Simulations of GMRT Servo-Mechanical System

Trupti Ranka

15 June, 2009

Abstract

For understanding the servo system and planning future developmental activities it is necessary to have computer simulations of the system. The simulation of Gaint Meterwave Radio Telescope (GMRT) servo-mechanical system with generic position controllers were carried in 1992 by B.C.Joshi [1]. These simulations were done before the actual antenna was build and hence model verification was not possible. In the view of new developmental activities being planned, it is necessary to modify these simulations to reflect the of servo-mechanical system now present at GMRT. Also since the antenna is now available for experiments it is necessary to verify this new model by comparing it with response of actual system.

Optimal controllers were designed for GMRT servo in [1] for future implementation. Optimal controllers have potential to give better antenna pointing accuracy even in presence of wind disturbance. The simulations for optimal controllers need to be modified in accordance with new plant matrices. In view of the new servo computer being developed for GMRT servo system there is now a realistic chance of implementing optimal controllers for GMRT servo system. The feasibility of the implementation needs to be verified with the help of simulations and lab experiments.

The points mentioned above have been discussed in this report.

Acknowledgments

I am grateful to Dr. Bhal Chandra Joshi for giving me an opportunity to work on this project. He has constantly motivated and guided me for this work for past 2 years. Our discussions for this work have been very educative for me. These discussions have not only helped me in understanding the servo-mechanical system of GMRT but has also broadened my general understanding of control systems. During my project he has encouraged me to think independently on various problems and explore the various possibilities of solving the problem.

The experience of working at GMRT has been very enriching. I would like to thank Mr. Suresh Sabhapathy who has helped me a lot for my day-to-day work at GMRT. He has helped me immensely in collecting various documents needed to calculate system parameters. My work would have not reached this stage without the help of all servo lab members especially Abhay Bhumkar, Datta Ghorpade, Amit Kumar, Santosh Bhaskar, Sanjay Belhekar, Kiran and Kavade who have helped me in performing various experiments on the antenna. A special thanks to Srinivasrao Beera for making my time spent in the lab very enjoyable.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation for simulation

This report presents the details and results of Giant Meterwave Radio Telescope (GMRT) servo system simulation. Simulation is a computer program which imitates the behaviour and response of a real system. In this case the system is the servo system of GMRT. The simulation programs attempt to generate outputs to various inputs as the actual servo system would do. These programs form a tool which can be used for a better understanding of the servo system and also for planning of new developmental strategies for the servo system. There are three major steps followed in making an accurate simulation of the real system. First the mathematical equations describing the system behaviour are derived. They are then used to create the simulation programs. Lastly, the simulation programs are validated by comparing its response with the response of the real system.

In the case of GMRT servo system, the main aim of the system is to safely point the antenna to a given location in the sky. For good quality astronomical images it is also necessary to achieve high pointing and tracking accuracy. These two objectives depend on the design of servo system as well as the values of various tunable parameters of the servo controller. Simulations can play a major role in both design of the servo system as well as coming with right tunable parameters to achieve the two objectives mentioned above. The direct uses of these programs are as listed below.

1. **Finding the best tuning parameters for the present system.**
   Various servo controller tuning parameters can be changed in the simulation, so as to find the parameters that give the minimum steady state tracking error and specified transient response. Generally, the tuning parameters are obtained by trail and error method on field, which can be very time consuming. The simulations gives us a set of tuning values to start with during the actual tuning of antenna. This saves a lot of time spent on tuning the antenna on field.

2. **Robustness study of the servo system.**
   GMRT consists of 30 radio telescopes. The physical parameters like, mechanical system parameters, motor parameters, electronic component values can change from antenna to antenna and also for the same antenna over a period of time. Though all 30 antennas have the same servo-mechanical design specification and manufacturing, their response

to sine sweep signal vary from each other. The effect on relative stability and tracking accuracy of servo system with respect to change in these parameters can be evaluated with the help of simulations.

3. **Study of tracking accuracy of the antenna under wind conditions and other disturbances.**
   The antenna is operated under effect of wind for about 80 percent of its operation time. Wind is a stochastic process and its influence on tracking accuracy can change from time to time. The simulations help us to study the performance of antenna servo even under the influence of varying wind conditions. The simulations help us to predict the decrease of tracking accuracy under various wind speeds.

4. **Design of better controllers.**
   Specifically, in the case of GMRT servo system an attempt is being made to design linear quadratic optimal controllers. These controllers can give the best possible tracking accuracy even under the influence of various disturbances.

5. **Implementation of optimal controllers.**
   The simulation programs for optimal controllers are being written in a software called Simulink. Simulink gives the facility to convert these simulation programs into appropriate C programs. These programs can then be loaded in the servo computer, where these controllers can be implemented in software form. There is no need to write separate code for implementation on hardware. The use of softwares like Matlab/Simulink can hence greatly expedite the implementation of optimal controllers.

6. **Design of Hardware in the Loop Simulator.**
   The antenna is not always available to engineers for performing various tests. This puts a heavy time constraint on the engineer to perform the tests on the antenna. If the simulations of antenna servo is present, then a setup can be designed where the actual antenna is replaced by the simulation program. The simulation program thus forms a virtual plant. The rest of the controller hardware can be interfaced with this program, and hence the controller and other associated hardware/software can be tested without the need to go to the actual antenna.

This work of simulation is largely based on earlier simulation work in [1]. This simulation work differs from earlier work in four major aspects. Firstly, the velocity and position loop controller simulation imitate the actual controllers in the present antenna. This is unlike, the generic controller design that was used in [1]. Thus the present simulations are a closer match to the actual system. Secondly, unlike in previous case this time antenna was actually available for model validation. Thus a clearer picture of the deficiencies in the simulations can be formed, which will aid the further improvement of the model. Thirdly, this time the basic plant modeling of the servo system has been carried out in Simulink rather than m-code. This will greatly facilitate the use of these simulations for further developmental activities. Lastly, the strategy that can followed to implement optimal controllers and testing their performance has been discussed.

1.2 Structure of the report

The report presents the equations (and derivation of the equations wherever necessary) that have been used for simulations. These equations are followed by the results of the simulation and their comparison with actual results obtained from the tests on antenna. Chapter 2 deals with simulation of present GMRT servo system which has brushed DC motor as its servo system. Chapter 3 deals with design of optimal controllers for GMRT servo system. Chapter 4 discusses the overall implications of various simulation results. It also discusses the ongoing work in simulation and feasible strategies for implementing optimal controller for GMRT servo system. Appendix A gives details of the Matlab/Simulink programs that have been used for simulations.

Chapter 2

Brushed DC motor and its servo system

2.1 Introduction

The present servo system at GMRT has brushed DC motors and associated control system to move each axis. From the control system perspective, the positioning of the antenna is done with the help of three nested loops. The final output of these loops is given to the servo amplifier which drives the motors and the antenna. These loops are current loop, velocity loop and position loop. The current loop is implemented in motor control card. The velocity loop is implemented in counter torque card while the position loop is implemented in servo computer in software form.

Each axis has two motors. The two motors are so driven that the backlash between the gears is eliminated. This is called as counter-torquing arrangement. The counter torquing is implemented in the counter torque card. Figure 2.1 shows the complete design of the servo loop. For simulation we consider that the counter torquing complete eliminates the backlash in the system. hence the two current loops can be replaced by a single loop as shown in figure 2.1

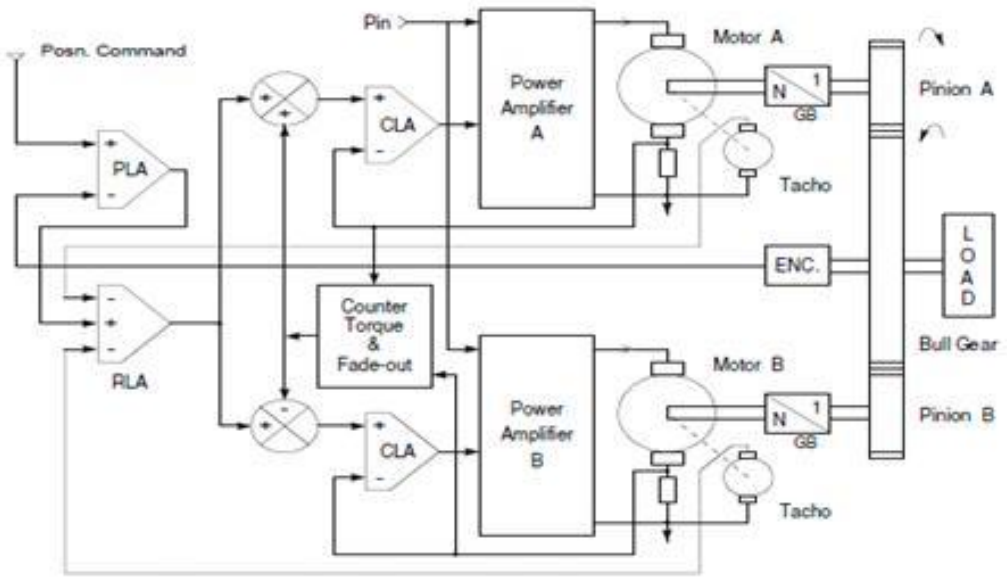

Figure 2.1: FBD of single axis servo-mechanical system with counter-torquing arrangement [2]

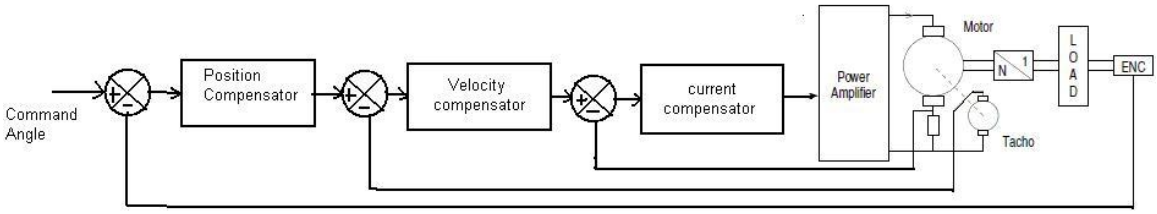

Figure 2.2: Equivalent FBD of single axis servo system assuming complete backlash compensation.

For understanding the system and guaranteeing the accuracy of simulation, the simulations are carried in two stages. In section 2.2 the simulation of only the motor and its control loops is described. The results of these are compared with that of test results. These simulation reflect only the motor and servo controller dynamics and exclude the antenna mechanical system dynamics. Once these simulations have been validated we move to simulate the servo system with the antenna mechanical system model. Section 2.3 describes the simulation of the servo system with antenna model. The simulations are then compared with test results.

2.2 Servo controller with a brushed DC motor

2.2.1 Derivation of Equations

We consider a detailed derivation of equations describing the DC motor, the current loop, the velocity loop and lastly the position loop.

Equation of Brushed DC motor

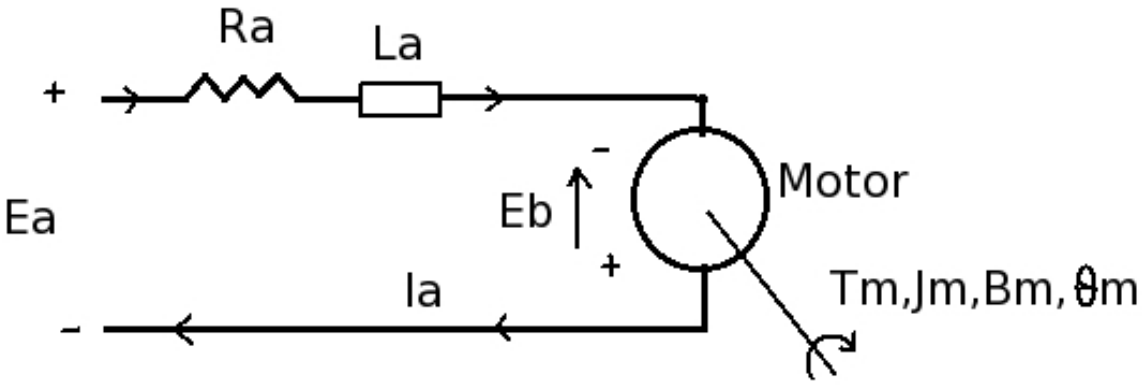

Figure 2.3: Diagram of Brushed DC motor

Figure 2.3 shows the equivalent electromechanical circuit for a brushed DC motor. The equations 2.1 - 2.3 can be derived from equivalent circuit [3]

$$\dot{I}_a = \frac{E_a}{L_a} - \frac{K_b \omega_m}{L_a} - \frac{R_m I_a}{L_a} \tag{2.1}$$

$$\dot{\omega}_m = \frac{K_t I_a}{J_m} - \frac{B_m \omega}{J_m} \tag{2.2}$$

$$\dot{\theta}_m = \omega_m \tag{2.3}$$

where,
I_a = Motor current
E_a = Motor input voltage
L_a = Motor winding inductance
K_b = Motor back EMF constant
R_m = Motor winding resistance
K_t = Motor torque constant
θ_m = Motor angular position
ω_m = Motor angular speed
J_m = Motor shaft inertia
B_m = Motor viscous friction

Equations for current loop

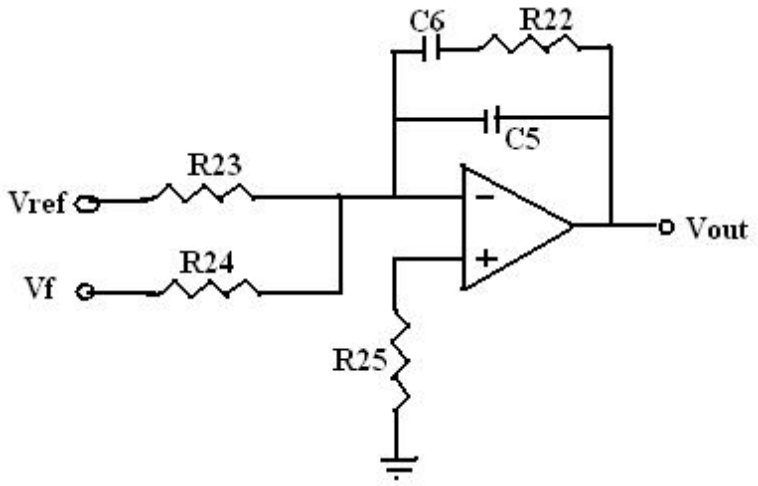

Figure 2.4: Current PI controller in MCC card

The current compensator for brushed DC motor is present in Motor control card. The current compensator is a PI compensator as shown in Figure 2.4. The maximum input from the counter torque card which is 10 volts is scaled to 4 volts using POT 24 in motor control card giving a current reference scaling factor of 0.4. The feed back is measured across the shunt of 0.005 ohm giving a feedback voltage of 0.4 Volts at maximum current of 80 amps. The PI compensator is also a summing amplifier. The input resistance to the PI compensator for reference voltage (R23) is 11 times higher than feedback voltage resistance (R24), thus approximately balancing the two inputs. According to Figure 2.4 the transfer function for current PI is given as follows,

$$G_f(s) = \frac{(R_{22}C_6s + 1)}{s[(C_5 + C_6) + R_{22}C_6C_5s]} \tag{2.4}$$

$$\begin{aligned} V_f &= R_{shunt}I_f \\ &= 0.005I_f \end{aligned} \tag{2.5}$$

Where,
$G_f(s)$ = Amplifier feedback impedance
V_f,I_f = Current loop feedback Voltage and current
Current loop PI is a summing amplifier thus,

$$V_{out}(s) = \frac{G_f(s)V_{ref}}{R_{23}} + \frac{G_f(s)V_f(s)}{R_{24}} \tag{2.6}$$

Where, V_{ref} = Reference input voltage (Cmd A or Cmd B)

The output of the current loop goes to the three phase SCR amplifier. The SCR amplifier is very fast as compared to the rest of the system. Hence, it can be considered as a simple gain factor. The 0-12 volts of current loop output is amplified to 0-150V by SCR amplifier which is then given to the motor. Hence, the gain factor K_{amp} is 150/12 = 12.5.

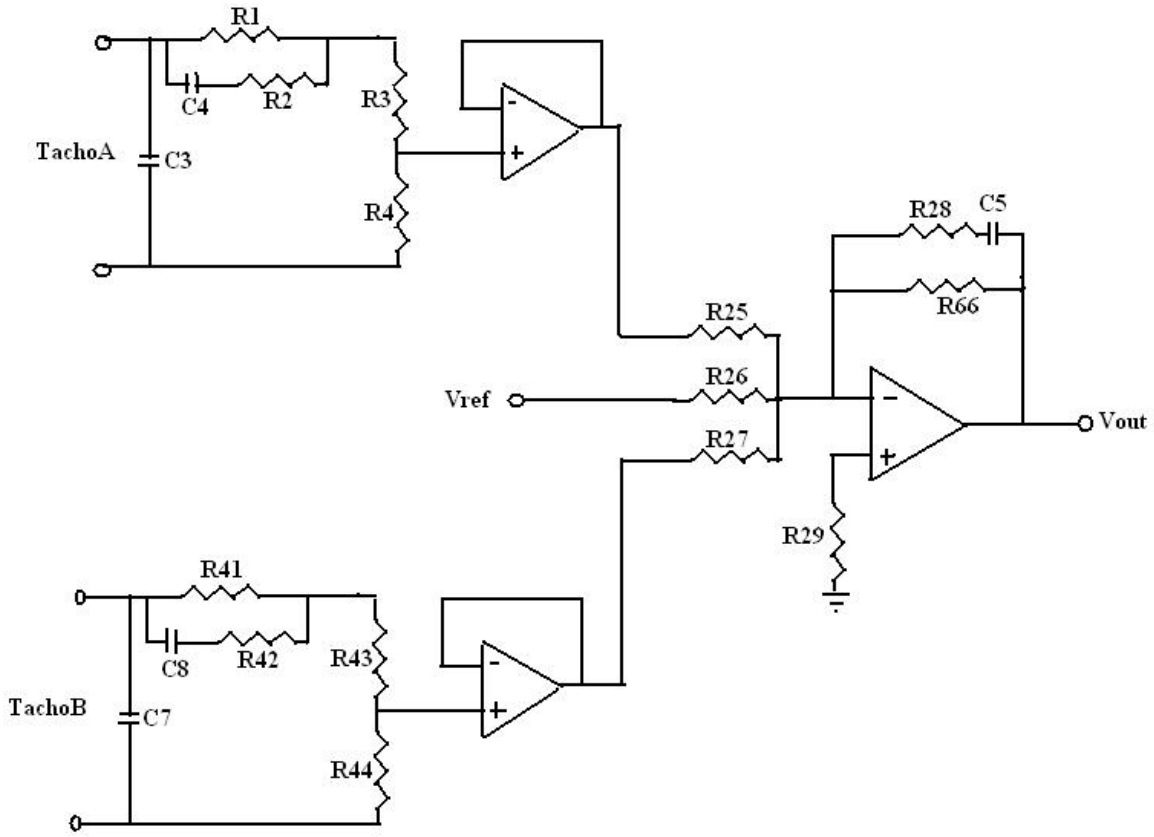

Figure 2.5: Circuit Diagram for Velocity compensator (brushed DC)

Equation for Velocity loop

The velocity compensator for brushed DC motor is implemented in counter torque card. Figure 2.5 shows the lead lag part of the counter torque card. This part forms the velocity compensator. The counter torque card takes three inputs:

1. Reference voltage, V_{ref}. This is the voltage corresponding to demand velocity generated from position loop output. Its range is -10 to 10 Volts. Here the input of 10 Volts corresponds to 2000 rpm of motor.

