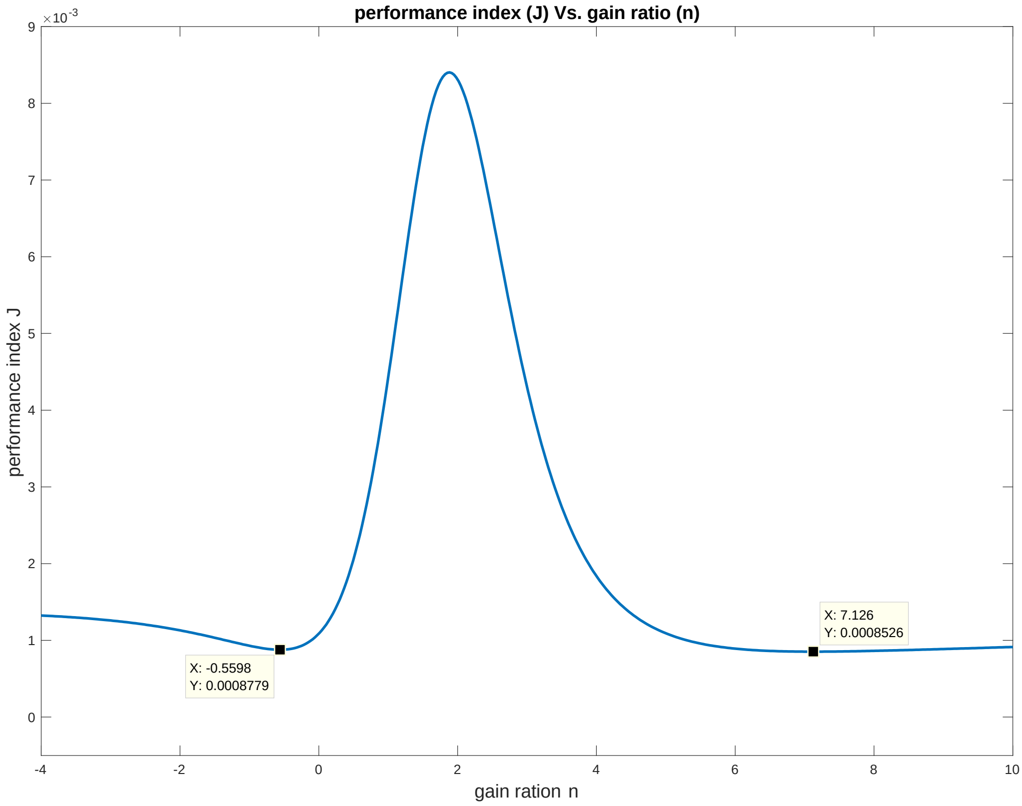

Figure 4.1.2: Performance Index J vs. gain ratio (n)

From the graph in figure (4.1.2) above, it can be shown that min J is when n = 7.1267.

Now using equation (3.3.31):

$k_1[\mathbf{Q}]\mathbf{h} = (b_{m-1}, b_{m-2}, \dots, b_0)^T$, After substituting with

Q from equation (4.16), and $\mathbf{b}(s)$ from (4.1.15), with $k_1 = 1$

The inner loop feedback gain h will be

$$\mathbf{h} = \mathbf{Q}^{-1}b = \left(\begin{bmatrix} 1.02 - 0.52n & -0.54 + 1.04n \\ 10.455 - 6.1152n & -1.404 + 10.66n \end{bmatrix}^{-1} \times \begin{bmatrix} 0.0297 \\ 0.33 \end{bmatrix}\right)_{n=7.1267}$$

$$\mathbf{h} = [h_1, h_2] = [-0.0019, 0.0036] \qquad \dots (4.1.18)$$

And the inner feedforward gain k with $n = 7.1267$ selected as a gain ratio, and $k_1$selected arbitrary as 1, $k_2 = nk_1$

$$\mathbf{k} = \begin{bmatrix} 1 \\ 7.1267 \end{bmatrix} \qquad \dots (4.1.19)$$

Moving on to the outer loop design, the outer loop as highlighted comprises of a forward and feedback path compensator

Letting

$$\mathbf{F} = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix}$$

And as the design requires, the steady state interaction of the system is not to exceed -10%, steady-state matrix is assumed to be

$S_s = \begin{bmatrix} 1 & -0.1 \\ -0.1 & 1 \end{bmatrix}$, from equation (3.3.7):

$$\mathbf{P} = (\mathbf{G}(0)^{-1} + \mathbf{k}(0) >< \mathbf{h}(0))\, \mathbf{S}_s(\mathbf{I} - \mathbf{F}\mathbf{S}_s)^{-1}$$

With: $G(0) = \begin{bmatrix} 1.02 & -0.52 \\ -0.54 & 1.04 \end{bmatrix}$, $\mathbf{k}(0) = \begin{bmatrix} 1 \\ 7.1267 \end{bmatrix}$, $h(0) = [-0.0019, 0.0036]$,

$S_s = \begin{bmatrix} 1 & -0.1 \\ -0.1 & 1 \end{bmatrix}$ and $\mathbf{F} = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix}$

Taking into consideration that **F** will be in the range $0 < f < 1$. Closed-loop system response will be studied and examined for for several values of **F**.

Let $f_1 = f_2 = 0.1$ that is

$F = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$, Then

$$\mathbf{P} = \begin{bmatrix} 1.3984 & 0.58123 \\ 0.59021 & 1.3993 \end{bmatrix}_{f_1 = f_2 = 0.1}$$

Conventional compensator design

To reach the conventional controller design, shown in figure (4.1.2), for easy implementation, feedback gain matrix **H** can be computed as in equation (3.3.17).

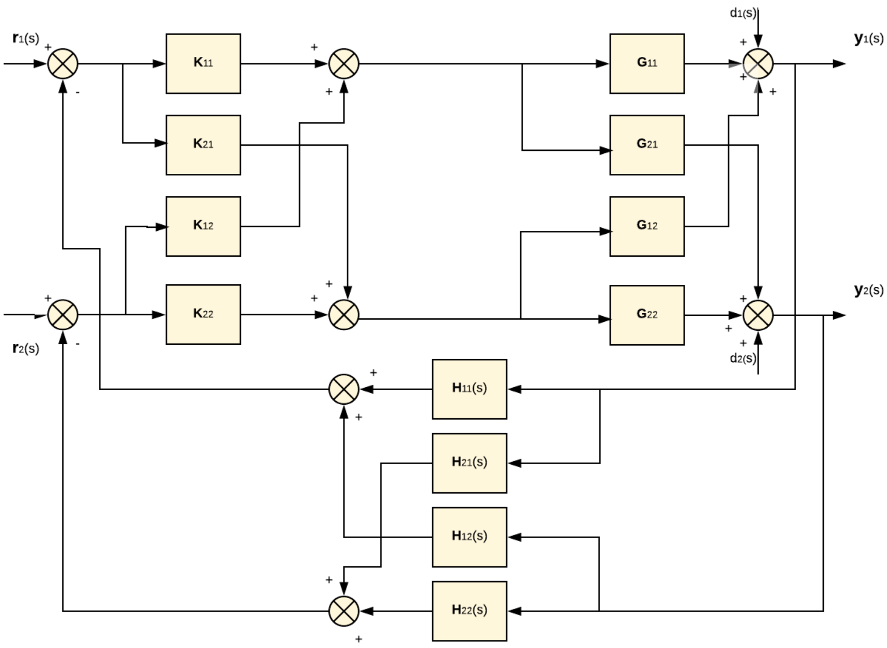

Figure 4.1.2: Block diagram showing least effort controller in the conventional structure

$$\mathbf{H}(\mathbf{s}) = \mathbf{P}^{-1}\mathbf{k}(\mathbf{s}) >< \mathbf{h}(\mathbf{s}) + \mathbf{F}$$

$$\mathbf{H}(\mathbf{s}) = \mathbf{P}^{-1}\begin{bmatrix} 1 \\ 7.1267 \end{bmatrix}\left[-0.0019\frac{11.76s+1}{2.6s+1}, 0.0036\right] + \mathbf{F}$$

For $f_1 = f_2 = 0.1$

$$\mathbf{P} = \begin{bmatrix} 1.3984 & 0.5812 \\ 0.5902 & 1.3993 \end{bmatrix}_{f_1=f_2=0.1}$$

And

$$\mathbf{H}(\mathbf{s}) = \mathbf{P}^{-1}\begin{bmatrix} 1 \\ 7.1267 \end{bmatrix}\left[-0.0019\frac{11.76s+1}{2.6s+1}, 0.0036\right] + \mathbf{F}$$

$$\mathbf{H}(\mathbf{s}) \approx \begin{bmatrix} 0.1149 & -0.0061 \\ -0.0113\left(\dfrac{11.76s+1}{2.6s+1}\right) & 0.1207 \end{bmatrix}_{f_1=f_2=0.1}$$

Similarly, for $f_1 = f_2 = 0.5$

$$\mathbf{P} = \begin{bmatrix} 2.4458 & 0.8296 \\ 0.8458 & 2.4459 \end{bmatrix}_{f_1=f_2=0.5}$$

And

$$\mathbf{H}(\mathbf{s}) \approx \begin{bmatrix} 0.5054 & -0.0023 \\ -0.0061\left(\dfrac{11.76s+1}{2.6s+1}\right) & 0.5111 \end{bmatrix}_{f_1=f_2=0.5}$$

And for $f_1 = f_2 = 0.8$

$$\mathbf{P} = \begin{bmatrix} 6.2472 & 0.1866 \\ 0.2326 & 6.2332 \end{bmatrix}_{f_1=f_2=0.8}$$ And

$$\mathbf{H}(\mathbf{s}) \approx \begin{bmatrix} 0.7990 & 0.00045 \\ -0.00216\left(\dfrac{11.76s+1}{2.6s+1}\right) & 0.8041 \end{bmatrix}_{f_1=f_2=0.8}$$

4.2 Design of H-infinity Controller

Based on the methodology presented in section 3.4 in the previous chapter, and Considering a mixed-sensitivity problem, direct implementation on the mixing system model shall be discussed in this section. The open loop transfer function as per the last chapter is

$$G(s) = \begin{bmatrix} \frac{1.02}{11.76s+1} & \frac{-0.52}{10.25s+1} \\ \frac{-0.54}{10.25s+1} & \frac{1.04}{2.6s+1} \end{bmatrix} \qquad \text{... (4.2.1)}$$

Selecting the weighting functions as

$$W_1 = \begin{bmatrix} \frac{0.1s+2}{5s+0.05} & 0 \\ 0 & \frac{0.1s+2}{5s+0.05} \end{bmatrix} \qquad \text{... (4.2.2)}$$

$$W_3 = \begin{bmatrix} \frac{s}{10} & 0 \\ 0 & \frac{s}{10} \end{bmatrix} \qquad \text{... (4.2.3)}$$

Let $W_2(s)$ be a “null” matrix. And, to avoid the problem of singularity, let’s assume that

$$W_2 = \begin{bmatrix} 10^{-8} & 0 \\ 0 & 10^{-8} \end{bmatrix} \qquad \text{... (4.2.4)}$$

The control problem here is to find a controller K that meets

$G(s), W_1, W_2$ and W_3 are all proper and bounded as $s \to \infty$.

After selecting the weighing filters, the system two-port augmented model can be created using the MATLAB® function (augtf), afterwhich the optimal $\mathcal{H}_\infty$ controller can be designed using MATLAB® function (hinfopt).

The minimum value of the variable γ was calculated with MATLAB® as in figure (4.2.1).

```
                  << H-Infinity Optimal Control Synthesis >>

 No      Gamma    D11<=1   P-Exist   P>=0   S-Exist   S>=0   lam(PS)<1    C.L.
-------------------------------------------------------------------------------
  1  1.0000e+00    OK        OK       OK      OK       OK       OK        STAB
  2  2.0000e+00    OK        OK       OK      OK       OK       OK        STAB
  3  4.0000e+00    OK        OK       OK      OK       OK       OK        STAB
  4  8.0000e+00    OK        OK      FAIL     OK       OK       OK        UNST
  5  6.0000e+00    OK        OK      FAIL     OK       OK       OK        UNST
  6  5.0000e+00    OK        OK      FAIL     OK       OK       OK        UNST
  7  4.5000e+00    OK        OK       OK      OK       OK       OK        STAB
  8  4.7500e+00    OK        OK       OK      OK       OK       OK        STAB
  9  4.8750e+00    OK        OK       OK      OK       OK       OK        STAB
 10  4.9375e+00    OK        OK       OK      OK       OK       OK        STAB
 11  4.9688e+00    OK        OK       OK      OK       OK       OK        STAB

     Iteration no. 11 is your best answer under the tolerance:   0.0100 .
```

Figure 4.2.1: MATLAB® result for the minimum of the variable γ = 4.9688

And the $\mathcal{H}_\infty$ controller transfer function:

$\mathbf{F}(\mathrm{s}) = \begin{bmatrix} \mathrm{F}_{11} & \mathrm{F}_{12} \\ \mathrm{F}_{21} & \mathrm{F}_{22} \end{bmatrix}$, where

$$\mathrm{F}_{11} = \frac{3983.8\ (s+161.3)\ (s+0.09756)\text{^}2\ (s+0.08503)\ (s+0.01)}{(s+161.3)\text{^}2\ (s+0.01)\text{^}2\ (s\text{^}2\ +\ 0.1722s\ +\ 0.007583)}$$

$$\mathrm{F}_{21} = \frac{524.7\ (s+161.3)\ (s+0.3846)\ (s+0.09756)\ (s+0.08503)\ (s+0.01)}{(s+161.3)\text{^}2\ (s+0.01)\text{^}2\ (s\text{^}2\ +\ 0.1722s\ +\ 0.007583)}$$

$$\mathrm{F}_{12} = \frac{505.27\ (s+161.3)\ (s+0.3846)\ (s+0.09756)\ (s+0.08503)\ (s+0.01)}{(s+161.3)\text{^}2\ (s+0.01)\text{^}2\ (s\text{^}2\ +\ 0.1722s\ +\ 0.007583)}$$

$$\mathrm{F}_{22} = \frac{863.84\ (s+161.3)\ (s+0.3846)\ (s+0.09756)\text{^}2\ (s+0.01)}{(s+161.3)\text{^}2\ (s+0.01)\text{^}2\ (s\text{^}2\ +\ 0.1722s\ +\ 0.007583)}$$

Chapter V: Simulation

5.1 Simulation of least effort controller

The closed-loop system model structure shown in figure (4.1.2) was implemented in MATLAB® SIMULINK® presented in appendix II figure (A.1). The model consists of two reference inputs r_1 and r_2. Let's say these are the hot and cold fluid inlet, respectively. Let's assume that the temp in the vessel is below the set mixture temperature. And the first reference input r_1 is used to achieve the required temperature in the vessel (i.e., Increase the mixture temperature to meet the required temperature). While the second reference input r_2 is kept non-changing. So, a single input will be used at once. That is r_1.

