POLITECNICO DI MILANO

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING

Msc. in Automation and Control Engineering

Coordinated Control of Voltage and Frequency in DVFS Capable Devices

Supervisor

Prof. Alberto LEVA

Master Thesis by:

Ömer ÖZTÜRK

Academic Year 2015-2016

To my lovely mother, father and sister...

I would like to express my deepest appreciation to my thesis supervisor Professor Alberto LEVA. Without his guidance and help this work would not have been possible. I *consider it an honor to work with Professor Alberto LEVA. Also I would like to thank to all my Professors and colleagues for their effort and help during my long journey at Polytechnic University of Milan. In addition, a thank you to my friend Bariscan DEDES, who supported me with his knowledge and friendship during this work.*

CONTENTS

LIST of FIGURES

LIST of TABLES

ABSTRACT

In today's data communication systems such as high-speed Ethernet transceivers, disk read/write channels, mobile digital receivers, memory interfaces, and so on, Phase Locked Loops (PLLs) are used. The first concepts on the subject date back to 1930 and a theoretical description of a PLL was published in the late '70s, however, the PLL has not spread rapidly due to manufacturing difficulties. Focusing for the purpose of this work on microprocessors, those that include a PLL also include a voltage regulator, given that operation at a certain frequency requires a minimum value of the supply voltage. The coordinated control of the two, however, is made difficult by the diversity between the dynamics of a PLL and that of a voltage regulator. In this work we study some possible techniques for coordinated control of the two systems just mentioned, so as to operate the microprocessor within the admissible voltage and frequency zone, without, however, excessively raise the first for reasons of conservatism, in order to limit consumption. The techniques are tested in simulation and critically compared, to identify the most suitable for typical use cases.

SOMMARIO

Negli odierni sistemi di comunicazione dati quali ricevitori/trasmettitori Ethernet ad alta velocità, canali di lettura/scrittura di dischi, ricevitori digitali mobili, interfacce di memoria e così via, si usano anelli ad aggancio di fase (Phase Locked Loops, PLL). I primi concetti in proposito risalgono al 1930 e una descrizione teorica di un PLL è stata pubblicata alla fine degli anni 70, tuttavia il PLL non si è diffuso rapidamente a causa di difficoltà realizzative. Concentrandosi per lo scopo di questo lavoro sui microprocessori, in quelli che includono un PLL è compreso anche un regolatore di tensione, dato che il funzionamento a una certa frequenza richiede un minimo valore della tensione di alimentazione. Il controllo coordinato dei due è però reso difficile dalla diversità tra la dinamica di un PLL e quella di un regolatore di tensione. In questo lavoro si studiano alcune possibili tecniche di controllo coordinato dei due sistemi menzionati in modo da far operare il microprocessore entro la zona di tensione e frequenza ammissibile, senza però alzare eccessivamente la prima per ragioni di conservatività in modo da limitare il consumo. Le tecniche sono provate in simulazione e confrontate criticamente per individuare la più adatta ai tipici casi di utilizzo.

Chapter 1

INTRODUCTION

1.1 Introduction

In nowadays' high-speed data communication systems such as Ethernet receivers, disk drive read/write channels, digital mobile receivers, high-speed memory interfaces , Phase Locked Loops are used. Mainly phase locked loop system is a feedback system which tries to generate an output signal whose phase is consistent with the phase of an input signal, phase locked loop can be analog or digital. Basically general structure of phase locked loops consist of three components which are phase detector, loop pass filter and voltage controlled oscillator.[1] Block diagram for the depicted general structure of PLL shown below:

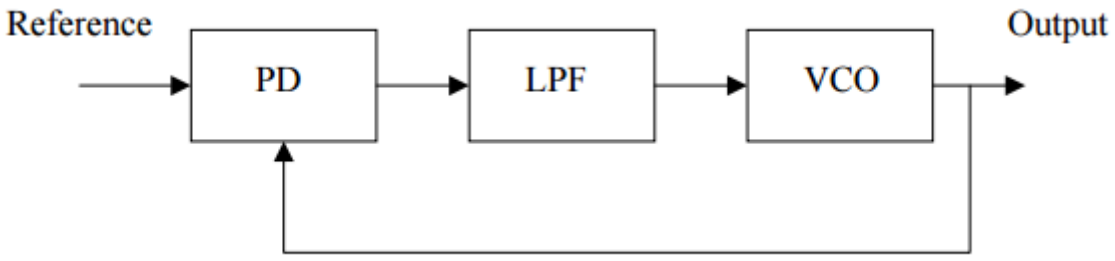

Figure 1: Phase locked Loop system

Main working principle behind PLL can be explain in some basic steps. First of all feedback system receives periodic signal as an input and phase detector compares its phase with output of the voltage controlled oscillator. Then output of the phase detector enters to loop low pass filter and output of the loop low pass filter behaves as control signal for the voltage controlled oscillator in order to drive it. Therefore as a result of feedback system the

voltage controlled oscillator will lock onto the reference signal which means that is it will become a tool for tracking a periodic signal while its phase and frequency varies.

An early concept of phase locked loops are described 1930's and the exact theoretical description of PLL was published in the late 1970's, but PLLs did not become widespread in modern communication circuits until the rapid development of integrated circuits due to the difficulty of realization. After integrated circuits developments, PLLs became more common in modern electronic systems and this improvement has enabled modern electronic systems to improve performance and reliability. [2]. For instance PLL application can be listed as frequency modulation stereo decoders, FM demodulation networks for FM operation, frequency synthesis that provides multiple of reference signal frequency. Also PLLs can be used in motor speed controls, tracking filters and frequency shift keying decodes for demodulation carrier frequencies.

Usage of PLLs in modern electronic circuits provide benefits to the whole system. Main advantage of PLL is that due to their working principle that mentioned before they synchronize the internal and external clocks. If the input clock of the system varies slightly the PLL frequency does not vary. This narrow frequency band is known as the dead zone of the phase detector. Inside this zone the feedback clock and the external clock are so close in phase that there are no correction pulses out of the phase-detector circuit, but the phase locked loops lock in very short time, due to this fast locking PLL based clock drivers are termed no dead zone.[3] On the other hand PLLs have some disadvantages. First of all due to complexity of their structure PLLs are expensive and also they are inherently noise sensitive[3].

In the modern electronic systems which has PLL inside, PLL requires power in order to work. Therefore PLLs should be powered by a voltage regulator. Voltage regulator is designed to automatically stabilize a constant voltage level, beside this voltage regulator also change and stabilize voltage level due to needs of the system. Therefore in circuit systems voltage regulator are present for two main reasons: First in order to regulate or vary the output voltage of the circuit or in order to stabilize the output voltage at desired constant level[4]. One type of the commercial voltage regulators can be seen below:

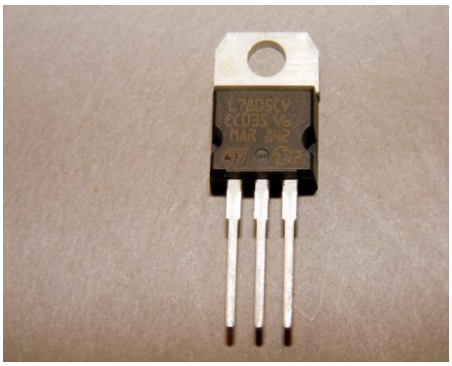

Figure 2: Voltage regulator example

Voltage regulators are widely used components for the important devices and systems. For instance voltage regulators take part in cars as alternators. Also computer power supplies are one type of voltage regulator which maintain a desired constant voltage level for the processor and other elements and also voltage regulators tries to stabilize the output of plant in central power station generator plants. Due to desired voltage levels and application types, different voltage regulators are present which can be listed as electromechanical regulator, electronic regulator, automatic voltage regulator, constant voltage transformer

1.2 Modelica Modeling Enviorment

In order to fulfill needs of the users, a lot of modeling and simulation tools are present in the market. While some tools are focusing on the one aspect, some certain tools are useful for more general cases. For instance Modelica is a hybrid modeling language which make possible to model complex systems covering multiple application domains[6]. Modelica which is provided and improved by the Modelica Association, is a modern , object-oriented, equation based language to conveniently model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. [5].

First generations of object oriented mathematical modeling languages and simulation systems are emerged in 1996[6] but due to mismatching between object oriented languages results were not successful. After this trials, a group of researchers and developers from universities and industries came together in order to define standardization and make object oriented language widespread. Therefore they found Modelica Association which is s a non-profit, non-governmental organization with the aim of developing and promoting the Modelica modeling language for modeling, simulation and programming of physical and

technical systems and processes. Version 1.0 of Modelica is released in 1997 and with a lot of changes and developments Modelica reached to today's version. Latest version of Modelica Language which has multi domain capability allows defining models in a declarative manner, modularly and hierarchically and combining various formalisms expressible in the more general Modelica formalism[6].

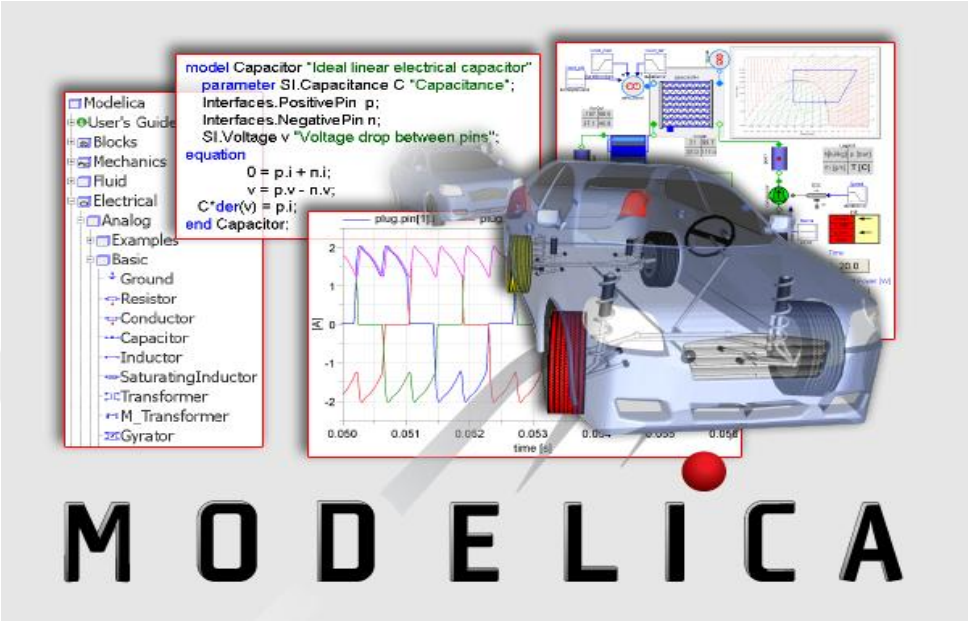

Figure 3: views from Modelica

Modelica Languages has depicted advantages over other available tools beside this Modelica is free and easy to access modeling and simulation language. Therefore due to all benefits of Modelica Language for this thesis work selected simulation and modeling tool is Modelica.

1.3 Problem Description

System which will be investigated for the thesis work consist of one phase-locked loop, one voltage regulator and main idea is behind this work is develop controller in order to balance different settling times of phase locked loop and voltage regulator. Phase- locked Loops are widely used in high speed data communications systems due to their fast locking times, bandwidth and stability characteristics. Present PLL in the investigated system is a type III third order charge pump PLL with transconductor-c(Gm-C) structure. As a result of this structure settling time of PLL decrease a lot. [7] Block diagram of the present PLL can be seen below:

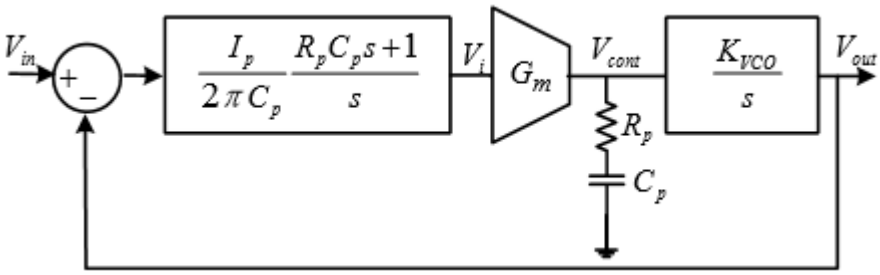

Figure 4: Schematic diagram of PLL

Voltage regulators are designed to automatically maintain a constant voltage level but dynamics of voltage regulators are really slower than PLL's. Therefore in the system which is asked to has synchronous work of PLL and voltage regulator, there will be mismatches between settling time of PLL and voltage regulator.

Normally in this kind of systems in order to reach desired frequency first we need to change voltage by voltage regulator which has slower dynamic. When voltage has reached to desired point then give input to PLL to reach desired value of the frequency. That operation is represented by black line in Figure. With aid of designed controller, we can approach to reaching desired frequency problem with new method that is changing voltage and frequency at the same time which is represented by red line in Figure. In the first approach changing voltage and frequency one by one requires more power than the second approach which leads to waste of power for whole system. Beside this for first approach total required time for the operation is sum of time required to change voltage and time required to change frequency. On the other hand total time for second method will be shorter than first method due to synchronous change in voltage and frequency. Therefore second approach is more efficient and faster than the first approach.

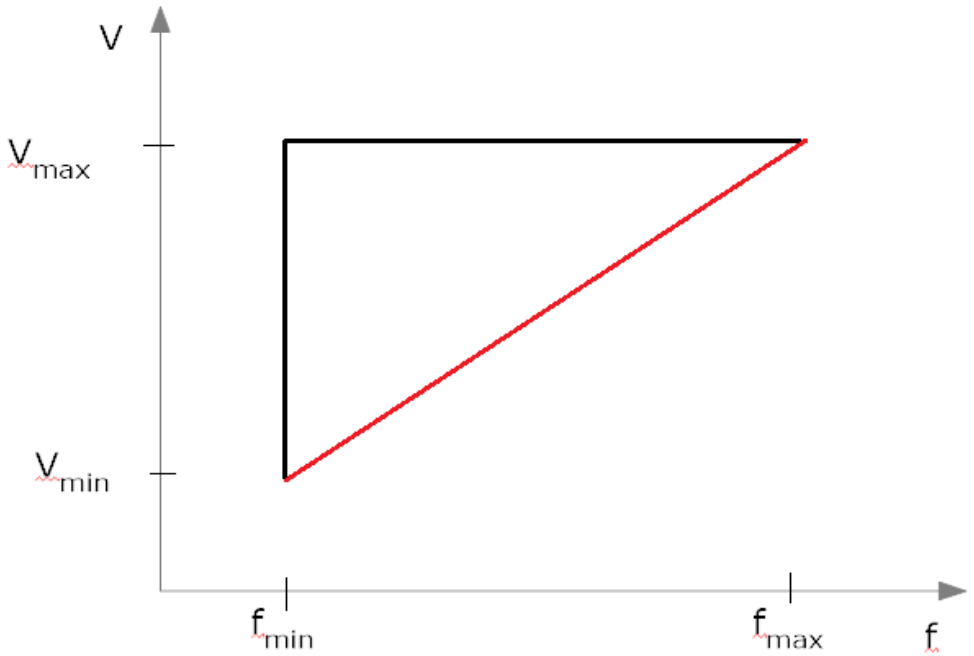

Figure 5: Representation of two methods for reaching desired frequency and voltage

1.4 Purpose and Organization of the Thesis

Target system for the thesis work consists of one phase locked loop and voltage regulator. As depicted in the section 1.3 phase locked loops have fast dynamics therefore their settling of it is short. On the other hand voltage regulators have slower dynamics than the phase locked loops this characteristic dynamics of the two components leads to a mismatch between settling times of two components. Because of mismatch problem, in order to reach desired frequency; first voltage should be changed by voltage regulator and then frequency will be changed by the phase locked loop. Changing voltage and frequency one by one instead of one shot will leads to waste of power and time for the system.

During this thesis work different control schemes such as PID, LQR and Feed forward will be combined to the phase locked loop and voltage regulator in order to synchronizes settling times. With the help of control mechanisms, settling time of voltage regulator will decrease and approach to the settling time of the phase locked loop. Therefore voltage and frequency change will be done simultaneously in order to save time and energy. In other words main aim behind this thesis work is with the help of different controller schemes trying

to solve this mismatches and balance settling time. In the course of thesis a system which has a controller, a phase locked loop and a voltage regulator created and simulated in the Modelica modeling environment. Beside this, application of different control schemes, gives chance to see effects and results of the different control mechanisms and compare the results of them. Block diagram of the generic system can be seen below:

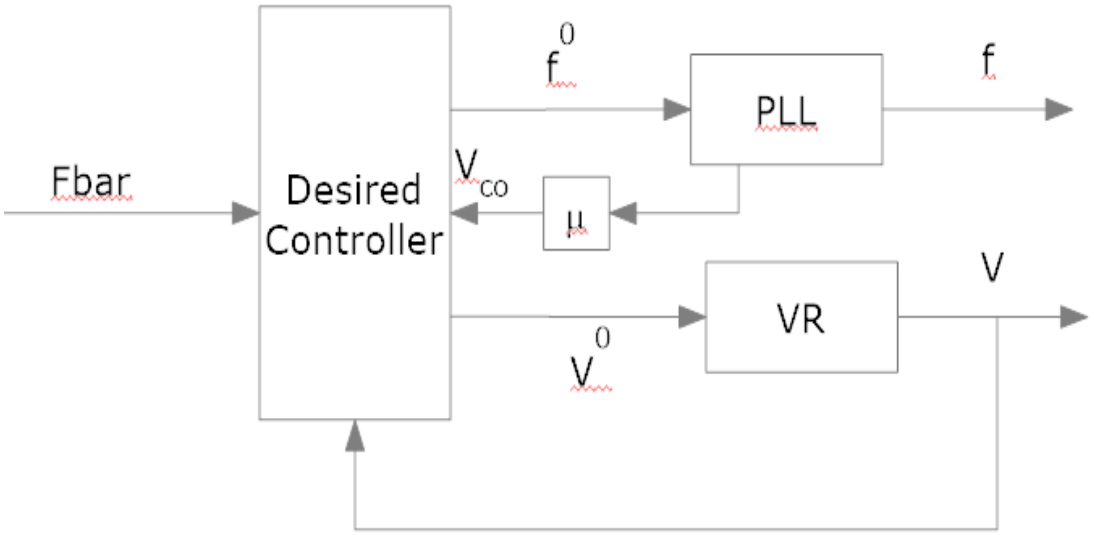

Figure 6: Generic view of the system

Organization of the work is in the following order:

Chapter 2: Introduces meaning of the mathematical models than generates state space models and transfer functions of the components also introduces control notion is introduced and finally describes different control schemes such as PID controller, linear quadratic regulator and feed forward action

Chapter 3: Mentions design steps of the different control schemes , implements different control schemes and combinations of them to the system of interests in the Modelica environment, generates step responses of the designed systems in the Modelica programming language .

Chapter 4: Concludes the work and defines future scope.

Chapter 2

MODELS OF SYSTEM COMPONENTS AND CONTROL SCHEMES

2.1 Introduction

In this chapter, mathematical models of the voltage regulator and phase locked loop are expressed in different forms also details of control schemes are introduced. Chapter two is organized as follows: firstly transfer functions of the of components are introduced which will be useful for the application of PID and feed forward control schemes. Secondly from transfer function of components, state space models are found which are key tools for the LQR controller scheme. Finally details of the used control schemes, applications of them and differences between them are explained in this chapter

2.2 Transfer functions

The transfer function also known as system function is ratio the output of a system to the input of a system in Laplace domain, considering its initial conditions and equilibrium point to be zero [8]. This ratio is generally denoted by $G(s)$ for example:

$$G(s) = \frac{Y(s)}{U(s)}$$

where Y(s) is denoting output of the system and U(s) is input of the system. In this representation the output is always equals to the transfer function times input:

$$Y(s) = G(s) * U(s)$$

Therefore block diagram of transfer function can be represented as follows:

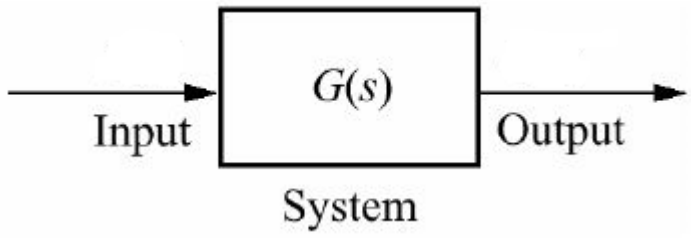

Figure 7: Generic block diagram of transfer functions

Transfer Functions are useful in application because of following reasons[10];

- Transfer function helps in analyzing stability of the whole system.
- By analyzing transfer function of the system, poles and zeros of the system can be determined easily.
- With the knowledge of the transfer function, response of the system to any input can be found easily.
- Transfer function is kind of mathematical model for the system and provides the gain of the system.