2. The other 2 inputs are tacho voltages for two motors (TachoA and TachoB). The tacho gives 34 volts per 2000rpm. The circuit network at the input of tacho voltage acts like lead compensator and also as a voltage divider which scales the tacho voltage from 34 volts to 10 volts for 2000rpm (motor full speed).

The tacho voltages along with reference voltage are the input to a summing amplifier which also acts as a lag network. At full speed the input from each tacho is scaled to 10 (by lead network voltage divider). Thus combined voltage of both the tachos is 20 volts for an input voltage of 10 volts (V_{ref}). To balance the voltages the input resistances (R25 and R27) to summing amplifier (lag compensator) for the tacho voltages is twice the input resistance (R26) for reference voltage. The transfer functions for counter torque card are as follows,

$$G_{s1}(s) = \frac{[R_4 + R_4(R_1 + R_2)C_4 s]TachoA}{(R_1 + R_3 + R_4) + [R_1R_2 + (R_1 + R_2)(R_3 + R_4)]C_4 s} \quad (2.7)$$

$$G_{s2}(s) = \frac{[R_{44} + R_{44}(R_{41} + R_{42})C_8 s]TachoB}{(R_{41} + R_{43} + R_{44}) + [R_{41}R_{42} + (R_{41} + R_{42})(R_{43} + R_{44})]C_8 s} \quad (2.8)$$

$$G_f(s) = \frac{R_{28}R_{56}C_5 s + R_{56}}{(R_{56} + R_{28})C_5 s + 1} \quad (2.9)$$

The compensator is a summing amplifier Hence:

$$V_{out}(s) = \frac{V_{ref}G_f(s)}{R_{26}} + \frac{G_{s1}(s)G_f(s)}{R_{25}} + \frac{G_{s2}(s)G_f(s)}{R_{27}} \quad (2.10)$$

Once the backlash has been overcome, both the motor will be at the same speed. Thus ideally both tacho input voltage are equal. The lead networks for input of both tachos are ideally the same.Thus,

TachoA = TachoB = Tacho, $G_{s1} = G_{s2} = Gs, R_{25} = R_{27} = R$

$$V_{out}(s) = \frac{V_{ref}(s)G_f(s)}{R_{26}} + \frac{2G_f(s)Gs(s)}{R} \quad (2.11)$$

Where, $G_f(s)$ = Amplifier feedback impedance
As the feedback input Tacho is subjected to extra transfer function Gs, the overall transfer function is represented as two terms for simplicity of implementation.

Equations for position loop

The position loop has been implemented in servo computer. The transfer function for position compensator is,

$$G_2(s) = \frac{G_{21}(T_{21}s + 1)(T_{22}s + 1)}{s(T_{23}s + 1)} \quad (2.12)$$

To implement the controller in digital form it is converted to discrete transfer function using bilinear transform and sampling time of 100 msec. The position error input to the compensator is pre-scaled by multiplying it by 1024. The output of the compensator is given to a 12 bit D/A converter with output voltage range of ±10 Volt. The input error to the compensator is in form of degrees. Hence the feedback has a scaling factor corresponding to radians to degree conversion. The encoder is a 17 bit absolute encoder. This adds a quantization of $360/2^{17}$ to the feedback loop. The present values of the compensator are as shown in table 2.2.1.

Hence the compensator equation is as follows:

$$G_2(s) = \frac{0.12(1 + 4.2s)}{s(1 + 0.4s)} \quad (2.13)$$

Parameter	Value
G_{21}	0.12
T_{21}	4.2
T_{22}	0
T_{23}	0.4

Table 2.1: Position compensator values

2.2.2 Simulation results and comparison with test results

The Brushed DC motor along with the compensator equations were simulated using simulink. The DC motor was implemented in state space form while the compensator in Laplace transfer function form. The details of simulation can be found in appendix A.2 figure A.1. To compare the simulation output with the actual response two tests were performed as follows,

DC motor parameter measurement

The various parameter in the equations describing the DC motor (Eqn: 2.1 - 2.3)vary from their values as given in specification sheet. By measuring the motor current and velocity for a given DC voltage input measurements these parameters can be estimated as follows [4],
At steady state (constant velocity) the term $\dot{I}_a$ becomes 0, so Eqn: 2.1 can be modified as,

$$\frac{E_a}{I_a} = K_b \frac{\omega_m}{I_a} + R_a$$

This is an equation of a straight line. If we plot a graph of ω_m/I_a Vs E_a/I_a, we will get a straight line with a slope equal to back emf constant K_b and y intercept as R_a as shown in figure 2.6.
Similarly, Eqn 2.2 can be written as,

$$\dot{\omega}_m = \frac{K_t I_a}{J_m} - \frac{B_m \omega}{J_m} - \frac{T_{sf}}{J_m}$$

Where the extra term T_{sf} is motor coulomb friction. At steady state (constant velocity), the term $\dot{\omega}_m$ becomes 0. Hence the equation can be modified as,

$$I_a = \frac{B_m}{K_t}\omega + \frac{T_{sf}}{K_t}$$

This is again an equation of straight line. Also we assume the value of K_t. But K_t can be independently estimated using torque transducer. Thus, by plotting ω/K_t Vs I_a we can measure motor viscous friction B_m and motor coulomb friction T_{sf}, as given in figure 2.7. Table 2.2 give the values of parameters estimated by experiment.

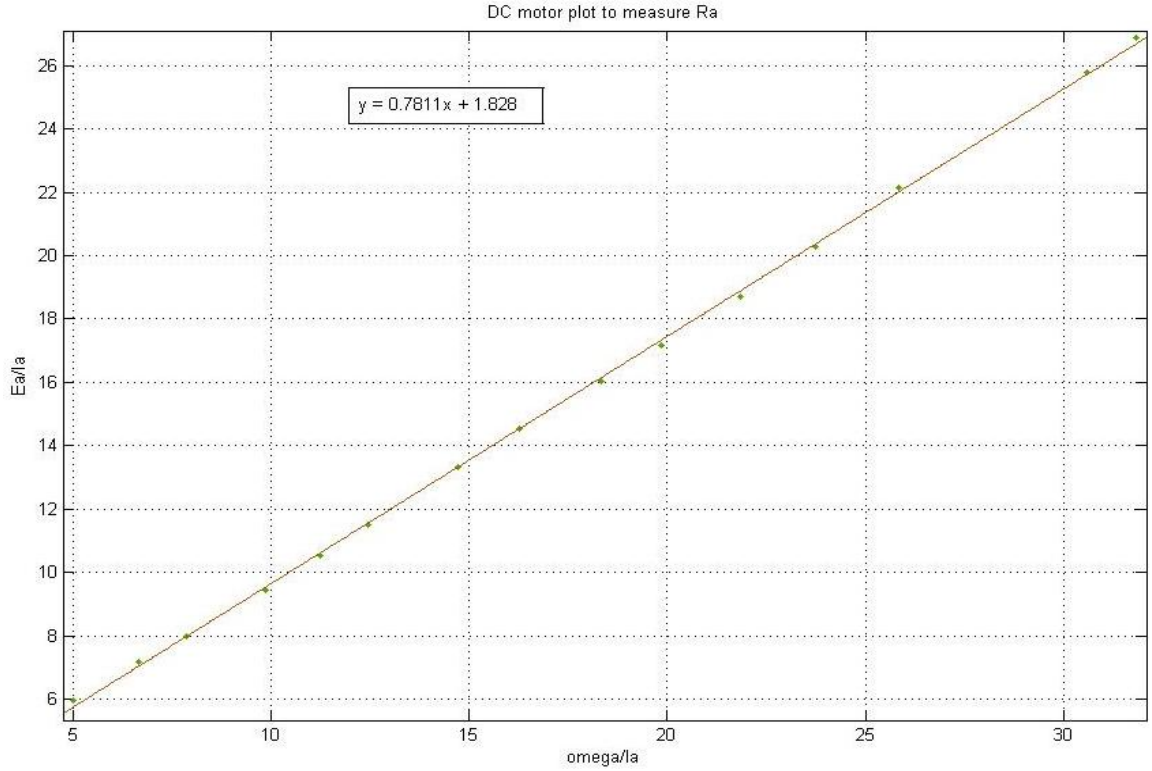

Figure 2.6: DC motor parameter estimation graph 1

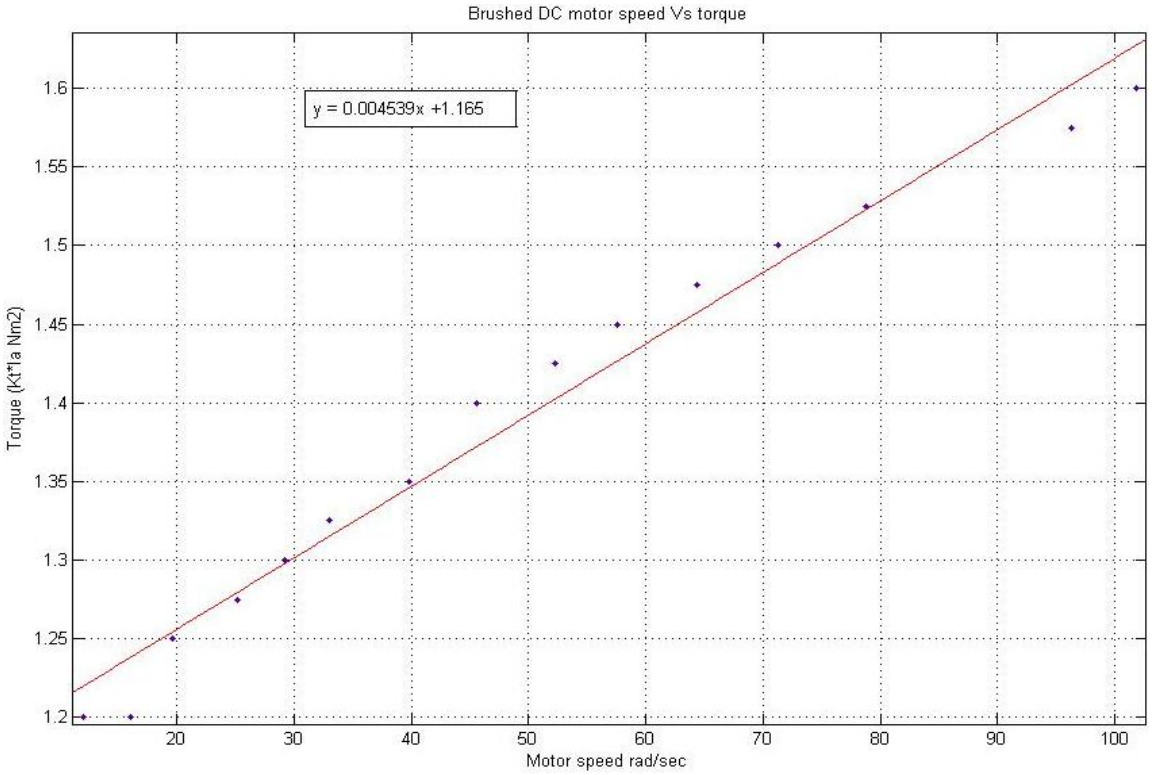

Figure 2.7: DC motor parameter estimation graph 2

Parameter	Value	Unit
Viscous Friction	.00453	$Nm/rads/sec$
Stiction	1.165	Nm
Resistance	1.828	Ω
Back EMF constant	0.07811	V/rads/sec

Table 2.2: Brushed DC motor Parameters as estimated by experiments

The above parameters are used for simulating the brushed DC motor. The simulation is modified to take into account the non-linear parameter of coulomb friction.

Step response of brushed DC motor along with servo system

The velocity step response of motor along with the servo system is measured and the data is recorded. This data is then read in Matlab and compared with with simulation response. A step voltage input using function generator is given at test input to counter torque card (velocity compensator). The parameters recorded are test ip voltage, TachoA, Tacho B (velocity), current cmd to A, current cmd to B (o/p of counter torque card). The motor current is measured in MCC card at current pin.

The servo velocity loop is designed considering the fact that 2 motors drive the load. But for testing the motor only one motor is excited. The other motor is not switched off, hence Tacho B and current in MCC card corresponding to motor 2 are 0. As a result the velocity to voltage scaling becomes twice. Or in other words now the 10 volts of input will not correspond to 2000 rpm but approximately to 4000 rpm. This additional scaling factor needs to be taken into account while comparing the actual and simulated response. Also during the test the velocity scaling pot P1 was adjusted so that 10 volts corresponded to 500 rpm instead of 2000 rpm. The Azimuth axis counter torque card was used for the experiment. Figure 2.8 and 2.9, show the comparison between the simulated and actual velocity and current cmd. Figure 2.10 shows the motor current. It is not compared with simulated as we have not simulated the dynamics of 3 phase SCR amplifier circuit which gives rise to pulsed current. Appendix B, figure A.1 gives the simulink model of motor plus servo system. Table 2.3 gives the compensator values for servo loop.