The second situation is that the temperature of the mixture in the vessel is above the set temperature, then the cold fluid line only "r_2" need to feed the vessel to regulate the mixture temperature.

In both situations, the mixture temperature will change, and that's the system output y_1. Also, the mixture head will change which is output y_2.

Testing the designed least effort controller achieved by simulating a unit step at the first system reference $r_1(s)$ and then on the second reference $r_2(s)$ and measuring the closed-loop system response on $y_1(s)$(the measured actual temperature of the mixture) and $y_2(s)$(the mixture head in the vessel).

figures (5.1.1) to (5.1.6) shows that the system response for a change in $r_1(s)$ and $r_2(s)$ with different values of f (f = 0.1, 0.5, 0.8).

Figure (5.1.1) shows the response for a unit step change in the first input $r_1(s)$, the output response of the closed-loop with $f_1 = f_2 = 0.1$ is reasonable but not up to the expected level.

The response is stably overdamped with 0 overshot for both outputs $y_1(s)$, $y_2(s)$. It can be noticed that both output responses settling time has improved (from 40s to 38s) for output

$y_1(s)$, but increased in a non-considerable manner for the output $y_2(s)$(from 37s in open-loop case to 39s after application of the designed controller).

While the first output response y_1 has a rising time of 22.8s (improved from 24s for the open-loop response). The second output response y_2 rise time improved from 22 (for open-loop system) to 18s.

Additionally, no major interaction noticed between the two outputs. The steady-state interaction didn't exceed the required 10% as per design. Both outputs settled at 1 and 0.1 as per steady state matrix S_s adjustment.

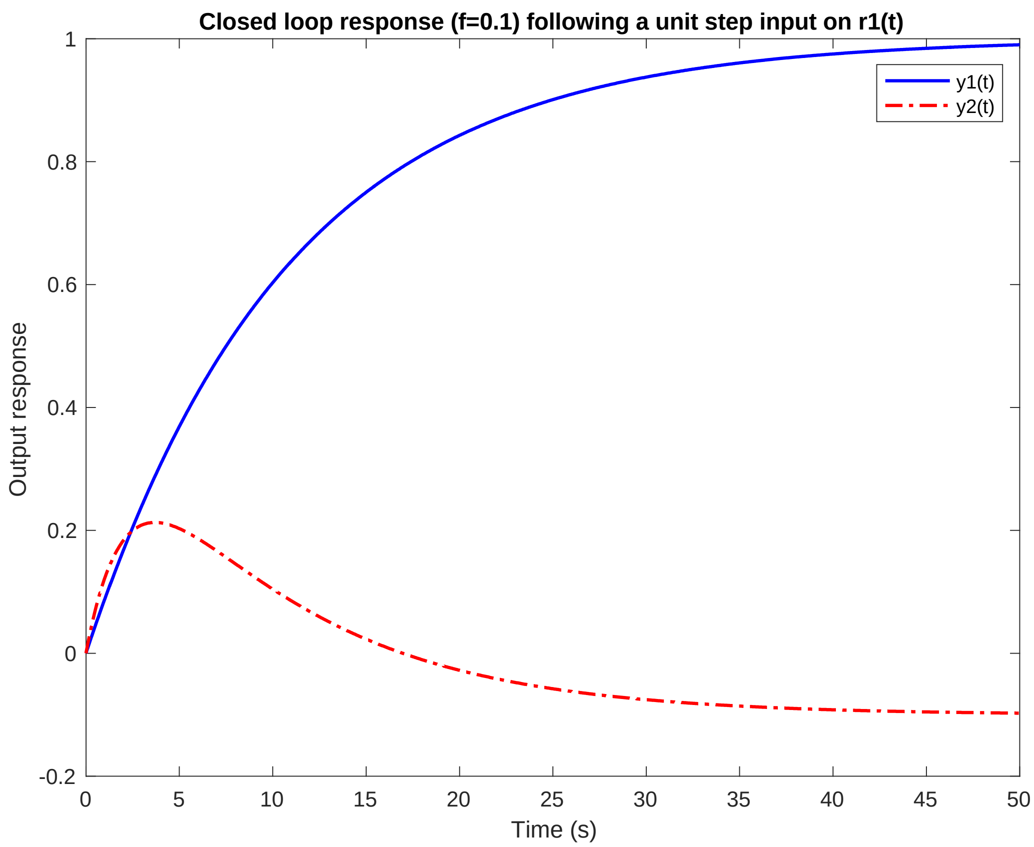

Figure 5.1.1: Response for a unit step change in $r_1(f = 0.1)$

On the other hand, figure (5.1.2) presents the response for a unit step change on the second input r2 with the same value of f ($f_1 = f_2 = 0.1$).

From figure (5.1.2) shown below, it can be noticed that the response is stable starting with 6% overshot for output $y_1(s)$and 7% overshot for output $y_2(s)$. The response has a noticeable improvement in terms of speed (from 22s to 9s for output $y_1(s)$and from 6s to 3s for output $y_2(s)$). On the other hand, the settling time is affected negatively from 36s to 40s for output $y_1(s)$, and from 10s to 25s for the output $y_2(s)$.

Additionally, no major interaction noticed between the two outputs. The steady-state interaction didn't exceed the required 10% as per design. Both outputs output $y_1(s)$and output $y_2(s)$ settled at 0.1 and 1 as per steady state matrix S_s adjustment.

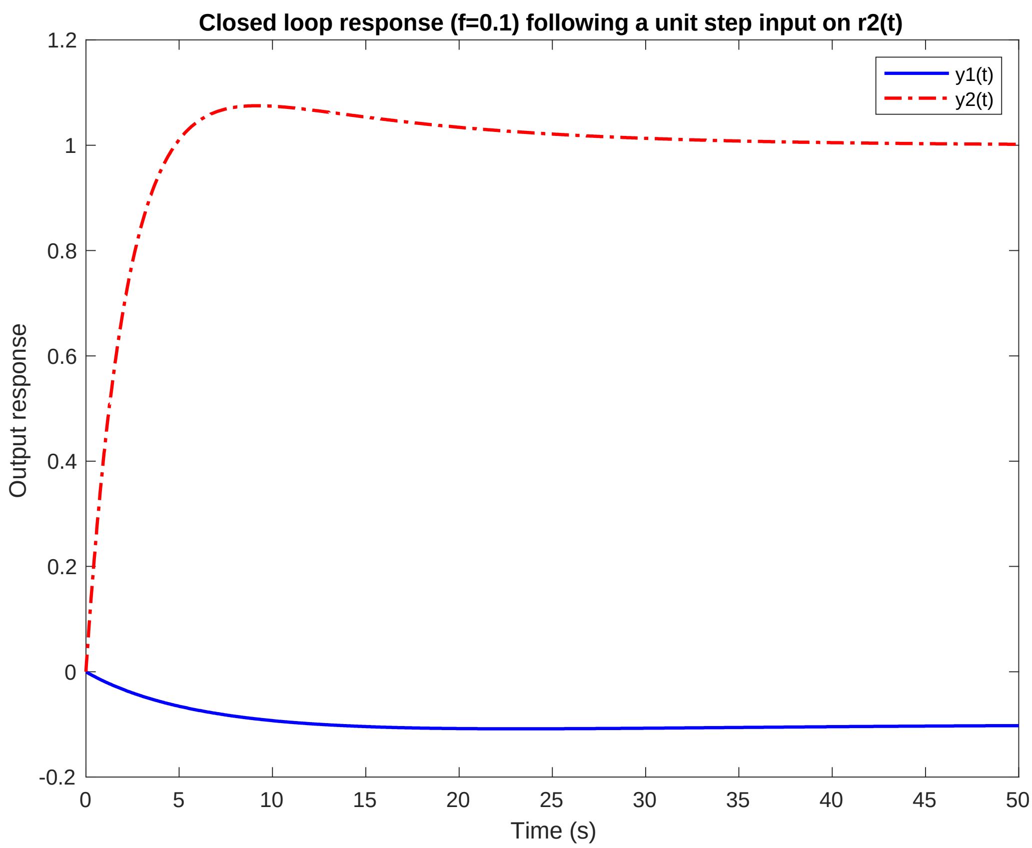

Figure 5.1.2: Response for a unit step change in $r_2(f = 0.1)$

Now the outer loop feedback increased to 0.5, and the system was simulated with the new controller adjustment, the output responses for two outputs $y_1(s)$, and $y_2(s)$ were plotted after exiting each input separately as shown in figure (5.1.3) for input $r_1(s)$ and figure (5.1.4) for input $r_2(s)$. Figure (5.1.3) shows more enhanced output $y_1(s)$, and $y_2(s)$ response against 1% step in input $r_1(s)$ in many aspects, as will be illustrated in the next paragraphs.

The response is stably overdamped with 0 overshot for both outputs $y_1(s)$, $y_2(s)$. It can be noticed that both output responses settling time has improved compared to the open-loop system and the previous system set-up (from 40s to 23s) for output $y_1(s)$, (from 37s in open-loop case to 25s) output $y_2(s)$. While the first output response $y_1$has a rising time of 12.5s (improved from 24s for the open-loop response). The second output response y_2 rise time improved from 22s (for open-loop system) to 12s. Additionally, no major interaction noticed between the two outputs. The steady-state interaction didn't exceed the required 10% as per design. Both outputs settled at 1 and 0.1 as per steady state matrix S_s adjustment.

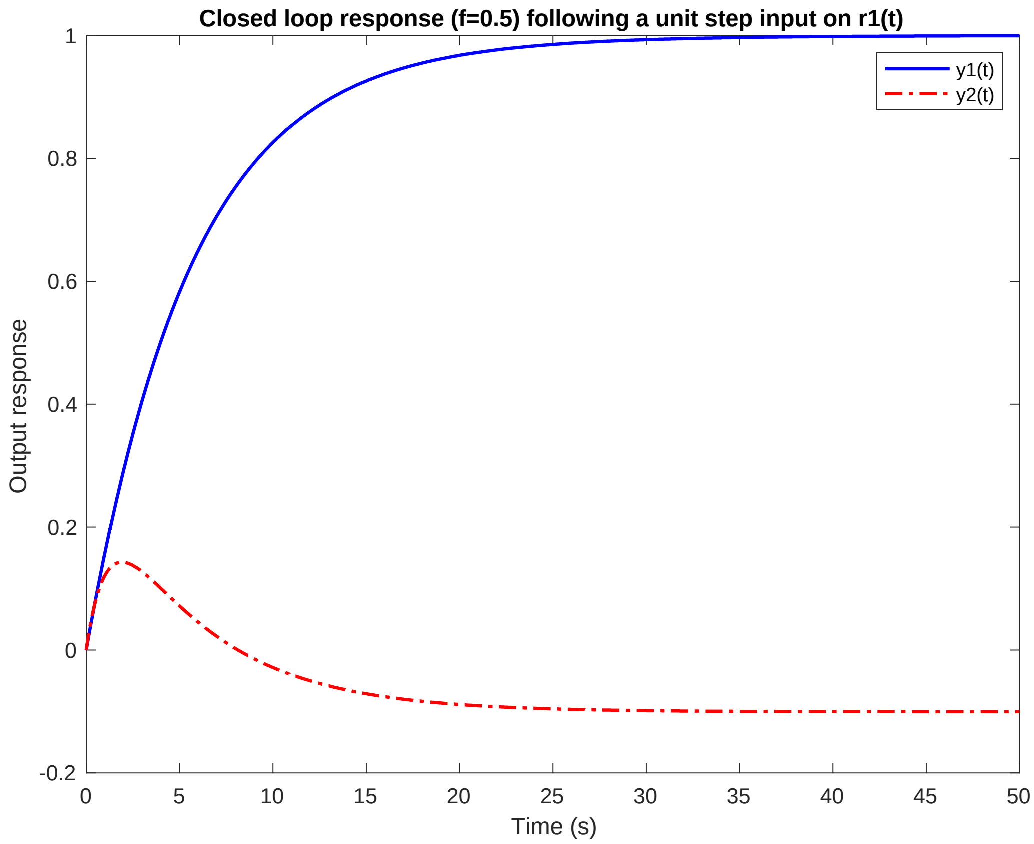

Figure 5.1.3: Response for a unit step change in $r_1(f = 0.5)$

On the other hand, figure (5.1.4) presents the response for a unit step change on the second input r2 with the same value of f ($f_1 = f_2 = 0.5$).

From figure (5.1.4) shown below, it can be noticed that the response is stable starting with 5% overshot for output $y_1(s)$ and output $y_2(s)$. The response has a noticeable improvement in terms of speed (from 22s to 5.4s for output $y_1(s)$and from 6s to 2s for output $y_2(s)$). On the other hand, the settling time is improved slightly for output $y_1(s)$ from 36s to 33s, and degraded from 10s to 18s for the output $y_2(s)$.

Additionally, no major interaction noticed between the two outputs. The steady-state interaction didn't exceed the required 10% as per design. Both outputs output $y_1(s)$and output $y_2(s)$ settled at 0.1 and 1 as per steady state matrix S_s adjustment.