2.2.1 Transfer function of the Phase Locked Loop and Voltage Regulator

. In order to design control scheme for our system we need to have transfer functions of the components of the system. Generic transfer function for the PLL is :

$$PLL(s) = \frac{1+s*\tau}{1+2*\frac{\xi_{pll}}{w_{npll}}*s+\frac{s^2}{w_{npll}^2}}$$

Generic transfer function for the Voltage regulator can be defined as :

$$VR(s) = \frac{1}{1+2*\frac{\xi_{vr}}{w_{nvr}}*s+\frac{s^2}{w_{npvr}^2}}$$

By definition, dynamics of the PLL are faster than dynamics of the voltage regulator. Therefore we can comment on the time constants and the natural frequencies of the components :

$$\frac{1}{w_{nvr}} \gg \tau, \frac{1}{w_{npll}}$$

In order to find real values for the real values of the transfer functions of the components, some formulas are exist which are useful for the second order systems. If typical second order transfer function is:

$$G(s) = \frac{1}{1+2*\frac{\xi}{w_n}*s+\frac{s^2}{w_n^2}}$$

According to valid formulas for typical second order transfer function:

$$T_r = \frac{1.8}{w_n},\ \ T_s = \frac{-(tolerance\ fraction)}{\xi * w_n},\ for\ tolerance\ fraction = 5\ \ T_s = \frac{3}{\xi * w_n}$$

After formulas, it's obvious that settling time and damping ratio is required for calculation of natural frequency. To find settling time and damping ratio step response is needed. Phase locked loop step response can be found in the paper of Habib Adrang and Hossein Miar Naeimi which named as " A type III Fast Locking Time PLL with Transconductor-C Structure:

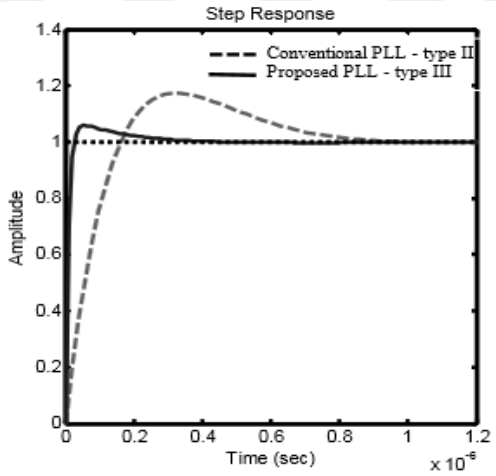

Figure 8: Step response of PLL

Also in the same paper, table which includes settling time and damping ratio of the phase locked loop can be found:

	Conventional PLL	Proposed PLL
Phase Margin	70°	85°
ω_{PM} (Mrad/S)	11.5	108
Rise Time	120 ns	18 ns
Settling Time	775 ns	215 ns
Overshoot	17.4	5.9
Damping ratio (ξ)	0.7	0.9

Table 1: Performance summary of PLL

After applying formulas to the values of the paper $result\ is\ w_{npp} = 15.5\ MHz$

In the system of interest the PLL has transfer function in the form of

$$PLL(s) = \frac{1+s*\tau}{1+2*\frac{\xi_{pll}}{w_{npll}}*s+\frac{s^2}{w_{npll}^2}}$$

which means that PLL has also one zero. In order to find position of zero therefore define the exact transfer function of the PLL bode diagram of it is needed. Also bode diagram is provided in the mentioned paper :

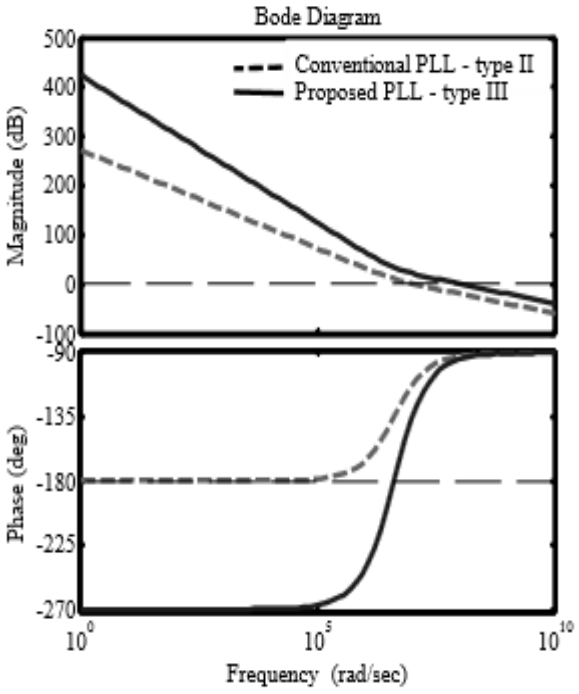

Figure 9: Bode diagram of the PLL

From the bode plot it is obvious that, there is a zero around $10^7 Hz$. Also its known that $f_z = \frac{1}{\tau_z}$ so $\tau_z = 10^{-7}s$. After calculation of zero, exact transfer function of the PLL is founded which is:

$$PLL(s) = \frac{1+1*10^{-7}*s}{1+12*10^{-8}*s+42*10^{-16}*s}$$

In the system of interest, other component is voltage regulator which is with two complex poles and no zero. Therefore generic transfer for the voltage generator is as follows:

$$VR(s) = \frac{1}{1 + 2 * \frac{\xi_{vr}}{w_{nvr}} * s + \frac{s^2}{w_{npvr}^2}}$$

As mentioned in the previous chapters dynamics of voltage regulators are really slower than phase locked loops. Used voltage regulator for the investigated system is oscillatory and its settling time is equals to 30ms. In order to achieve oscillatory property of the voltage regulator, its damping ratio is selected as 0.5. Depicted valid formulas for the fast locked loop are also valid for the voltage regulator. As a result of formulas $w_{nvr} = 200Hz$ then transfer function of voltage regulator with real values is:

$$VR(s) = \frac{1}{1+5*10^{-3}*s+25*10^{-6}*s}$$

2.3 State Space Model

In control engineering field, state space represents mathematical model of the system. State space representation consist of set of input, output and state which are defined by first order differential equations. State space representation is combination of state equation and output equation. The internal state of the system is explicitly accounted for by an equation known as the state equation and output of the system is provided in terms of a combination of the current system state, and the current system input, through the output equation. Main idea in the state space model is state of the system which means current value of internal elements of the system and state changes separately to the output of the system. Generic view for the state space representation can be seen below:

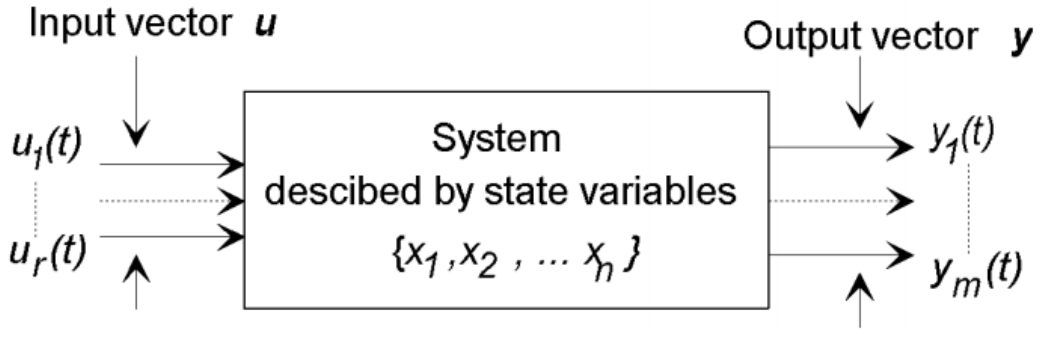

Figure 10: Generic view for the state space representation

In state space representation, in order to model system input variables, output variables and state variables should be defined in the vector form. In SISO systems there is just one

input but MIMO cases there are multiple inputs and all of them should be defined in the vector form. Again in MIMO systems there is several output values for the system and output variables should be independent of one another, and only dependent on a linear combination of the input vector and the state vector. Finally The state variables represent values from inside the system, that can change over time. Input variables are denoted with u, output variables are denoted with y and state variables are denoted with x.

State space model of the physical system is formed by two main equations. One of them represents the state of the system and the other is for the output of the system. State space equations can be seen below:

$$\dot{x}(t) = A * x(t) + B * u(t)$$
$$y(t) = C * x(t) + D * u(t)$$

where $\dot{x}(t)$ is first derivative of the state vector of the system. In the state space equations matrix A is known as system matrix and it explains relation between state change and current state. Matrix B is control matrix of the system and explains the effects of the input variables over state change. Matrix C is the output matrix for the system and shows relation between state variables and output variables. Finally matrix D is known as feed-forward matrix and it relates input variables with system output. Dimensions of these matrices are related with the dimensions of the dimensions of the input variable vector, output variable vector and state variable vector. Effects of matrices can be seen in open loop model of state space below:

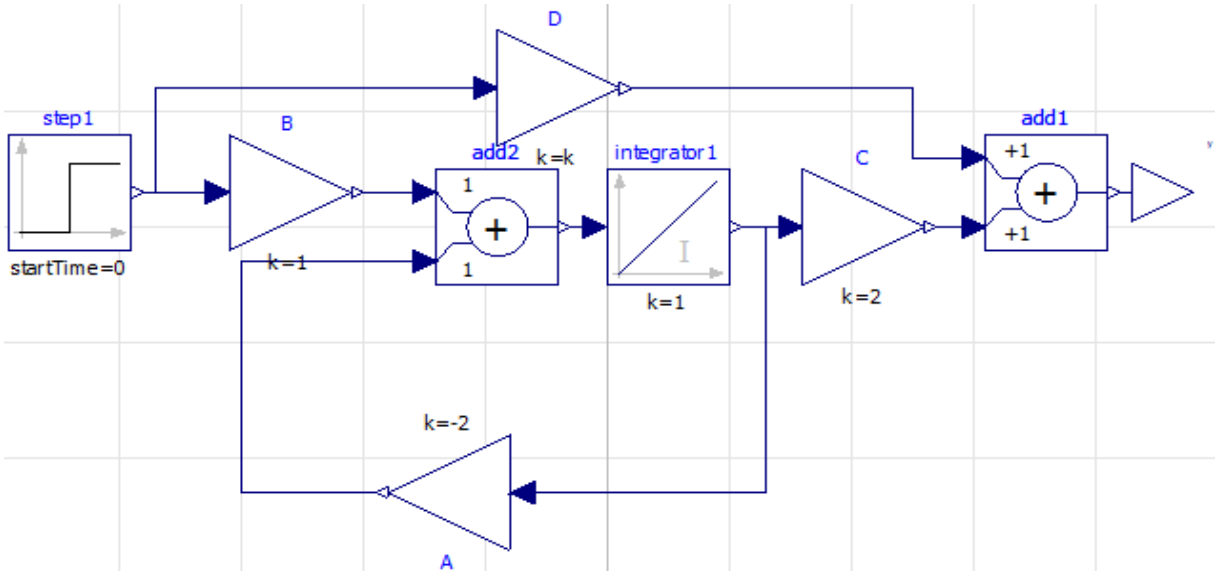

Figure 11: State Space representation in Modelica software

Using state space representation instead of other methods has several advantages and they can be listed as follows:

- With the help of state space model, it is easy to handle with MIMO systems.
- the system model includes the internal state variables, the output variable and the input variable.
- Results of the state space model is in time domain.
- The effect of initial conditions can be easily incorporated in the solution.

2.3.1 State Space Model of the Phase Locked Loop and Voltage Regulator

As mentioned in the previous chapter state space models have some advantages over other modeling concepts. Therefore for some control schemes such as LQR, transfer functions of the system components should be converted in the state space model. There are some methods to achieve conversion between state space and transfer function. State space models of the systems are not unique that means a system has many state space representations. Thus there are many different methods for converting transfer function model into state space model.

One of the most straightforward method for converting from the transfer function of a system to a state space model is to generate a model in controllable canonical form. The term of controllable canonical form is related with the control theory and property of controllable canonical form is that the resulting model is guaranteed to be controllable. In order to see how this method works, first consider the third order differential transfer function which is :

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^2 + b_1 s + b_2}{s^3 + a_1 s^2 + a_2 s + a_3}$$

In order to start calculation first multiply both sides by $X(s)/X(s)$ then solve $Y(s)$and $U(s)$ in the form of $X(s)$.

$$Y(s) = (b_0 s^2 + b_1 s + b_2) * Z(s) \qquad y = b_0 \ddot{x} + b_1 \dot{x} + b_2 x$$

$$U(s) = (s^3 + a_2 s^2 + a_1 s + a_3) * Z(s) \qquad u = \dddot{x} + a_1 \ddot{x} + a_2 \dot{x} + a_3 x$$

After this point we can select x and its two derivatives as state variables:

$$q_1 = x \quad \dot{q_1} = \dot{x} = q_2$$

$$q_2 = \dot{x} \quad \dot{q_2} = \ddot{x} = q_3$$

$$q_3 = \ddot{x} \quad \dot{q_3} = \dddot{x} = u - a_1 q_3 - a_2 q_2 - a_3 q_1$$

If we rearrange the equations:

$$y = b_o \ddot{x} + b_1 \dot{x} + b_2 x = b_0 q_3 + b_1 q_2 + b_2 q_1$$

Than state space equations become:

$$\dot{x}(t) = Ax(t) + Bu(t) = \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{vmatrix} \mathrm{x(t)} + \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix} u(t)$$

$$y(t) = Cx(t) + Du(t) = |b_2 \quad b_1 \quad b_0| \mathrm{x(t)} + 0.\mathrm{u(t)}$$

This calculation was for the consider the third order differential transfer function and its easily generalize to general n^{th} order transfer function which is:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n}$$

After applying same idea to the general n^{th} order transfer function, the controllable canonical form is:

$$A = \begin{vmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{vmatrix} \quad B = \begin{vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{vmatrix}$$

$$C = |b_n - a_n b_0 \quad b_{n-1} - a_{n-1} b_0 \quad \dots \quad b_2 - a_2 b_0 \quad b_1 - a_1 b_0| \quad D = b_0$$

Transfer function of system components are computed in the previous section. As a next step state space models of the system components should be computed. Transfer functions of the system components as follows:

$$PLL(s) = \frac{1+1*10^{-7}*s}{1+12*10^{-8}*s+42*10^{-16}*s}$$

$$VR(s) = \frac{1}{1+5*10^{-3}*s+25*10^{-6}*s}$$

In order to apply depicted formula which transforms transfer function into state space model, MATLAB has valid command. MATLAB is another commercial modeling and simulating software. MATLAB command for the conversion is [A,B,C,D] = tf2ss(b,a), it provides A,B,C,D matrixes of the state space equation as a result of running. State space equations matrices for the voltage regulator as follow:

$$A = \begin{bmatrix} -200 & -40000 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0 \quad 40000], \quad D = 0$$

Result for the phase locked loop is :

$$A = \begin{bmatrix} -0 & -2.381 * 1.0e+14 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0 \quad 2.381 * 1.0e+14], \quad D = 0$$

2.4 Control Schemes

Control Theory is an interdisciplinary subfield of the science which is a branch for the engineering. Main idea behind control theory is influencing the dynamic behavior of the system of interest. In control theory, a controller tries to make systems' output track the reference which is desired output of the system. Controller achieves this by manipulating input of the system. Control systems do four main actions which are measure, compare, compute and correct. Many examples for the controllers can be found from daily lives of individuals such as cruise controllers of automobiles, automatic doors and automatics lightings.

In control theory, two different structures for the controllers are present. First one is open loop controller which is also known as non feedback controller. Open loop controller manipulates input of the system only by considering state and the model of the system. Generic block diagram of the open loop controllers can be seen below:

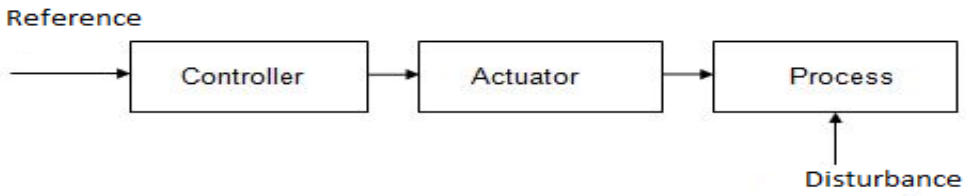

Figure 12: Block Diagram of the open loop controller

On the other hand closed loop controller uses feedback path. Therefore while providing manipulated input to the system of interest, it also considers output of the system. As a result of depicted structure closed loop controllers have advantages over open loop controllers. Advantages are listed as follows:

- Good rejection of disturbances,
- Unstable process can also be stabilized with help of feedback path and
- Reduced sensitivity to variations in the parameters.
- Obvious improvement in the performance of the controller

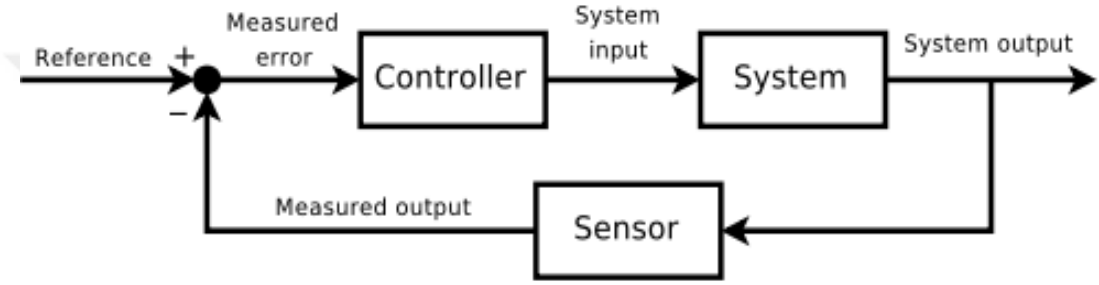

Figure 13: Block diagram of closed loop controller

In order to overcome different control problems, variety of control methods are invented such as PID, feed forward, linear quadratic regulator, model predictive control, smith predictor etc. Some of these methods are used as key tool in this work, in order to balance settling times of two components and achieve synchronous running. In order to create complete documentation, in the previous sections that methods will be expressed.

2.4.1PID controller

One of the used control schemes for the component of the system is a proportional-integral-derivative controller which known as PID controller. PID controller is a control loop feedback mechanism widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point. [12] The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

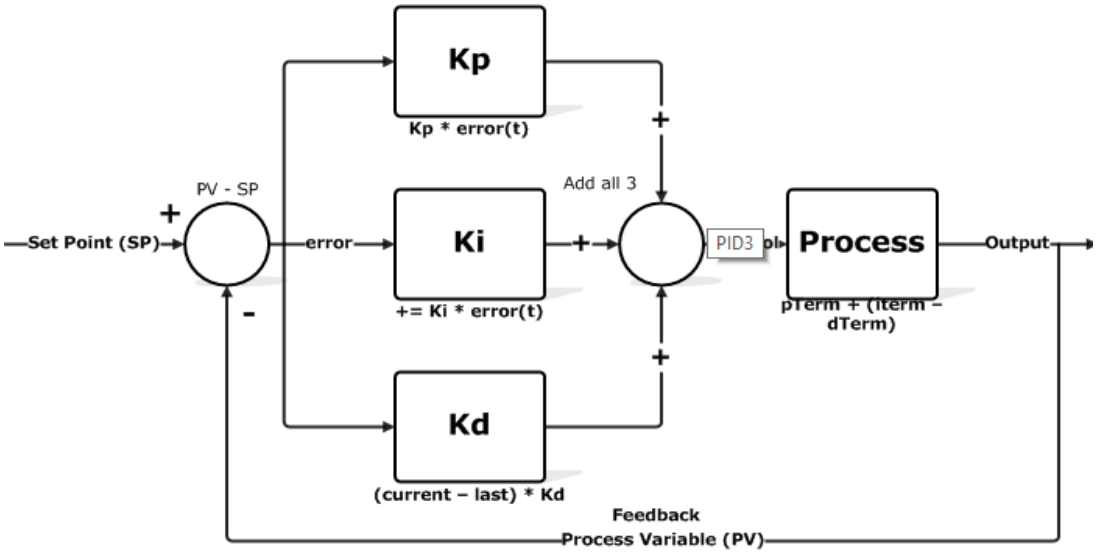

Figure 14: General Scheme for PID Controller

Basic controller equation for PID controller can be express in two different forms. First one is parallel form other one ideal form. Equation for the parallel form and ideal form as follows respectively[12]:

$$C(s) = K_p + \frac{Ki}{s} + K_d * s$$

$$C(s) = K_p(1 + \frac{1}{K_i s} + K_d s)$$

Where K_p is proportional gain, Ki is integral gain and K_d is the derivative gain. Each gain has significant facilities for the control purposes. Proportional gain provides an overall control action which is proportional to the error signal[12]. Integral gain reduces steady state error of the system output[12]. Finally derivative gain improves transient response of the whole system[12]. Effects of changing PID gains individually on system performance can be seen on below table[12]:

Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Table 2: Effects of changes on PID gains on system performance

In order to achieve useful results from controller, these gains should be tuned according the transfer function of plant. There are different methods to tune PID parameters, such as Zigler-Nichols method, Tyreus Luyben method and Cohen Coon method. In this work, parameters are tuned according to Zigler Nichols method, on MATLAB controller design tool user interface which work whit the command of "sisotool(G)". After obtaining gains of PID controller, model will be created on the Modelica modeling enviorment. Created model will be consist of transfer function and PID controller block of the Modelica software, example for the model can be seen below:

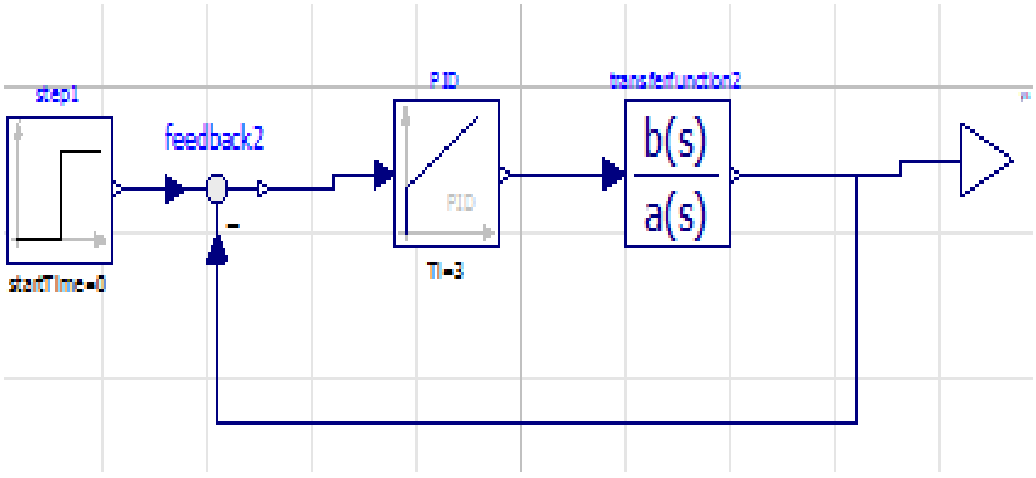

Figure 15: Model with PID controller and transfer function in Modelica software

2.4.2 Feedforward Control

An other useful controller scheme for the aim of balancing settling times is feed forward control. The disturbance is detected as it enters the process and an appropriate change

is made in the manipulated variable such that the controlled variable is held constant. Therefore in contrast to feedback control, feed forward control takes the corrective action as soon as a disturbance entering the system is detected. In other words control element responds to change in command or measured disturbance in a pre-defined way. Generic block diagram for the feed forward control scheme can be seen below:

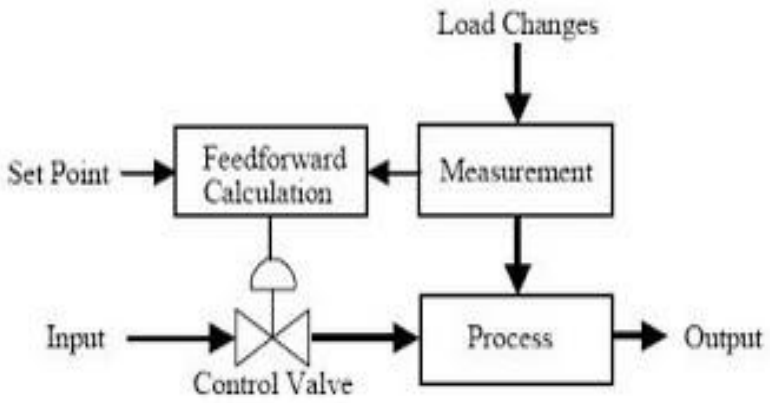

Figure 16: Example of feed forward Controller

Feed forward controller has some advantages over other control types. First of all control output of the feed forward controller does not depends on the process feedback, so feed forward controller never leads to oscillation. Also required energy for the feed forward controller is lower than other control schemes. Therefore the feed forward controller can achieve stability lower cost and lighter weight.

In some control problems, the feedback controller of a PID controller could not achieve desired controller performance. In such cases combining the feedback controller with feed forward one can increase the whole controller. Mainly in such combination of controllers, knowledge of system can be fed forwardly and combine it with the output of the PID controller in order to reach desired controller performance. Also for this work combination of the PID controller and the feed forward controller is an useful option. Created scheme for the combinatorial model in Modelica modeling environment can be seen in below figure

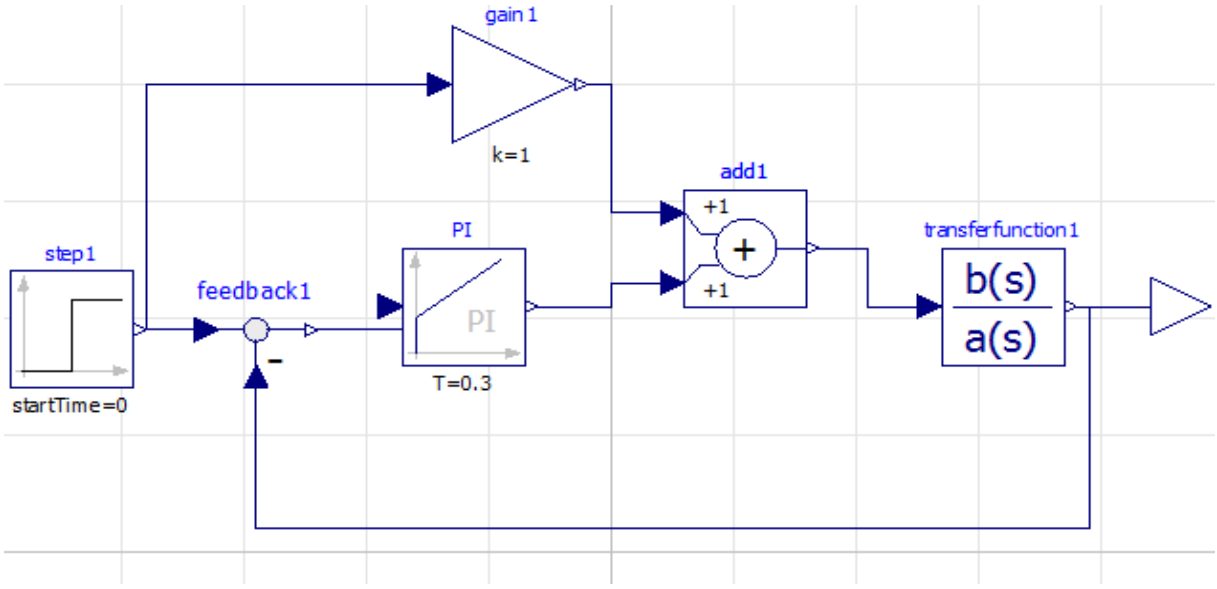

Figure 17: Model with PID +feed forward controller transfer function in Modelica software

2.4.2 Linear Quadratic Regulator

Linear quadratic regulator is a type of regulator connected with theory of optimal control. The theory of optimal control tries to achieve operation of dynamic system at minimum cost where cost function is quadratic function[15]. General scheme for Linear quadratic regulator can be seen below:

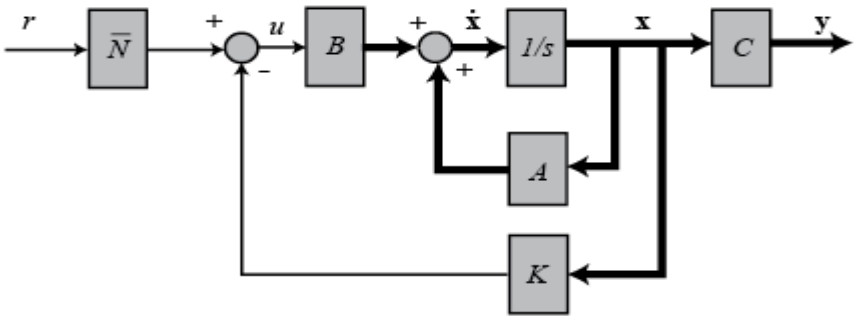

Figure 18: General Scheme For Linear Quadratic Regulator

LQR assumes linear dynamical system which is a set of linear differential equations. and quadratic cost function as follows:

$$J(x_t, y_t) = \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$

Where $Q \geq 0, R > 0$ positive-semi definite and positive-definite matrices and $u(t)$ is such to minimize the cost function. It is supposed to design the state feedback control signal to stabilize the system[14]:

$$u(t) = Kx(t)$$

The design of vector of gains K is tradeoff between the transient response quality and control effort which is managed with initialization of matrices Q, R. Increasing the magnitude of Q more would make the tracking error smaller, but would require greater control force $u(t)$.More control effort generally corresponds to greater cost. Optimal control problem consist of finding control signal to minimize cost function subject to dynamic equations of linearized system:

$$\min_u J$$

$$s.t.\ \dot{x} = Ax + Bu$$

Matrices Q, R, are chose relatively to each other to meet transient response specifications, magnitude constraints on state variables and control effort. They represent the weights of terms in minimization process, so in that sense, variables that should be slower, more damped and without overshoots should have bigger weights and therefore bigger corresponding elements in mentioned matrices. After choosing matrices, solution of the Riccati equation provides with gains that multiply state variables. In order to reach LQR gains there is a valid MATLAB command which is K=lqr(A,B,Q,R).

With the help of gain matrix K which is result of the Riccati equation transient requirements are met but there is a steady state error. In order to solve this error desired steady-state value of the states should be computed and multiply that by the chosen gain K and new value becomes reference for computing input. This can be done by adding a constant gain $\bar{N}$ after the reference.

Linear quadratic regulators have some advantages over other control schemes. Firstly stability of the system is guaranteed by LQR if there is proper model of the system and all the states in the system is known. Also the controller is automatically generated by simply selecting a couple of parameters. LQR is also straightforward to use for multivariable systems; the design procedure is essentially the same as for single-input-single-output systems On the other hand in some cases LQR has some drawbacks. For instance, obtaining an

analytical solution to the Ricatti equation is quite difficult in some cases. Beside this if the model of the system is not complete it can be really difficult get a controller which achieves desired controller performance. Although it's disadvantages, due to depicted advantages LQR is really useful for some control problems. Created general model which consist of state space model of the system and LQR in Modelica modeling environment can be seen below:

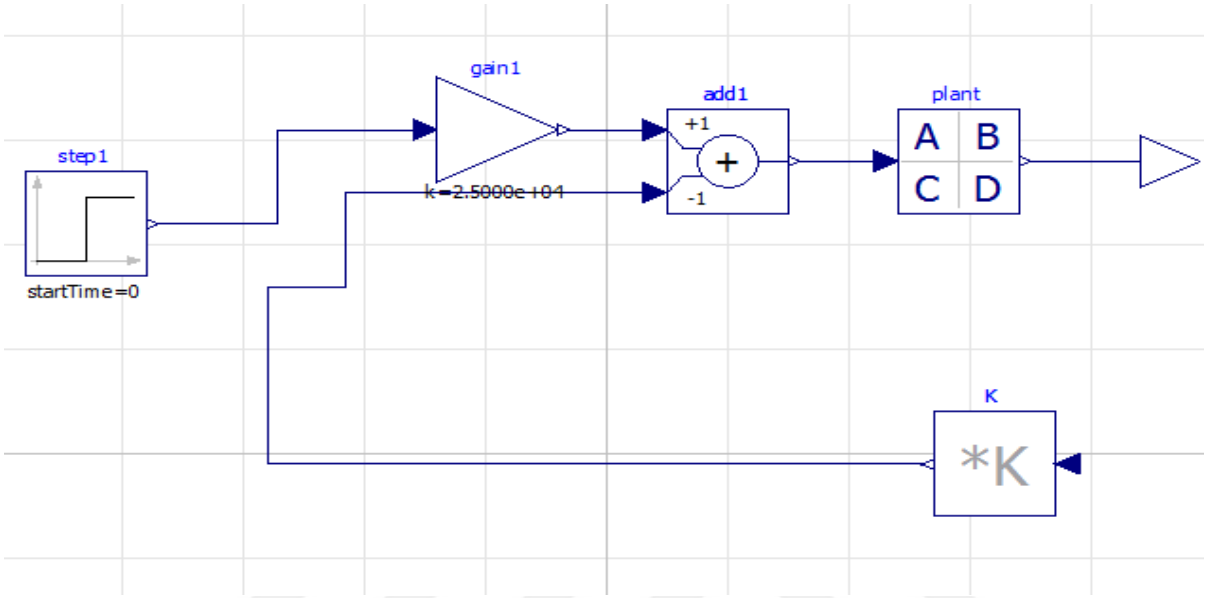

Figure 19: Model for LQR in Modelica

2.5 Conclusion

In this chapter, transfer function and state spaces models of the voltage regulator and phase locked loop are introduced. Beside this, details, advantages, drawbacks and Modelica modeling environment models of the PID controller, feed forward controller and linear quadratic regulator are presented. Therefore all blocks for the implementation of investigated systems are expressed.

Chapter 3

DESIGN AND IMPLEMENTATION OF CONTROL SCHEMES

3.1 Introduction

In this chapter, firstly open loop models of the voltage regulator and phase locked loop will be provided. Than different control schemes will be designed according to the mathematical models of the systems. After design step control schemes will be applied to the investigated system and results of the control schemes will be investigated.

3.2 Open Loop Models of the Components

In this section of the chapter open loop models of the voltage regulator and phase locked loop will be investigated. Aim of this investigation is observe difference of the dynamics of two component and also it is possible to check correctness of transfer functions which are calculated in chapter two from the open loop results. Therefore results of the open loop models will provide evidence for the main problem.

In order to model transfer function of the voltage regulator, numerator and denumerator of the transfer function block of the Modelica software filled with the obtained values in the chapter two. Model for the open loop configuration of the voltage regulator is present below figure:

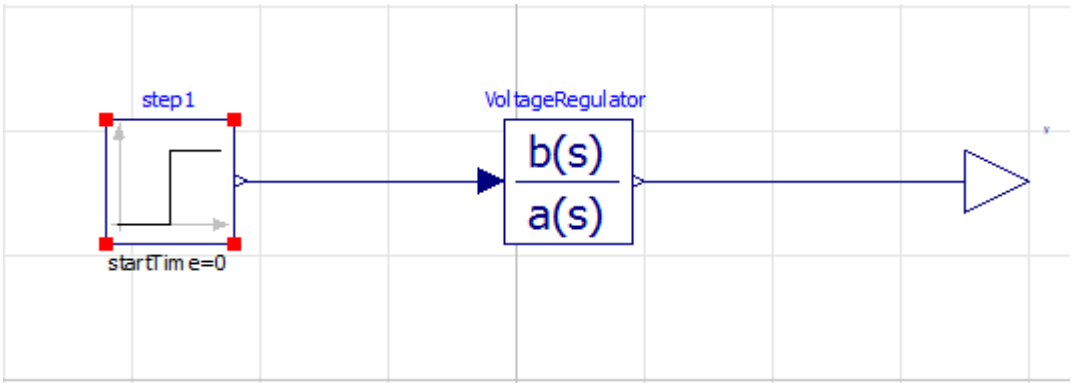

Figure 20: Open Loop Modelica Model Of the Voltage Regulator

In the figure it can be seen that unity step input is applied to the transfer function in order to observe step response of the voltage regulator than find the settling time of the system component. Step response of the voltage regulator as follow:

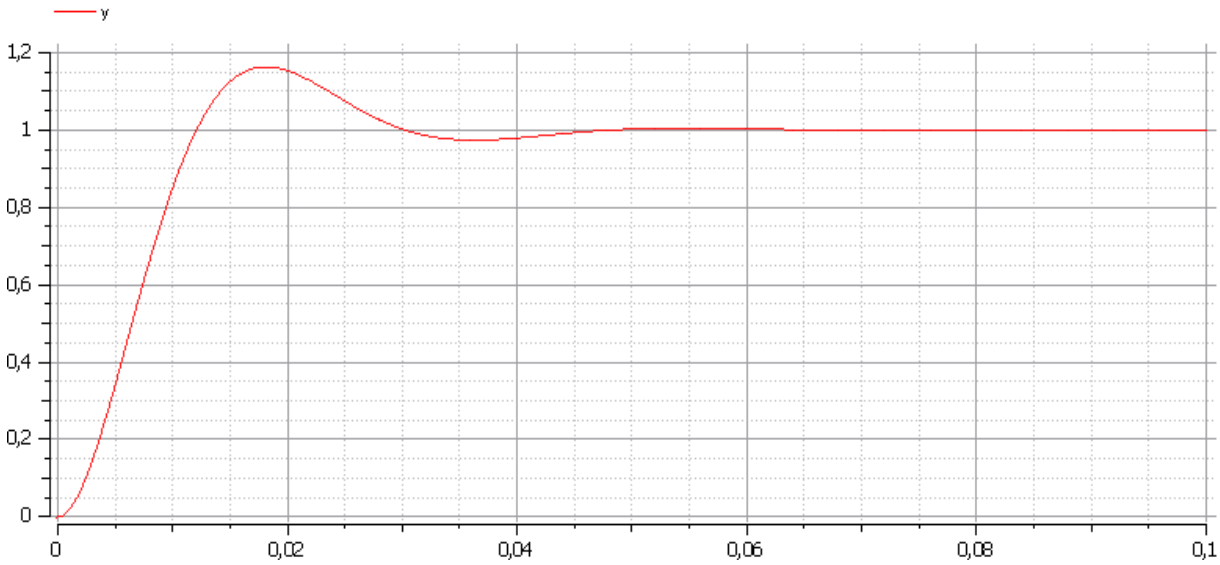

Figure 21: Open Loop Response of Voltage Regulator

In Figure, it is obvious that settling time and oscillatory curve is similar to desired voltage regulator. Therefore obtained values of the transfer functions from depicted formulas in chapter two are acceptable.

As next step, open loop model of the phase locked loop is created. In Modelica modeling enviorment, calculated values of the transfer function for phase locked loop is entered to the numerator and denumerator of the transfer function block. As an input unitary step is applied to the system in order to observe step response of the phase locked loop, open loop model of the PLL can be seen in below Figure:

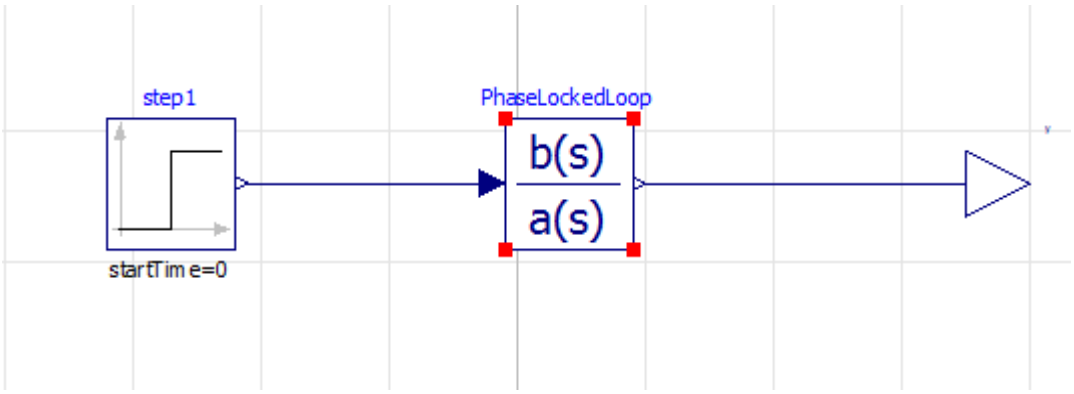

Figure 22: Open Loop Modelica Model Of the PLL

In order to observe settling time and dynamic performance of the phase locked loop, step response is required. As a result of unity step input, step response of the PLL as following figure below:

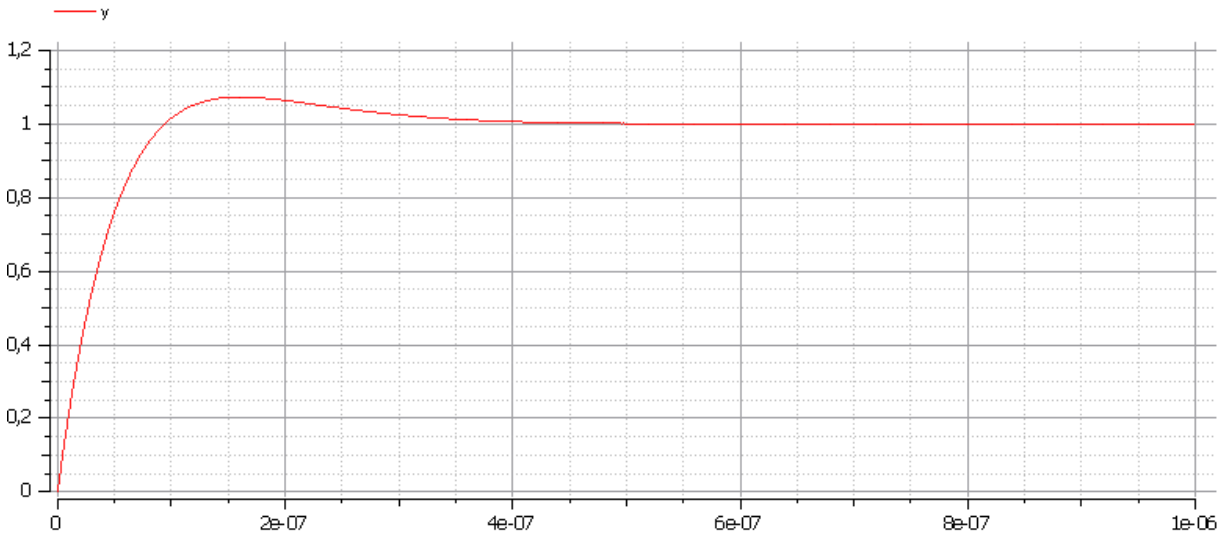

Figure 23: Open Loop Response of PLL

From the step response of the phase locked loop, it is obvious that settling time and oscillatory curve is similar to the depicted on the paper of Habib Adrang and Hossein Miar Naeimi which named as " A type III Fast Locking Time PLL with Transconductor-C Structure. Therefore, obtained transfer function model for the phase locked loop is matching with the component of the system of interest.