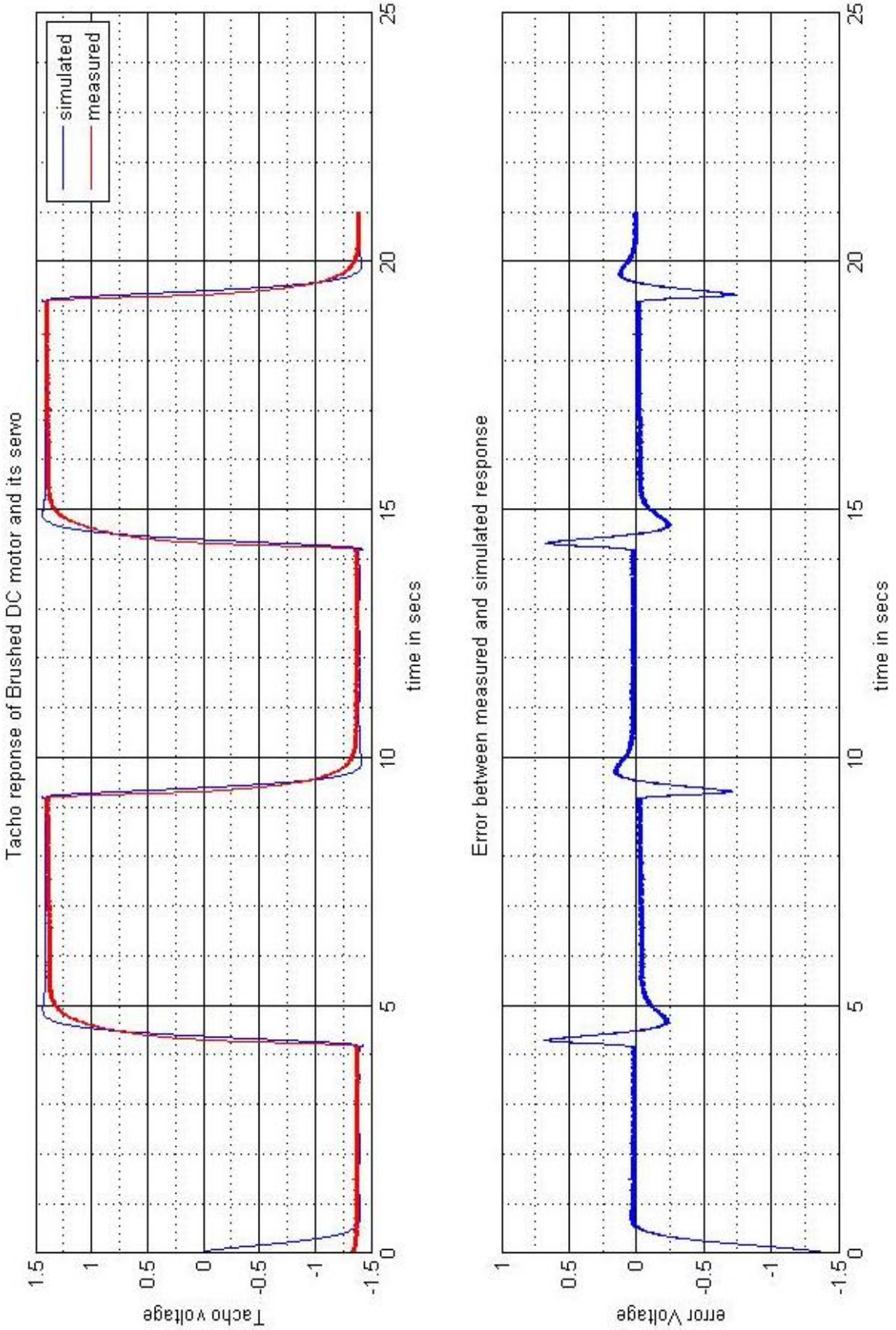

Figure 2.8: Brushed DC motor only velocity step response Tacho A

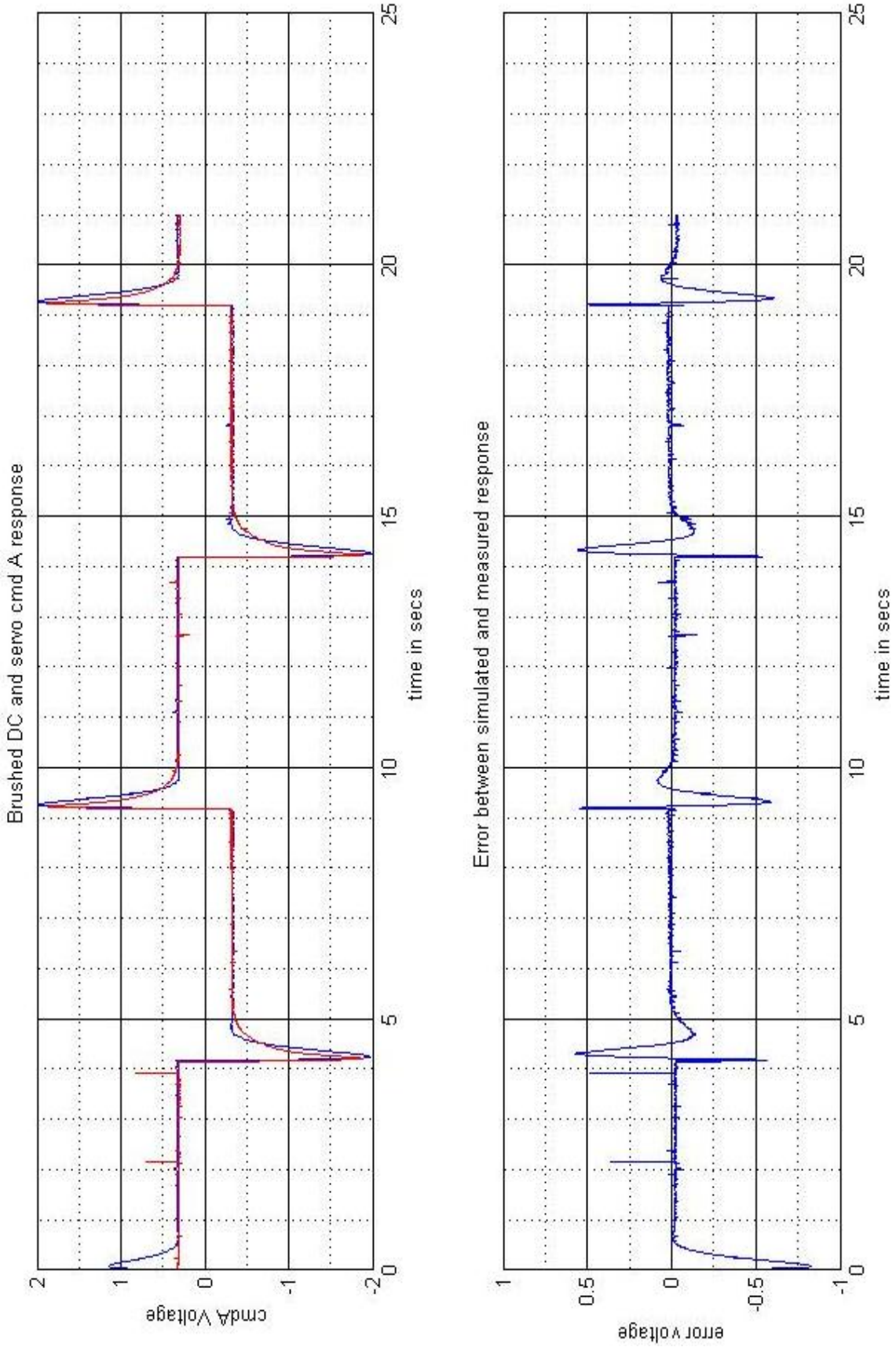

Figure 2.9: Brushed DC motor only velocity step response cmd A

Parameter	Value
R_1	24 $K\Omega$
R_2	24 $K\Omega$
R_3	24 $K\Omega$
R_4	20 $K\Omega$
C_4	0.22 μF
R_{25}	27.4 $K\Omega$
R_{26}	13.7 $K\Omega$
R_{27}	27.4 $K\Omega$
R_{28}	180 $K\Omega$
R_{66}	100 $K\Omega$
C_5	0.33 μF
R_{22}	100 $K\Omega$
R_{23}	110 $K\Omega$
R_{24}	10 $K\Omega$
C_5	0.01 μF
C_6	0.22 μF

Table 2.3: Servo loop parameters

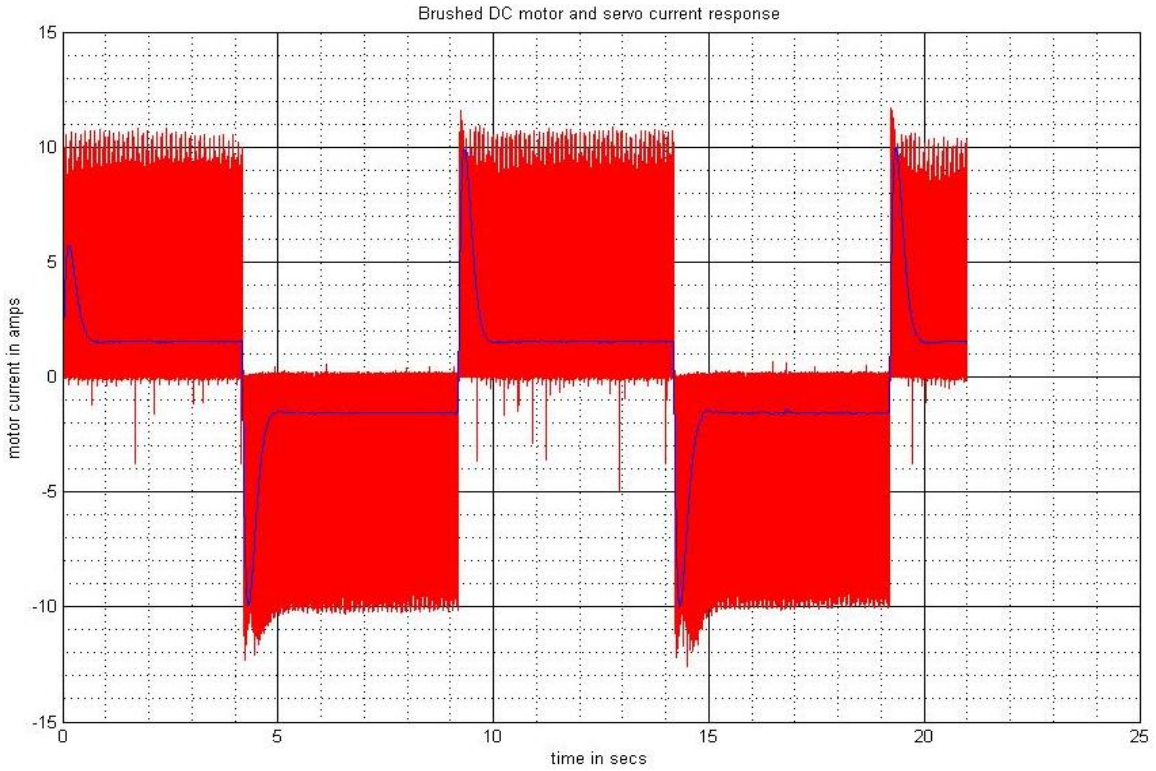

Figure 2.10: Brushed DC motor only velocity step response motor current

2.3 Brushed DC motor servo and GMRT mechanical model

In this section we discuss the simulation of brushed DC servo system with complete antenna mechanical model. In the simulation, in addition to time domain response, we also plot the bode plots of the system to understand the relative stability of the system. Similarly to measure the stability of the actual system frequency tests are carried on the antenna along with the velocity step response. The frequency tests give us the measure of system bandwidth and resonance frequency of the system. A more detailed frequency test is then proposed which can help us plot the bode plot of the system. This in effect will give us the stability margins of the system. The frequency tests of the actual system can also give us additional information about differences between the simulated and actual system. This information can further be used to refine the simulation programs.

2.3.1 Note on equations

System Matrices

To simulate the GMRT model it is necessary to find the equations governing the motion of the antenna mechanical structure. The complete description and derivation of these equations and resulting state matrices A,B,C,D can be found in [1]. Figure 2.11 shows the various parts of GMRT mechanical structure. For deriving these equations a lumped model of mechanical system is considered as shown in Figure 2.12. The state equations are derived using Lagrange energy equations.. Eqn 2.17, 2.18, 2.19, 2.20 give the state matrices A,B,C,B respectively for the mechanical system. Table 2.4 gives the explanation of various parameters used in state space matrices. This mechanical model is coupled with the equations for the servo controller as derived in section 2.2

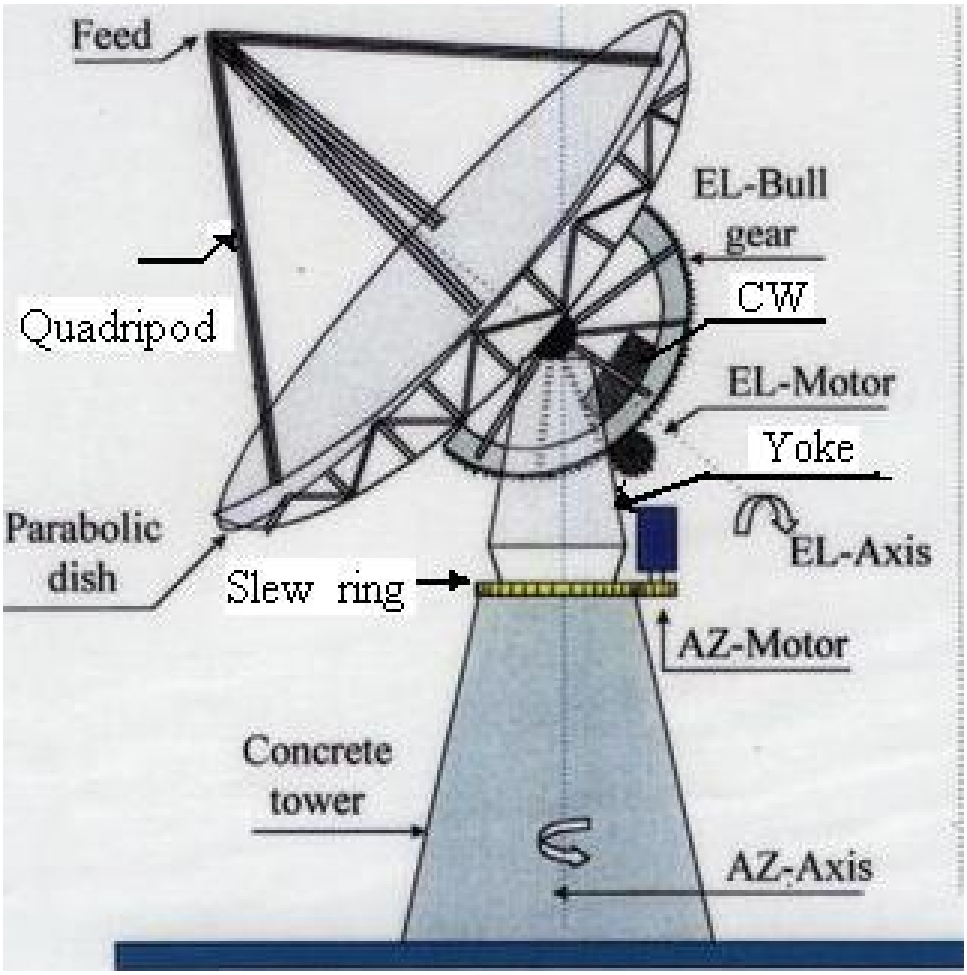

Figure 2.11: GMRT mechanical structure

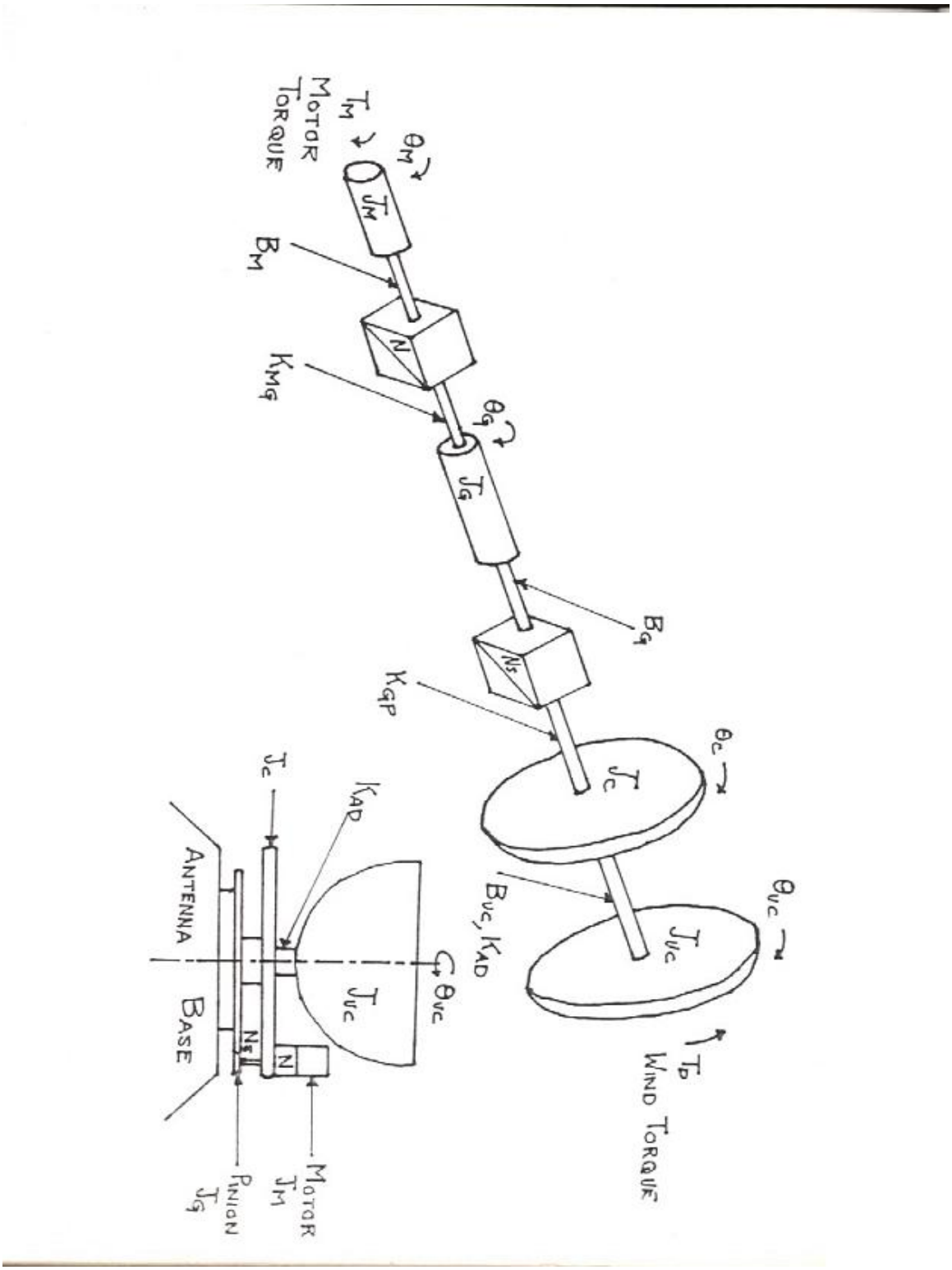

Figure 2.12: Lumped Model of GMRT mechanical system [1]

The state space equations for a system is given by [3]

$$\dot{X} = AX + Bu \tag{2.14}$$
$$Y = CX + Du \tag{2.15}$$

For present mechanical model, the state vector X is given as;

$$X = \begin{bmatrix} \theta_m \\ \dot{\theta}_m \\ \theta_g \\ \dot{\theta}_g \\ \theta_c \\ \dot{\theta}_c \\ \theta_{uc} \\ \dot{\theta}_{uc} \\ \dot{I}_a \end{bmatrix} \tag{2.16}$$

Where, θ_m = Motor angular position
$\dot{\theta}_m$ = Motor angular speed
θ_g = Gearbox output shaft position
$\dot{\theta}_g$ = Gearbox output shaft speed
θ_c = Controlled inertia position
$\dot{\theta}_c$ = Controlled inertia speed
θ_{uc} = Uncontrolled inertia position
$\dot{\theta}_{uc}$ = Uncontrolled inertia speed
I_a = Motor current The A, B, C, D matrices for present case is given by,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-K_{mg}}{J_m N^2} & \frac{-B_m}{J_m} & \frac{k_{mg}}{J_m N} & 0 & 0 & 0 & 0 & 0 & \frac{K_t}{J_m} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{K_{mg}}{J_g N} & 0 & \frac{-Kmg - K_{gp}/N_s^2}{J_g} & \frac{-B_g}{J_g} & \frac{K_{gp}}{J_g N_s} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{K_{gp}}{J_c N_s} & 0 & \frac{-(K_{gp}+K_{ad})}{J_c} & \frac{-(B_c+B_{uc})}{J_c} & \frac{K_{ad}}{J_c} & \frac{B_{uc}}{J_c} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{K_{ad}}{J_{uc}} & \frac{B_{uc}}{J_{uc}} & \frac{-K_{ad}}{J_{uc}} & \frac{-B_{uc}}{J_{uc}} & 0 \\ 0 & \frac{-K_b}{L_a} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-R_m}{L_a} \end{bmatrix} \tag{2.17}$$

$$B = \begin{bmatrix} 0 \\ \frac{E_a}{L_a} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.18}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.20}$$

Mechanical Parameters

The values for mechanical parameters have been taken from [1] and Tata Consulting Engineer's (TCE) design report. For complete details of mechanical system interested reader can refer to TCE's detailed design engineering notes. The four mechanical elements : Motor plus gearbox inertia J_m, Pinion inertia J_g, controlled inertia J_c, uncontrolled inertia J_{uc} and the associated viscous friction and spring constants are formed by clubbing together various mechanical elements of the dish. Table 2.4 give the details of the values and meaning of each parameter. The results of only azimuth axis simulation are presented. Similar scheme was be used for simulation of elevation axis.

Parameters	Mechanical elements	Value	Unit
Motor Inertia J_m	motor and gearbox inertia	0.066768	Kgm^2
Pinion Inertia J_g	Pinion	3.5	Kgm^2
Controlled Inertia J_c	Slew ring, yoke, cradle, and counter weight (CW), bullgear, reflector	$2 \times 0.995 \times 10^6$	Kgm^2
Uncontrolled Inertia J_{uc}	Quadripod and feed system	10.9×10^6	Kgm^2
Motor Viscous Friction B_m	-	.005424	Nm/rad/sec
Pinion Viscous friction B_g	-	4379.2	Nm/rad/sec
Controlled Inertia viscous friction B_c	-	7.8336×10^6	Nm/rad/sec
Uncontrolled Inertia viscous friction B_{uc}	-	1.34368×10^7	Nm/rad/sec
Motor to pinion spring constant K_{mg}	-	10^7	Nm/rad
Pinion to controlled inertia spring constant K_{gp}	stiffness of slew ring	5.0864×10^9	Nm/rad
Controlled inertia to uncontrolled inertia spring constant K_{ad}	Quadripod stiffness	1.428×10^9	Nm/rad
Slew ring gear Ratio N_s	-	12.6	-
Gear box gear ratio N	-	1488	-

Table 2.4: Azimuth axis mechanical parameters

2.3.2 Comparison of simulation results with test results

Bode plots

To understand the system bandwidth and system response bode plot of each loop was plotted. The 3 loops in GMRT servo system are cascade loops. In cascaded loops, tuning is started with the innermost loop and then the subsequent outer loops are tuned. If the inner loop bandwidth is much higher than outer loop bandwidth then the inner loop acts like a DC gain for the outer loop. Hence the tuning of outer loop does not affect the tuning of inner loop.

For faster error correction it is necessary that the innermost loop i.e current loop must have the highest bandwidth. Its bandwidth should be 5 to 10 times higher than the velocity loop bandwidth. Similarly the velocity loop bandwidth should be 5-10 times higher than position loop bandwidth.