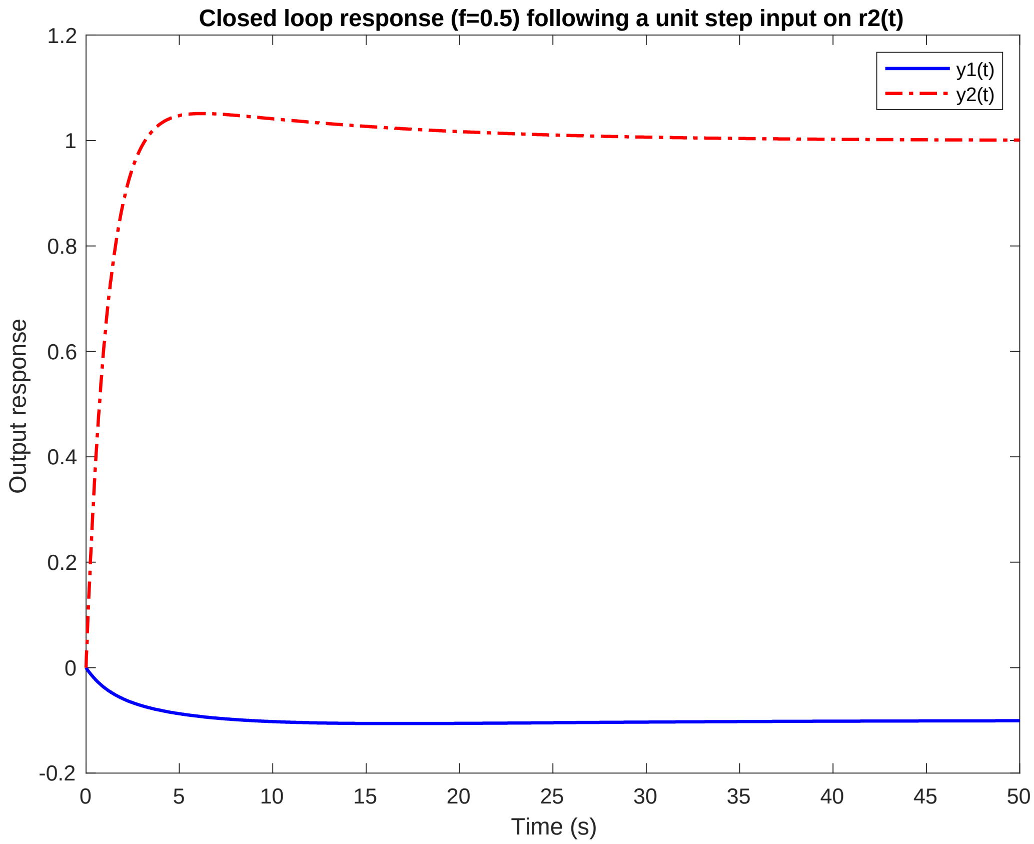

Figure 5.1.4: Response for a unit step change in r_2(f = 0.5)

The outer loop feedback increased to 0.8, and the system was simulated with the new controller adjustment, the output responses for two outputs $y_1(s)$, and $y_2(s)$ were plotted after exiting each input separately as shown in figure (5.1.5) for input $r_1(s)$ and figure (5.1.6) for input $r_2(s)$. Compared to the previous two designs, figure (5.1.5) shows the best output $y_1(s)$, and $y_2(s)$ response against 1% step in input $r_1(s)$, this will be illustrated in the next paragraphs. The response is stably overdamped with 0 overshot for both outputs $y_1(s)$, $y_2(s)$. It can be noticed that both output responses settling time has improved compared to the open-loop system and the previous system set-ups (from 40s to 8s) for output $y_1(s)$, (from 37s in open-loop case to 9s) output $y_2(s)$. While the first output response y_1 has a rising time of approximately 4s (improved from 24s for the open-loop response). The second output response y_2 rise time improved from 22s (for open-loop system) to 1s only. Additionally, no major interaction noticed between the two outputs beyond the pre-set limit. The steady-state interaction didn't exceed the required 10% as per design. Both outputs settled at 1 and 0.1 as per steady state matrix S_s adjustment.

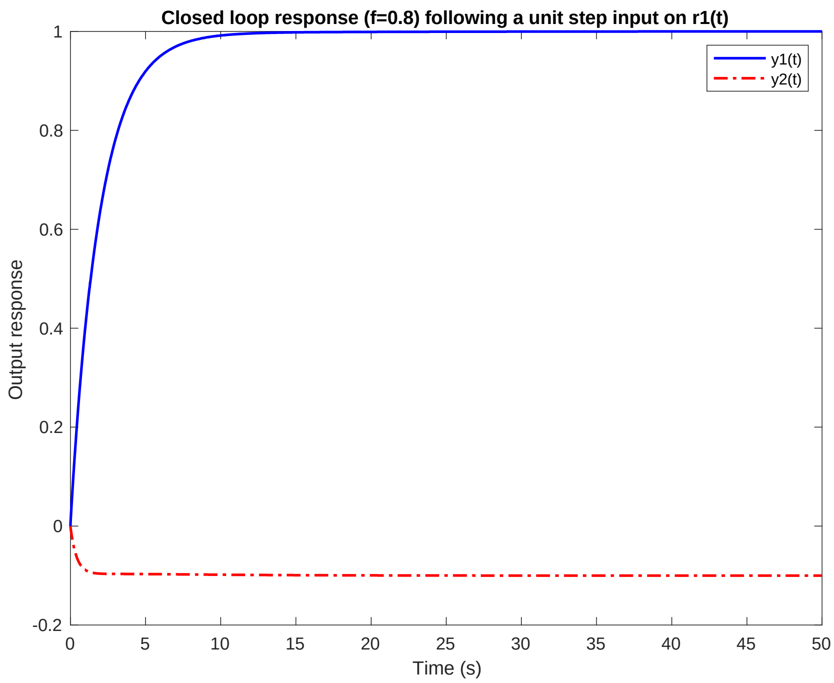

Figure 5.1.5: Response for a unit step change in r_1(f = 0.8)

On the other hand, figure (5.1.6) presents the response for a unit step change on the second input r2 with the same value of f ($f_1 = f_2 = 0.8$).

From figure (5.1.6) shown below, it can be noticed that the response is stable starting with 10% overshot for output $y_1(s)$ and 2.5% overshot for output $y_2(s)$. The response has a noticeable improvement in terms of speed (from 22s to 0,54s for output $y_1(s)$and from 6s to 0.8s for output $y_2(s)$). On the other hand, the settling time is improved dramatically for output $y_1(s)$ from 36s to 4s, and from 10s to 6s for the output $y_2(s)$.

Additionally, no major interaction noticed between the two outputs. The steady-state interaction didn't exceed the required 10% as per design. Both outputs output $y_1(s)$and output $y_2(s)$ settled at 0.1 and 1 as per steady state matrix S_s adjustment.

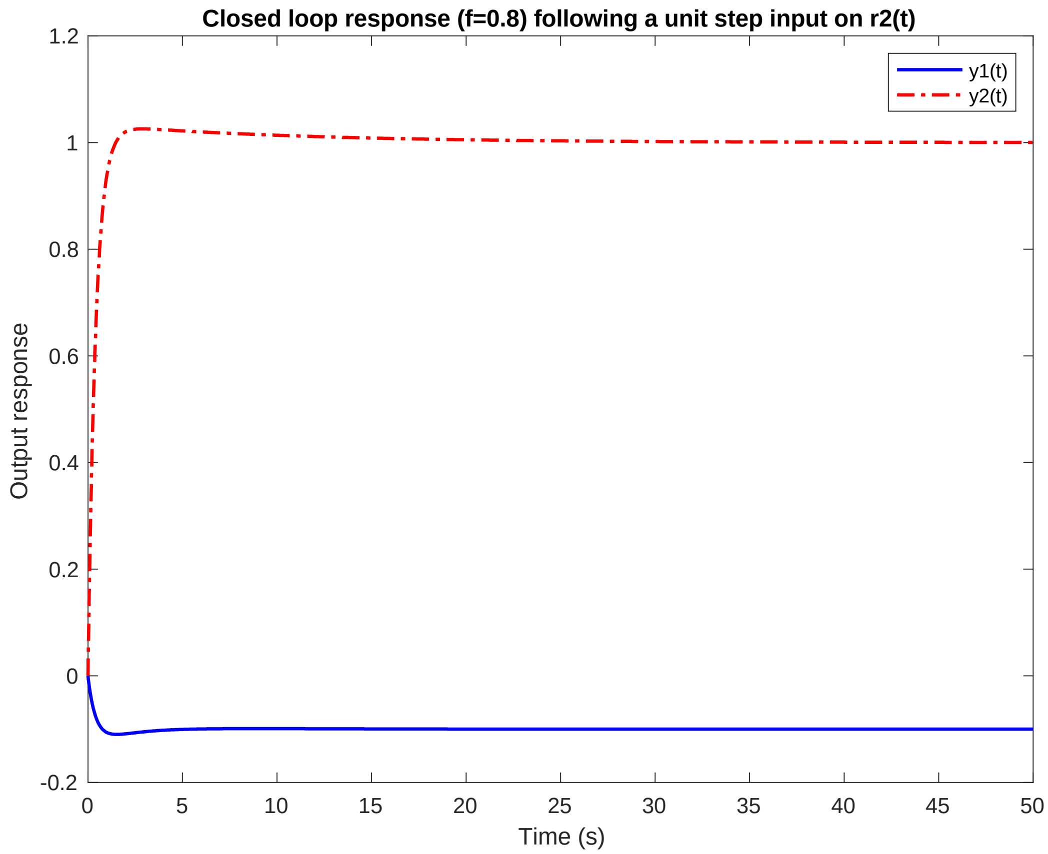

Figure 5.1.6: Response for a unit step change in $r_2(f = 0.8)$

Finally, figure (5.1.7, 5.1.8) below show the output response for $f = 0.1, 0.5, 0.8$ overlapped for easy visual comparison.

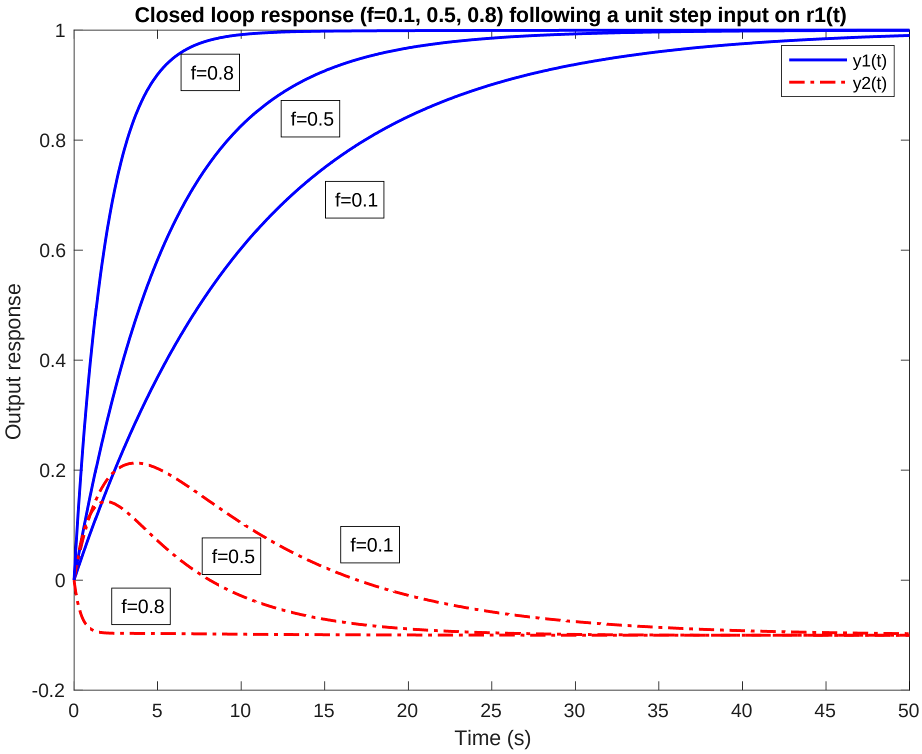

Figure 5.1.7: Response for a unit step change in r_1(f = 0.1, 0.5, 0.8)

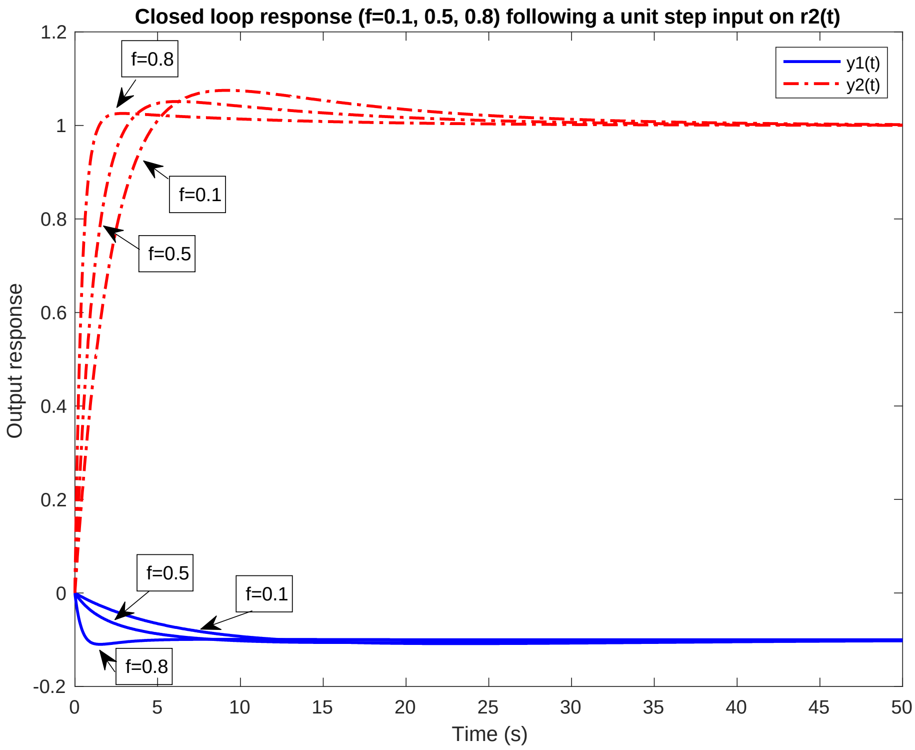

Figure 5.1.8: Response for a unit step change in r_1(f = 0.1, 0.5, 0.8)

To be noted that even though the speed of response and settling time improved with increasing f from 0.1 to 0.8, increasing f added up to the control energy used by the controller, so there shall be a trade-off between the energy consumed by the controller and enhancing the system response, to be highlighted that the change in controller energy in not in a considerable amount. Refer to figure (5.1.9), (5.1.10).