In order to control correctness of the transformation from transfer function model to the state space model which is done in the chapter two, open loop configuration should be applied also to the state space models. Firstly obtained values for the A, B,C and D matrices of the voltage regulator is entered to the state space block in the Modelica modeling

environment and unity step input is applied as an input to the system. Created model can be seen below:

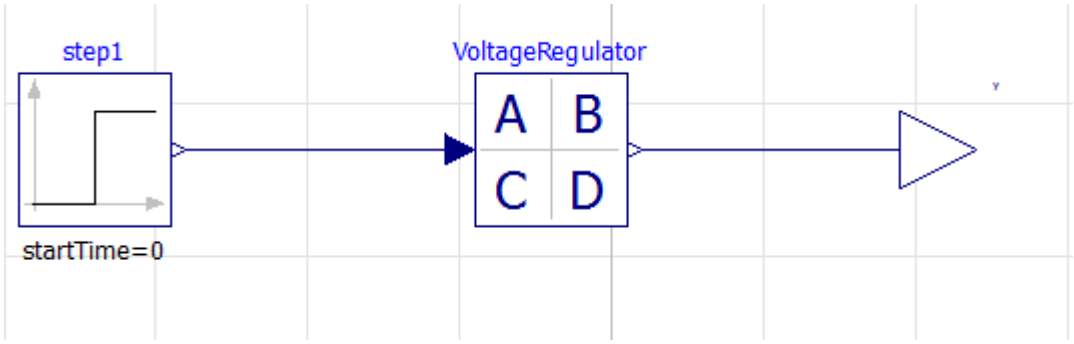

Figure 24: Open Loop State Space Modelica Model Of the Voltage Regulator

Step responses of the transfer function model and state space model should be compared, in order to check correctness of the transformation from transfer function model to the state space model. Step response of the state space model of voltage regulator as following figure:

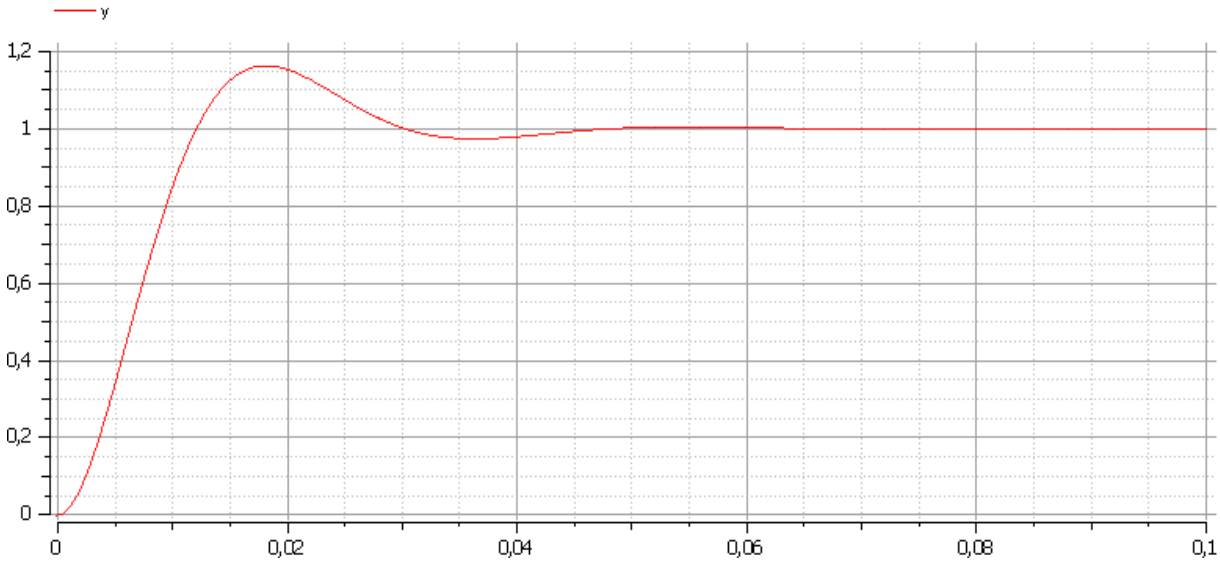

Figure 25: Open Loop Response State Space model of Voltage Regulator

From figure 25, it is obvious that resulting step response from state space model and step response from transfer function model are exactly same. Therefore transformation between them is correct.

Secondly correctness of the transformation for phase locked loop should be checked. Obtained values for the A, B,C and D matrices of the phase locked loop that are calculated in the chapter two is entered to the state space block in the Modelica software and unity step

input is applied as an input to the system. Open loop of PLL with state space model and resulting step response can seen be below respectively:

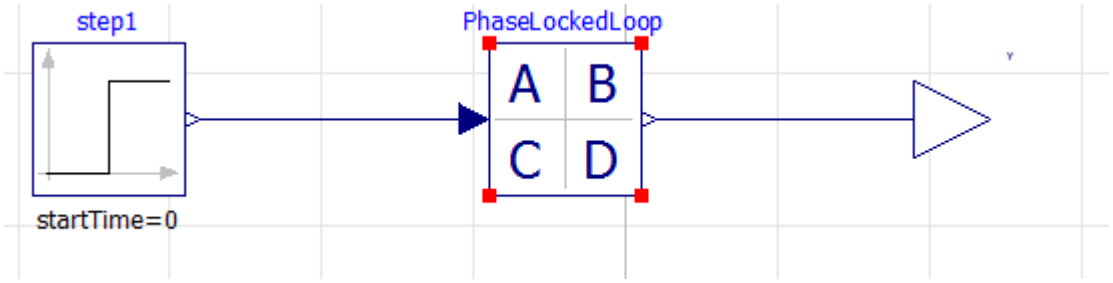

Figure 26: Open Loop State Space Modelica Model Of the Phase Locked Loop

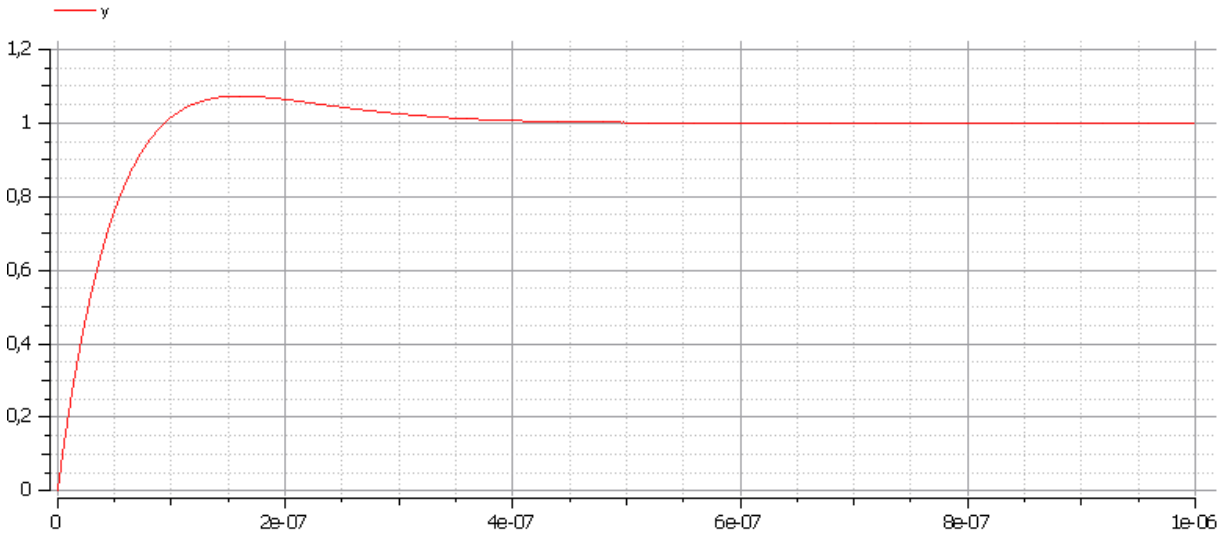

Figure 27: Open Loop Response of PLL with State Space Model

Also for this system component, by comparing figures it is obvious that resulting step response from state space model and step response from transfer function model are exactly same. Therefore used transformation method is valid.

An other observation from the step response figures of the voltage regulator and phase locked loop, as expected, there is a big gap between settling times of the system components. Due to slower dynamics of the voltage regulator, it's settling time is around 30ms. On the other hand settling time of the phase locked loop is around 400ns. Mentioned differences between settling times leads to wasting power and time problem in the system of interest. Balancing settling times of the components and eliminate depicted mismatch is the main problem of the this thesis work. Therefore step responses of the system components are proof for the existence of the problem. In order to observe the problem, both of the system components can be settled down in the same configuration than settling times of both

components can be observed. Therefore required open loop configuration which consist of transfer function model of the system components can be seen in the following figure:

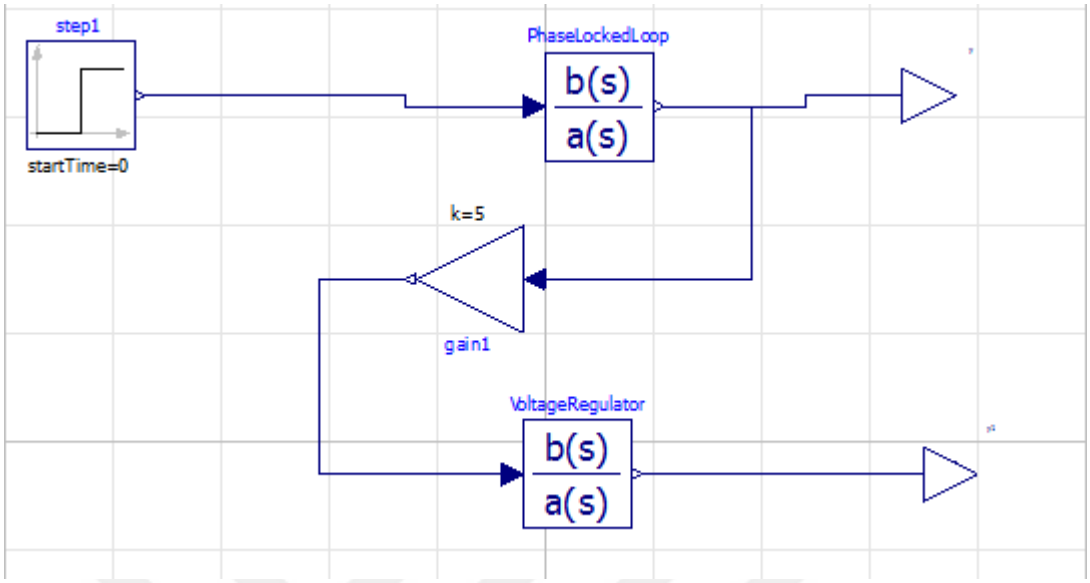

Figure 28: Open Loop Configuration with both of the system components

Step response of the open loop configuration can be seen below. And on the figure depicted problem is obvious.

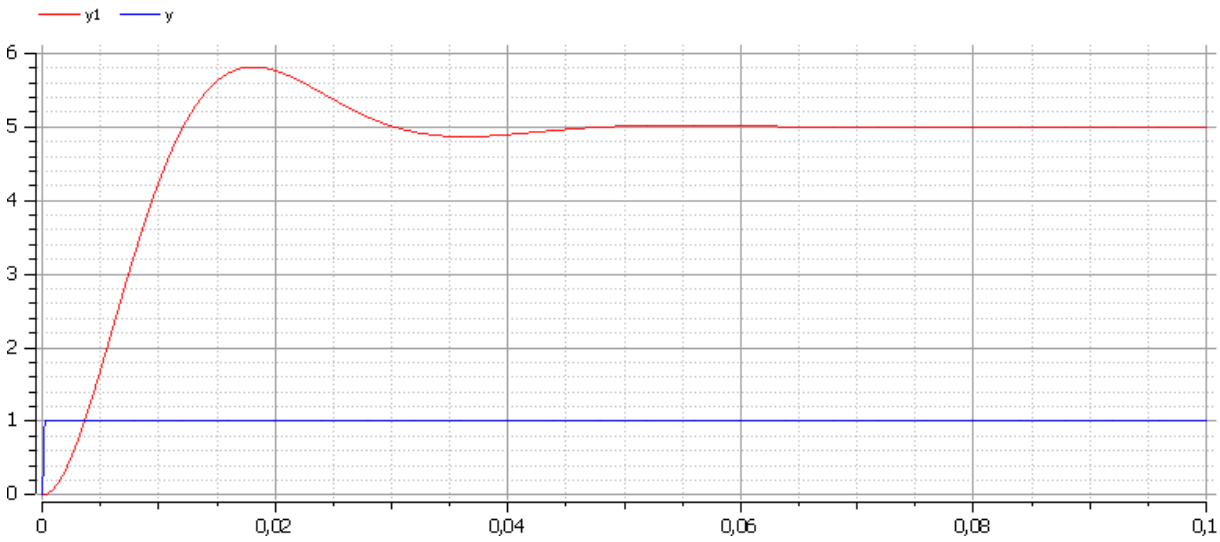

Figure 29: Step response of the Open Loop Configuration with both of the system components

3.3 Design and Implementation of the PID Controller

Proportional-integral-derivative controller which known as PID controller is one of the most widely used controller mechanism in the today's industry. Also PID controller is a

significant tool in order to satisfy problem of this work. PID controller finds the error value between desired set point and measure process output. Than in order to achieve good tracking of set point, PID controller tries to minimize the error value. Structure of the PID controller in Modelica modeling environment can be seen below figure:

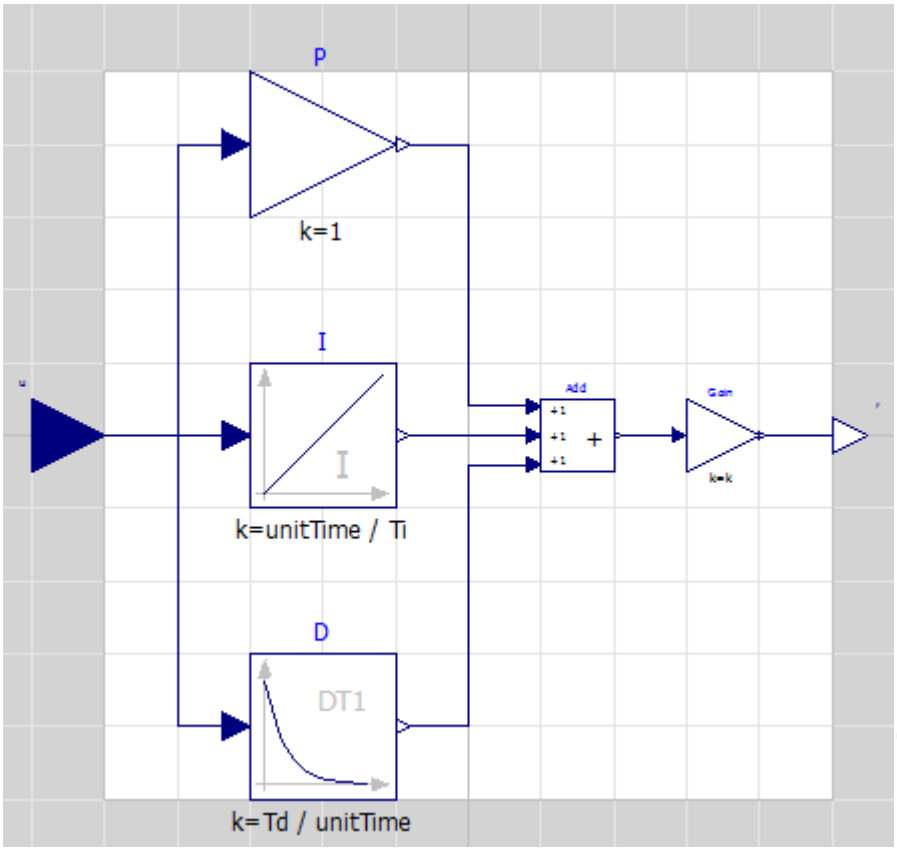

Figure 30: Internal Structure of the PID controller

As a result of internal structure of the PID controller, basic valid controller equation is:

$$C(s) = K_p(1 + \frac{1}{K_i s} + K_d s)$$

As depicted in the chapter two performance of the controller depends on the consistency of the PID controller gains with the system components. In order to achieve high consistency level, controller gains should be tune according to transfer functions the system components. . There are different methods to tune PID parameters, such as Zigler-Nichols method, Tyreus Luyben method and Cohen Coon method.

The Ziegler-Nichols' closed-loop tuning method which is developed by John G. Ziegler and Nathaniel B. Nichols is a fairly accurate heuristic method of tuning a PID

controller. The Ziegler Nichols method determine gains which provides good performance for the PID controllers for a wide range of common industrial processes[16].

The Ziegler-Nichols' closed loop method depends on the experiments which are done on the control system of interest, which can be seen below figure[18]:

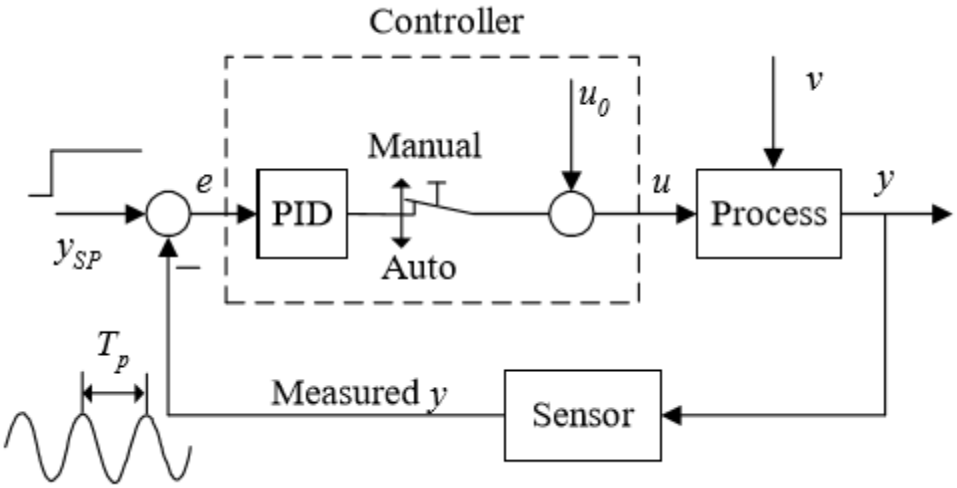

Figure 31: Working principle of the Ziegler-Nichols method

Working principles of the Ziegler-Nichols' closed loop method can be listed as follows[18]:

- As a first step, system process should be in the specified operating point of the control system or as close to as possible.
- Turn the controller to Proportional only mode. Turn both the integral and derivative modes off by setting gains as $K_i = \infty$ and $K_d = 0$. And as a first controller set up, select K_p equals to zero.
- From $K_p = 0$, increase K_p until existence of sustained oscillations in the signals in the control system. This K_p value is the ultimate gain of the controller system which is also known as critical gain. And ultimate gain is denoted by K_{pu}.
- As a next step, ultimate period which is denoted by P_u should be measured from the sustained oscillations.
- Finally controller gains should calculated according to the table below:

	K_p	T_i	T_d
P controller	$0.5K_{p_u}$	∞	0
PI controller	$0.45K_{p_u}$	$\frac{P_u}{1.2}$	0
PID controller	$0.6K_{p_u}$	$\frac{P_u}{2}$	$\frac{P_u}{8} = \frac{T_i}{4}$

Table 3: Formulas for the controller gains in the Ziegler Nichols method

During PID controller implementation parts of this thesis work, Ziegler Nichols method is selected as main tuning method. Depicted calculations for the controller gains are done by the MATLAB control and estimation tool manager which works with the MATLAB command, sisotool(G). Valid user interference can be seen in the following figure

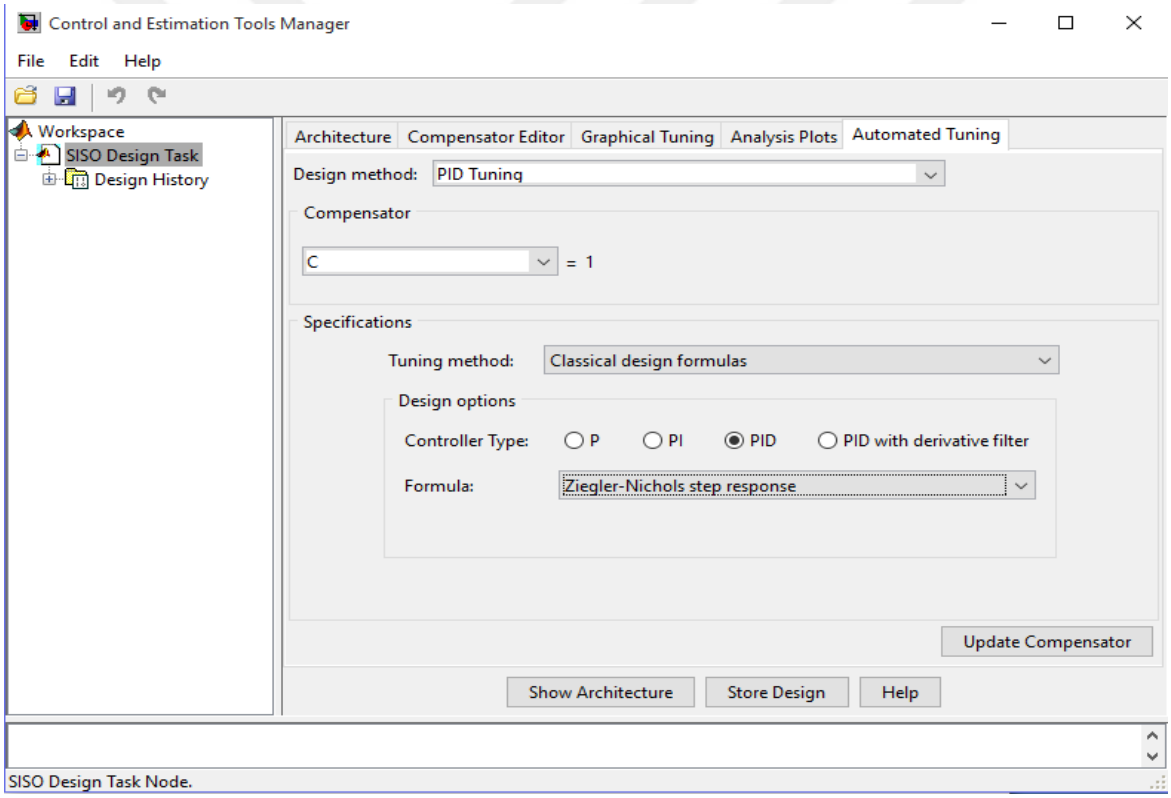

Figure 32: MATLAB control and estimation tool manager

After calculating the proportional gain, integral gain and derivative gain according to the transfer function of the system components PID controllers applied to the system of interest. In order to balance settling times of the voltage regulator and phase locked loop than save the system from wasting power and time.