The bandwidth for the loops is decided by constraint on position loop bandwidth. The first resonance frequency of the mechanical structure is between 1Hz-2Hz. To avoid the excitation of system at resonance frequency it is necessary to attenuate the system gain at resonance frequency. Hence the bandwidth of the position loop should below 1 Hz. On the other hand, to quickly correct the positioning errors induced by wind disturbance torques the system must have as fast response as possible hence a high position loop bandwidth is required. Because of these two conflicting factors the position loop bandwidth is designed around 0.5 Hz. Therefore the velocity loop bandwidth should be around 5 Hz. Current loop bandwidth is much higher that velocity loop bandwidth and is decided by the industrial amplifier circuit which also has current loop PI.

Figure 2.13, 2.14 and 2.15 show the bode plots for uncompensated and compensated current loop. Figure 2.16, and 2.17 show the bode plots for uncompensated and compensated velocity loop. In the following bode plots the steady state closed loop gain may vary from 0dB depending on input/output scaling factors. Bandwidth is measured as frequency at a gain -3dB below zero frequency gain of closed loop bode plot.

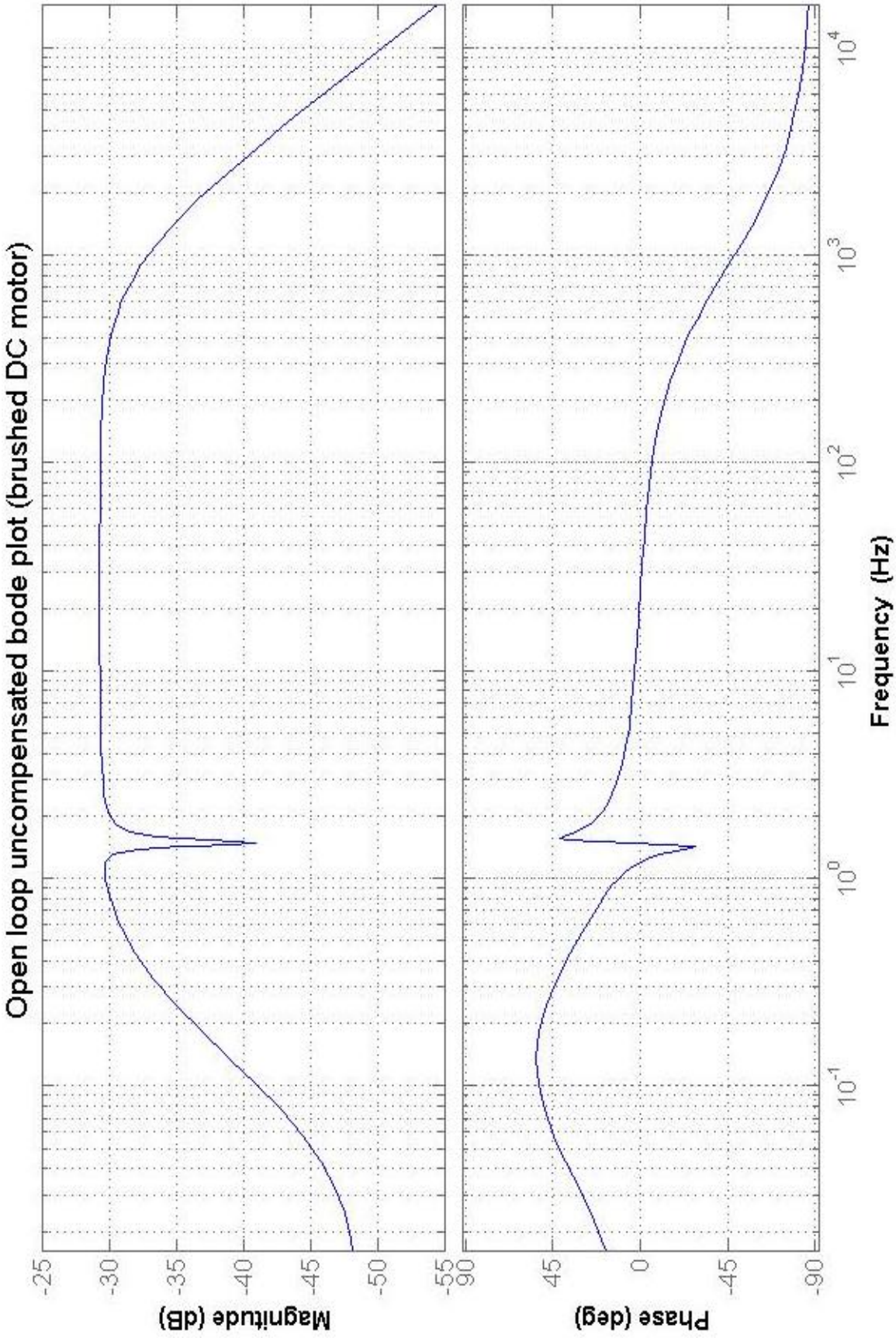

Figure 2.13: Bode plot: Uncompensated open current loop

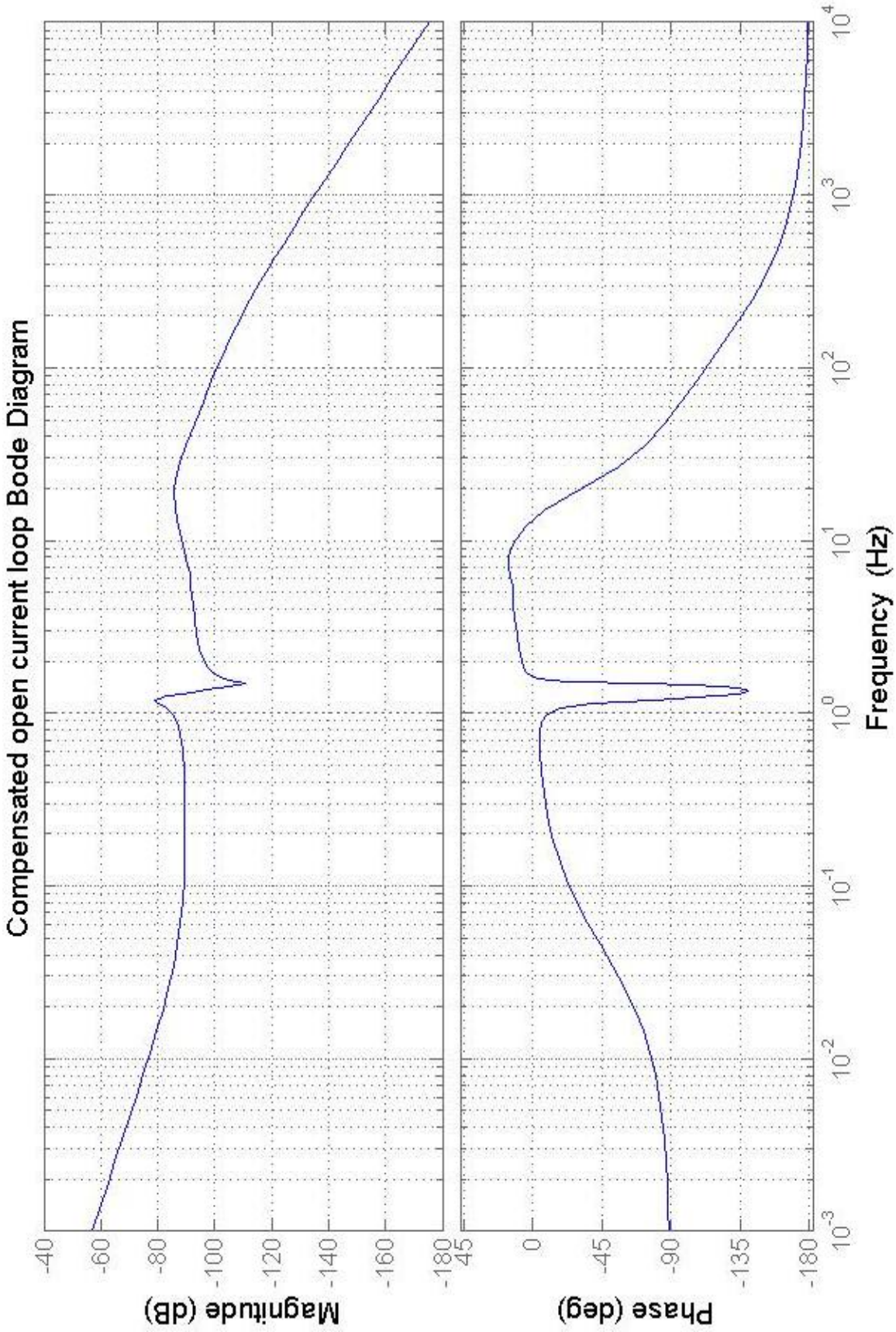

Figure 2.14: Bode plot: Compensated open current loop

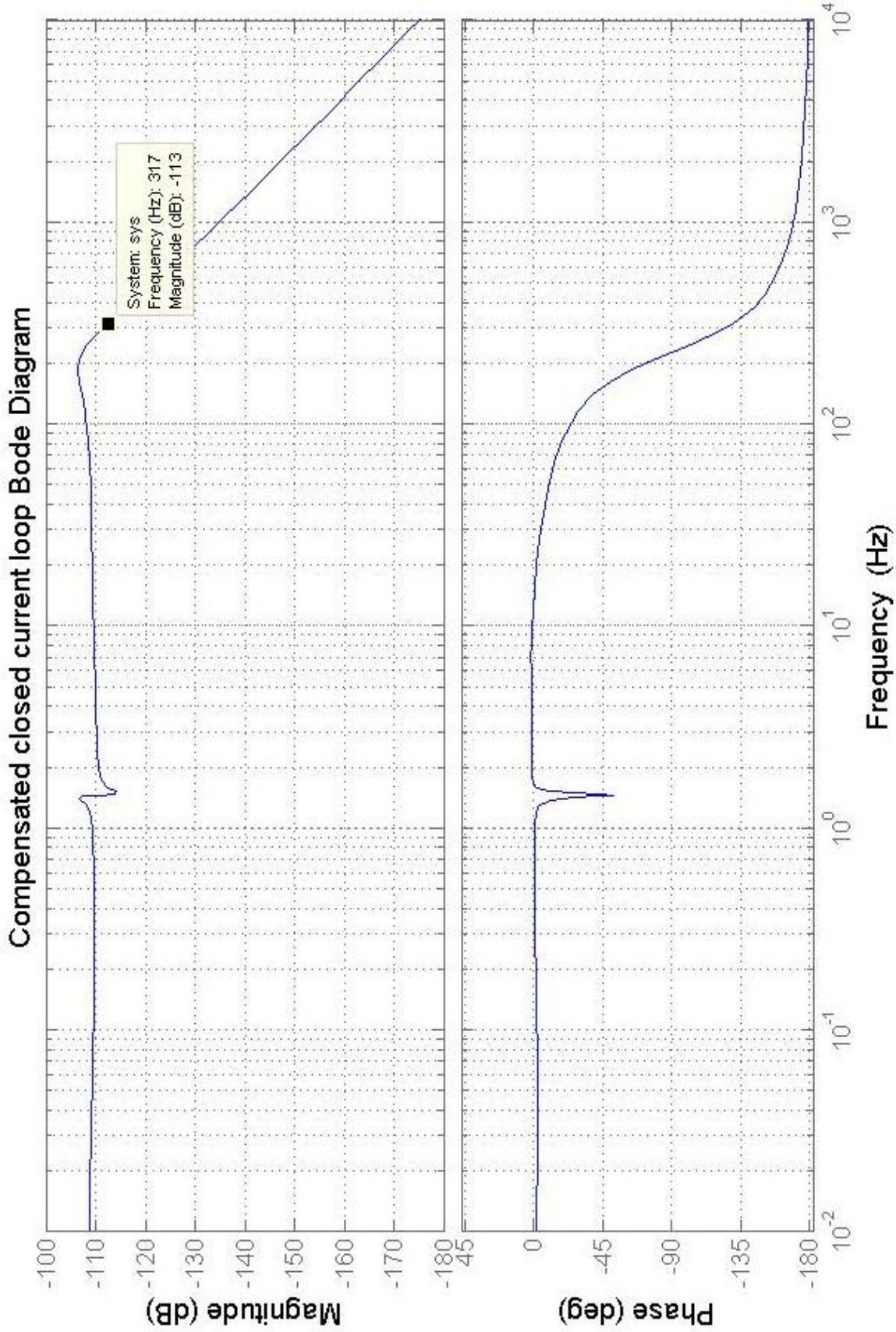

Figure 2.15: Bode plot: Compensated open current loop

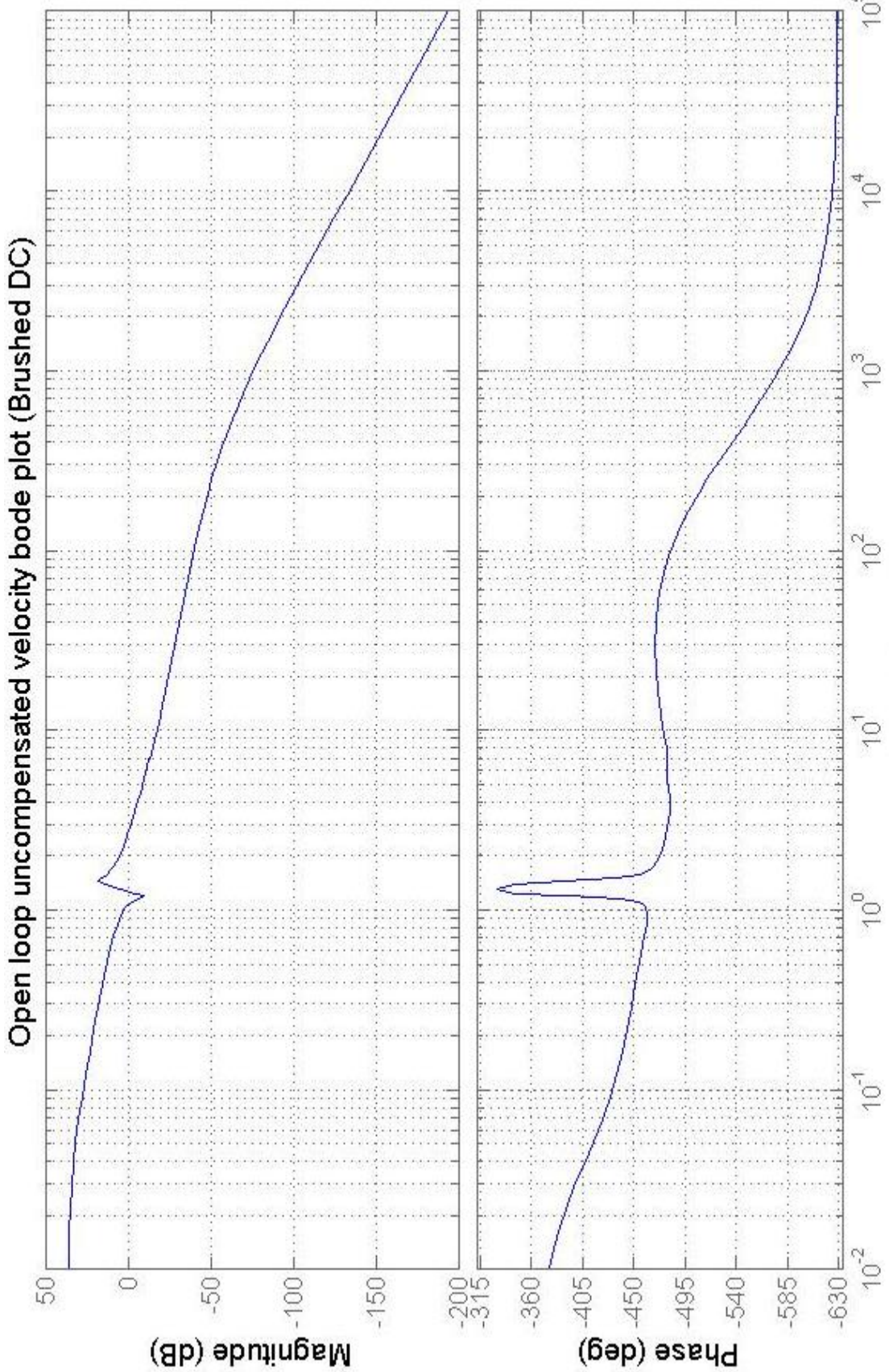

Figure 2.16: Bode plot: Uncompensated open velocity loop

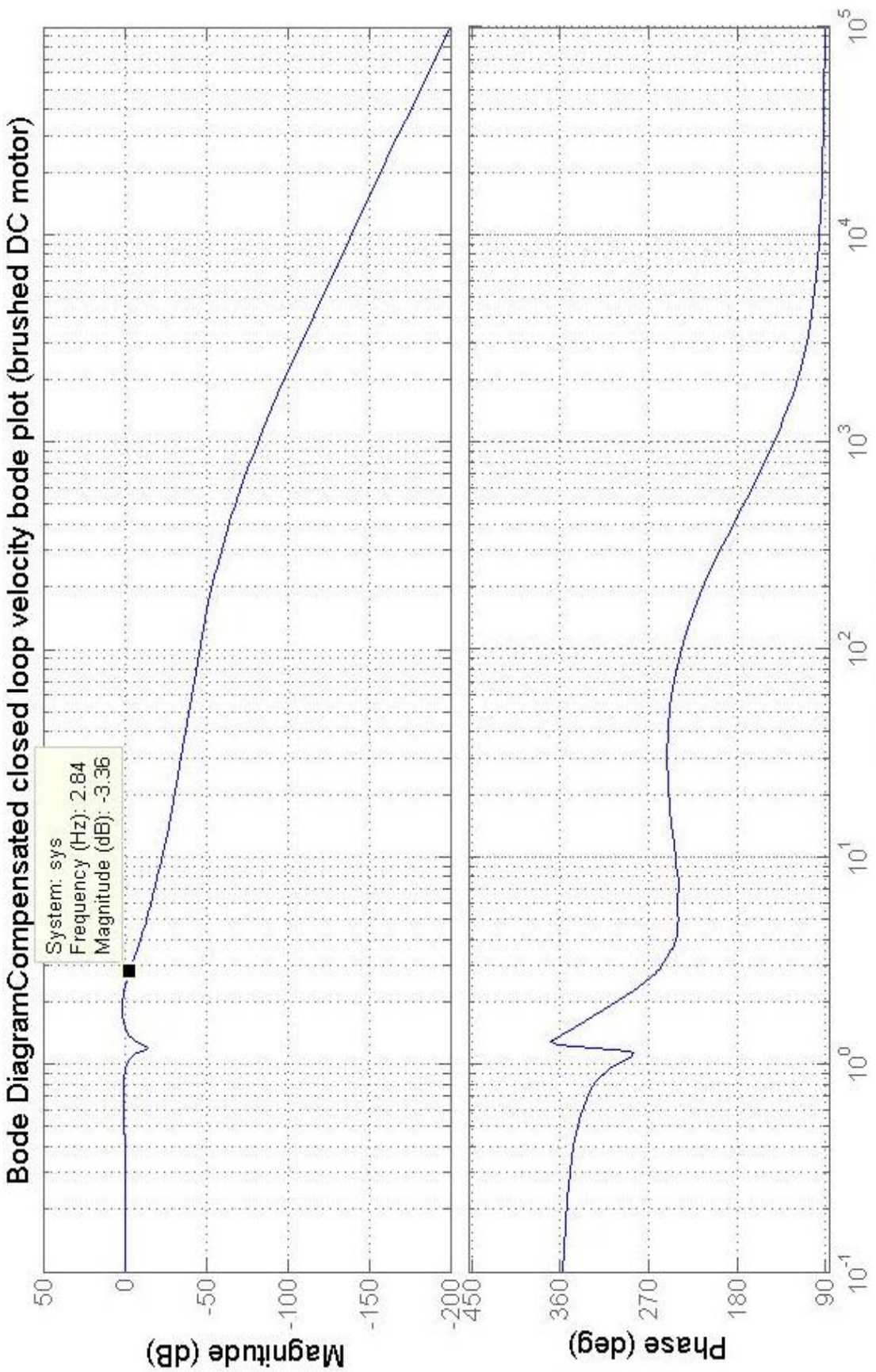

Figure 2.17: Bode plot: Compensated closed velocity loop

Frequency Tests on antenna

To measure the frequency response of the system, the velocity loop is excited with a sine sweep signal. Sine sweep signal is given to test input and the output is measured at Tacho A and Tacho B in counter torque card. The sweep is a logarithmic sweep with formula,

$$f(t) = f_0 \beta^t \tag{2.21}$$

$$\beta = \left(\frac{f_T}{f_0}\right)^{\frac{1}{T}} \tag{2.22}$$

where, f_0 = initial frequency
f_T = final frequency
T = Time of final frequency
Figure 2.18 and 2.20 shows the sine sweep response of the antenna for azimuth and elevation axis respectively. It can be seen that the system has an anti-resonance frequency at about 1Hz and resonance frequency at about 1.2 Hz. The system bandwidth is about 3.24 Hz for azimuth axis. Figure 2.19 gives the simulated sine weep response. The comparison of figure 2.18 and 2.19 gives the differences in simulated and actual response of the system at each frequency. Figure 2.21 shows the bode plot of current loop as given in Industrial amplifier manual. Table 2.5 gives the bandwidth for measured and simulated, current and velocity loop.