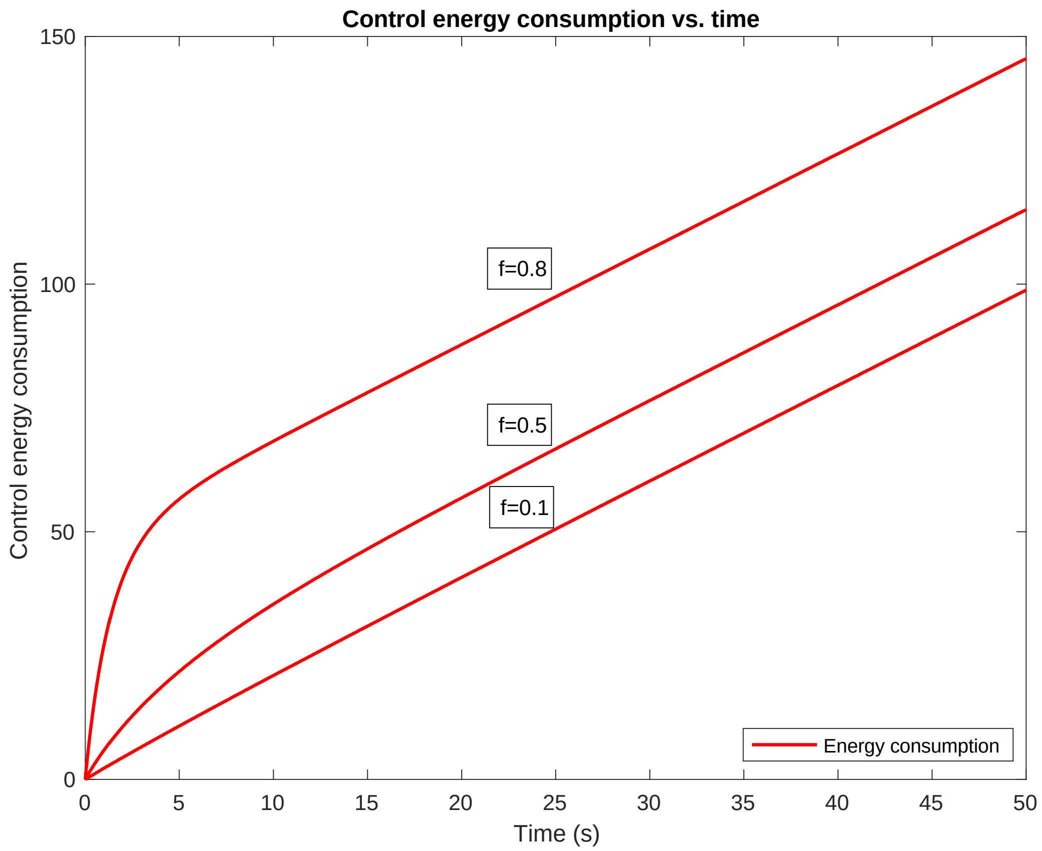

Figure 5.1.9: Control energy consumption for a unit step change on $r_1(s)$

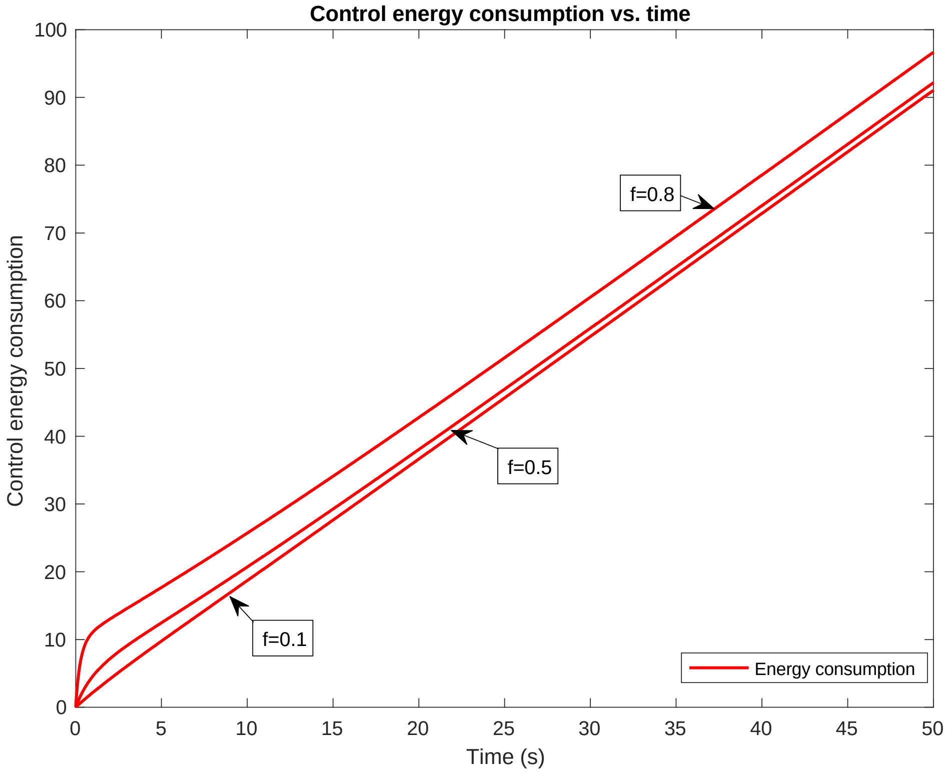

Figure 5.1.10: Control energy consumption for a unit step change on $r_2(s)$

The controller robustness against disturbances was inspected by exposing the system to a negative step disturbance signal at the first output $y_1(s)$. The system closed-loop response is simulated for different values of f and the result was plotted in figure (5.1.11)

It is very clear that how the system is improving by increasing the values of f from 0.5 to 0.8 where the recovery from disturbance is the highest and fastest when $f = 0.8$, (system recovered by 80%) for which the system is considered to have an acceptable disturbance rejection property.

Moreover, a disturbance signal is introduced again at the second output $y_2(s)$, The system closed-loop response is simulated for different values of f and the result was plotted in figure (5.1.12). at $f = 0.8$ the system also recovered by more than 80% which is considered a good performance.

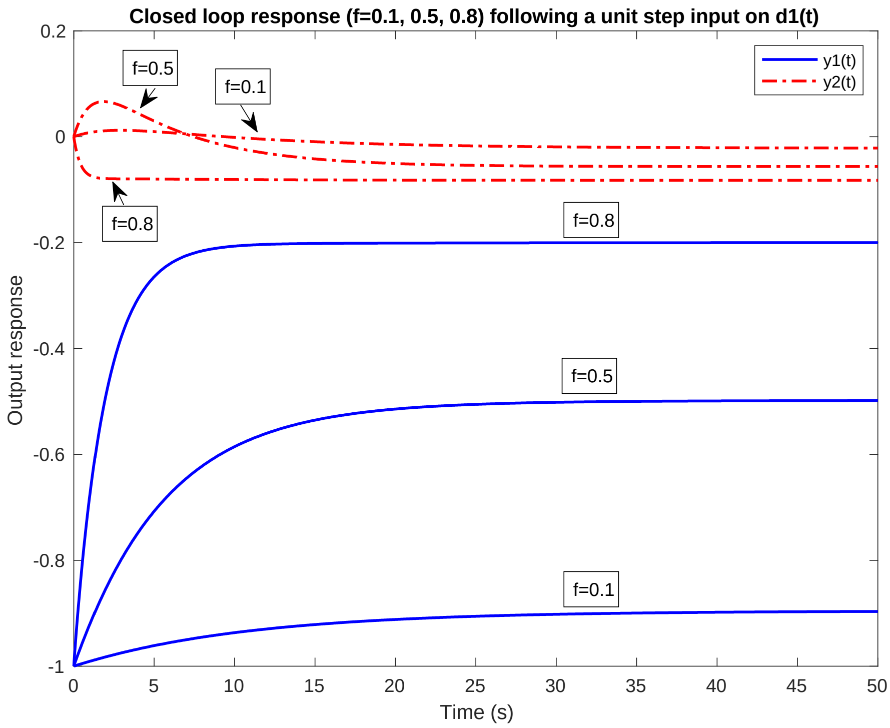

Figure 5.1.11: Response following a negative step change in $\delta_1(s)$ $(f = 0.1, 0.5, 0.8)$

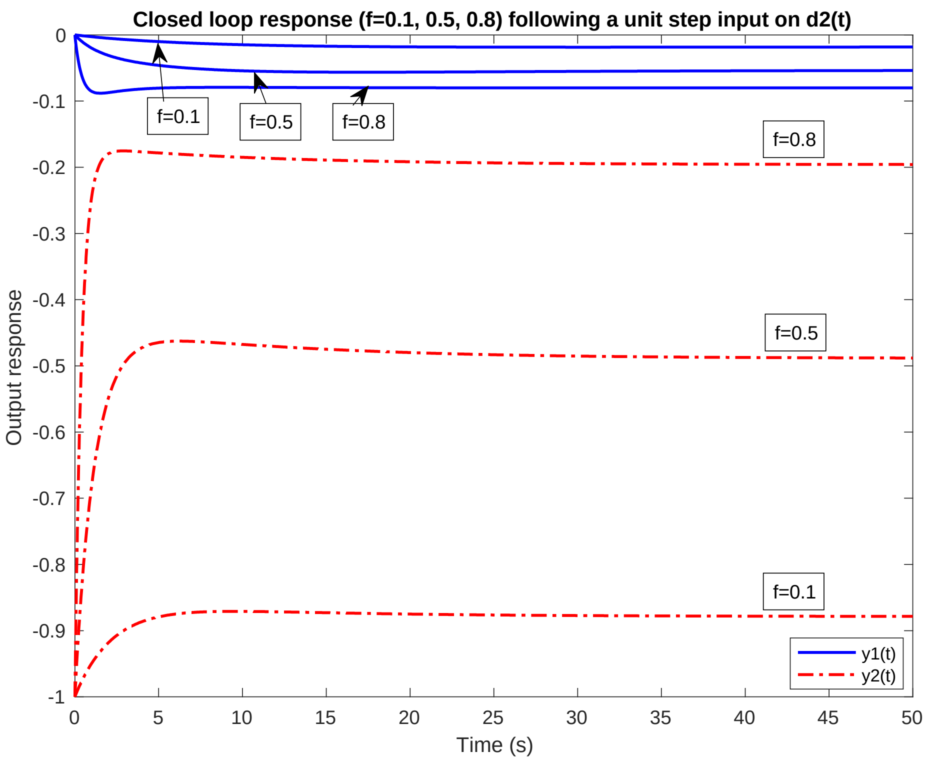

Figure 5.1.12: Response following a negative step change in $\delta_2(s)$ $(f = 0.1, 0.5, 0.8)$

5.2 Simulation of H-infinity controller

The closed-loop of the system including an H-infinity controller was implemented in MATLAB® SIMULINK® presented in appendix II figure (B.1). The model consists of two reference inputs r_1 and r_2. Similar to least effort controller case, testing the designed H-infinity controller achieved by simulating a unit step at the first system reference input $r_1(s)$ and then on the second reference input $r_2(s)$ and measuring the closed-loop system response on $y_1(s)$ (the measured actual temperature of the mixture) and $y_2(s)$(the mixture head in the vessel).

Figure (5.2.1) presents how the system respond to a unit step change in the first reference input $r_1(s)$, the system response is good and meats the expected level.

The response is stable overdamped with 0% overshot for output $y_1(s)$, It can be noticed that, settling time for output $y_1(s)$ has improved (from 40s to 2s).

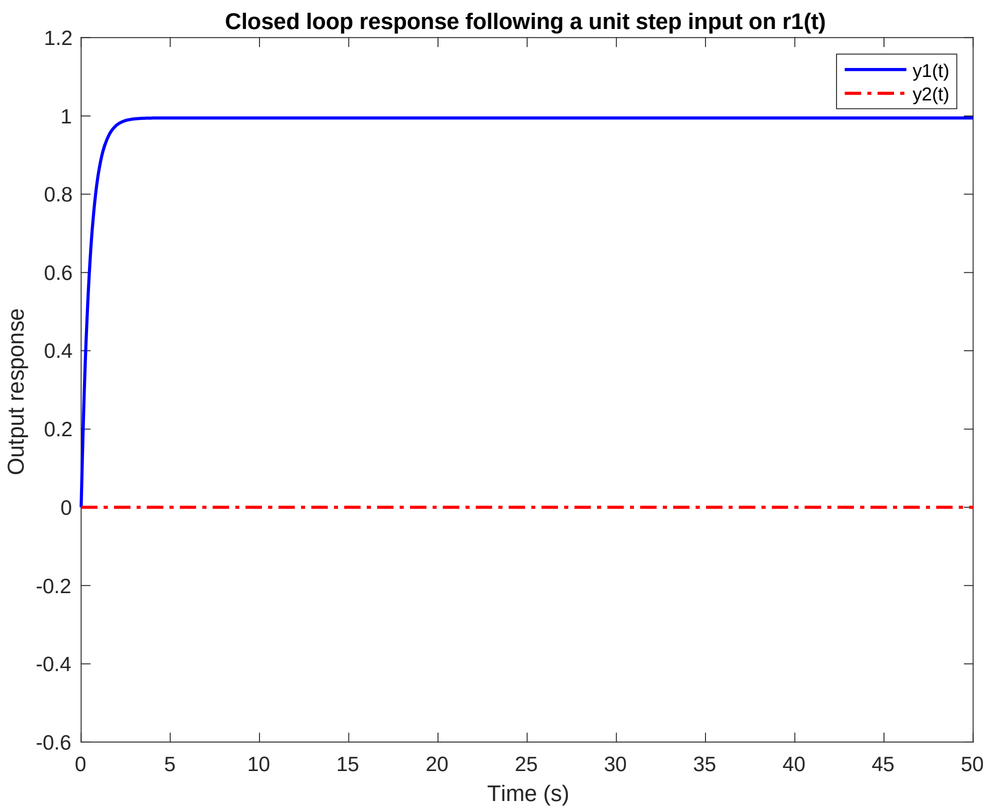

Figure 5.2.1: Response for a unit step change in r_1

While the open-loop first output response y_1 has a rising time of 24s, y_1 with H-infinity controller has a rise time of 1.1s, which is considered as a tremendous improvement.

Additionally, the open-loop system steady state interaction problem was completely solved. The interaction between the two outputs $y_1(s)$ and $y_2(s)$is almost 0. $y_1(s)$ settled at 99.5% and output $y_2(s)$ settled down at 0%. The control energy used to achieve the control action mentioned above is plotted in figure (5.2.2) below.