3.3.1 PID Implemented to Voltage Regulator No Controller for PLL

In the first implementation of the PID controller, PID controller is implemented to the voltage regulator and phase locked loop is leaved as open loop in the configured system. In order to implement PID controller, calculated gain values for proportional gain, integral gain and derivative gain according to the transfer function of voltage regulator is entered to the PID block of the Modelica software. Created model and step response of the model are as follows:

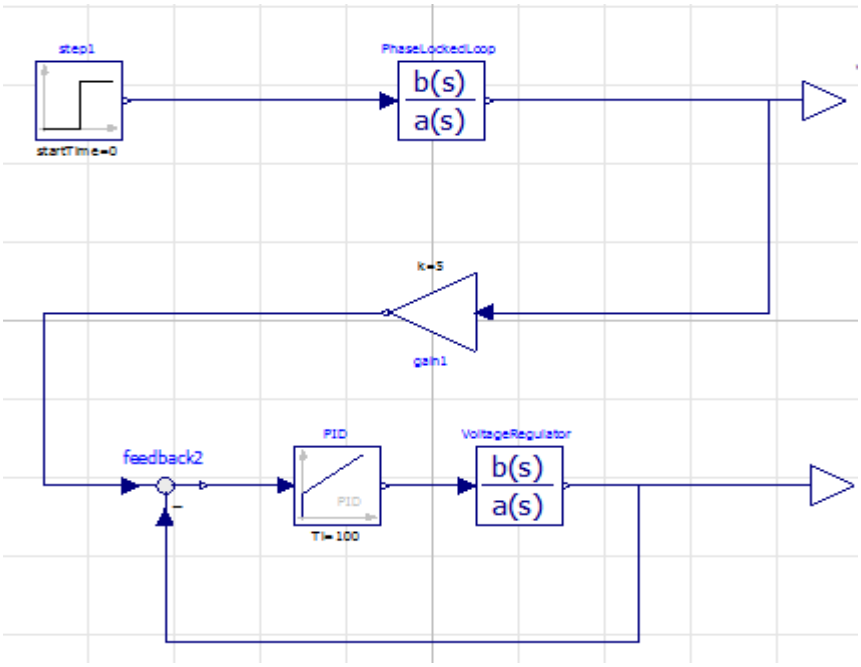

Figure 33: Voltage regulator with PID controller PLL no controller

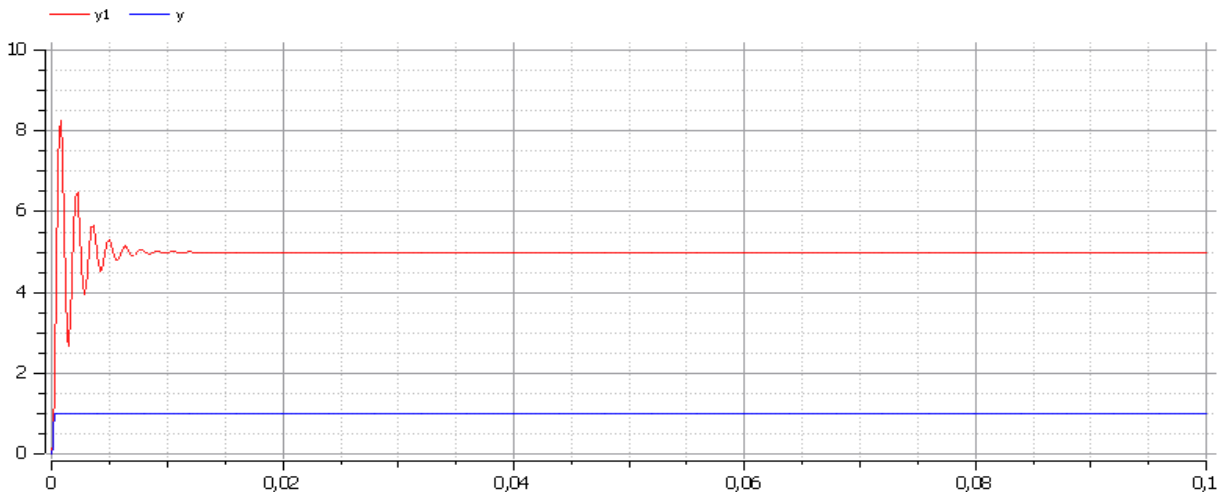

Figure 34: Step Response Voltage regulator with PID controller PLL no controller

As expected, presence of the PID controller effect the settling time of the voltage regulator. By attempting to minimize the error value between a measured process variable and a desired set point, PID controller achieves to decrease settling time of the voltage regulator. In the open loop configuration, settling time of the voltage regulator was 30ms, with the feedback effect of the PID controller it is around 10ms. Gap between settling times of the system components has decreased due to presence of the PID controller.

3.3.2 PID Implemented to Both Voltage Regulator and PLL

As a second setup, PID controller implemented to the both of the voltage regulator and phase locked loop. PID controller block of Modelica modeling environment is filled with obtained values of the PID controller gains for the system components. Configured system and change in the step response of the phase locked loop are as follows:

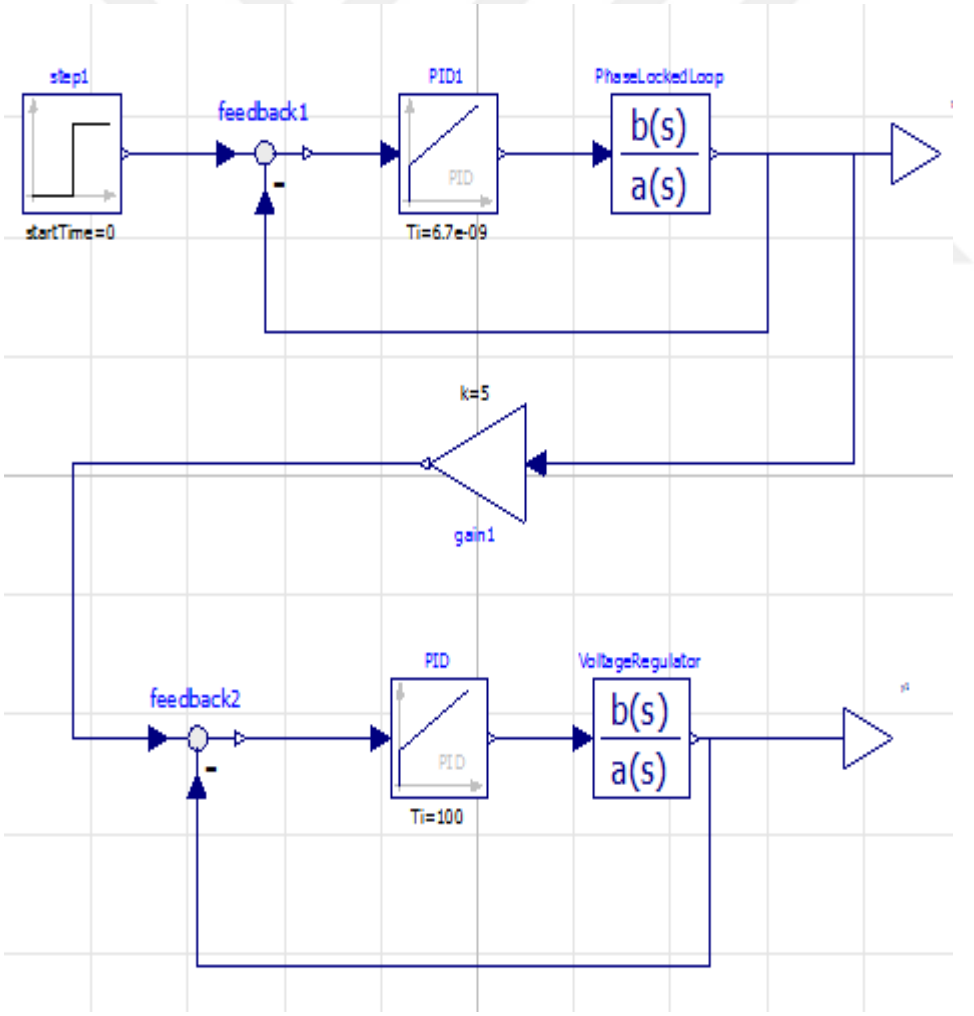

Figure 35: Voltage regulator and phase locked loop with PID controller

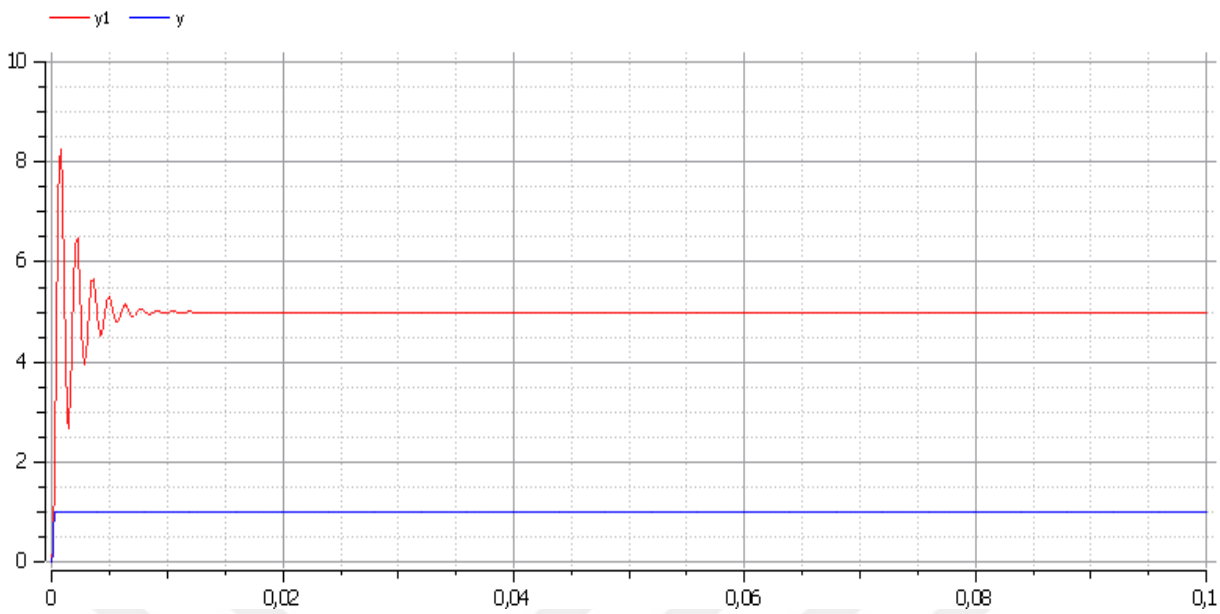

Figure 36: Step response Voltage regulator and phase locked loop with PID controller

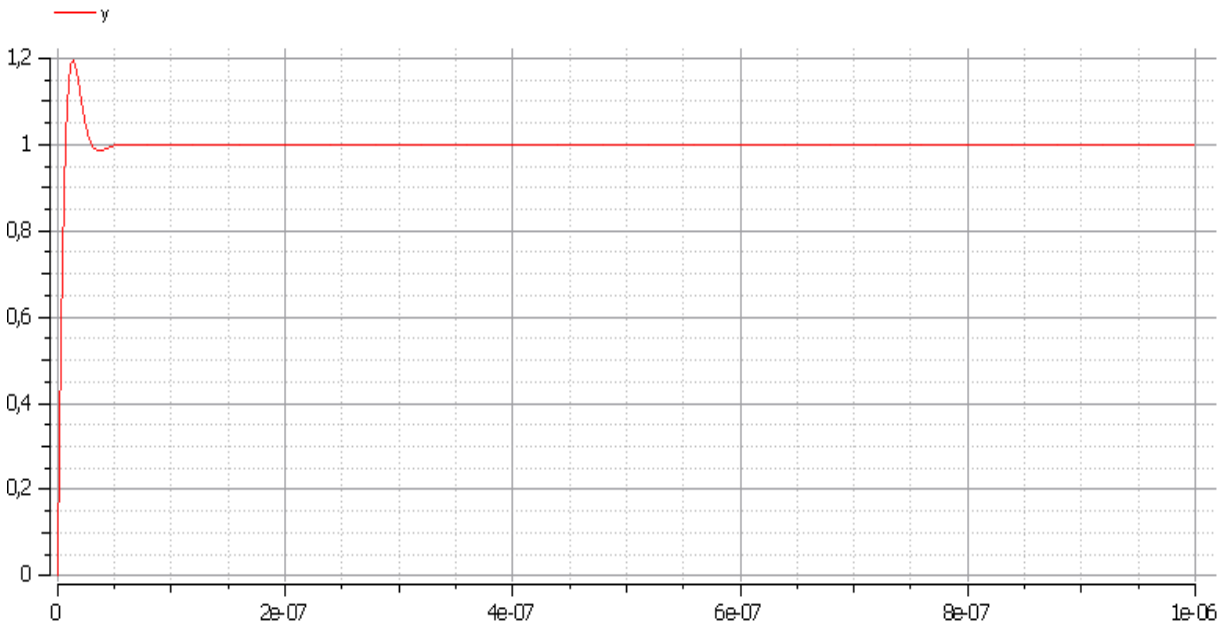

Figure 37: Step response of phase locked loop with PID controller

From Figure 37, it is obvious that settling time of the phase locked loop is decreased. Also in this case, PID controller does its job which is maintaining the output at a level so that there is no difference between the process variable and the set point. Depicted working principle of the PID controller leads to this decrease. But it is obvious that still there is a big gap between settling times that leads to power and energy loss for the system of interest.

3.4 Design and Implementation of the LQR

Linear quadratic regulators are based on the optimal control theory. As depicted in the chapter 2, the theory of optimal control tries to achieve operation of dynamic system at minimum cost where cost function is quadratic function. In the linear quadratic regulator design process quadratic cost functions is assumed as follows:

$$J(x_t, y_t) = \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$

In the cost function, $Q \geq 0, R > 0$ matrices are positive-semi definite and positive-definite matrices respectively and $u(t)$ is selected in order to minimize the cost function. It is supposed to design the state feedback control signal to stabilize the system:

$$u(t) = Kx(t)$$

Optimal control problem consist of finding control signal to minimize cost function subject to dynamic equations of linearized system:

$$\min_u J$$

$$s.t.\ \dot{x} = Ax + Bu$$

Before applying linear quadratic regulator to the system of interest, controllability of the system should be verified. For the system to be completely state controllable, the controllability matrix must have rank n where the rank of a matrix is the number of independent rows (or columns). The controllability matrix of the system takes the form shown below.

$$C = [A \vdots AB \vdots A^2B \vdots \cdots \vdots A^{n-1}B]$$

In order to generate controllability matrix of the system, there is a valid MATLAB command which is ctrb. Also there is a MATLAB command which is rank(sys) in order to find the rank of the matrix. Used MATLAB command lines to check controllability of the system of interest can be seen below:

```
 sys_ss =
ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',
outputs);
```

```
co = ctrb(sys_ss);
controllability = rank(co)
```

The next step in the design process is to find the vector of state-feedback control gains $\mathbf{K}$ assuming that we have access all four of the state variables. This can be accomplished in a number of ways. For this work, used option is to use the lqr command of the MATLAB which returns the optimal controller gain assuming a linear plant, quadratic cost function, and reference equal to zero. Used MATLAB command line can be seen below:

```
K = lqr(A,B,Q,R)
```

While designing vector of gains K, matrices Q and R play significant role. Matrices Q , R, are chose relatively to each other to meet transient response specifications, magnitude constraints on state variables and control effort. They represent the weights of terms in minimization process, so in that sense, variables that should be slower, more damped and without overshoots should have bigger weights and therefore bigger corresponding elements in mentioned matrices. Main problem of the system of interest is difference between settling time of the components. Therefore during the work K matrices should be designed in order to decrease depicted gap.

The final step in the design process is adding pre compensation to the controller. Presence of gain matrix K which is result of the depicted MATLAB command, designed controller meets transient requirements, on the other hand this controller configuration led to a steady state error. Therefore desired steady-state value of the states should be computed and multiply that by the chosen gain K and new value becomes reference for computing input. This can be done by adding a constant gain $\bar{N}$ after the reference. Valid MATLAB command lines, in order to generate pre compensation gain can be seen below:

```
s = size(A,1);
          Z = [zeros([1,s]) 1];
          N = inv([A,B;C,D])*Z';
          Nx = N(1:s);
          Nu = N(1+s);
          Nbar=Nu + K*Nx
```

3.4.1 LQR Implemented to Voltage Regulator No Controller for PLL

In the first implementation of the linear quadratic regulator, LQR is implemented to the voltage regulator and phase locked loop is leaved as open loop in the configured system.