Loop	Simulated Bandwidth	Measured Bandwidth
Current loop	333Hz	10 Hz
Velocity loop	2.68 Hz	3.24 Hz

Table 2.5: Simulated and measured loop bandwidths for brushed DC motor and its servo system

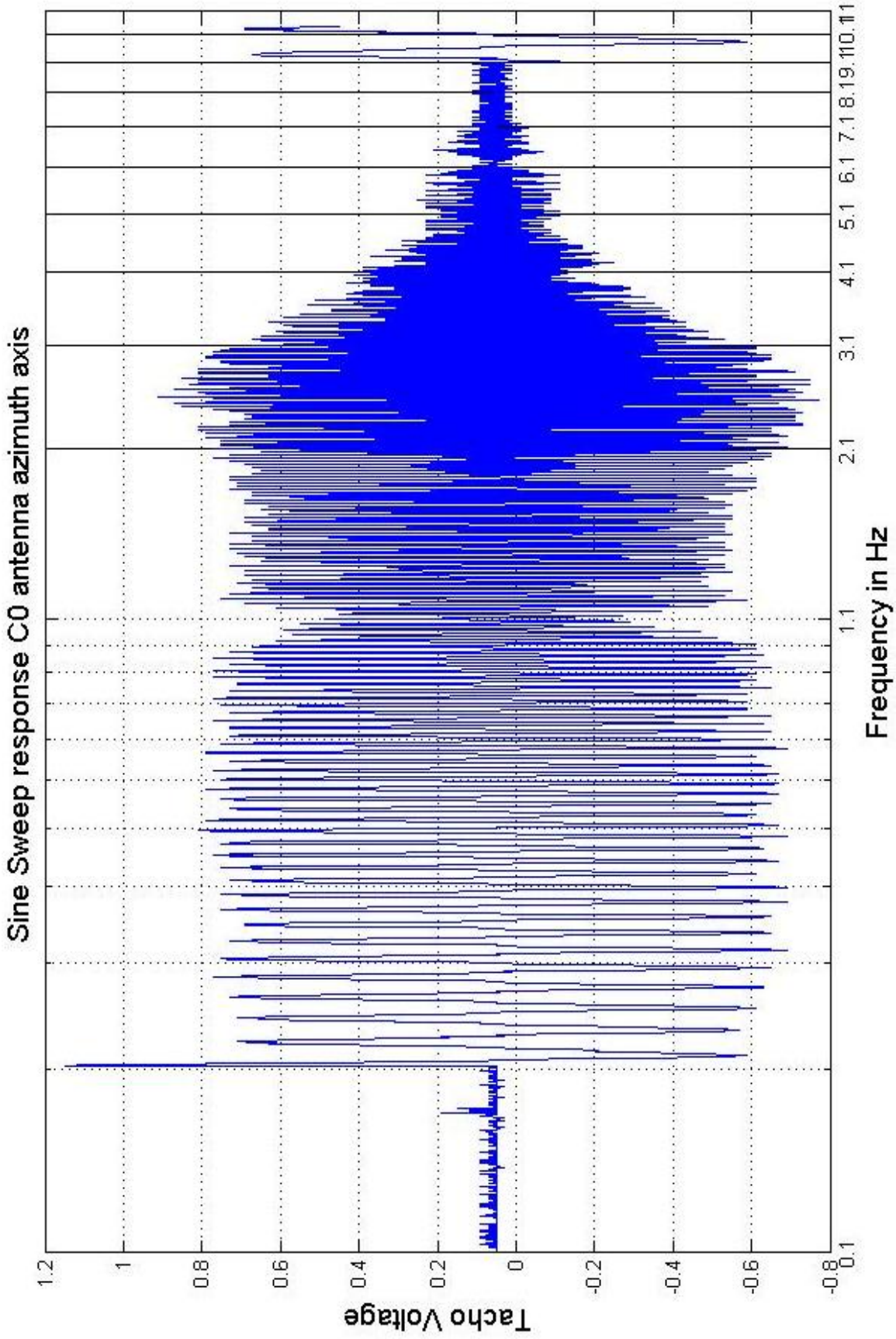

Figure 2.18: Response of C00 antenna azimuth axis to sine sweep velocity input

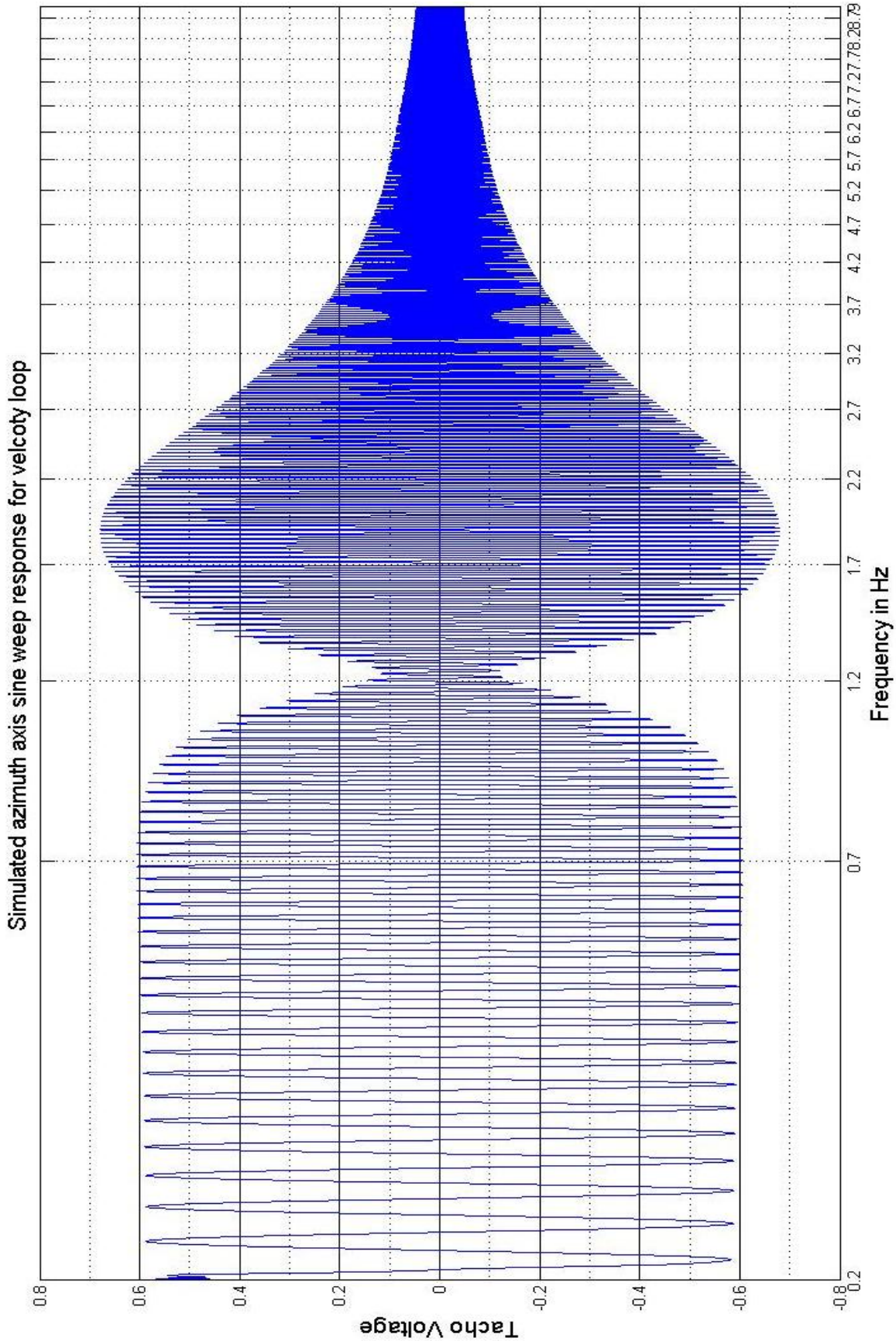

Figure 2.19: Simulated sine sweep response for azimuth axis

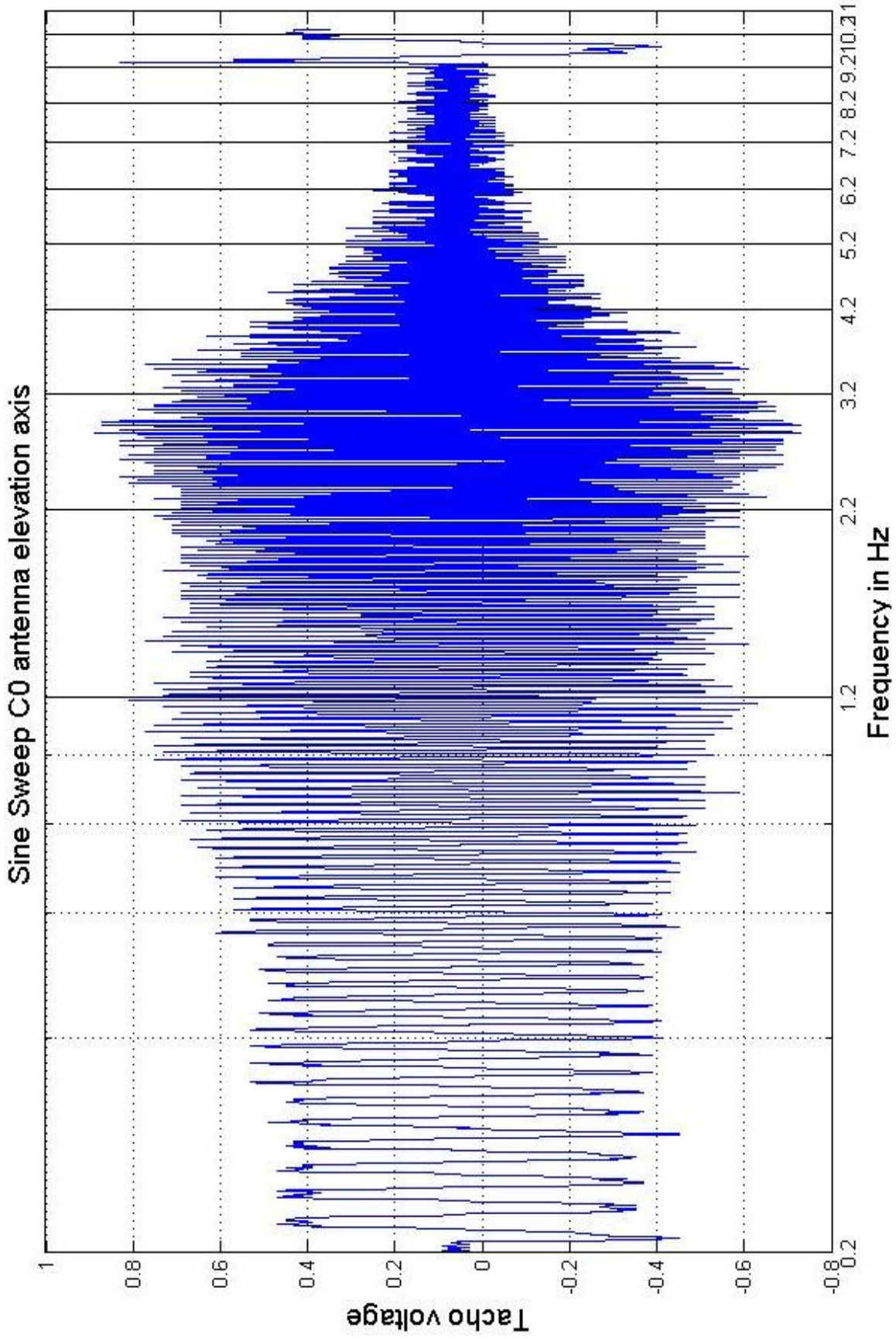

Figure 2.20: Response of C0 antenna elevation axis to sine sweep velocity input

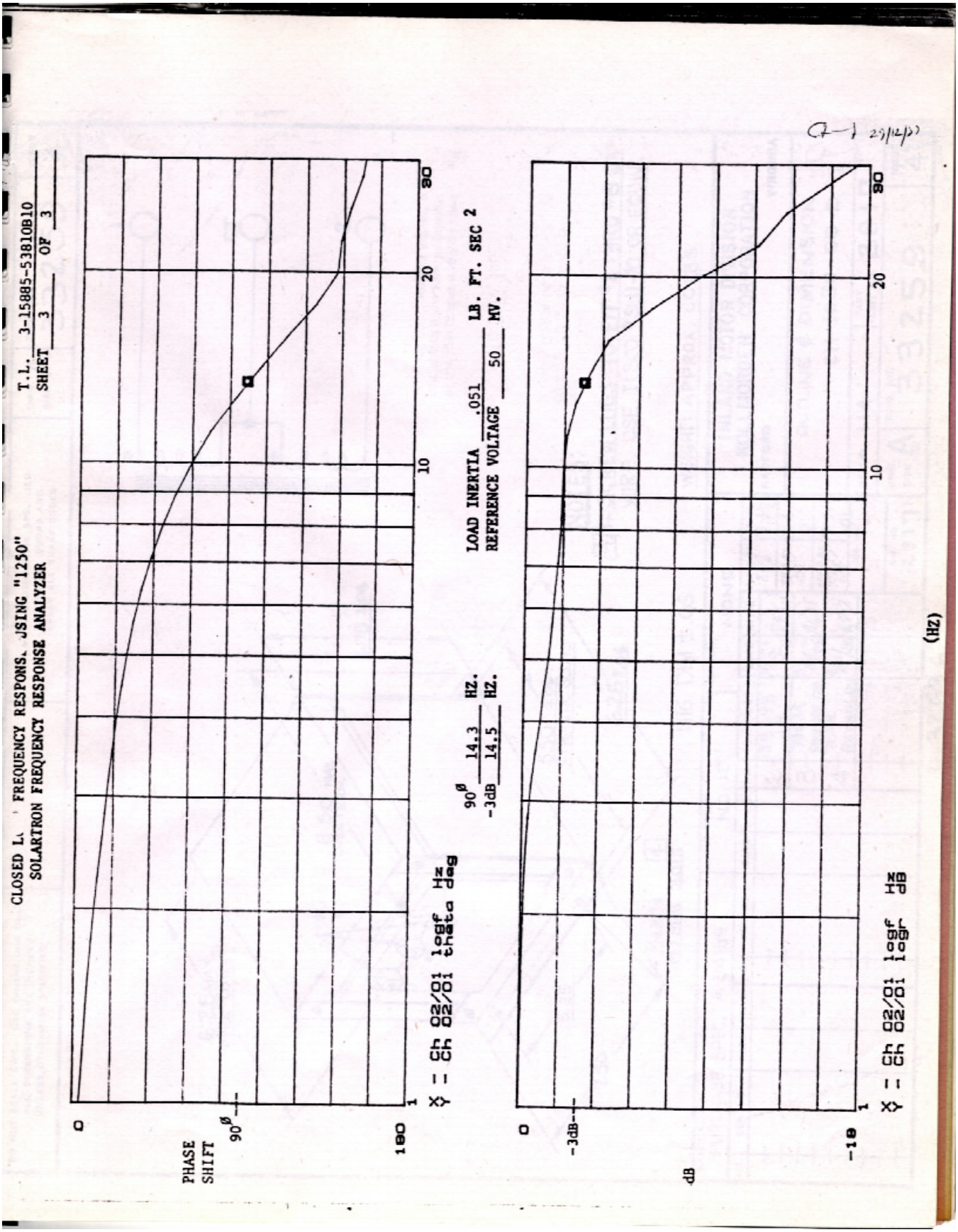

Figure 2.21: Measured current loop bode plot

The the measurement of the sine sweep response of the system gives us information about only the magnitude response of the system. It is difficult to extract phase response information of the system from sine sweep response. Phase response is necessary to calculate the the relative stability of the system in terms of its gain margin and phase margin. To overcome this problem we can excite the antenna with a range of singlr frequency sine signals. Here the antenna is excited with a sine signal of single frequency at a time and its response (which will be the sine wave of same frequency but with different magnitude and shifted in phase) is recorded. By computing the Fast Fourier Transform (FFT) of the input signal and the output signal we can get the information of system's magnitude gain and phase shift for that particular frequency. By plotting the magnitude gain and phase response of the system at all the frequency which were used to excite the system, the system's bode plot can be plotted. Figure 2.22 and 2.23 show the sine sweep response and measured bode plot for C02 azimuth axis velocity loop. It can be seen that the magnitude response of bode plot matches with the sine sweep response. Similarly, figure 2.24 and 2.25 show the sine sweep response and measured bode plot for elevation axis. A good match in magnitude response can be found in this case too. The matlab m-code for computing FFT of input and output signal, and generating bode plot can be found in Appendix A.