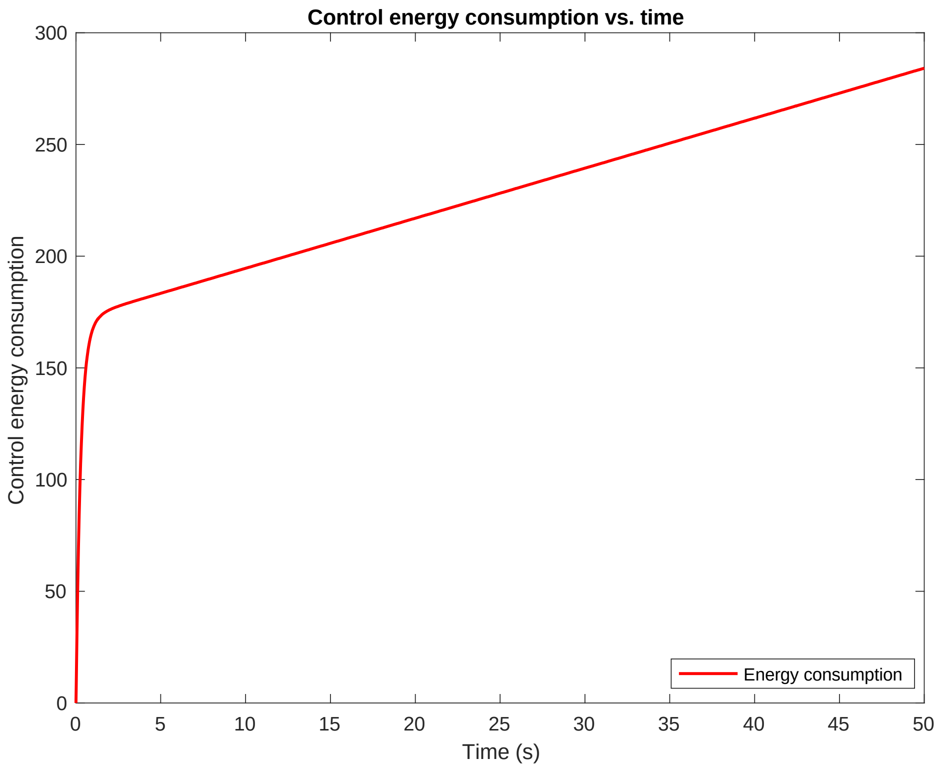

Figure 5.2.2: Control energy consumption for a unit step change on $r_1(s)$

From figure (5.2.2), it can be noticed that the graph jumps in the first 1s from 0 to 175, after which the energy curve slope decreases and become constant stable curve till the end of the control period.

Once again, the closed-loop system performance was measured for 1% step change on the input $r_2(s)$ and the output curve was plotted for both the outputs $y_1(s)$ and $y_2(s)$ as shown in figure

(5.2.3) below. Figure (5.2.3) shows the system response to a unit step change in the input $r_2(s)$, the output response of the closed-loop is good and exceeds the expected level.

The response is stable overdamped with 0% overshot for output $y_2(s)$, It can be noticed that, settling time for output $y_2(s)$ has improved (from 10s to 1.95s).

While the open-loop first output response y_1 has a rising time of 5.7s, y_2 with H-infinity controller has a rise time of 1.1s, which is considered a very good improvement.

Additionally, the open-loop system steady state interaction problem was completely solved. The interaction between the two outputs $y_1(s)$ and $y_2(s)$is almost 0. $y_2(s)$ settled at 99.5% and output $y_1(s)$setteled down at 0%.

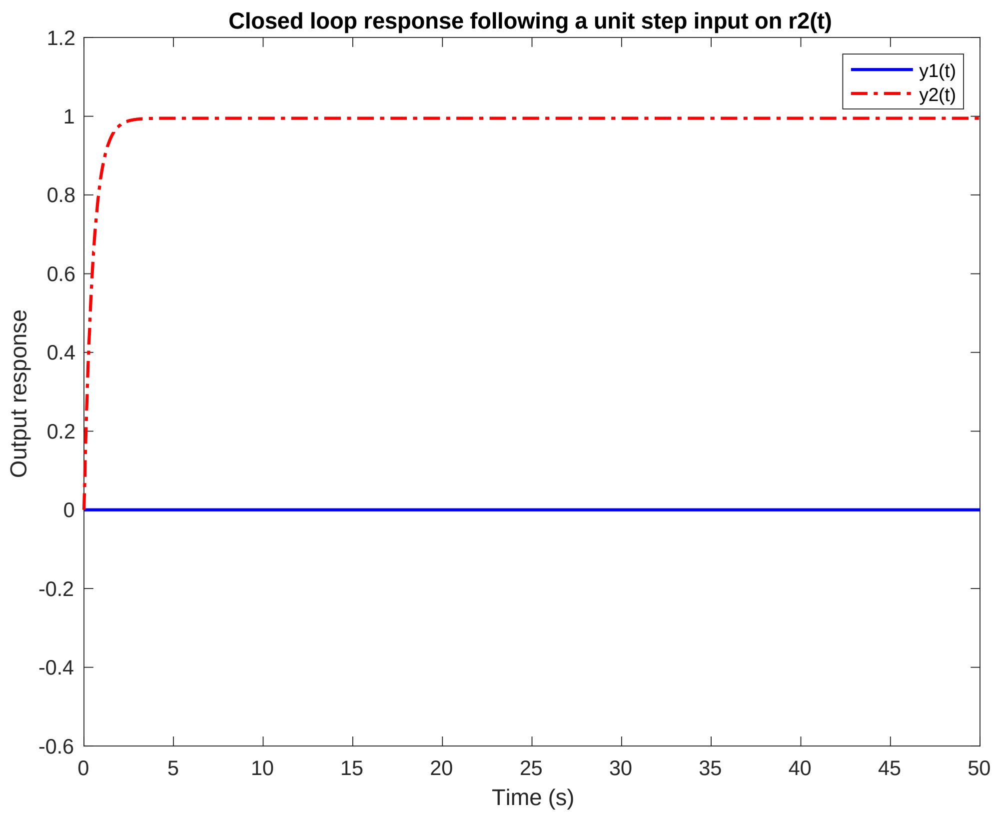

Figure 5.2.3: Response for a unit step change in r_2

The control energy used to achieve the above-mentioned control action is plotted in figure (5.2.4) below.

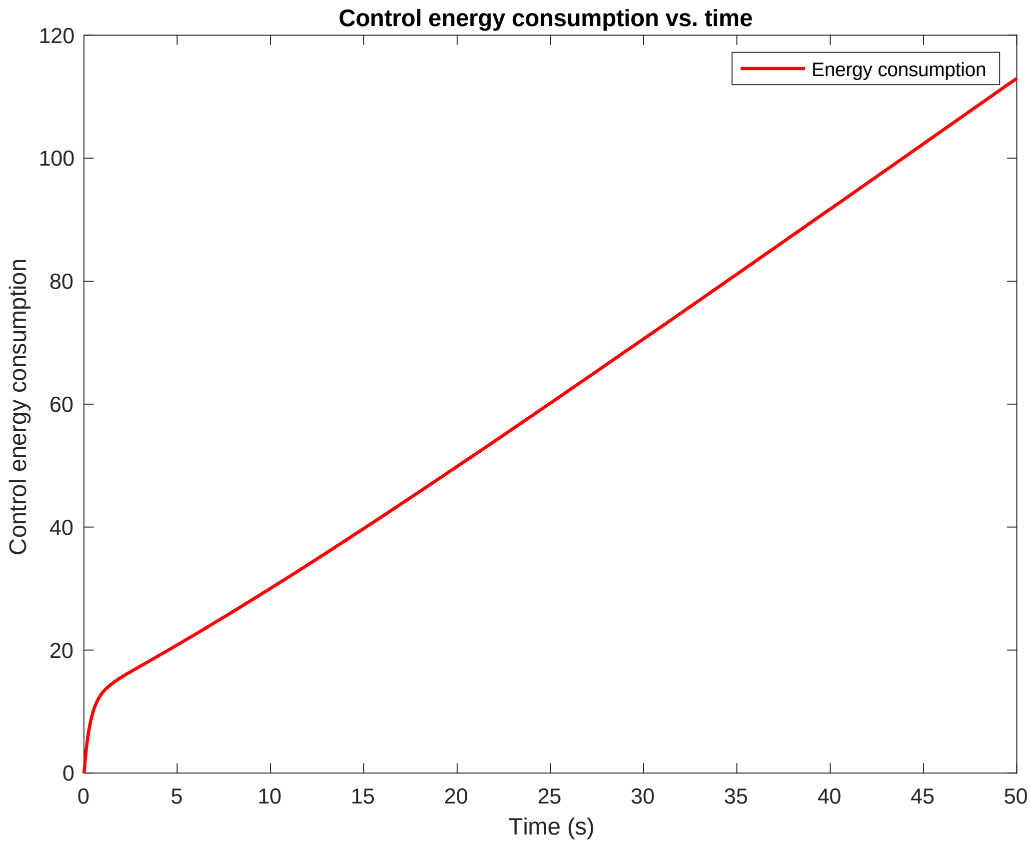

Figure 5.2.4: Control energy consumption for a unit step change on $r_2(s)$

The controller robustness against disturbances was inspected by exposing the system to a negative step disturbance signal at $y_1(s)$. The closed-loop response of the system is simulated, and the result was plotted in figure (5.2.5).

It is very clear that how the system recovery from disturbance beyond expectations and very fast (around 1 second). System recovered by 100% and achieved a very good disturbance rejection property.

Moreover, a disturbance signal is introduced again at the second output $y_2(s)$, The system closed-loop response is simulated, and the result was plotted in figure (5.2.6). The system also achieved 100% recovery.

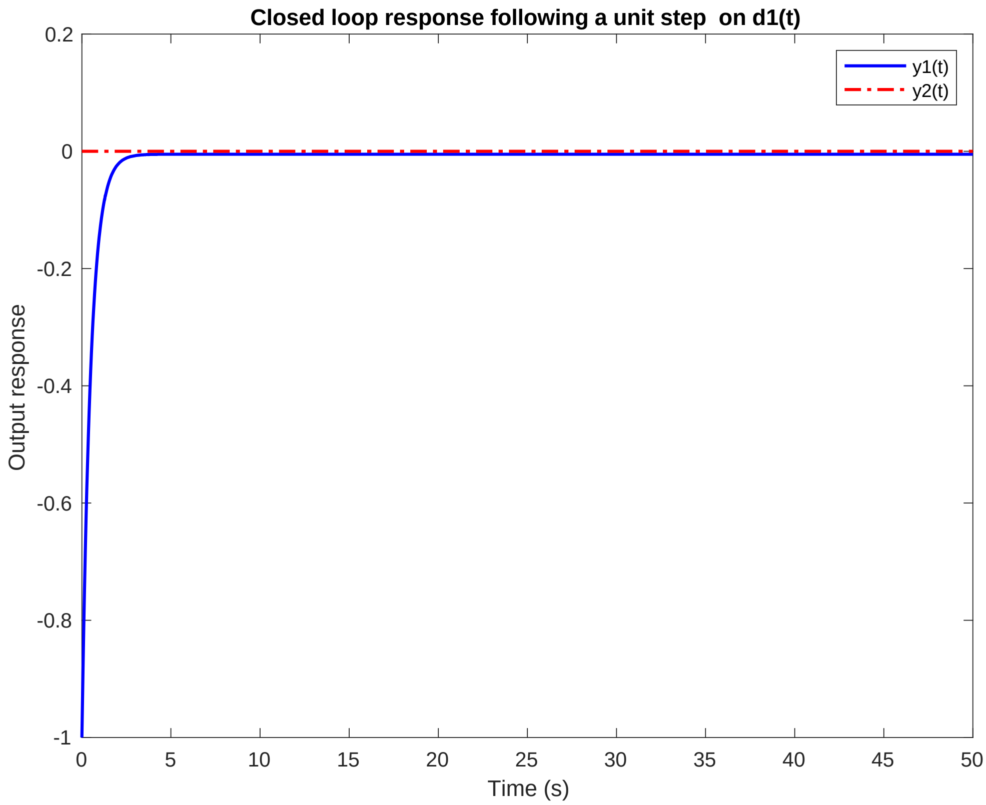

Figure 5.2.5: Response following a negative step change in $\delta_1(s)$

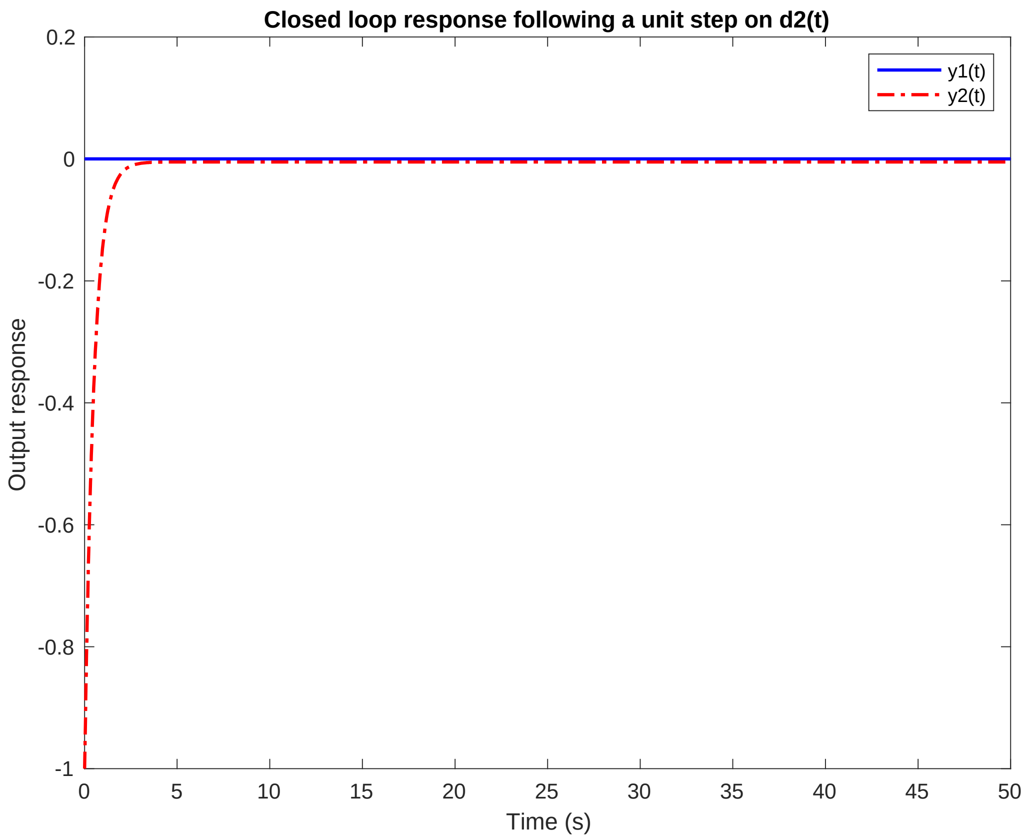

Figure 5.2.6: Response following a negative step change in $\delta_2(s)$

Chapter VI: A Comparison study

Every control design method for designing a controller for a multivariable system has its advantages and disadvantages. A comparison study of the two used design techniques will be held based on four main factors. The first comparison criterium is the closed-loop response offered by the controller. The second criterium of evaluation is the simplicity of the controller and the feasibility for real-life applications. The third criterium is the controller disturbance suppression. That is, how the controller reacted to a unit step change of disturbance on each output separately. The last factor selected for the comparison study is the energy used by the controller in order to produce a control action.

6.1 Least effort controller

From the British school of modern control, Least Effort Control method is used to design a controller for the open system matrix in the frequency domain. One flaw is that the transfer function must be in the form of equation (3.3.18). To achieve this form some approximation is required, and it is to be assured that the transient and steady state of the approximation matches the original system.