Linear quadratic regulator is designed according to the previous section with the help of the MATLAB software. In order to implement linear quadratic regulator, mathematical model of the voltage regulator is demonstrated by the state space model of the Modelica software, generated gain matrix K is applied to the system by matrix gain block of the Modelica modeling environment and calculated pre compensation value is entered to the gain block of the Modelica software. Input of the gain matrix K is defined by the one line of the Modelica code which can be seen below:

```
K.u = plant.x;
```

Created model and step response of the model are as follows:

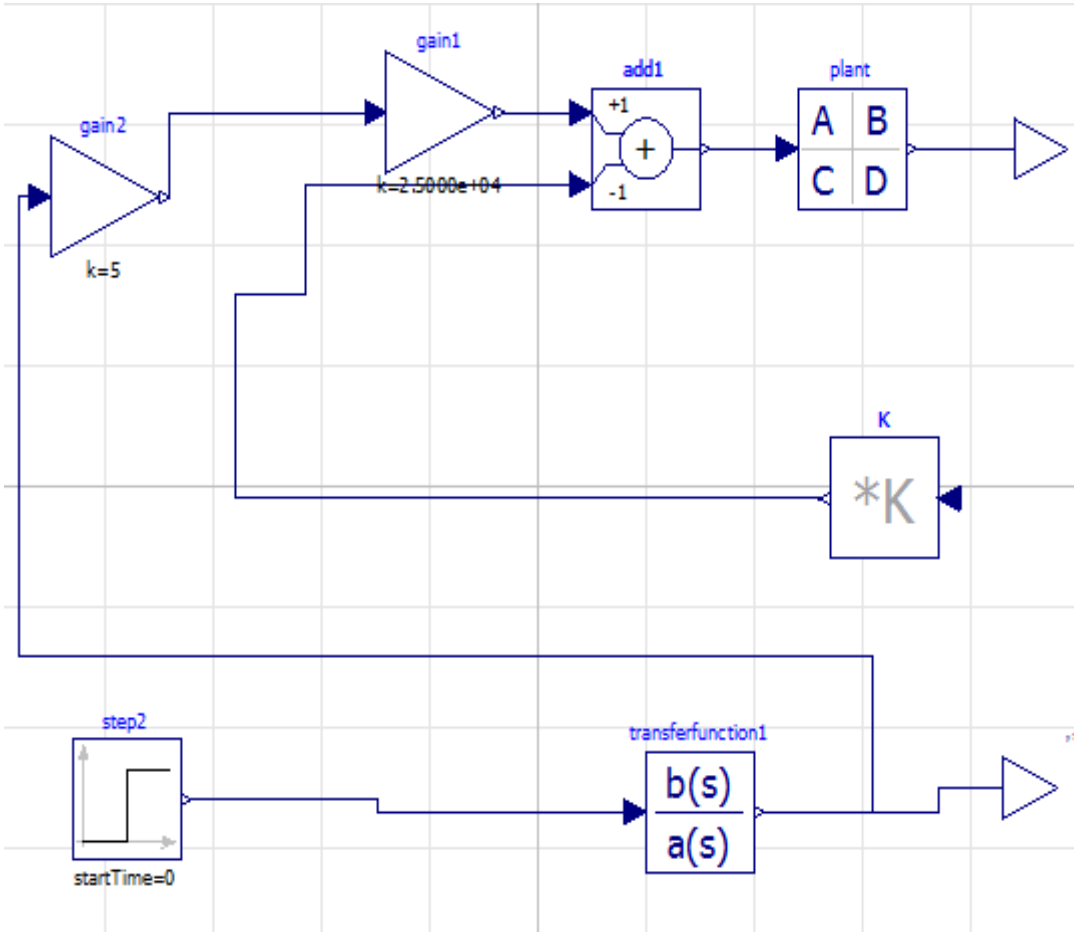

Figure 38: Voltage regulator with LQR and PLL is in open loop form

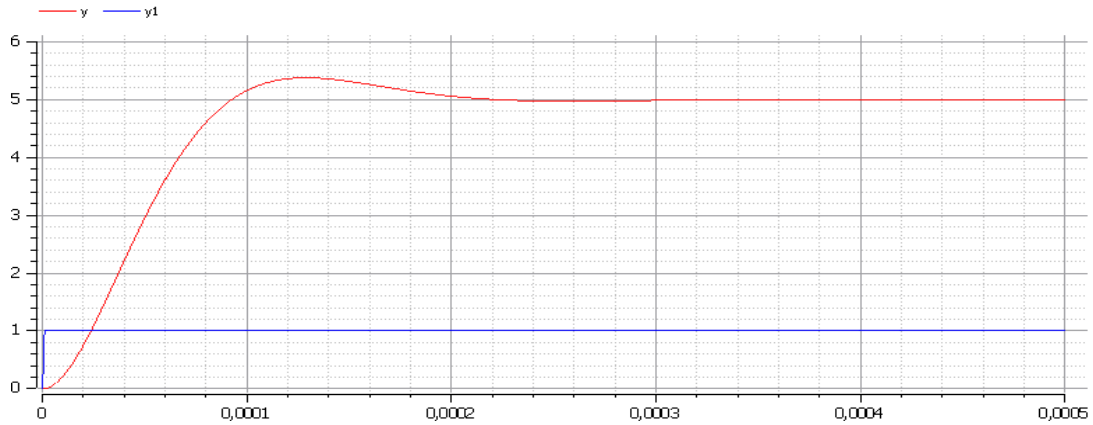

Figure 39: Step response of the voltage regulator with LQR and PLL is in open loop form

From Figure 39, it is obvious that presence of the linear quadratic regulator effects the settling time of the voltage regulator. By trying to achieve operation of dynamic system at minimum cost where cost function is quadratic function, linear quadratic regulator led to depicted decrease in the settling time of the voltage regulator. In this configuration settling time of the voltage regulator is around 0.0002s. Gap between settling times of the system components has decreased due to presence of the LQR

3.4.2 LQR + PID Implemented to Voltage Regulator and No Controller for PLL

As a second setup, linear quadratic regulator and PID controller implemented to the voltage regulator and phase locked loop is leaved as open loop in the designed control system. Matrix gain K and pre compensation gain of the linear quadratic regulator are generated in the MATLAB environment. Mathematical model of the voltage regulator is implemented with the state space block of the Modelica modeling environment and calculated gains are implemented by the gain blocks of the Modelica software. In order to complete implementation of the linear quadratic regulator, input of the gain matrix K is expressed by the code line which is as follows:

```
K.u = plant.x;
```

In this configuration, additional PID controller is added to the voltage regulator. In order to implement PID controller, calculated gain values for proportional gain, integral gain and derivative gain according voltage regulator is entered to the PID block of the Modelica programming language. Configured system and the step response of the system of interest are as follows:

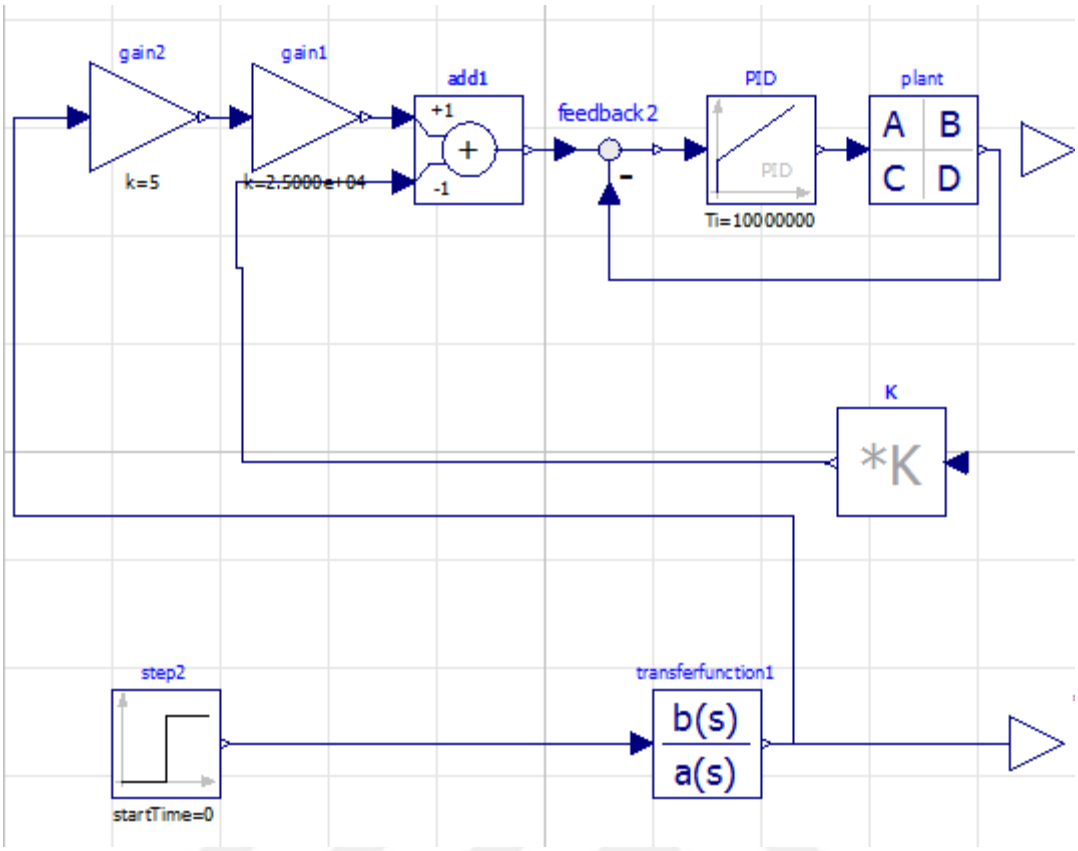

Figure 40: Voltage regulator with LQR+PID and PLL is in open loop form

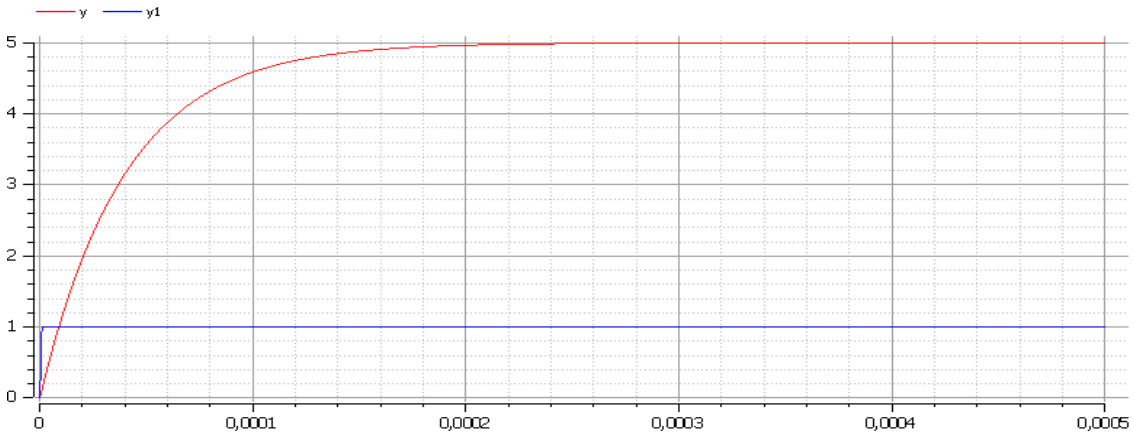

Figure 41: Step response of voltage regulator with LQR+PID and PLL is in open loop form

From step response of the configuration, it is obvious that presence of the PID controller and linear quadratic regulator effects the settling time of the voltage regulator. Linear quadratic regulator, by trying to minimize depicted cost function, decrease settling time of the voltage regulator. Also By attempting to minimize the error value between a measured process variable and a desired set point, PID controller tries to decrease settling time of the voltage regulator. In this configuration settling time of the voltage regulator is

around 0.0002s and also it is obvious that presence of the PID controller eliminate overshot which was present in the Figure111.

3.4.3 LQR Implemented to Voltage Regulator PI Controller For PLL

As a third configuration with linear quadratic regulator, LQR is implemented to the voltage regulator and PID controller implemented to the phase locked loop. As depicted in the previous subsections; gain matrix K and pre compensation gain for linear quadratic regulator are defined in the MATLAB environment. Also gain values for the PID controller are generated with the help of MATLAB software. System of interest configured with state space transfer function, PID controller and gain blocks of the Modelica modeling environment. Designed system and step response of the system of interest can be seen below:

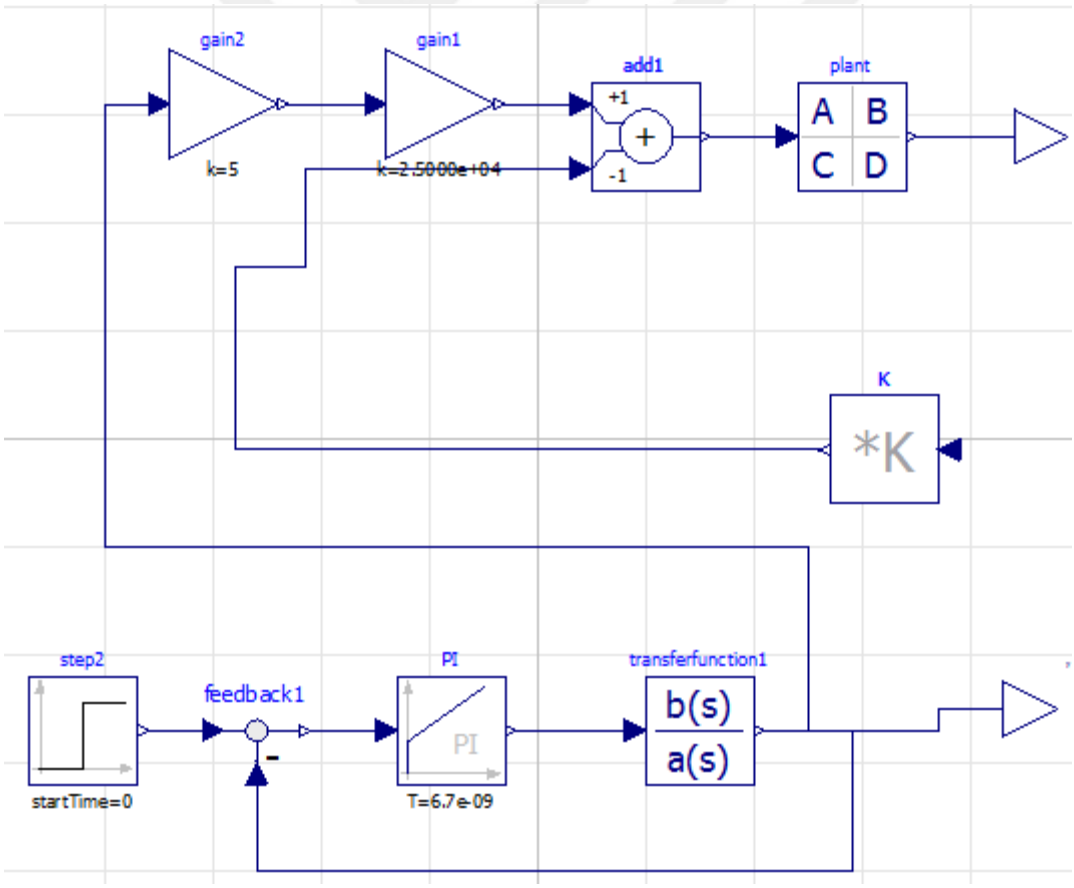

Figure 42: Voltage regulator with LQR and PLL with PI controller

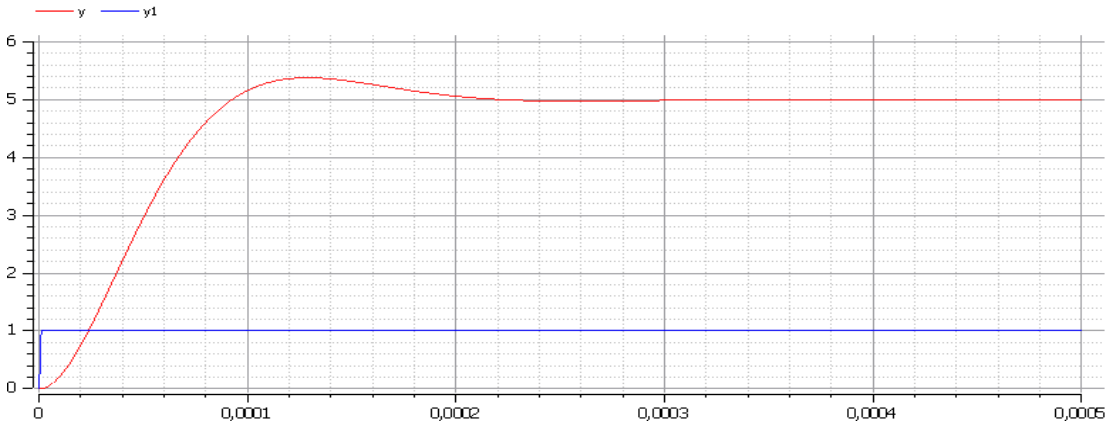

Figure 43: Step response of Voltage regulator with LQR+PID and PLL with PI controller

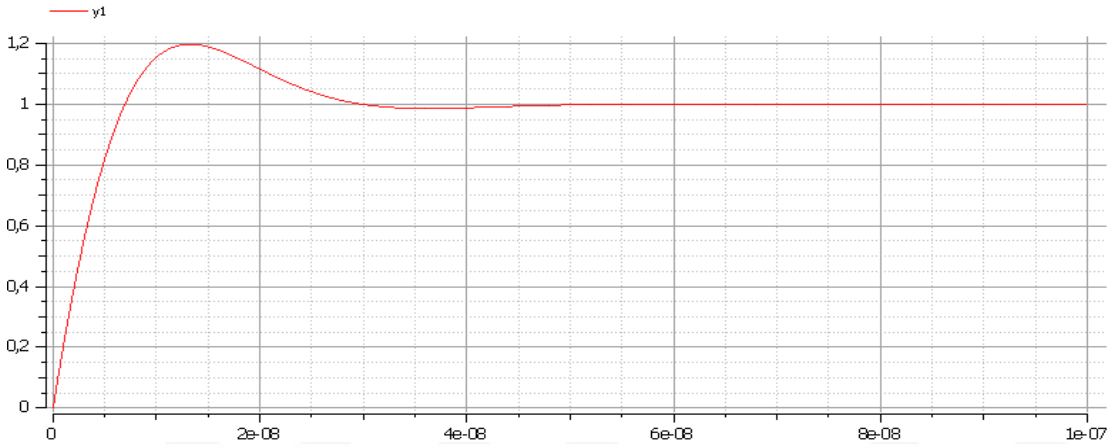

Figure 44: Focused Step response of PLL with PI controller

As a result of step response of the designed model, Linear quadratic regulator decreases the settling time of the voltage regulator. In open loop configuration settling time of the voltage regulator was around 30ms and with the help of designed controllers it decreased around 0.0002s. PI controller does its job which is maintaining the output at a level so that there is no difference between the process variable and the set point so it decrease the settling time of the phase locked loop. Gap between settling times of the system components has decreased due to presence of the LQR. Difference between settling times which is source of power and time loss is decreased due to designed controllers.

3.4.4 LQR + PID Implemented to Voltage Regulator PI Controller for PLL

As a final configuration with linear quadratic regulator, additional PID controller is added to the LQR controller of the voltage regulator. PI controller implemented to the phase locked loop. Mathematical model of the voltage regulator is expressed by the state space

block and mathematical model of the PLL is expressed by the transfer function block of the Modelica modeling environment. PID controller blocks of Modelica software which are applied to the both of the components, are filled with obtained values of the PID controller gains for the system components. Also in order to implement LQR, required gain matrix and pre compensation gain are calculated and entered to the gain blocks of Modelica modeling environment. Configured system and change in the step response created model are as follows:

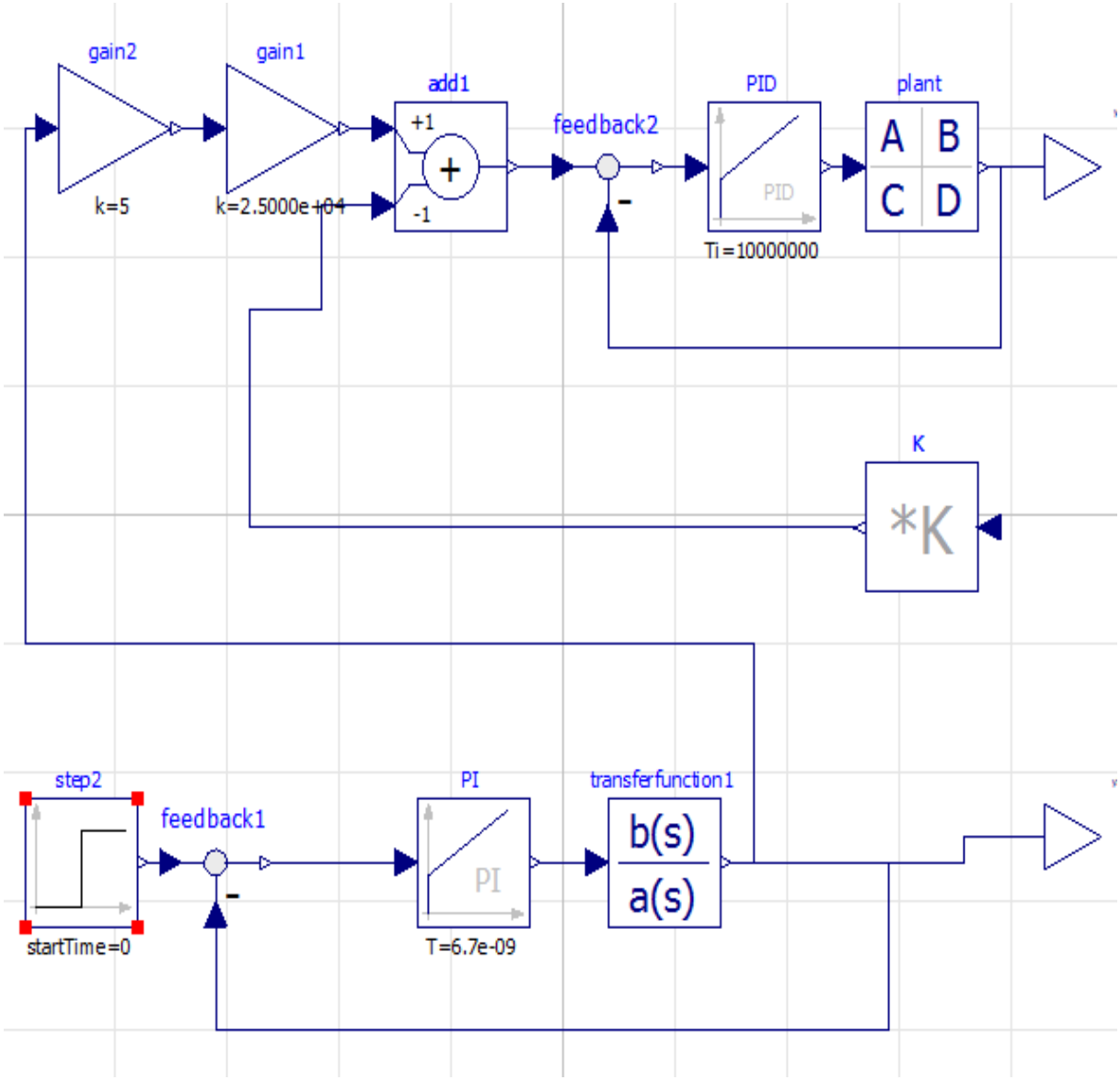

Figure 45: Voltage regulator with LQR+PID and PLL with PI controller

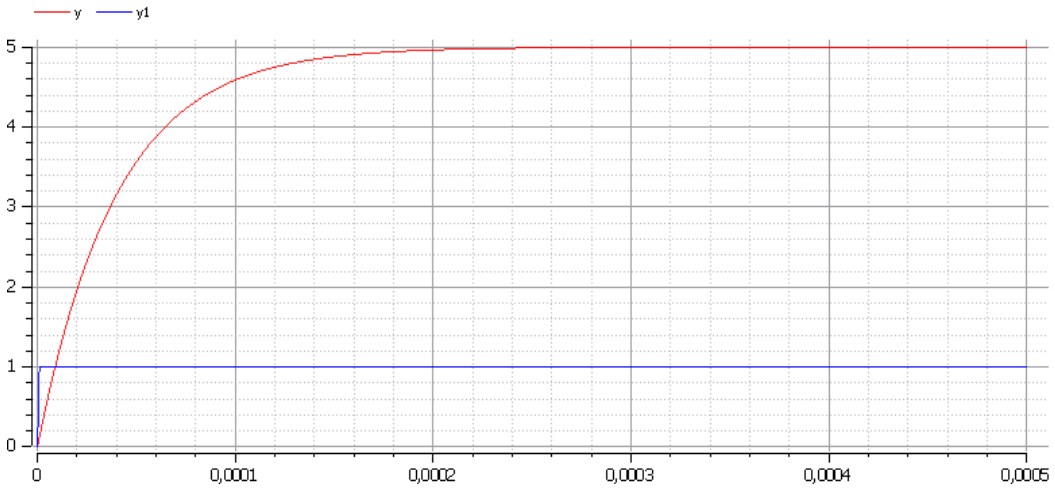

Figure 46: Step response Voltage regulator with LQR+PID and PLL with PI controller