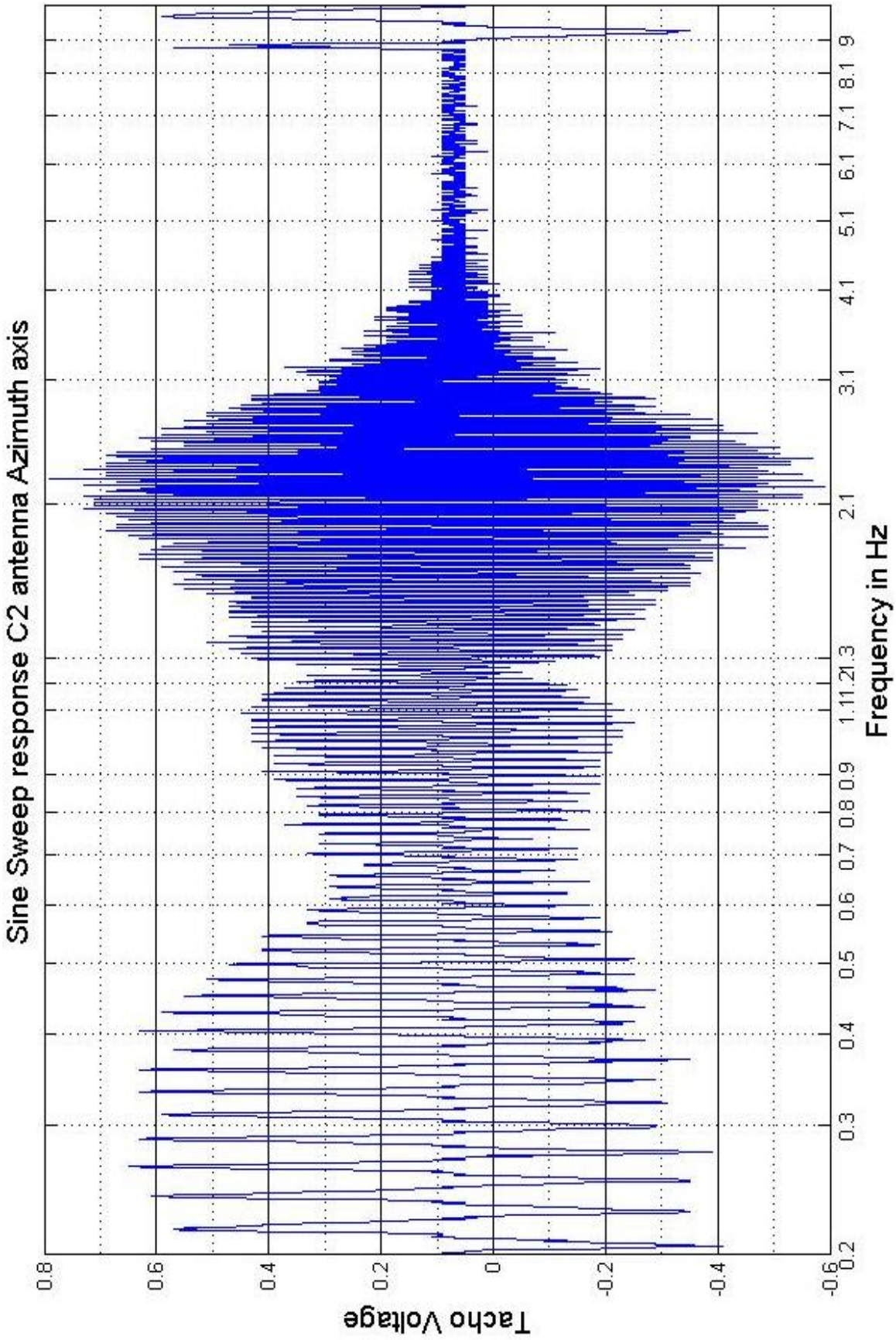

Figure 2.22: Sine sweep response for C2 azimuth axis

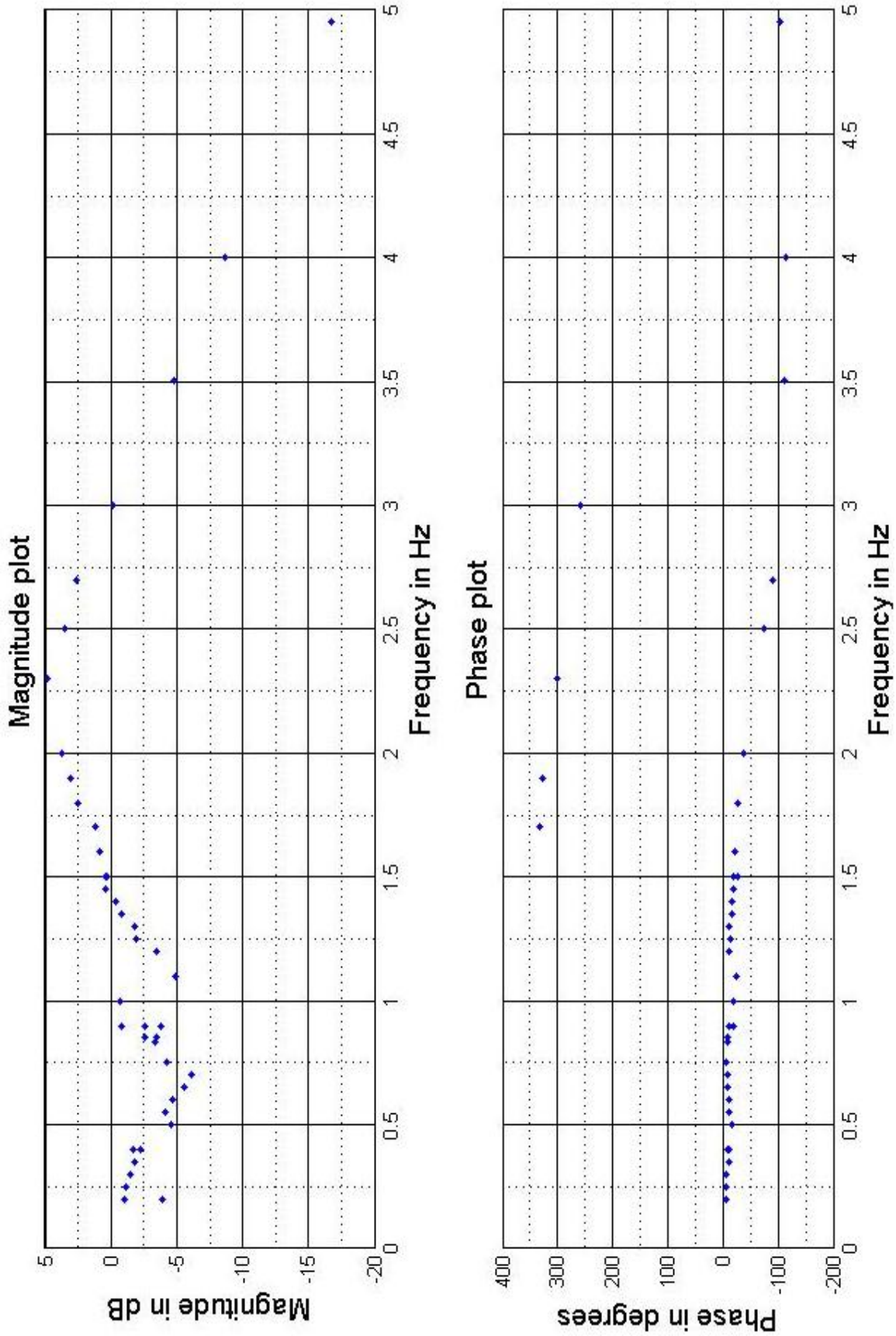

Figure 2.23: Experimentally measured bode plot for C02 azimuth axis

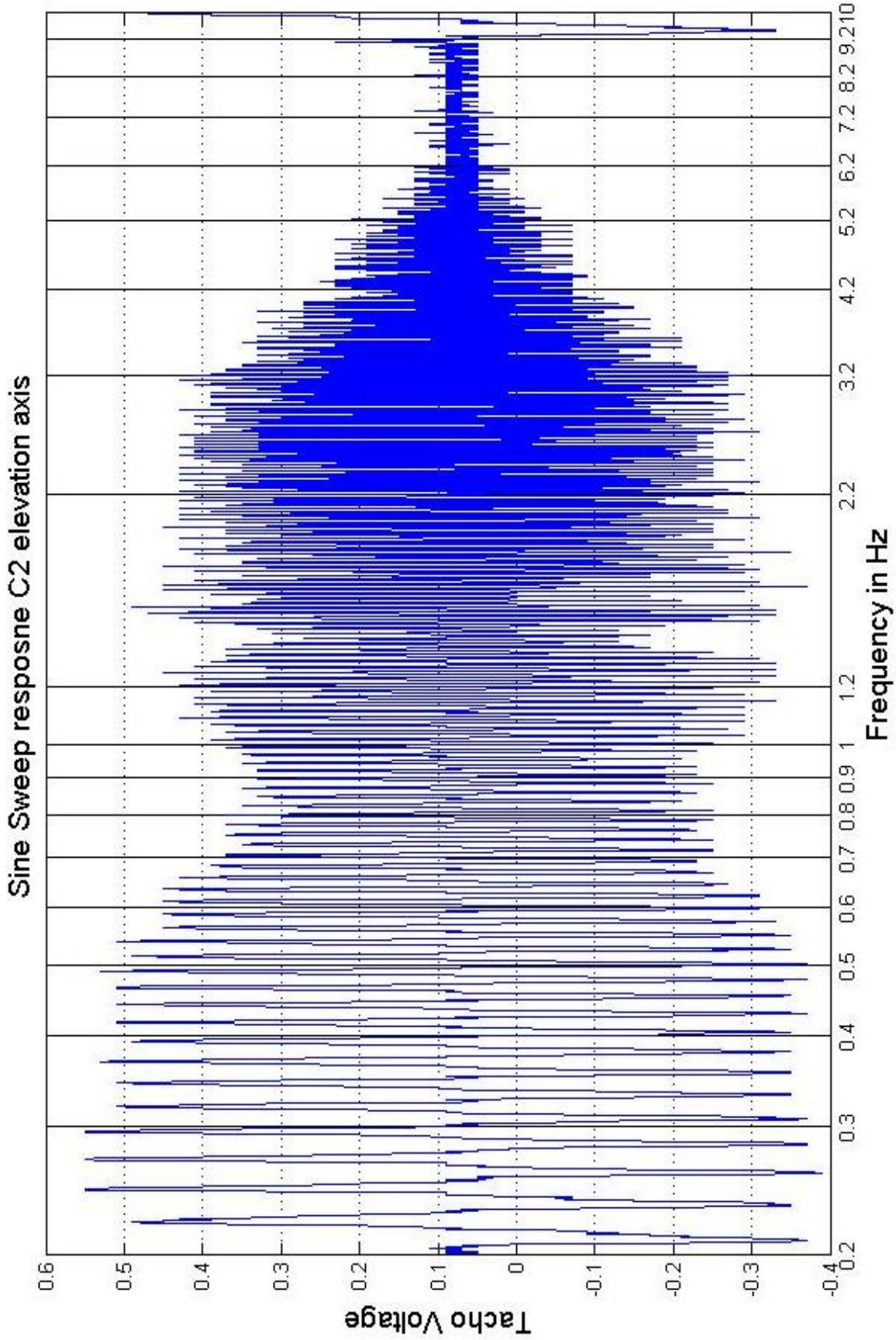

Figure 2.24: Sine sweep response for C02 elevation axis

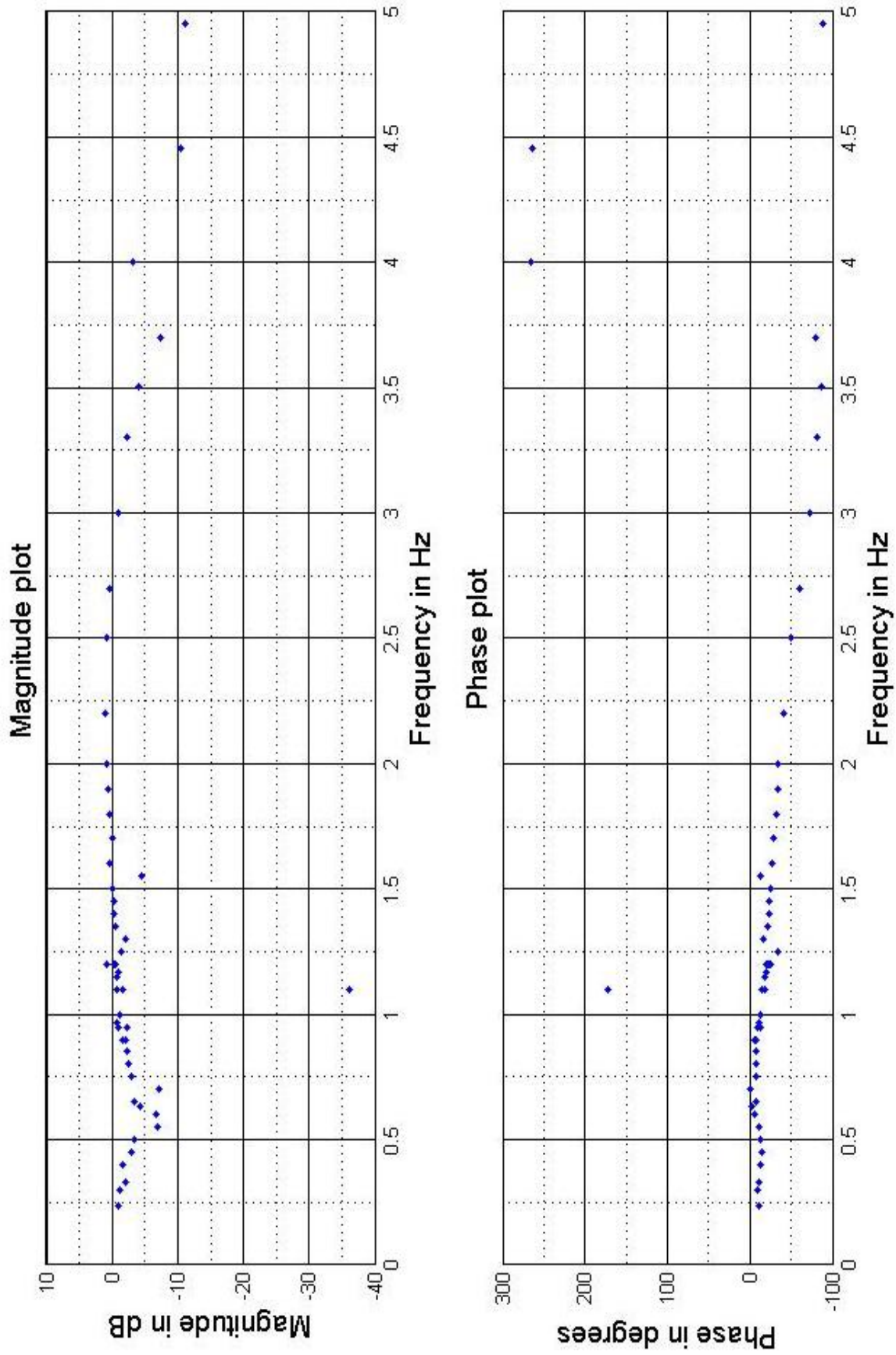

Figure 2.25: Experimentally measured bode plot for C02 elevation axis

Comparison of step responses

The velocity step response of the antenna is carried out in similar fashion as described in section 2.2. The graph of the velocity step is shown in Figure 2.26. Figure 2.27 and 2.28 show the cmd A response and motor current response respectively. Figure A.3 shows the simulink model of brushed DC motor servo system with GMRT mechanical model. Simulated step response of position loop is shown in figure 2.29. Figure 2.30 shows position ramp response. The present servo system does not have facility to measure system position step, ramp or frequency response. Hence no comparison between simulated and measured data has been presented. The ramp response shows a steady state tracking error of $\approx 5''$. This error is below the $10''$resolution of encoder and hence acceptable. The step response shows a 44%. This is slightly above the specified overshoot of 40%. The step response for the system shows 64 sec settling time. This is well beyond the specified settling time of 10 secs for the system. The simulated response needs to be verified with actual step response of the system. According to the measured response corrective steps must be taken to improve servo loop tuning.