6.2 H-infinity Controller

H-infinity control, on the other hand, is a modern school method from the American school. The design problem is about finding the suboptimal feedback design optimization problem in which the control task is oriented about designing a stabilizing controller that minimizes the H-infinity norm of the closed-loop transfer function. First, weighting matrices (filters) are defined, which is vital to the final performance of the system. Then the controller is designed by solving an optimization problem mathematically or numerically through genetic algorithms GAs (Goldberg 1989) or loop shaping.

6.3Closed-loop response

The closed-loop control system response for a unit step changes on input $r_1(s)$ and input $r_2(s)$ was presented in the previous chapter, for both the controllers. Comparing the output $y_1(s)$ and $y_2(s)$ responses to input $r_1(s)$ in figures (5.1.5) and (5.2.1) shows that the H-infinity controller transcends the least effort controller but not in a considerable manner. The same can be noticed when comparing figures (5.1.6) and (5.2.3) related to the input $r_2(s)$.

Regarding output interaction, while least effort controller meat the design requirement at less than 10% for both the inputs, H-infinity controller achieved a good result at approximately completed de-coupling.

6.4 Disturbance rejection

With respect to disturbance rejection, although H-infinity controller showed perfect rejection of the injected disturbance signal (100%) refer to figures (5.2.5) and (5.2.6), least effort controller with f=0.8 setting showed 80% disturbance rejection as shown in figures (5.1.9) and (5.1.10) which is also a good achievement that can be improved by increasing f in the cost of energy consumption.

6.5 Consumption of energy

Energy consumption can be calculated by the following formula (R. Whalley and M. Ebrahimi, 2006).

$$E(t) = \int_{t=0}^{t=1} (u_1(t)^2 + u_2(t)^2).dt$$

A Simulink® model shown in figures (A1), and (B1) in the appendix was used to calculate the energy cost by both controllers to perform the control job. For input $r_1(s)$, as shown in figures (5.1.9) and (5.2.2), H-infinity controller is near twice the cost of least effort controller. Also, for input $r_2(s)$, as shown in figures (5.1.10) and (5.2.4) least effort controller energy consumption is around 17% of H-infinity controller.

6.6 Simplicity & ease of implementation

It can be noticed that while Least Effort controller has a simple form consisting of feedback and feedforward simple gains only. However, the H-infinity controller has a complicated structure with transfer functions of 5th order. Closed-loop controller model reduction techniques shall be used to make such controller applicable for real life applications. That makes the least effort controller a good choice in real applications.

Chapter VII: Conclusions and Recommendations

7.1 Conclusions

In this research two control methods (least-effort, and H-infinity) were highlighted. They were implemented to a 2×2 feed mixing system, and the system response to step inputs and disturbances on the outputs was studied.

MATLAB® control system toolbox and Simulink® were used to simulate, study, design and verify the performance of the two controllers under study.

The design method of the Least Effort Controller is based on a systematic method which gives the freedom for the designer to improves specific performance criteria in multivariable systems. Because it consists of a combination of two loops, inner and outer loop, the designer can adjust the inner loop to improve the system dynamics, while the outer loop improves the disturbance rejection properties and sets output interaction to the required design level. Most importantly, both loops control action is based on the least control effort.

Moreover, least effort controller is very simple when it comes to application, while H-infinity controller consists of transfer functions of 5^{th} order, least effort controller is as simple as a set of constant gains. This does ease not only the implementation process, but also the lifetime maintenance cost.

The main difficulty that may face Least Effort Controller is when the system transfer function is not simple and cannot be written as per the required form. For more than 2×2 systems finding the optimum solution will not be as easy as it is for 2×2 systems. However, it this can be overcome by applying numerical methods for solving the optimization problem.

The system response has shown a dramatic change regarding system dynamics including speed related parameters as settling time and rise time. Moreover, the steady state interaction was as per the design requirement, and the system shows a good performance regarding disturbance rejection.

On the other hand, the H-infinity controller design method has shown an adequate response and recognizable behavior regarding enhancing system dynamics and improving steady state interaction. Disturbance suppression far exceeded Least effort controller. But all of this came on the cost of control energy consumption. One more disadvantage of this control method is the complexity of the controller which raise the importance of model reduction techniques with such a controller.

7.2 Recommendations

- Least Effort controller has shown good performance with simple implementation of constant gains; less energy cost was noticed that's why it is recommended as a control system for plants where control energy is essential. Constructing the controller depends on well-defined systematic procedure, and the response was meeting the design requirements. This control method is recommended for feed mixing system.
- Like optimal LQR (lqr), (lqg), and H-infinity controller (hinfopt), an algorithm for least effort controller (LEQ) can be constructed in MATLAB® control system toolbox. Which will simplify the control job for more complex systems.
- Least effort controller shall be extended to be implemented on digital systems control problem.
- H-infinity control shown superior performance in both transient and steady state parts. However, the complication of the controller model and the higher energy consumption by it made it stand behind lest effort controller. Also, additional studies need to be done to develop a systematic method of deriving the system weighing filters.

References

Bibel, J. & Malyevac, D. (1992). Guidelines for the Selection of Weighting Functions for H-INFINITY Control. *Naval Surface Warfare Center Dahlgren Division*, (NSWCDD/MP-92/43).

Bode, H.W., (1940) *Feedback amplifier design*, Bell Systems Technical Journal, vol.19, 1940, pp.42.

Chiang, R. & Safonov, M. (2006). *Robust Control Toolbox for use with MATLAB®*. Natick, Mass.: MathWorks.

Dasgupta, D. and McGregor, D. (1991). *A structured genetic algorithms*. University of Strathclyde, Department of Computer Science.

Dingyu X, YangQuan C, and Derek P, (2007), *Linear Feedback Control Analysis and Design with MATLAB®*, Siam, Philadelphia.

Dorato, Peter (1987), A Historical Review of Robust Control, *IEEE Control Systems Magazine*.

Dorf, R.C., and Bishop, R.H. (2008), *Modern Control Systems*. Pearson Education Inc. New York.

Doyle, J.C., and Stein, G. (1981), *Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis*, *IEEE Transaction son Automatic Control*, Vol. AC-26.

Doyle, J.C., Glover, K., Khargonekar, P.P. and Francis, B.A. (1988), State-Space Solutions to Standard H_2 and H, Control Problems, *Proceedings of 1988 American Control Conference,* Atlanta, GA.

Dukkipati, R. (2006). *Analysis and design of control systems using MATLAB®*. New Delhi: New Age International (P) Limited, Publishers.

Dutton, K., Thompson, and S. and Barraclough, B. (1997), *The Art of Control Engineering.* Pearson, Prentice Hall.

El-Hassan, T. (2012). *A Design Study for Multivariable Feedback Control System Regulation for Aircraft Turbo-Jet Engines*. MSc. Dissertation, BUiD, Dubai.

Routh, E (1877), *A treatise on the stability of a given state of motion*, in , London, Macmillan, And Co.

Evans, Walter R. (1965), *Spirule Instructions*, Whittier, CA: The Spirule Company

Flugge-Lotz, Irmgard, Titus, Harold A. (1962). Optimum and Quasi-Optimum Control of Third and Fourth-Order Systems. *Stanford University Technical Report* (134):8–12.

Goldberg, D, *Genetic algorithms in search, optimization, and machine learning*. in, Boston, Addison-Wesley, 1989.

Gopal M. (1993), *Modern Control Systems theory*, new age internationals (P) Limited publishers, New Delhi

Green M. (1992), H_∞ controller synthesis by *J*-lossless coprime factorization, *SIAM J. Control and Opt.,* 30, PP 522-547.

Grimble, M.J. and Biss, D. (1988), Selection of Optimal Control Weighting Functions to Achieve Good H, Robust Designs, *Proceedings of IEE International Conference Control 88,* Conference Publication No. 285.

Gleaves, Albert (1921). *A history of the transport service*. New York, George H. Doran Company.

H. Bosgra, O., Kwakernaak, H. & Meinsma, G. (2008). *Design Methods for Control Systems*. Enschede: Dutch Institute of Systems and Control. Accessed 30 May 2018. http://wwwhome.math.utwente.nl/~meinsmag/dmcs/dmcs0708.pdf

H. Nyquist (1932), *Regeneration theory*, Bell System Technical Journal, vol. 11, pp. 126–147, 1932

Horowitz, I.M., (1963) *Synthesis of linear systems*. Academic Press, 1963.

Horowitz, I.M., Sidi, M., (1972) *Synthesis of feedback systems with large plant ignorance for prescribed time domain tolerances*. International Journal of Control, vol.16, 1973, pp.287-309.

Horowitz (1982). *Quantitative feedback theory*. IEE Proc. Pt. D, 129:215–226, November 1982.

Introduction: Root Locus Controller Design, Control Tutorials for MatLAB and SIMULINK. MATLAB® 7.14, . Accessed 5 Oct 2018.

James, H.M., Nichols, N.B., Phillips, R.S., (1947) *Theory of servomechanisms*, volume 25 of MIT Radiation Laboratory Series, McGraw-Hill, New York.

John e. Bibel & D. Stephen Malyevac, (1992), *Guidelines for The Selection of Weighting Functions For H-Infinity Control,* Naval Surface Warfare Center Dahlgren Division, Dahlgren, Virginia.

Johnson, C.D. (1997), *Process Control Instrumentation Technology,* (5th edition). New Jersey: Prentice Hall, Inc.

Kalman, R.E., (1960a) *Contributions to the theory of optimal control*. Bol. Soc. Mat. Mexicana, vol.5, 1960, pp.102-119.

Kalman, R.E., (1960b) *A new approach to linear filtering and prediction problems*. ASME J. Basic Eng., vol.82, 1960, pp.34-45.

Kalman, R.E., Bertram, J.E., (1960) *Control system analysis and design via the "second method"of Lyapunov. I. Continuous-time systems*, ASME J. Basic Eng., 1960, pp.371-393.

Kalman, R.E., Bucy, R.S., (1961) *New results in linear filtering and prediction theory*. ASME J. Basic Eng., vol.80, 1961, pp.193-196.

Khargonekar P., Petersen I, and Rotea, M. (1987), H_∞ optimal control with state feed- back, *IEEE Trans. Aut. Control*, AC-33, PP 783-786.

Khargonekar P. (1991), State-space H_∞ control theory and the LQG problem, In: A.C. Antoulas (Editor), *Mathematical System Theory - The influence of R.E. Kalman,* Springer-Verlag, Berlin etc.

Krause J. M. (1992), Comments on Grimble's comments on Stein's comments on rolloff of H_∞ optimal controllers, *IEEE Trans. Aut. Control*, 702-702.

Kwakernaak H. (1986), *A polynomial approach to minimax frequency domain optimization of multivariable feedback systems*, *Int. J. Control*, PP120-156.

K. Zhou, J.C. Doyle and K. Glover (1996), *Robust And Optimal Control*, New Jersey: Prentice Hall.

Lyapunov, A.M., (1877) *general problem of movement stability*. Ann. Fac. Sci. Toulouse, vol.9, 1907, pp.203-474.

Limebeer, D., Anderson, B. (1988), *An interpolation theory approach to* H_∞ *controller degree bounds*, *Linear Algebra and its Applications,* PP 348-380.

Limebeer, D., Anderson, B., Khargonekar, P., and Green, M., *A game theoretic approach to* H_∞ *control for time-varying systems*, to appear in *SIAM J. Control*.

Mayr, Otto (1970). *The Origins of Feedback Control*. Clinton, MA USA: The Colonial Press, Inc.

Michalewicz, Z. (1992), *Genetic Algorithms + Data Structure = Evolution Programs*, New York: Springer-Verlag.

Maciejowski, J. (1996). *Multivariable feedback design*. 1st edition. Wokingham [u.a.]: Addison-Wesley.

Maxwell, J (1868), On Governors. in *Proceedings of the Royal Society of London*, 16, 270-283.

McFarlane D. C. and K. Glover (1990), *Robust Controller Design using Normalized Coprime Fac- tor Descriptions*, Lecture Notes in Control and Information Sciences, 138, Springer-Verlag, Berlin.

Meinsma G. (1991), *Polynomial methods in* H_∞ *optimal control*, Memorandum 1008, Dept. Appl. Mathematics, University of Twente, The Netherlands.

Meinsma G. (1992), *Optimal solutions to a two-block* H_∞ *problem*, Memorandum 1080, Dept. Appl. Mathematics, University of Twente, The Netherlands.

Meinsma G., *Optimal solutions to a two-block* H_∞ *problem*, accepted for publication in *Int. J. of Robust and Nonlinear Control.*

Meinsma G. (1993), *On* H_∞ *optimal solutions to two-block problems*, to appear in the proceed- ings of the *ECC-'93*, Groningen.

Minorsky, Nicolas (1922). Directional stability of automatically steered bodies. *Journal of the American Society of Naval Engineers*. 34 (2): 280–309. 10.1111/j.1559 3584.1922.tb04958.x

Ogata K. (2010). *Modern Control Engineering.* 5th edition. Upper Saddle River, New Jersey.

Ohlmeyer, E.J. (1990), *Robust Control Theory: CurrentStatus and Future Trends,* Naval Surface Warfare Center, Dahlgren, VA.

Seborg D., Edgar, T., and Mellichamp, D. (2004), *Process Dynamic and Control*, Wiley, 2nd edition.

Swan, Tony (April 2013). Ford's Assembly Line Turns 100: How It Really Put the World on Wheels. Car and driver. Retrieved 26 March 2017.

Tailor Made Mixing Systems - GEA Group [online]. [Accessed 3 November 2018]. Available at https://www.gea.com/en/products/Tailor-Made-Mixing-Systems.jsp

Ullmann, Fritz (2005). *Ullmann's Chemical Engineering and Plant Design*, Volumes 1–2. John Wiley & Sons.

Vaccum processing units - UNIMIX [online]. [Accessed 7 November 2018]. Available at https://www.ekato.com/en/products/process-plants/bb

Van Valkenburg, M. E (1984). University of Illinois at Urbana-Champaign, *In memoriam: Hendrik W. Bode (1905-1982)*, IEEE Transactions on Automatic Control, Vol. AC-29, No 3., pp. 193–194

Vardulakis A. I. G. (1991), *Linear Multivariable Control*, John Wiley & sons, Chichester, etc..

Whalley, R and Ebrahimi, M, (2006), *Multivariable System Regulation*, Proc. IMechE, Part C, Journal of Mechanical Engineering Science, Vol. 220, No.5, pp. 653-667.

Whalley, R., (2016), *Multivariable systems and control*, ENGG506 Lecture notes distributed in the module, the British University in Dubai, Dubai.

Whalley R.; Ebrahimi M., (2000), *Process System Regulation with Minimum Control Effort*, Proceedings of the IMechE Part E Journal of Process Mechanical Engineering, Volume 214, pp. 245-256.