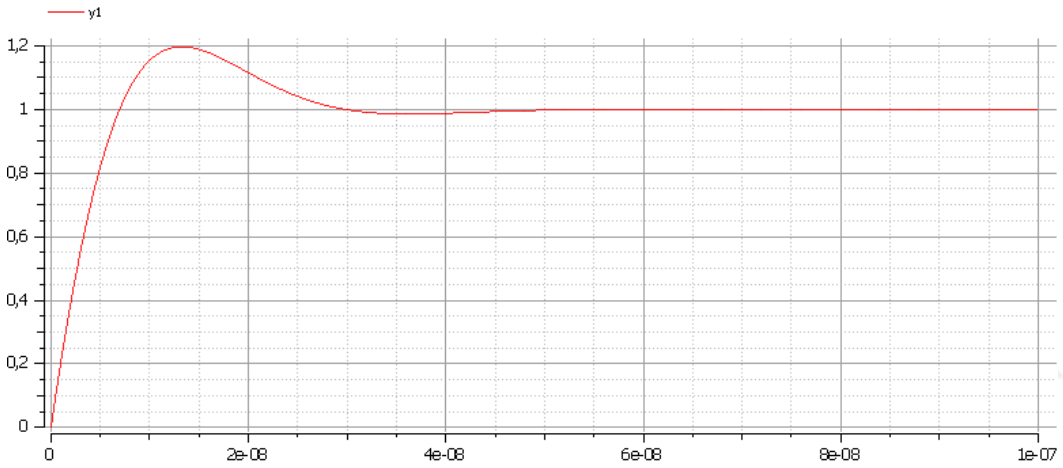

Figure 47: Focused Step response of PLL with PI controller

From step response of the designed model. combination of linear quadratic regulator and PID controller which applied to the voltage regulator, led to decrease in the settling time of the voltage regulator. Settling time of the voltage regulator was 30ms in the open loop configuration, after designed controller it decreased to the 0.0002s. Also in the combination presence of the PID controller eliminates the overshoot and makes step response of the voltage regulator smoother. Beside this PI controller is designed for the phase locked loop so settling time of the PLL is also decreased and become smoother.

3.5 Design and Implementation of the Feed Forward Controller

Significant difference of the feed forward controllers than the feedback controller is that it detects disturbance as it enters the process. In other words the feed forward control acts the moment a disturbance occurs instead of waiting for a change in process variable. Due to

this difference feed forward controller acts quickly and directly eliminate the effect of a disturbance. To sum up the feed forward controller deals with the disturbance of the system and the feedback controller tries to solve other issues which can cause difference between set point and the process variable.

In a lot of control problems which are present today's industries, combination of feedback and feed forward controllers are used. Combination of two different methods generally provides better controller performance for the system. In combinational case control signal is composed of two terms which are the feedback component and feed forward component from the set point Set point change in this case will be detected before it will have an effect on system. In this case feed forward term is used to rapidly suppress set point changes. However, this term must be used always with feedback one as feedback control system is able to suppress unmeasured disturbances and minimize error value. As a next step of implementation of controller, also for this thesis work combination of feedback and feed forward controllers is an useful tool.

3.5.1 Feed Forward + LQR + PID Implemented to Voltage Regulator and No Controller for PLL

In the first implementation of the feed forward controller, combination of feed forward controller, linear quadratic regulator and PID controller is applied to the voltage regulator. In this configuration phase locked loop is leaved as open loop in the configured system. According to state space model of the voltage regulator, gain matrix K and pre compensation gain is generated in the MATLAB software. Also gain values for proportional gain, integral gain and derivative gain are calculated according to the mathematical model of the voltage regulator. In order to implement combination of controllers, mathematical model of the voltage regulator is demonstrated by the state space model of the Modelica modeling environment. Calculated PID gains are entered to the PID block, calculated gain matrix and pre compensation gain are applied as gain block and feed forward effect is implemented as gain block. Finally phase locked loop is expressed with the transfer function block of the Modelica software. Created model and step response of the model are can be seen below:

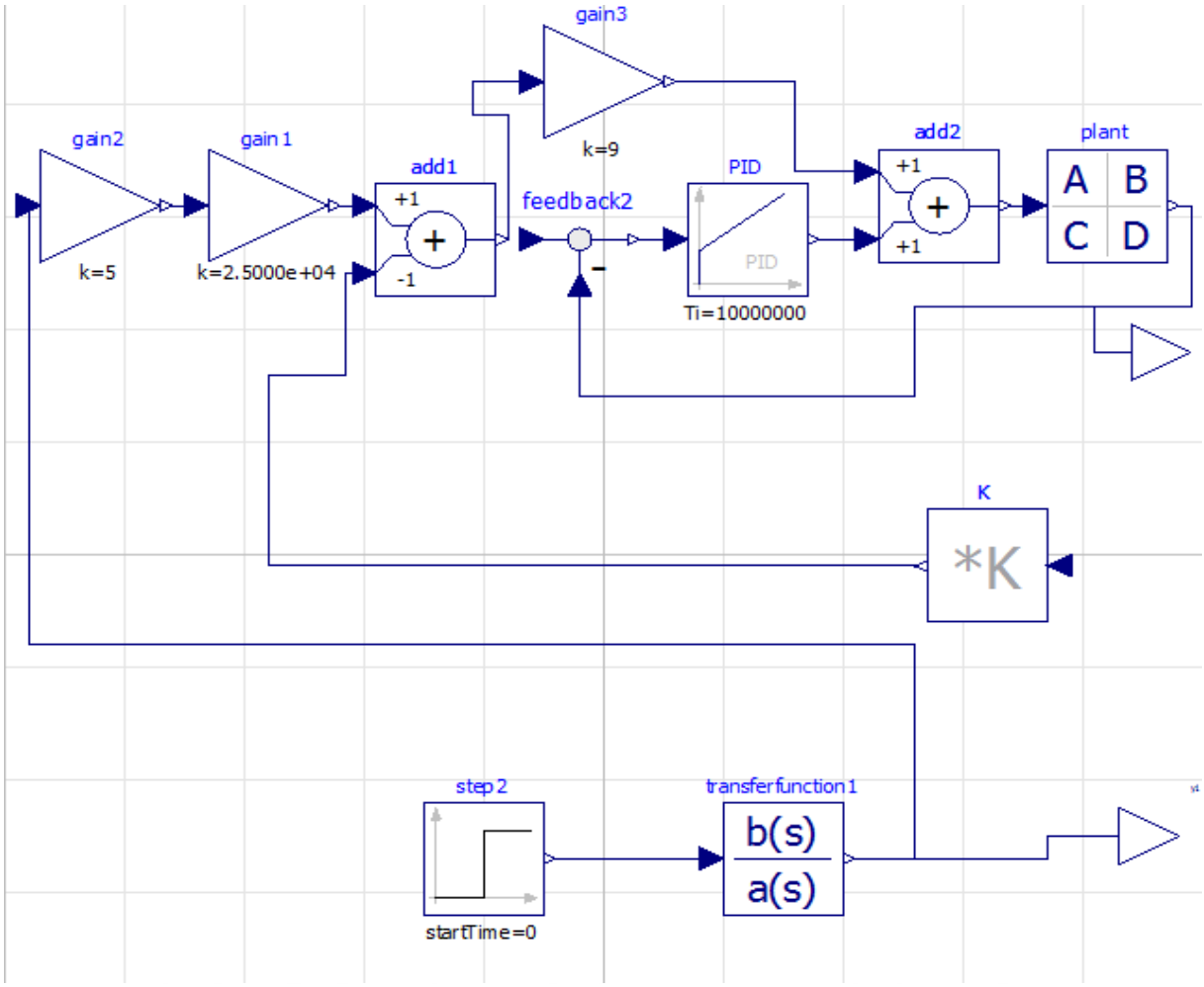

Figure 48: Voltage regulator with LQR+PID+FF and PLL is in open loop form

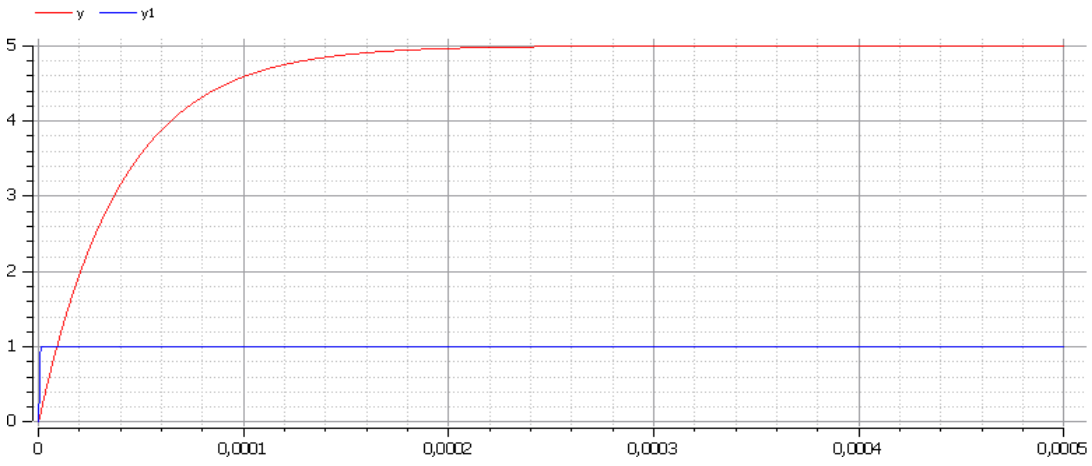

Figure 49: Voltage regulator with LQR+PID+FF and PLL is in open loop form

In the step response of the designed model, combination of feed forward, linear quadratic regulator and PID controller decreases the settling time of the voltage regulator. In this combination each controller does different jobs and as a result of all of them settling time of the voltage regulator decreases to 0.0002s from 30ms and ne overshoot exists in the step response of the voltage regulator. Main source of problem of this thesis work is difference between settling time of voltage regulator and phase locked loop which led to waste of time and power. As a result of implementing combination of three different controller method, gap between settling times of the system components has decreased.

3.5.2 Feed Forward + LQR + PID Implemented to Voltage Regulator PI Controller for PLL

In the final configuration which consist of feed forward controller, combination of feed forward controller, linear quadratic regulator and PID controller is applied to the voltage regulator and PI controller is applied to the phase locked loop. Mathematical model of the PLL is implemented with the transfer function block and calculated PID gains which are calculated due to the transfer function of the PLL are entered to the PID block. Matrix gain K and pre compensation gain of the linear quadratic regulator which are valid for the voltage regulator, are generated in the MATLAB environment and entered to the designed system with the gain blocks. Also for the voltage regulator, calculated PID gains which are calculated due to the mathematical model of the voltage regulator are entered to the PID block. Finally feed forward effect is implemented as gain block. Designed system and step response of the system of interest can be seen below:

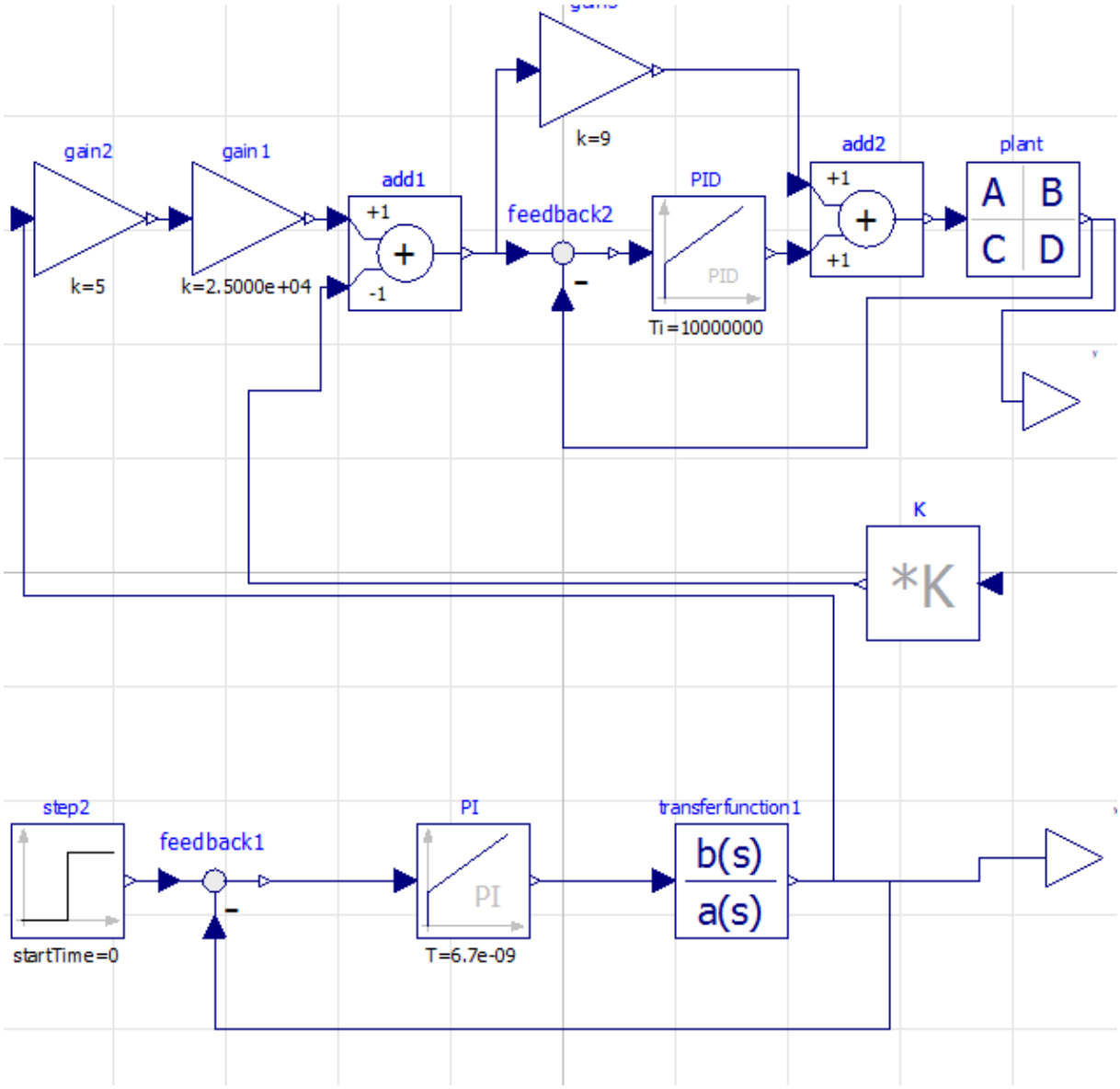

Figure 50: Voltage regulator with LQR+PID+FF and PLL with PI controller

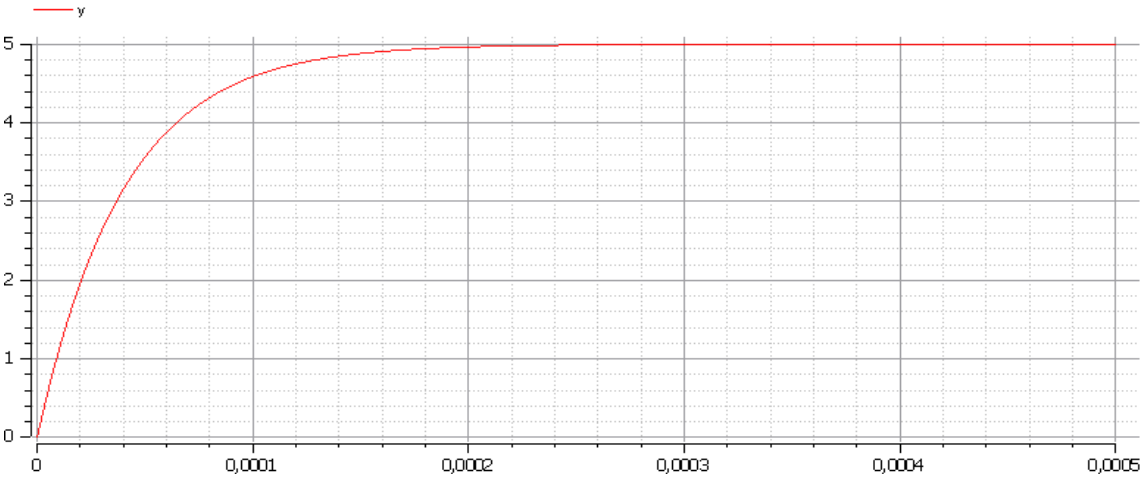

Figure 51: Step response of Voltage regulator with LQR+PID+FF and PLL with PI controller

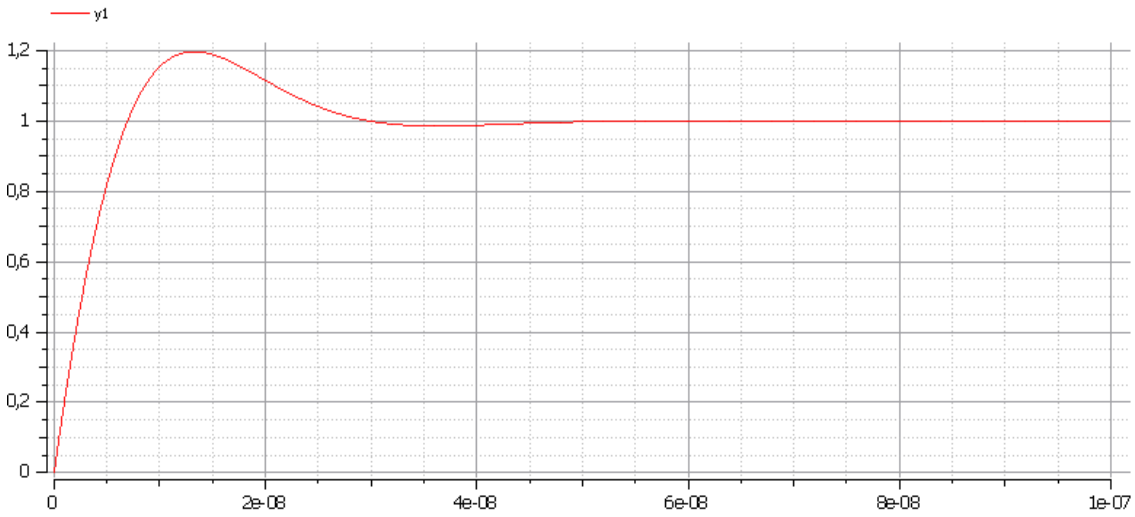

Figure 52: Focused Step response of PLL with PI controller

In the final model which consist of feed forward action, combination of feed forward controller, linear quadratic regulator and PID controller is applied to the voltage regulator and PI controller is applied to the phase locked loop. As a result of combination of controllers, settling time of the voltage regulator decreases to 0.0002s from 30ms as the similar to the previous configuration. Also in this case, PID controller does its job which is maintaining the output at a level so that there is no difference between the process variable and the set point. Depicted working principle of the PID controller leads to this decrease in the settling time of the PLL.

3.6 Addition Anti- Windup Implementation to PID Controller

An other option for the controlling aims is adding anti-windup compensation to the PID controller. Feed forward signals are added after controller blocks which may lead to windup effect. This can be explained simply by observing the fact that feed forward signals are in general big due to the fast reaction and then are pushed through saturation block. Therefore, it has been necessary to implement anti-windup technology to avoid this phenomenon. Windup is caused by the interaction of integral action and saturations. When control variable reaches saturation limit, feedback loop is broken and system runs as an open loop because the actuator will remain at its limit independently of the process output. However, the integral part will continue to integrate the error so the integral term will become very large, it “winds up”. After the mentioned winding, it takes a long period for system to return in normal mode. This effect can have source in the change of reference value. In order to solve this problem anti wind-up compensation should be added to the PID controller.