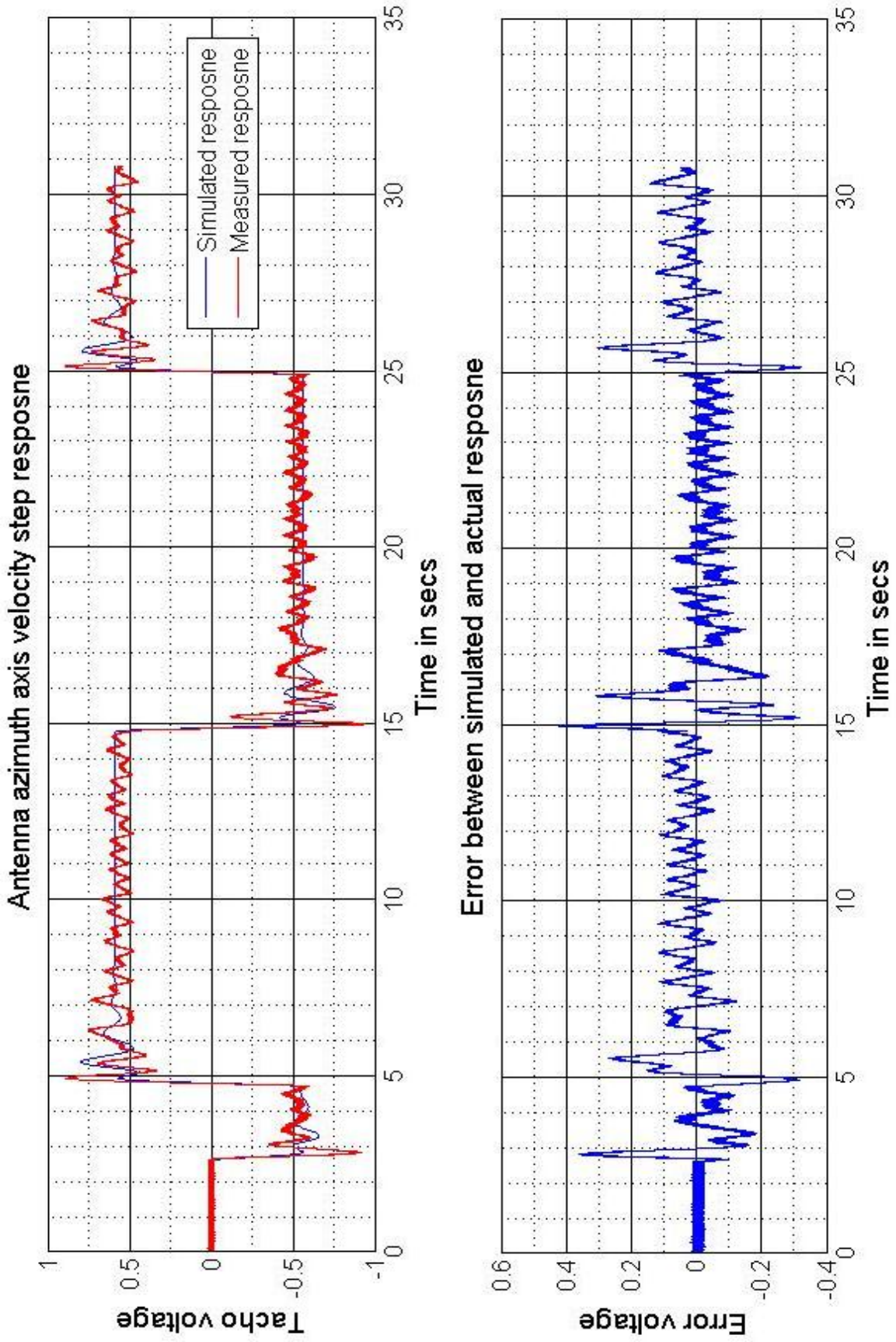

Figure 2.26: Velocity step response of antenna for brushed DC motor servo system

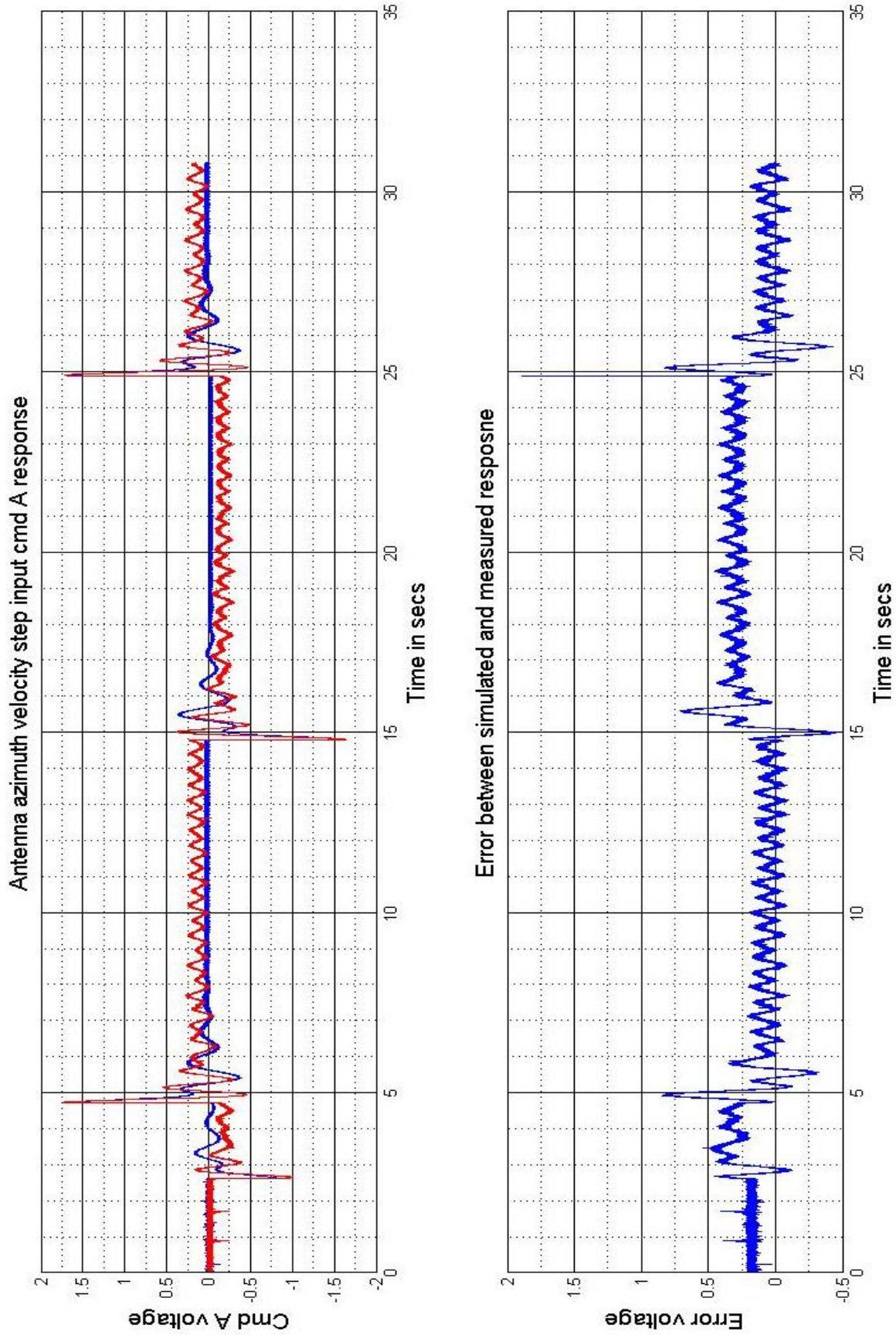

Figure 2.27: Simulated versus measured response of cmdA for velocity step response of antenna

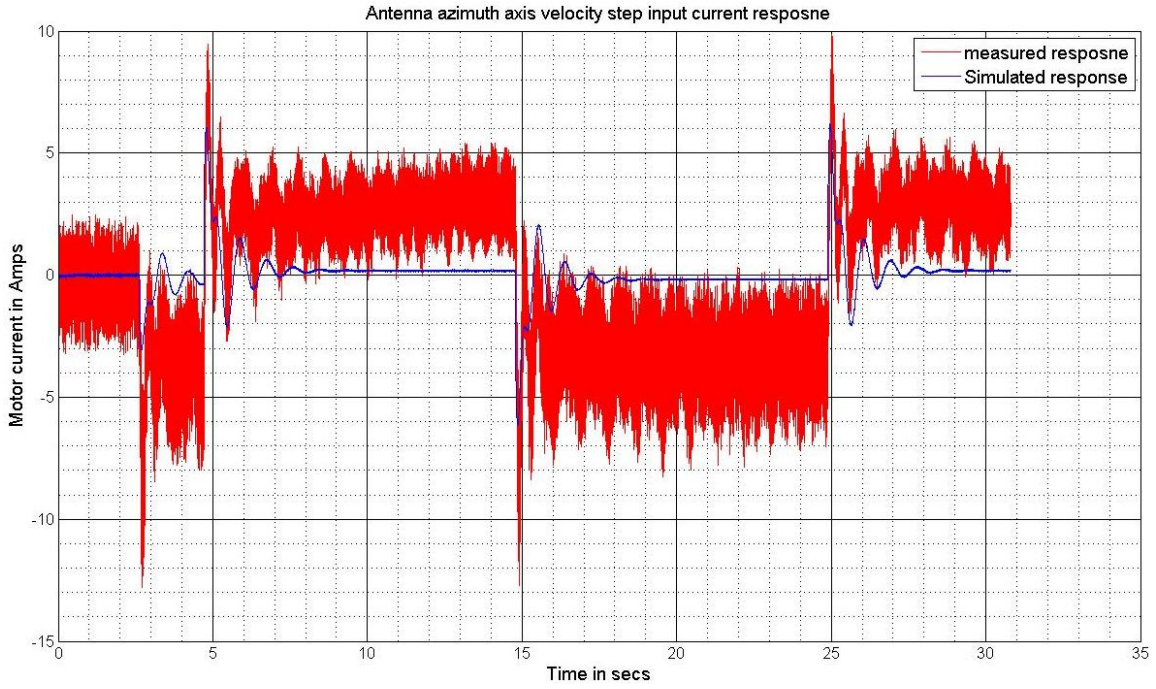

Figure 2.28: Simulated versus measured response of motor current for velocity step response of antenna

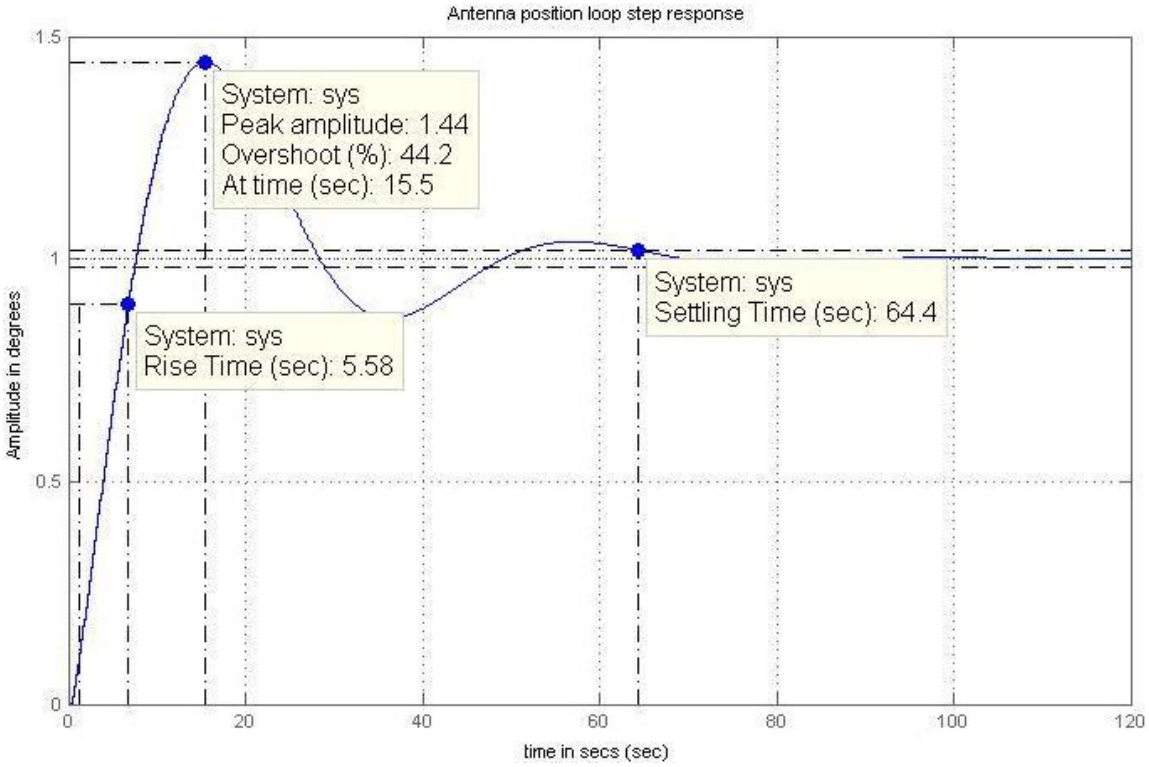

Figure 2.29: Position loop step response

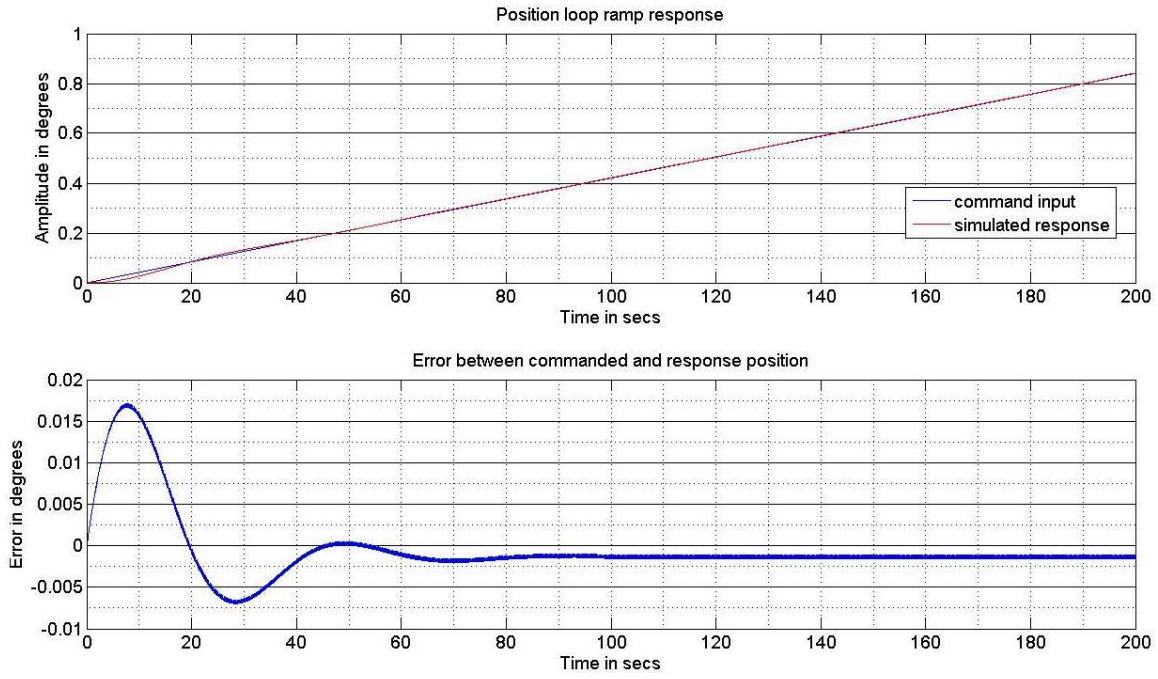

Figure 2.30: Position loop ramp response

2.4 Conclusions

A few immediate observations can be made from the simulated versus measured output. Though the steady state state response of the measured and simulated output match, there are differences in transient response. In frequency domain this is seen as difference in sine sweep response after the anti resonance frequency. For current loop the bandwidth given in industrial amplifier manual is about 10 Hz as against $\approx$ 300Hz in simulated response. On the other hand the measured bandwidth for C2 antenna azimuth axis velocity loop is about 3.24 hz as against 2.68 Hz of simulated response. Once these discrepancies in higher frequency response have been removed, we can expect the transient response of the system to match.

The estimated resistance of brushed DC motor are very different from those mentioned in motor specification sheet. It was also seen that resistance varies largely from motor to motor. The bandwidth of current loop is very sensitive to resistance value of motor. For velocity loop the response of of system after anti-resonance is found sensitive to resistance value. The parameters of motors actually connected to the antenna cannot be measured. So in simulation we take estimated parameters of motors in lab. If the same motors were to be used in antenna while measuring its step response better simulation response can be obtained. The motors used in antenna are very old. Due to regular servicing of motor its torque constant may change over a period of time. Hence it is necessary to verify the torque constant of the motor K_t using a torque transducer.

The sine sweep response (in terms of resonance, anti-resonance frequency and system bandwidth) and the velocity step response vary from antenna to antenna. The main reason for this can be ascertained to the differences in parameters of mechanical elements. It is a Known fact

that mass of counter weight varies from antenna to antenaa. The anti resonance frequency is found sensitive to uncontrolled inertia and axis complaince spring constant parameter. A method to estimate these parameters needs to be established so that they can be estimated for each antenna separately and the simulation values can be changed accordingly.

An attempt was made to generate a complete bode plot for C02 antenna azimuth and elevation axis. The bode plots so generated are plotted in figure 2.23 and 2.25. It can be seen that there is some erratic change in phase response of the system (both in az and el axis) which is not be possible in actual antenna. The m-code program was cross-checked with simulated data. In case of simulated data the program gave accurate output. This proves that the m-code is correct. Hence the reason for this sudden change in phases of the system response maybe due to some noise corrupting the acquired data. The accurate reason must be determined and the test should be performed again to get an accurate bode plot. It is also seen that the phase plot does not cross the 180 degree line upto 5hz. Hence gain margin of the system cannot be determined. In next test we should check response even beyond 5 Hz till the phase crosses 180 degree line or till we establish that the system reaches 180 degree asymptotically.

Chapter 3

Design of optimal controllers

3.1 Introduction

As discussed in the introduction of this report, one of the aims of simulation is to design optimal controller for the GMRT servo system. As discussed in [5] optimal controllers of linear Quadratic Gaussian (LQG) type improve the pointing accuracy of telescope in presence of wind disturbances. As shown in [5] there are 3 configurations for implementing optimal controller for a servo system. We opt for the first configuration. Here the position loop is replaced by an optimal controller. Out of the 3 choices this may not be the best choice in terms of improving pointing accuracy, but this is the easiest to implement.The optimal control problem is discussed for brushed DC motor servo system. The simulation is on the lines of optimal control simulations for GMRT servo system as discussed in [1]. For in detail discussion of optimal control problem for GMRT system interested reader can refer to [1]. For theory of optimal controllers reader can refer to [6] and [7]. Here too we first simulate a linear quadratic regulator (LQR) for GMRT servo system. Since in case of GMRT it is important to track a source in sky, we progress to simulation of linear quadratic tracking regulator (LQTR). Finally, considering the fact that all states of system are not available for measurement on the antenna we design linear Gaussian estimator to estimate the system states and then combine it with linear quadratic servo regulator.