Whalley R. & Ebrahimi M., (2004), *Automotive Gas Turbine Regulation*, IEEE Trans. Contr. Syst. Technol., pp. 465-473.

Whalley, R and Ebrahimi, M, (2006), *Multivariable System Regulation*, Proc. IMechE, Part C, Journal of Mechanical Engineering Science, Vol. 220, No.5, pp. 653-667.

Whalley, R.; Abdul-Ameer, A.; Ebrahimi, M. & Nikranjbar, A. (2010). *Winder Control using a Ward-Leondard System*. Int.Journal Industrial Systems Engineering, vol.6, No.2

Whalley, R., (2012), *Integrity Analysis for Multivariable Feedback Systems*, The 8th WSEAS International Conference on dynamical systems and control, Portugal,

Whalley, R., (2016), *State Space and Multivariable systems*, ENGG505 Lecture notes distributed in the module, the British University in Dubai, Dubai, September 2015.

Xue, D., Chen, Y. & Atherton, D. (2007). *Linear feedback control*. Philadelphia: Society for Industrial and Applied Mathematics.

Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, vol. 26 (2), pp. 301-320.

Appendix I: MATLAB® code

A. Least Effort controller MATLAB® code

```
%% A Program For Designing A Least Effort Controller For
%% pneumatic feed mixing system

clear all
clc
format short
format loose

disp('************************************************************')
disp('************************************************************')
disp(' This code is made to design a least effort comtroller for')
disp(' pneumatic feed mixing system')
disp('************************************************************')
disp('************************************************************')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('************************************************************')
disp('************************************************************')
disp(' Press Enter to start ....')
disp('************************************************************')
disp('************************************************************')

pause

%% The Open-Loop Transfer Function

syms n s

G11=tf([1.02],[11.76 1]);
G12=tf([-0.52],[10.25 1]);
```

```
G21=tf([-0.54],[10.25 1]);
G22=tf([1.04],[2.6 1]);

disp('The open-loop transfer function of the system, G(s), is')
G = [G11, G12; G21, G22];

%another form of G
G=[1.02/(11.76*s+1), -0.52/(10.25*s+1); -0.54/(10.25*s+1), 1.04/(2.6*s+1)];

pretty (vpa (G))

%% The Open-Loop system response

t=0:0.0001:50;
y1=step(G11,t);
y2=step(G12,t);
y3=step(G21,t);
y4=step(G22,t);

disp('================================================================')
disp('Press Enter to Plot the Open-loop system response')
disp('and display response info.')
disp('---------------------------------------------------------------')
pause

figure(1)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
disp('================================================================')
disp('from input1')
disp('---------------------------------------------------------------')
y1info = stepinfo(y1,t)
y2info = stepinfo(y2,t)
grid on

legend('y1(t)','y2(t)')
title('Open-loop response for a unit step input on u1(t)')
```

```
xlabel ('Time (s)')
ylabel ('output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);

figure(2)
plot(t,y3,'b',t,y4,'r-.','LineWidth',1.5)
disp('============================================================')
disp('from input2')
disp('------------------------------------------------------------')
y1info = stepinfo(y3,t)
y2info = stepinfo(y4,t)
grid on

legend('y1(t)','y2(t)')
title('Open-loop response for a unit step input on u2(t)')
xlabel ('Time (s)')
ylabel ('output response')
ax = gca;
ax.FontSize = 10;

axis([0,50,-0.6,1.2]);

%% Showing G(S) as G(S) = L(s) .A(s) / d(s) . R(s) . Gamma(s)
G=1/(26.65*s^2+12.85*s+1)*[1.02/(11.76*s+1), -0.52/(10.25*s+1); -0.54/(10.25*s+1), 1.04/(2.6*s+1)];
A = [1.02*(10.25*s+1), -0.52*(11.76*s+1); -0.54*(2.6*s+1), 1.04*(10.25*s+1)];
A = vpa (A);
d = (10.25*s+1)*(2.6*s+1);
d = vpa (d);
L=[1/((11.76*s+1)*(10.25*s+1)), 0; 0, 1/((2.6*s+1)*(10.25*s+1))];
L = vpa (L);
R = eye(2);
Gamma = eye(2);
```

```
% To Display A(S), d(s), L(s) and R(s)
disp('============================================================')
disp('Press Enter to display A(s), d(s), L(s), R(s) and Gamma(s)')
disp('------------------------------------------------------------')
pause
disp('A(s) is')
pretty(A)
disp('d(s) is')
pretty(d)
disp('L(s) is')
pretty (L)
R
Gamma

%% Forming the inner product <h.A(s).k>

syms n real k1 k2 h1 h2

hAk = [h1 h2]*A*[k1 k2]';
hAk = subs(hAk, k2, n*k1);
hAk = subs(hAk, k1, 1);

disp('============================================================')
disp('Press Enter to display the inner product<h.A(s).k>')
disp('------------------------------------------------------------')

pause

disp('The inner product is:')
pretty(hAk)

%% Presenting the system Q matix

Q=[(1.02-0.52*n) (-0.54+1.04*n);(10.455-6.1152*n) (-1.404+10.66*n)];

disp('============================================================')
```

```
disp('Press Enter to display the matrix Q')
disp('----------------------------------------------------------------')

pause

disp('The Q matrix is:')
pretty (vpa(Q))

%% Finding the steady-state value of the transfer function
disp('================================================================')
disp('Press Enter to display the steady-state of transfer function')
disp('----------------------------------------------------------------')
pause
G0 = limit(G,s,0);
pretty (vpa (G0))

%% Designing the inner loop

syms G0 b0 b

G0 = subs(G,s,0);
b = [0.09; 1];
b(1,1)=0.09;
b(2,1)=1;

%Choose the zero of numerator b(2,1)
disp('================================================================')
disp('Entering the value of the zero of b(s)/d(s)=-1')
disp('----------------------------------------------------------------')
X1=input('Enter the value of b(1,1) [0.09]:');

if isempty(X1)
    disp ('defult b(1,1) sellected : 0.09')
    X1 = 0.09;
end

```

```
b(1,1) = X1

sysbsds=tf([b(2,1) b(1,1)],sym2poly(d));

%root locus to sellect b0
disp('============================================================')
disp('Press Enter to display the root locus of b(s)/d(s)=-1')
disp('------------------------------------------------------------')
pause

figure(3)
rlocus(sysbsds)
ax = gca;
ax.FontSize = 10;
grid on

disp('============================================================')
display('The poles of bs/ds are:')
disp('------------------------------------------------------------')

p=pole(sysbsds)

% Choose b0 from root locus
% setting the defult to 0.33
syms b0

disp('============================================================')
disp('Entering the value of b0')
disp('------------------------------------------------------------')
X2=input('Enter the value of b0 [0.33]:');

if isempty(X2)
disp ('defult b0 sellected : 0.33')
X2 = 0.33;
end

```

```
b0 = X2;
b=b0*b
%% Finding the performance index, J
syms n
J=(1 + n^2) * (b)' * (inv(Q))' * inv(Q) * b; % performance index
disp('============================================================')
disp('Press Enter to display the performance index,J')
disp('------------------------------------------------------------')
pause
pretty(vpa(J))
disp('============================================================')
disp('Press Enter to display the graph of J as a function of n')
disp('------------------------------------------------------------')
pause
figure(4)
ezplot(J,[-4,10, -0.5*10E-4,9*10E-4]);
ax = gca;
ax.FontSize = 10;
grid
title ('preformance index "J" vs. Gain ratio "n"')
ylabel('Performance index J')
xlabel('Gain ratio n')
%% Finding the derivative of J to find the minimum
```

```
dJ=diff(J);

r=solve (dJ);
rnum = vpa (r);

% to find min
djmin = subs (J, n, r);
djminnum = vpa (djmin);

%J1=diff(J);
[num, den] = numden (dJ);
J1 = num / den;

% Finding the values of 'n' for which J is minimum
syms x
dJ = subs(dJ, n, x);
n = solve(dJ);
n = double(n);
disp('=============================================================')
disp('Press Enter to display n for which J has an extremum')
disp('-------------------------------------------------------------')
pause

N = sort(n)
% Findind the corresponding values of J
dJ = subs(J,n);

%% Finding the value of n at which J is the minimum
disp('=============================================================')
disp('Press Enter to display the value of n at which J is minimum')
disp('-------------------------------------------------------------')

for nn = 1 : length(n)
if isreal(n(nn))==0 n(nn) = inf;
end
end
```

```
n = n (isfinite(n));
J = subs(J,n);
pause

n = n(find (J == min(J)))
disp('============================================================')
disp('Press Enter to display the corresponding value of matrix Q')
disp('------------------------------------------------------------')
pause

Q = subs(Q,n);
pretty (vpa(Q))

%% Finding the value of (h1, h2)
disp('============================================================')
disp('Press Enter to display the value of h1, h2')
disp('------------------------------------------------------------')
pause

k1=1;
h=inv(Q)*b/k1;
pretty (vpa(h))
h = double (h);

%% Finding h(s)

disp('============================================================')
disp('Press Enter to display h(s)')
disp('------------------------------------------------------------')
pause

hs(1,1) = h(1,1)*(11.76*s+1)/(2.6*s+1);
hs(1,2) = h(2,1);

pretty (vpa(hs));
```

```
%% Finding the value of k
disp('============================================================')
disp('Press Enter to display k')
disp('-----------------------------------------------------------')
pause

k2=n*k1;
k=[k1 k2]'

%% Defining Ss
disp('============================================================')
disp('Press Enter to display the value of Ss')
disp('-----------------------------------------------------------')

pause

Ss=[1 -0.1;-0.1 1]

%% Defining the matrix F
disp('============================================================')
disp('Entering the value of f')
disp('-----------------------------------------------------------')
r1=input('Enter the value of f (0<f<1)[0.8]: ')

if isempty(r1)
    disp ('defult (f) sellected : 0.8')
    r1 = 0.8;
end

f = r1;


F=[f 0;0 f]

```

```
%% Outer loop feed-forward gain
disp('============================================================')
disp('Press Enter to display the feed-forward')
disp('gain matrix of the outer loop,p')
disp('----------------------------------------------------------')
pause

P=(inv(G0)+k*h')*Ss*inv((eye(2)-F*Ss));
pretty (vpa(P))


%% Conventional outer loop feedback gain
disp('============================================================')
disp('Press Enter to display the system feed-back')
disp('gain matrix of the outer loop, H')
disp('----------------------------------------------------------')
pause

H=(inv(P)*(k*hs))+F;
pretty (vpa(H));

HH=inv(P)*([1; 7.1267]*[-0.001939*((11.76*s+1)/(2.6*s+1)), 0.00356])+F;
pretty (vpa(HH))

%% Simulation and plotting the system response

% defining the system inputs
syms r1 r2 d1 d2

disp('============================================================')
disp('Entering the value of r1')
disp('----------------------------------------------------------')
r11=input('Enter the value of r1 (0 or 1)[0]:')

if isempty(r11)
    disp ('defult (r1) sellected : 0')
```

```
    r11 = 0;
end

r1=r11;

disp('=============================================================')
disp('Entering the value of r2')
disp('-------------------------------------------------------------')
r22=input('Enter the value of r2 (0 or 1)[0]:')

if isempty(r22)
    disp ('defult (r2) sellected : 0')
    r22 = 0;
end

r2=r22;

% defining the system disturbance

disp('=============================================================')
disp('Entering the value of d1')
disp('-------------------------------------------------------------')
d11=input('Enter the value of d1 (0 or 1)[0]:')

if isempty(d11)
    disp ('defult (d1) sellected : 0')
    d11 = 0;
end

d1=d11;

disp('=============================================================')
disp('Entering the value of d2')
disp('-------------------------------------------------------------')
d22=input('Enter the value of d2 (0 or 1)[0]:')

```

```
if isempty(d22)
    disp ('defult (d2) sellected : 0')
    d22 = 0;
end

d2=d22;

%Running the simulation
disp('||||||||||||||||(Enter to run Simulation )|||||||||||||||||||')
pause
disp('||||||||||( Simulation running, please wait ... )||||||||||||')

sim('least_effort')

%% Plotting the system response

if r1 == 1
disp('=========================================================')
disp('Press Enter to plot system response to unit step input on r1(t)')
disp('----------------------------------------------------------')
pause

figure(5)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
y1info = stepinfo(y1,t)
y2info = stepinfo(y2,t)
grid on
legend('y1(t)','y2(t)')
title('Closed-loop response for a unit step input on r1(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end
```

```
if r2==1
disp('============================================================')
disp('Press Enter to plot system response to unit step input on r2(t)')
disp('------------------------------------------------------------')
pause

figure(5)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
y1info = stepinfo(y1,t)
y2info = stepinfo(y2,t)
grid on
legend('y1(t)','y2(t)')
title('Closed-loop response for a unit step input on r2(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end
if d1 == 1
disp('============================================================')
disp('Press Enter to plot system response to unit step on d1(t)')
disp('------------------------------------------------------------')
pause

figure(5)
plot(t, y1, 'b', t, y2, 'r-.', 'LineWidth', 1.5)
y1info = stepinfo(y1,t)
y2info = stepinfo(y2,t)
grid on
legend('y1(t)','y2(t)')
title('Closed-loop response for a unit step input on d1(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
```

```
axis([0,50,-0.6,1.2]);
end
if d2==1
disp('============================================================')
disp('Press Enter to plot system response to unit step on d2(t)')
disp('------------------------------------------------------------')
pause
figure(5)
plot(t, y1,'b',t, y2, 'r-.', 'LineWidth', 1.5)
y1info = stepinfo(y1,t)
y2info = stepinfo(y2,t)
grid on
legend('y1(t)','y2(t)')
title('Closed-loop response for a unit step input on d2(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end
disp('============================================================')
disp('Press Enter to plot system Energy consumption')
disp('------------------------------------------------------------')
pause
figure(6)
plot(t,E,'r','LineWidth',1.5)
grid on
legend('Energy consumption')
title('Control energy consumption vs. time ')
xlabel ('Time (s)')
ylabel ('Control energy consumption')
ax = gca;
ax.FontSize = 10;

disp('||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
```

```
disp('|||||||||||||||||||||(End of program)||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
```