Modelica modeling environment has LIMPID block which is a PID controller with limited output, anti-windup compensation and set point weighting. LIMPID block has some extra properties than the normal PID block of the Modelica software, that properties are as follows:

- When controller reaches to its limits, added anti-windup compensation works and drive integrator state to zero. Therefore controller keeps its output in limits.
- Due to limiting on the high frequency gain of the derivative part of PID controller, controller avoids to the excessive amplification of the noises.
- Set point weighting is present.

Internal structure of the depicted Modelica block can be seen below:

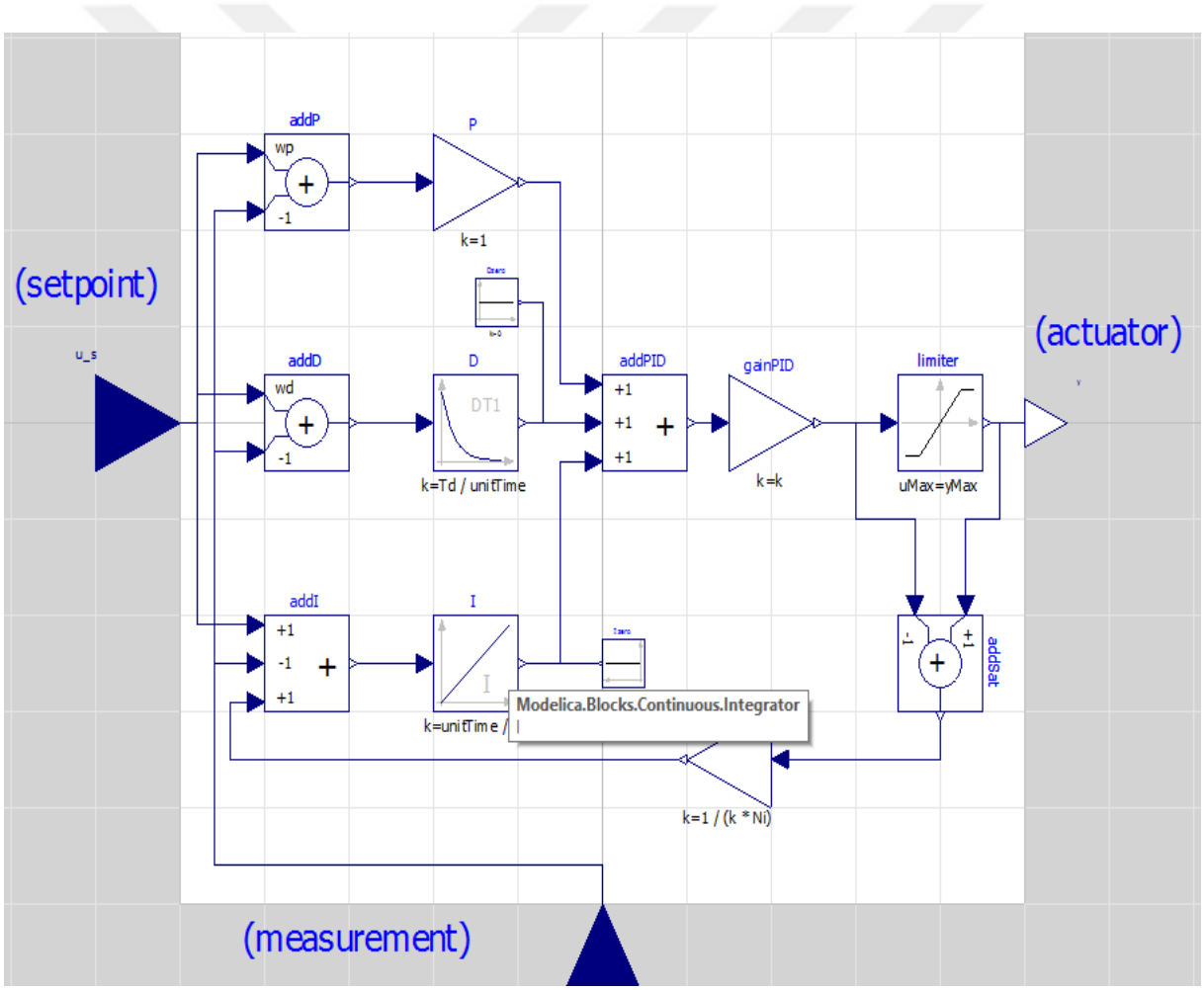

Figure 53: Internal structure of LIMPID block

3.6.1 Anti-Windup PID Implemented to Voltage Regulator PI Controller for PLL

In the first implementation of the anti-windup compensation, LIMPID block is applied to the voltage regulator and PI controller is applied to the phase locked loop. Mathematical models of the voltage regulator and phase locked are expressed by the transfer function block of the Modelica modeling environment. PID controllers are implemented by entering calculated gain values for proportional gain, integral gain and derivative gain according to the both of the system components, to the blocks of the Modelica software. Created model and step response of the model are as follows:

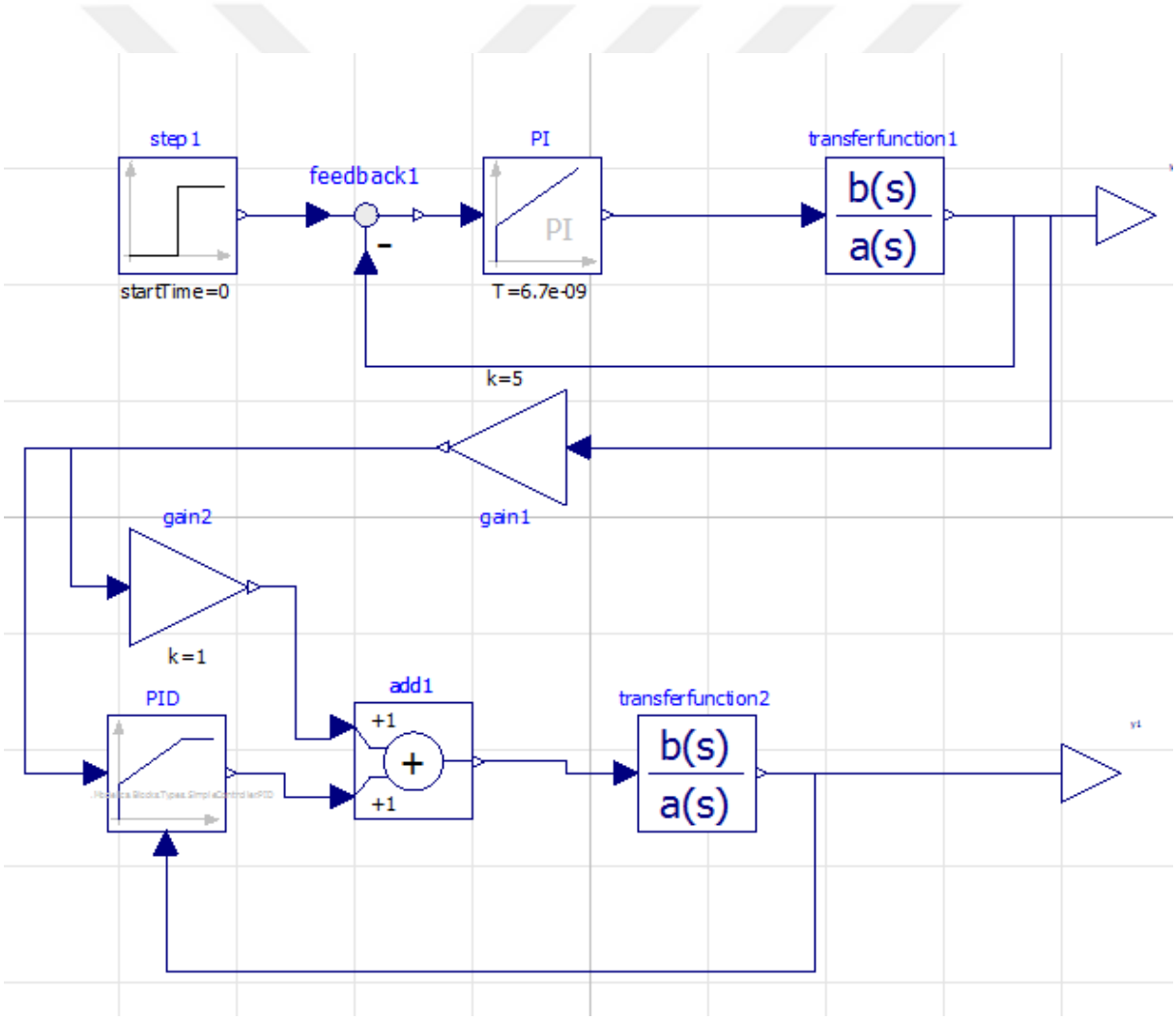

Figure 54: Voltage regulator with LIMPID block and PLL with PI controller

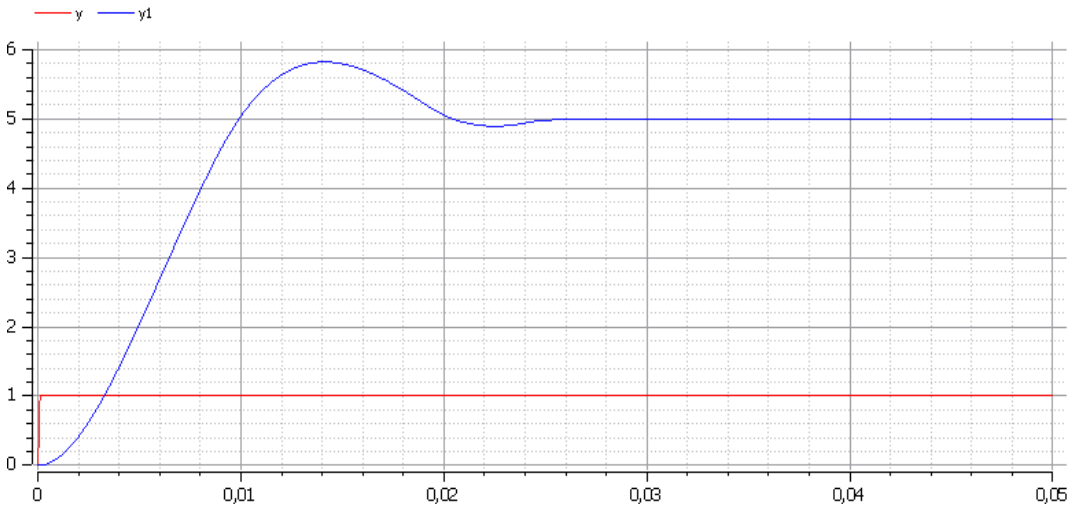

Figure 55: Step response of Voltage regulator with LIMPID block and PLL with PI controller

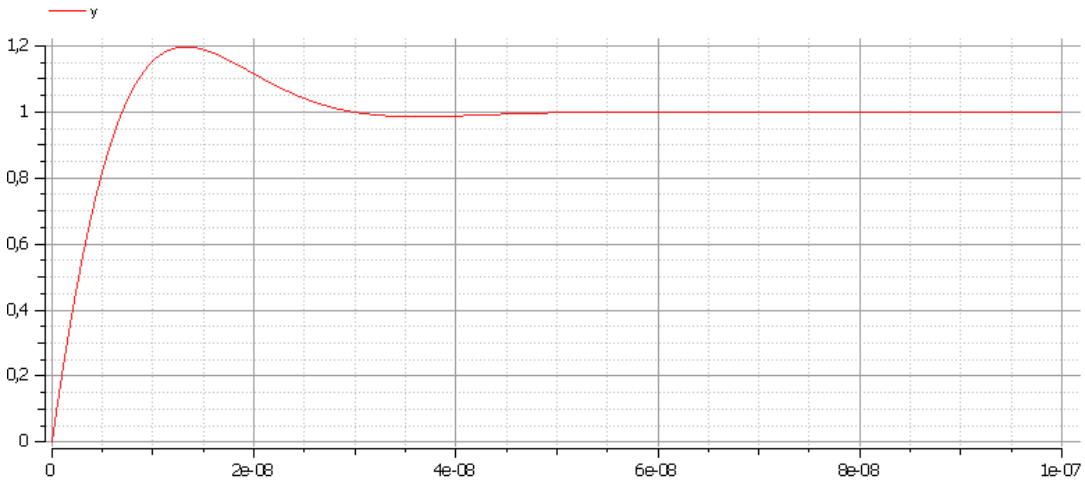

Figure 56: Focused Step response of PLL with PI controller

From the figures of the step response of the components, it is easy to says that PI and LIMPID blocks of the Modelica does their jobs and decrease the settling time of the both components and make their step response curves smoother. In the open loop configuration settling time of the voltage regulator was 30ms but with the help of PID controller with anti-windup compensation it decreased to the around 20ms.

3.7 Comparison Between Designed Controllers

As mentioned in the problem description section; in order to eliminate waste of time and energy of the system of interest , difference between settling times of the voltage

regulator and phase locked loop should be vanished or decreased. In order to understand most useful system for desired aim; settling time, overshoot and steady state errors of the step response of the designed system should be compared. Table which consist performance properties of the designed configuration is can be seen below

Designed Model	Settling Time	Overshoot	Steady State Error
1. Both components in open loop	≈ 30ms for VR ≈ 400ns for PLL	%15 for VR %5 for PLL	0 for VR 0 for PLL
2. PID applied to VR no controller to PLL	≈ 12ms for VR ≈ 400ns for PLL	%50 for VR %5 for PLL	0 for VR 0 for PLL
3.PID applied to both components	≈ 12ms for VR ≈ 60ns for PLL	%50 for VR %20 for PLL	0 for VR 0 for PLL
4.LQR applied to VR no controller to PLL	≈ 200μs for VR ≈ 400ns for PLL	%5 for VR %5 for PLL	0 for VR 0 for PLL
5. LQR + PID to VR no controller to PLL	≈ 200μs for VR ≈ 400ns for PLL	No for VR %5 for PLL	0 for VR 0 for PLL
6.LQR applied to VR PI applied to PLL	≈ 200μs for VR ≈60ns for PLL	%5 for VR %20 for PLL	0 for VR 0 for PLL
7.LQR+PID to VR PI applied to PLL	≈ 200μs for VR ≈ 60ns for PLL	No for VR %20 for PLL	0 for VR 0 for PLL
8.FF+PID+LQR to VR no controller to PLL	≈ 200μs for VR ≈ 400ns for PLL	No for VR %5 for PLL	0 for VR 0 for PLL
9. FF+PID+LQR to VR PI applied to PLL	≈ 200μs for VR ≈60ns for PLL	No for VR %20 for PLL	0 for VR 0 for PLL
10. Anti-windup PID to VR PI applied to PLL	≈ 20ms for VR ≈ 60ns for PLL	%15 for VR %5 for PLL	0 for VR 0 for PLL

Table 4: Performance summary of the designed configurations

From the Table 4 it is obvious that in the configuration eight gap between settling times of the voltage regulator and phase locked loop is minimum and also overshot values for the both of the components are in acceptable limits. Therefore designed system, which has combination of PID controller, feed forward action and linear quadratic regulator is the best for the desired aim.

3.8 Conclusion

In this chapter, step responses of the investigated system are provided. Firstly open loop configuration of the model is investigated and open loop settling times are defined. After that, design steps of the different control schemes such as PID controller, linear quadratic regulator are defined. Than control systems are defined with the help of explained control schemes. Finally step response of the control systems are investigated and the performance of them are compared

Chapter 4

CONCLUSION AND FUTURE WORK

During this thesis work, first of all mathematical models of the voltage regulator and phase locked loop are defined in the forms of transfer function and state space model. After reaching mathematical models of the components, different control schemes and their combinations are applied to the mathematical models in order to balancing settling times of the components and eliminating the waste of time and power of the system.

In details, in the first chapter general knowledge about system and Modelica modeling environment is provided and then problem is described. After that in second chapter of the work, meaning of the mathematical models are introduced than state space models and transfer functions of the components are introduced. Also in chapter two, control notion is introduced and different control schemes such as PID controller, linear quadratic regulator and feed forward action are described. Finally in third chapter, design steps of the different control schemes are mentioned and different control schemes and combinations of them are implemented to the system of interests in the Modelica environment. Step responses of the designed systems are generated in the Modelica software. And step responses of the designed configurations are compared and most useful for the problem is selected.

To sum up, obtained step responses of the designed configurations match the expectations from the control schemes. Best performance values are reached with the combination of the PID controller, linear quadratic regulator and feed forward. With this

combination difference between settling times of the components is decreased and also overshoots of the step responses are vanished.

Future scope of this work can be implementing the control schemes and their combinations to the more realistic models of the voltage regulator and phase locked loop. And also as future work more predictive controller schemes can be selected as a part of designed system

Appendix

Matlab code for convert phase locked loop transfer function into state space model

```
clc;
clear all;
close all;

n = [0 0.0000001 1];
d = [0.0000000000000042 0.00000012 1];
sys = tf(n,d);
[A,B,C,D] = tf2ss(n,d)
```

Matlab code for convert voltage regulator transfer function into state space model

```
clc;
clear all;
close all

n = [0 0 1];
d = [0.000025 0.005 1];
sys = tf(n,d);
[A,B,C,D] = tf2ss(n,d)
```

Matlab code for tuning tool for PID gains of the voltage regulator

```
clc;
clear all;
close all;

n = [0 0 1];
d = [0.000025 0.005 1];
sys = tf(n,d)
[C_pi, info]= pidtune(sys, 'pi')
sisotool(sys)
```

Matlab code for tuning tool for PID gains of the phase locked loop

```
clc;
clear all;
```

```
close all;

n = [0 0.0000001 1];
d = [0.0000000000000042 0.00000012 1];
sys = tf(n,d);
[C_pi, info]= pidtune(sys, 'pi')
sisotool(sys)
```

Matlab code for LQR gains

```
clc;
clear all;
close all

n = [0 0 1];
d = [0.000025 0.005 1];
sys = tf(n,d)

[A,B,C,D] = tf2ss(n,d)

states = {'x' 'x_dot' };
inputs = {'u'};
outputs = {'x'};

sys_ss =
ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',
outputs);

poles = eig(A)
co = ctrb(sys_ss);
controllability = rank(co)

Q = C'
Q(2,2)=1000000000000000000


R = 1;
K = lqr(A,B,Q,R)

s = size(A,1);
          Z = [zeros([1,s]) 1];
          N = inv([A,B;C,D])*Z';
          Nx = N(1:s);
          Nu = N(1+s);
          Nbar=Nu + K*Nx
```

Bibliography

[1] Mark Readman, *Phase Locked Loops,* control system principles.co.uk, 2011.

[2] Guan-Chyun Hsich, James C. Hung, *Phase-Locked Loop Techniques--A Survey,* IEEE Transactions On Industrial Electronics, Vol. 43, No. 6, December 1996 (pp. 609 - 615)

[3] *Phase- Lock Loop-Based(PLL) Clock Driver: A Critical Look at Benefits versus Costs,* Texas Instruments, March 1997.

[4] Chester Simpson, *Linear and Switching Voltage Regulator Fundamental Part 1,* Texas Instruments,2011.

[5] Dong He, Quingyu Xiong, Weiren Shi, Xin Shi, Jaiqi Zhang, *Building Energy and Control System Modelinf and Simulation Using MODELICA*, 26th Chinese Control and Desicion Conference(CCDC), 2014.

[6] Peter Fritzson, *Modelica- A Cyber- Physical Modeling Language and the OpenMODELICA Environment,* 978-1-4577-9538-1/11/$26.00 ©2011 IEEE.

[7] Habib Adrang, Hossein Miar Naeimi, *A type III Fast Locking Time PLL with Transconductor-C Structure*, 978-1-4244-4480-9/09/$25.00 ©2009 IEEE

[8] Sigurd Skogestad, Ian Poslethwaite, *Multivariable Feedback Control Analysis and Design,* John Wiley & Sons, 4th edition.

[9] *Engineering Mathematics: Open Learning Unit Level 1,* Chapter 20.7 Transfer Functions.

[10] Smarajit Ghosh, *Control Systems Theory and Applications,* Pearson Education, 2007

[11] John C. Doyle, Keith Glover, Pramod P. Khargonekar, Bruce A. Francis, *State-Space Solutions to Standars H_2 and H_∞ Control Problems,* OO18-9286/89/08OO-0831$01.OO© 1989 IEEE (pp. 831-847).

[12] Kiam Heong Ang, Gregory Chong, Yun Li, *PID Control System Analysis, Design and Technology,* Ieee Transactions on Control Systems Technology, Vol. 13, No. 4, July 2005.

[13] John Doyle, Bruce Francis, Allen Tannenbaum, *Feedback Control Theory,* Macmillan Publishing Co., 1990.

[14] Andrew Lamperski, Noah J. Cowan, *Time-changed Linear Quadratic Regulator*, 2013 European Control Conference (ECC), Switzerland, July 2013.

[15] R. E. Kalman, *Contributions to The Theory of Optimal Control,* Bias , Baltimore, Martland, (pp. 149 - 166)

[16] C. C. Hang, K. J. Aström, W. K. Ho, *Refinements of the Ziegler- Nichols tuning Formula,* IEE PROCEEDINGS-D, Vol. 138, NO. 2, MARCH I991

[17] Finn Haugen, *Ziegler-Nichols' Closed-Loop Method,* Tech Teach, July 2010