B. H-infinity controller MATLAB® code

```
%% A Program For Designing H-infinity Controller For
%% pneumatic feed mixing system
clear all
clc
format short
format loose
disp('*************************************************************')
disp('*************************************************************')
disp(' This code is made to design an H-infinity comtroller for')
disp(' pneumatic feed mixing system')
disp('*************************************************************')
disp('*************************************************************')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('*************************************************************')
disp('*************************************************************')
disp(' Press Enter to start ....')
disp('*************************************************************')
disp('*************************************************************')
pause
%% The Open-Loop Transfer Function
syms n s
s=tf('s');
G11=tf([1.02],[11.76 1]);
G12=tf([-0.52],[10.25 1]);
G21=tf([-0.54],[10.25 1]);
G22=tf([1.04],[2.6 1]);

disp('The open-loop transfer function of the system, G(s), is')
G = [G11, G12; G21, G22]
```

```
%% to find the state space matrices in its minimum realization.
sys=ss(G,'min');
[A,B,C,D] = ssdata(sys);

% Displaying A,B,C and D for open-loop
disp('============================================================')
disp('Press Enter to display A,B,C,D for open-loop system')
disp('------------------------------------------------------------')
pause
(A)
(B)
(C)
(D)

%% The Open-Loop system response

t=0:0.0001:50;

y1=step(G11,t);
y2=step(G12,t);
y3=step(G21,t);
y4=step(G22,t);

disp('============================================================')
disp('Press Enter to Plot the Open-loop system response')
disp('and display response info.')
disp('------------------------------------------------------------')
pause

figure(1)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
disp('============================================================')
disp('from input1')
disp('------------------------------------------------------------')
y1info = stepinfo(y1,t)
```

```
y2info = stepinfo(y2,t)
grid on

legend('y1(t)','y2(t)')
title('Open-loop response for a unit step input on u1(t)')
xlabel ('Time (s)')
ylabel ('output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);

figure(2)
plot(t,y3,'b',t,y4,'r-.','LineWidth',1.5)
disp('==========================================================')
disp('from input2')
disp('----------------------------------------------------------')
y1info = stepinfo(y3,t)
y2info = stepinfo(y4,t)
grid on

legend('y1(t)','y2(t)')
title('Open-loop response for a unit step input on u2(t)')
xlabel ('Time (s)')
ylabel ('output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);

%% To define the weighting functions

W1 = [(0.1*s+2)/(5*s+0.05), 0; 0, (0.1*s+2)/(5*s+0.05)];
W2 = [tf(1e-8), 0; 0, tf(1e-8)];
W3 = [s/10, 0; 0, s/10];

% Displaying the weighting functions
disp('==========================================================')
```

```
disp('Press Enter to display the weighting functions')
disp('------------------------------------------------------------')
pause
(W1)
(W2)
(W3)
%% H-infinity Controller design
% To define the augmented system
SysTss = augtf(G,W1,W2,W3);

% Displaying the augmented system
disp('============================================================')
disp('Press Enter to display the augmented system model')
disp('------------------------------------------------------------')
pause
SysTss

%The two-port format of the augmented system
[a,b1,b2,c1,c2,d11,d12,d21,d22]=branch(SysTss);

% To apply H-infinity controller on the defined augmented system
disp('============================================================')
disp('Press Enter to design an H-infinity controller')
disp('for the augmented system model')
disp('------------------------------------------------------------')
pause
[gamma, Gc] = hinfopt (SysTss)

%% Defining A,B,C,D of Gc
[Ac,Bc,Cc,Dc] = ssdata(Gc);

%% defining num, den of the elemts of H-infinity controller
[numg11,deng11] = tfdata(Gc(1,1));
[numg21,deng21] = tfdata(Gc(2,1));
[numg12,deng12] = tfdata(Gc(1,2));
[numg22,deng22] = tfdata(Gc(2,2));
```

```
% To display the controller transfer function
disp('============================================================')
disp('Press Enter to display H-infinity controller transfer funtion')
disp('----------------------------------------------------------')
pause

zpk(Gc)

%% Simulation and plotting the system response
% defining the system inputs
syms r1 r2 d1 d2

disp('============================================================')
disp('Entering the value of r1')
disp('----------------------------------------------------------')
r11=input('Enter the value of r1 (0 or 1)[0]:')
if isempty(r11)
    disp ('defult (r1) sellected : 0')
    r11 = 0;
end
r1=r11;
disp('============================================================')
disp('Entering the value of r2')
disp('----------------------------------------------------------')
r22=input('Enter the value of r2 (0 or 1)[0]:')

if isempty(r22)
    disp ('defult (r2) sellected : 0')
    r22 = 0;
end
r2=r22;
% defining the system disturbance
disp('============================================================')
disp('Entering the value of d1')
disp('----------------------------------------------------------')
```

```
d11=input('Enter the value of d1 (0 or 1)[0]:')
if isempty(d11)
    disp ('defult (d1) sellected : 0')
    d11 = 0;
end
d1=d11;
disp('============================================================')
disp('Entering the value of d2')
disp('------------------------------------------------------------')
d22=input('Enter the value of d2 (0 or 1)[0]:')

if isempty(d22)
    disp ('defult (d2) sellected : 0')
    d22 = 0;
end
d2=d22;

%Running the simulation
disp('|||||||||||||||(Enter to run Simulation )|||||||||||||||||')
pause
disp('||||||||||( Simulation running, please wait ... )||||||||||||')
sim('H_infinity_thesis')

%% Plotting the system response
if r1 == 1
disp('============================================================')
disp('Press Enter to plot system response to unit step on r1(t)')
disp('------------------------------------------------------------')
pause

figure(3)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
y1info = stepinfo(y1,t)
grid on

legend('y1(t)','y2(t)')
```

```
title('Closed-loop response for a unit step input on r1(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end

if r2==1
disp('================================================================')
disp('Press Enter to plot system response to unit step on r2(t)')
disp('---------------------------------------------------------')
pause

figure(3)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
y2info = stepinfo(y2,t)
grid on
legend('y1(t)','y2(t)')
title('Closed-loop response for a unit step input on r2(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end
if d1 == 1
disp('================================================================')
disp('Press Enter to plot system response to unit step on d1(t)')
disp('---------------------------------------------------------')
pause
figure(3)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
y1info = stepinfo(y1,t)
grid on
legend('y1(t)','y2(t)')
```

```
title('Closed-loop response for a unit step input on d1(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end
if d2==1
disp('============================================================')
disp('Press Enter to plot system response to unit step on d2(t)')
disp('------------------------------------------------------------')
pause
figure(3)
plot(t,y1,'b',t,y2,'r-.','LineWidth',1.5)
y2info = stepinfo(y2,t)
grid on
legend('y1(t)','y2(t)')
title('Closed-loop response for a unit step input on d2(t)')
xlabel ('Time (s)')
ylabel ('Output response')
ax = gca;
ax.FontSize = 10;
axis([0,50,-0.6,1.2]);
end
disp('============================================================')
disp('Press Enter to plot system Energy consumption')
disp('------------------------------------------------------------')
pause
figure(4)
plot(t,E1,'r','LineWidth',1.5)
grid on
legend('Energy consumption')
title('Control energy consumption vs. time ')
xlabel ('Time (s)')
ylabel ('Control energy consumption')
ax = gca;
```

```
ax.FontSize = 10;

disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||(End of program)||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
disp('|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||')
```

Appendix II: Simulation Model

A. Least Effort Simulation Model

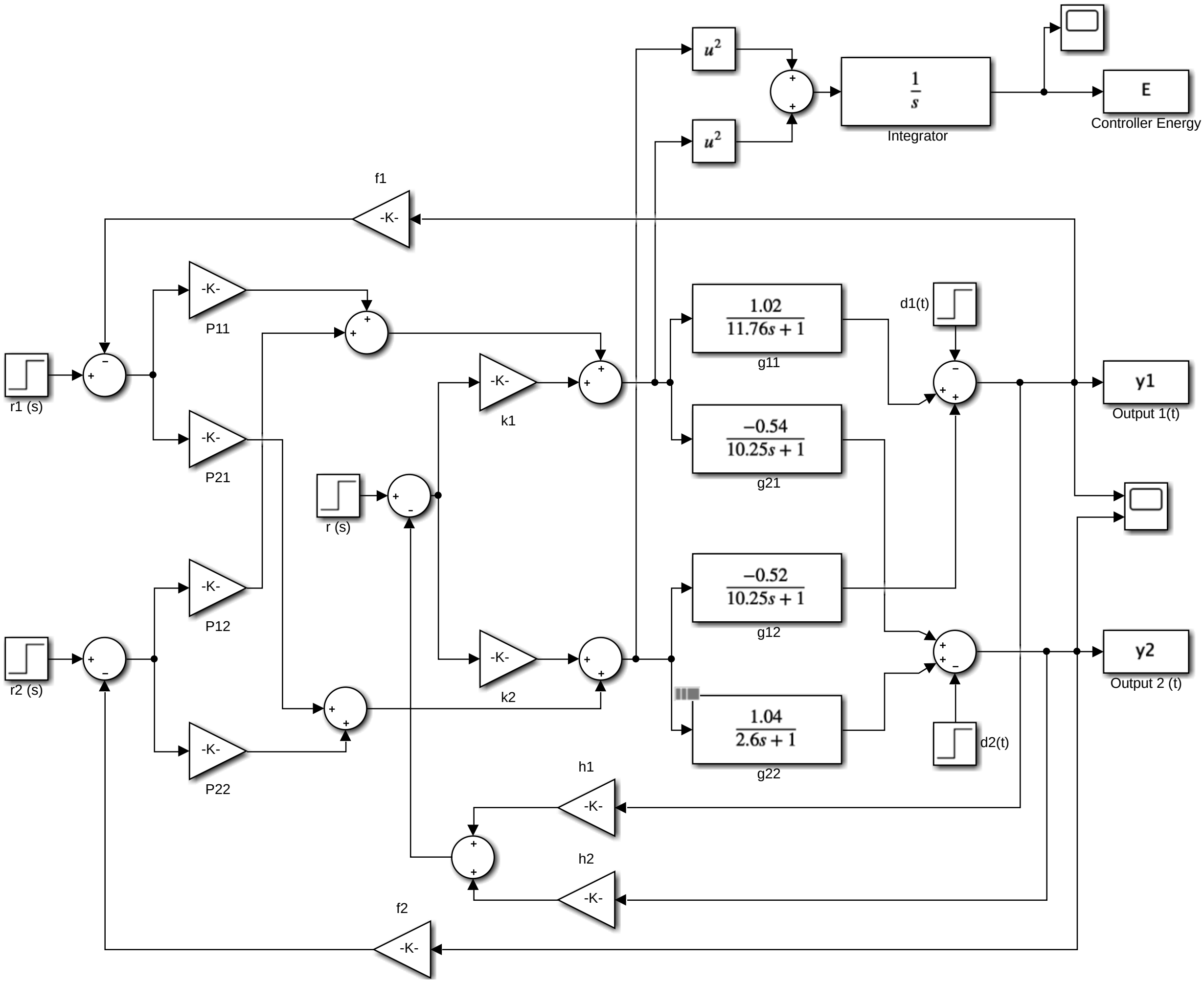

Figure A.1: least effort controller simulation model

B. H-infinity controller Simulation Model

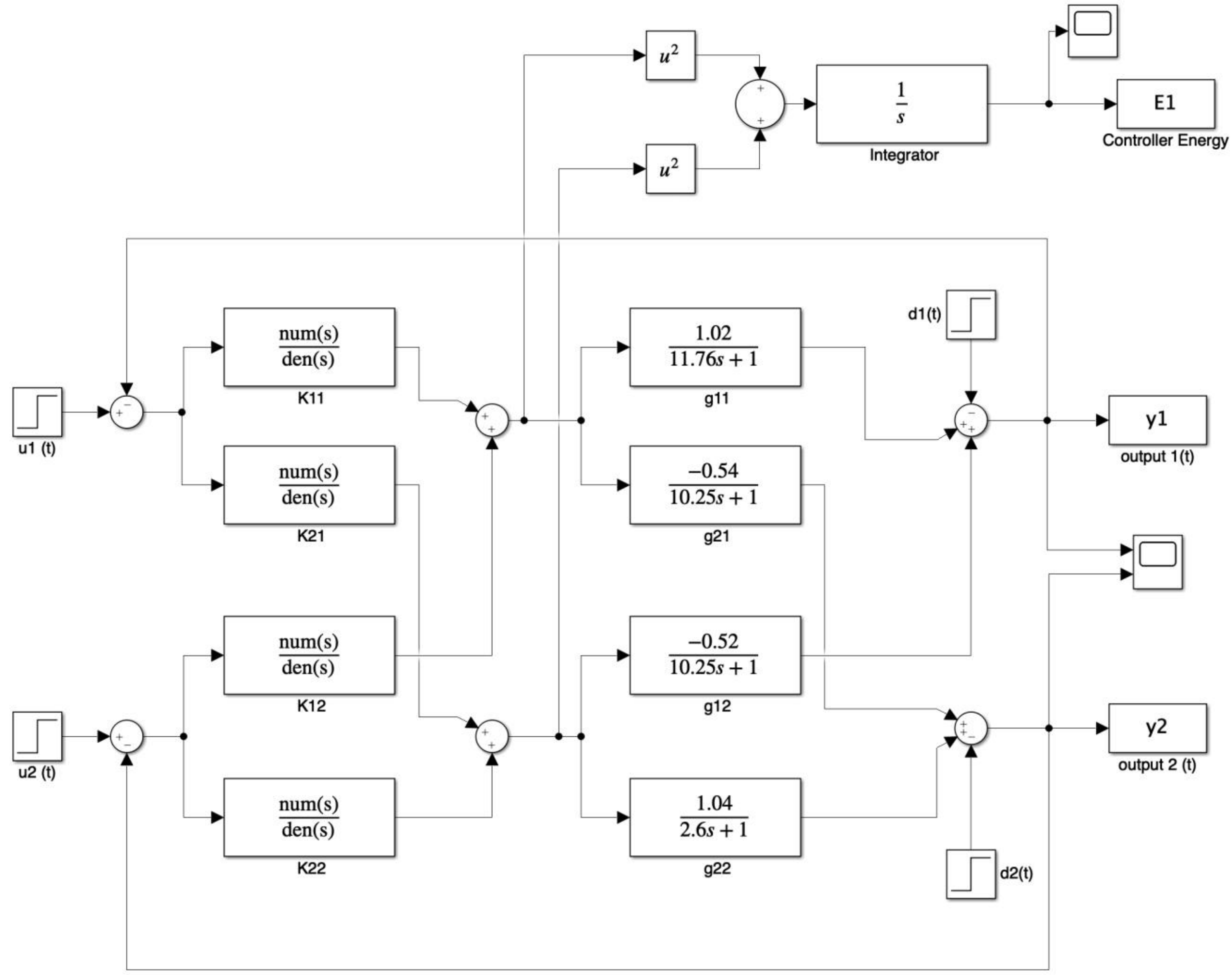

Figure B.1: H-infinity controller simulation model