3.2 Derivation of state space equations for present servo system

Optimal controllers are designed as state feedback controllers. To design state feedback controllers it is necessary that the complete system model is expressed as a state space model. For optimal controller the system consists of plant along with current and velocity loop. In the design it is decided to replace the position controller with optimal controller. Hence it is necessary to convert the transfer functions as described in eqn 2.6, 2.11 into their state matrices and augment them with plant matrices as described by eqn 2.17, 2.18, 2.19 and 2.20. It is seen that current loop dymanics are much faster than rest of the system. Hence for the simplicity of implementation current loop is considered as a constant gain. The current loop is replaced by an equivalent DC gain of 16amp/Volt of velocity loop output. Figure 3.1 shows the state space representation of the complete system. Matrices A_2 , B_2, C_2, P, and Cr correspond to the

plant. Matrices A_1 , B_1, C_1, D_1 velocity lag compensator and A_3 , B_3, C_3, D_3 correspond to velocity lead compensator. The matrices for lead lag compensator are derived by representing velocity transfer function in observer canonical form. The derivation for observer canonical form is as follows,
Consider eqn 2.7

$$G_{s1}(s) = \frac{[R_4 + R_4(R_1 + R_2)C_4 s]TachoA(s)}{(R_1 + R_3 + R_4) + [R_1R_2 + (R_1 + R_2)(R_3 + R_4)]C_4 s}$$

Let,

$$\begin{aligned} Z_{w0} &= \frac{R_4(R_1 + R_2)}{[R_1R_2 + (R_1 + R_2)(R_3 + R_4)]} \\ Z_{w1} &= \frac{R_4}{[R_1R_2 + (R_1 + R_2)(R_3 + R_4)]C_4} \\ P_{w1} &= \frac{(R_1 + R_3 + R_4)}{[R_1R_2 + (R_1 + R_2)(R_3 + R_4)]C_4} \end{aligned}$$

Therefore,

$$G_{s1}(s) = \frac{Z_{w0}s + Z_{w1}}{s + P_{w1}} \tag{3.1}$$

$$A_3 = [-P_{w1}] \tag{3.2}$$

$$B_3 = [Z_{w1} - Z_{w0}P_{w1}] \tag{3.3}$$

$$C_3 = [1] \tag{3.4}$$

$$D_3 = [Z_{w0}] \tag{3.5}$$

Similarly considering eqn 2.9 we define,

$$\begin{aligned} P_{z1} &= \frac{1}{(R_{66} + R_{28})C_5 + 1} \\ Z_{z0} &= \frac{R_{28}R_{66}}{(R_{66} + R_{28})C_5 + 1} \\ Z_{z1} &= \frac{R_{66}}{(R_{66} + R_{28})C_5 + 1} \end{aligned}$$

Hence,

$$G_f(s) = \frac{Z_{z0}s + Z_{z1}}{s + P_{z1}} \tag{3.6}$$

$$A_1 = [-P_{z1}] \tag{3.7}$$

$$B_1 = [Z_{z1} - Z_{z0}P_{z1}] \tag{3.8}$$

$$C_1 = [1] \tag{3.9}$$

$$D_1 = [Z_{z0}] \tag{3.10}$$

The current loop along with electrical part of motor are considered as a gain K_f hence the matrices A_2, B_2 and C_2 represent the mechanical part of the plant and are given as follows,

$$A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-K_{mg}}{J_m N^2} & \frac{-B_m}{J_m} & \frac{k_{mg}}{J_m N} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{K_{mg}}{J_g N} & 0 & \frac{-Kmg - K_{gp}/N_s^2}{J_g} & \frac{-B_g}{J_g} & \frac{K_{gp}}{J_g N_s} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{K_{gp}}{J_c N_s} & 0 & \frac{-(K_{gp}+K_{ad})}{J_c} & \frac{-(B_c+B_{uc})}{J_c} & \frac{K_{ad}}{J_c} & \frac{B_{uc}}{J_c} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{K_{ad}}{J_{uc}} & \frac{B_{uc}}{J_{uc}} & \frac{-K_{ad}}{J_{uc}} & \frac{-B_{uc}}{J_{uc}} \end{bmatrix} \tag{3.11}$$

$$B_2 = \begin{bmatrix} 0 \\ \frac{K_t}{J_m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.12}$$

$$C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.13}$$

$$K = \frac{2K_{tacho}}{R_{25}} \tag{3.14}$$

$$N = \frac{1}{R_{26}} \tag{3.15}$$

The matrices can be augmented to give complete system matrices as follows (refer figure 3.1),

$$\begin{aligned} \dot{X} &= A_2 X + B_2 U_2 \\ Y &= C_2 X \\ U_2 &= C_1 Z + D_1 U_1 \end{aligned}$$

$$
\begin{aligned}
\dot{Z} &= A_1 Z + B_1 U_1 \\
U_1 &= R\,N - F \\
F &= C_3 W + D_3 K U_3 \\
U_3 &= C_2 X \\
\dot{W} &= A_3 W + B_3 K C_2 X \\
F &= C_3 W + D_3 K C_2 X \\
U_1 &= R\,N - C_3 W - D_3 K C_2 X \\
\dot{Z} &= A_1 Z + B_1 [R\,N - C_3 W - D_3 K C_2 X] \\
\dot{Z} &= -B_1 D_3 K C_2 X + A_1 Z - B_1 C_3 W + R\,N B_1 \\
U_2 &= C_1 Z + D_1 [R\,N - C_3 W - D_3 K C_2 X] \\
U_2 &= -D_1 D_3 C_2 K X + C_1 Z - D_1 C_3 W + D_1 R\,N \\
\dot{X} &= A_2 X + B_2 [-D_1 D_3 C_2 K X + C_1 Z - D_1 C_3 W + D_1 R\,N] \\
\dot{X} &= A_2 X - B_2 D_1 D_3 C_2 X + B_2 C_1 Z - B_2 D_1 C_3 W + B_2 D_1 R\,N \\
\dot{X} &= [A_2 - B_2 D_1 D_3 C_2 K] X + B_2 C_1 Z - B_2 D_1 C_3 W + B_2 D_1 R\,N
\end{aligned}
$$

$$
\begin{bmatrix} \dot{X} \\ \dot{Z} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} A_2 - B_2 D_1 D_3 K C_2 & B_2 C_1 & -B_2 D_1 C_3 \\ -B_1 D_3 K C_2 & A_1 & -B_1 C_3 \\ B_3 K C_2 & 0 & A_3 \end{bmatrix} \begin{bmatrix} X \\ Z \\ W \end{bmatrix} + \begin{bmatrix} B_2 D_1 N \\ B_1 N \\ 0 \end{bmatrix} R \quad (3.16)
$$

$$
Y = C_2 X \quad (3.17)
$$

The current in motor is given by,

$$
\begin{aligned}
I_a &= \frac{U_2}{K_f} \\
I_a &= \frac{1}{K_f} \begin{bmatrix} -D_1 D_3 C_2 K & C_1 & -D_1 C_3 \end{bmatrix} [X \;\; W \;\; Z]' + \frac{1}{K_f} [D_1 N] R \quad (3.18)
\end{aligned}
$$

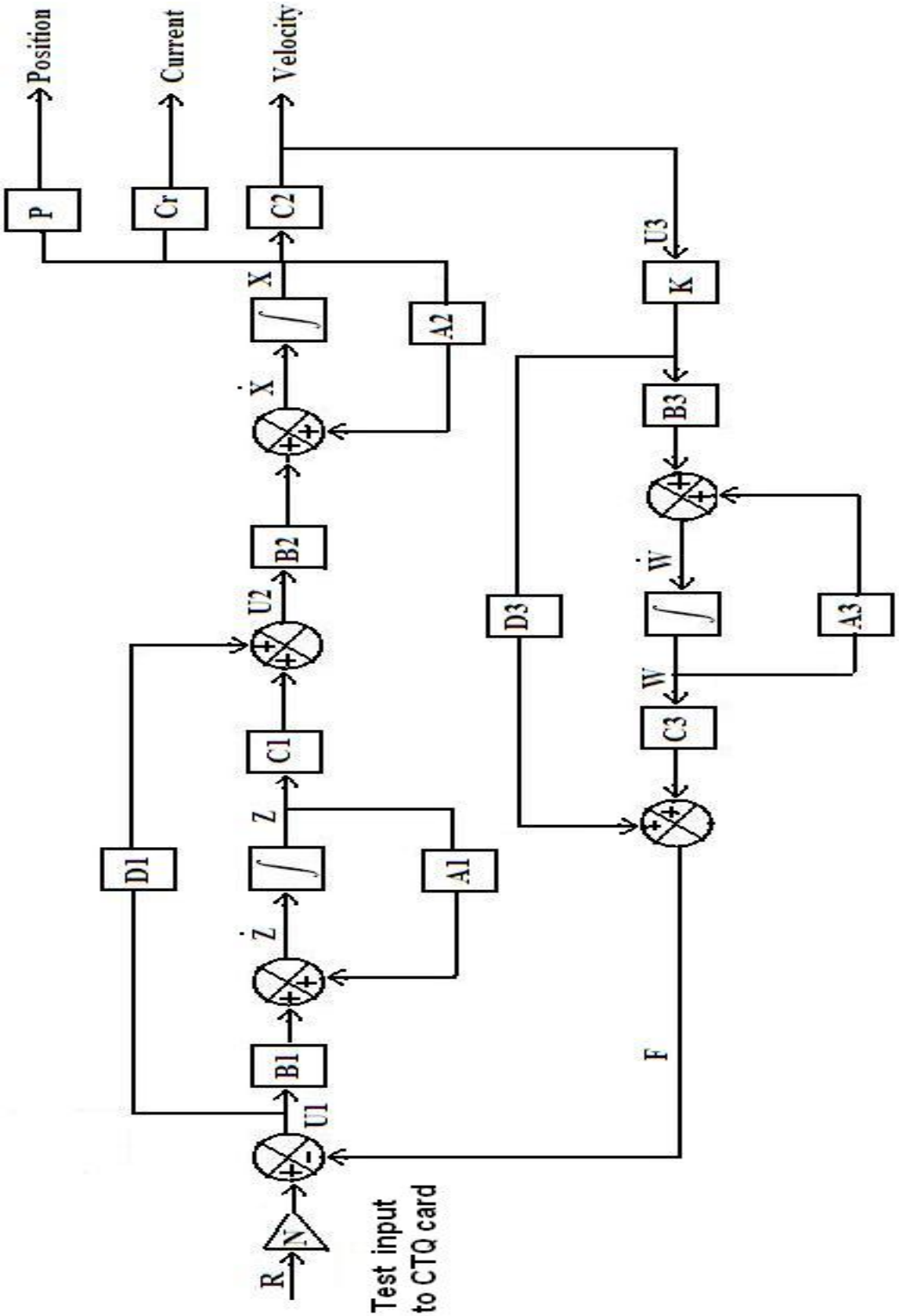

Figure 3.1: State space representation of servo system

3.3 Linear Quadratic Regulator (LQR)

The Linear Quadratic optimal control has two major type of control aims: The regulator problem and the tracking problem. The qualitative statements of these 2 types of problems can be stated verbatim as given in [7]

> *Qualitative statement of the regulator problem*
> Suppose that initially the plant output or any of its derivatives,is nonzero. Provide a plant input to bring the output and its derivatives to zero. In other words, the problem is to apply a control to take the plant from nonzero state to zero state. This may typically occur where the plant is subjected to unwanted disturbances that perturb its output.
>
> *Qualitative statement for tracking problem*
> Suppose that the plant output or its derivative, is required to track some prescribed function. Provide a plant input that will cause this tracking.

We first tackle the problem of regulator in this section and that of tracking in next section. Though we do not use a regulator in our final design, a simulation of regulator gives us insight in the easy of controllability of each state of the system.

3.3.1 Note on equations for LQR

To design a LQR we have to deal with two issues. First we have to define a performance index which has to be optimized (minimized) and then find an input u^* which will regulate the system giving optimum (minimum value) of performance index. Consider a system given by,

$$\dot{x} = Ax + Bu \tag{3.19}$$

$$y = C'x + Du \tag{3.20}$$

We consider the problem of infinite time regulator. For this performance index for the system is given as,

$$PI = \int_0^\infty (u'Ru + x'Qx)dt \tag{3.21}$$

where Q and R are weight matices.
The PI can be represented as,

$$PI = x'(t)Px(t) \tag{3.22}$$

Where P is a positive definite symmetric matrix. The value of u^* minimizing eqn 3.21 is found by deducing Hamiltonian-Jacobi equation for the PI as given in [6] and [7]. The control input u^* which will minimize the PI is given by,

$$u^* = -R^{-1}BPx(t) \tag{3.23}$$

where P is a positive definite symmetric matrix satisfying equation 3.24,

$$-\dot{P}(t) = P(t)A(t) + A'(t)P(t) - P(t)B(t)R^{-1}(t)B'(t)P(t) \tag{3.24}$$

Eqn 3.24 is the matrix riccati equation. Solving it gives the value for P. Then, u^* is given as,

$$u^* = -Kx(t) \tag{3.25}$$
$$K = -R^{-1}BP$$

3.3.2 Simulation results for LQR

The LQR for GMRT servo is simulated by writing m-code in matlab. Appendix A gives the m-code used for simulation. Equations 3.26, 3.27 and 3.28 give the values for weight matrices Q an R used for simulation and feedback gain matrix K calculated by the simulation. The weight matrices are decided such that the response minimizes PI while the constraints on the motor speed and current are not by violated. The maximum possible motor speed is 2000 rpm i.e 209.33 rads/sec Where as the maximum allowable current in motor is 80 Amps. Another constraint for GMRT servo is the maximum torque that can be applied to gearbox by motor. Its value is 20 Nm which corresponds to 40 Amps of motor current. Hence the final limit on motor current works out to be 40 Amps. In case of high load condition, the load is equally shared by both the motors. So in high load situations, the maximum torque that can be applied to the both the gearboxes is 20+20 = 40 N-m. In simulation we use only one motor instead of 2 motors. Hence maximum allowable torque for single motor in simulation is 40 N-m which puts a maximum current constraint of 80 Amps on the motor. Figures 3.2 and 3.3 give simulated results for motor current and dish position. It is seen that the maximum motor current is around 79 Amps. Equations 3.26 and 3.27 give the weight matrices Q and R used in simulation and equation 3.28 give the calculated feedback gain vector K.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.26}$$

$$R = [1] \tag{3.27}$$

$$K = [0.0076 \;\; 0.0002 \;\; 0.0073 \;\; 0.0000 \;\; 0.5197 \;\; 0.2630 \;\; -2.6507 \;\; 1.4928 \;\; 0.0040 \;\; -18.9278] \tag{3.28}$$

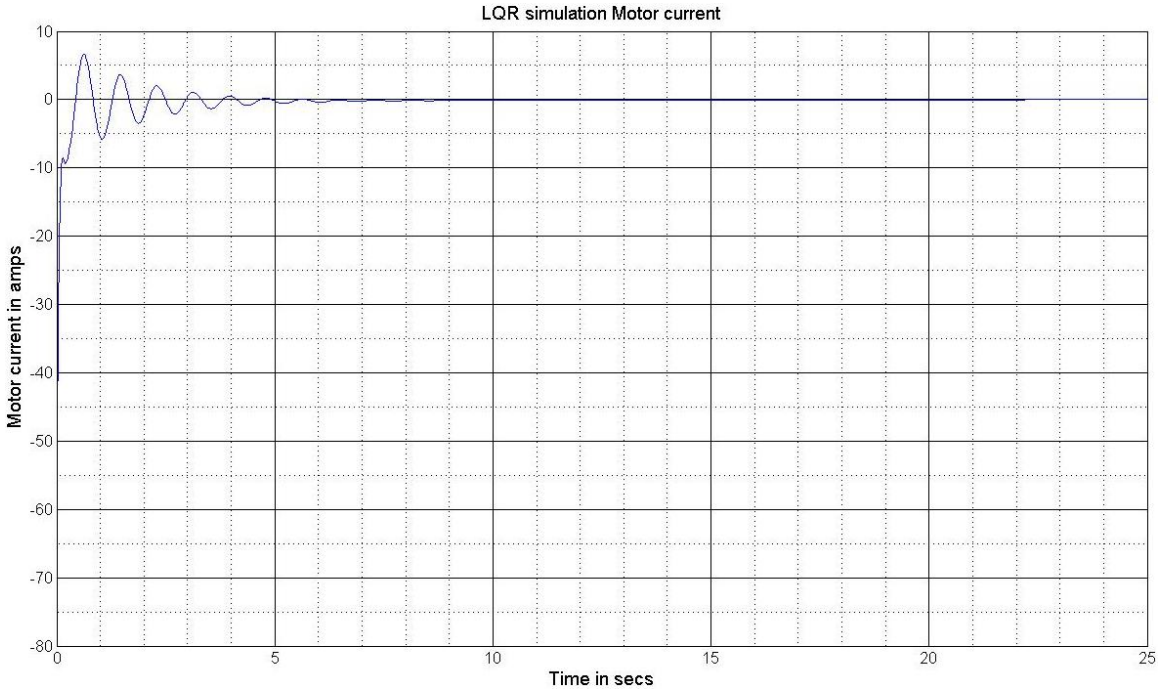

Figure 3.2: Response of LQR: Motor current

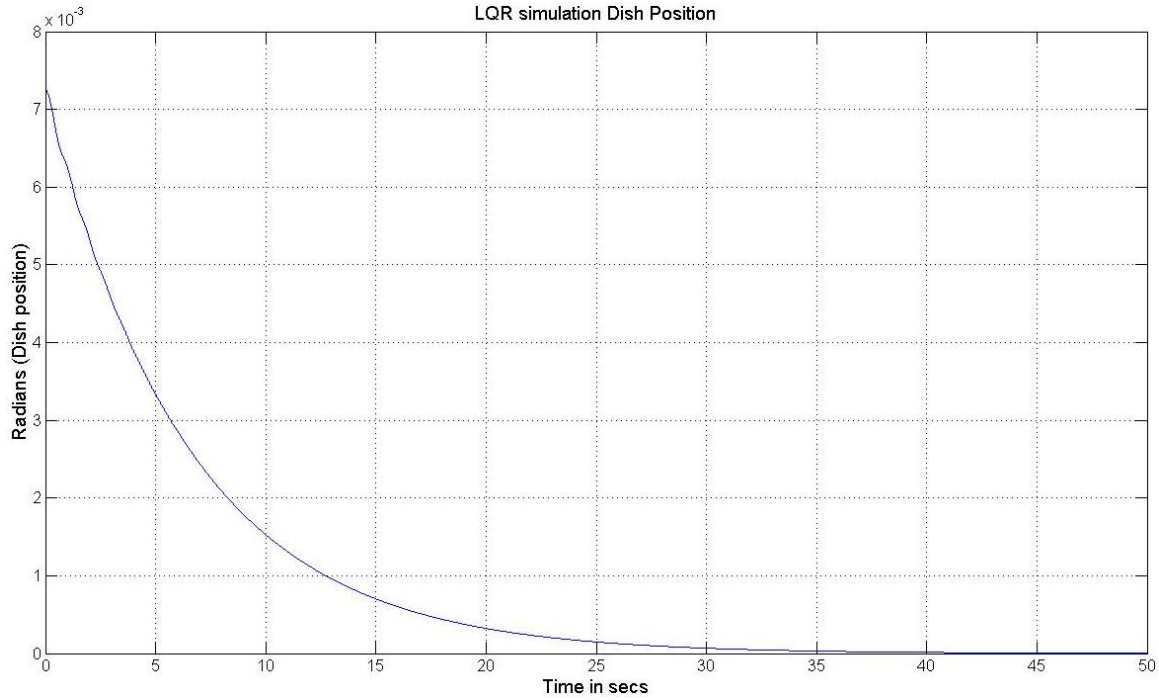

Figure 3.3: Response of LQR: Dish Position

3.4 Linear Quadratic Tracking Regulator

For GMRT system we need to track a source in sky. Hence, the antenna must constantly follow a given input trajectory. We can convert a tracking problem into a regulator problem by defining an input reference model whose output is the trajectory that the system needs to follow. The reference model is such that it has zero input and some initial state. The response of the reference model to this non zero initial state creates the trajectory that the system has to follow. By augmenting the reference model to the system model we create a new system which has zero input and some non zero initial state (in reference model). Our aim now is to find an input u^* which bring system from nonzero state to zero state as well as minimize the PI. Now this is a standard regulator problem. Figure 3.4 shows the block diagram of a system with reference model. The next subsection defines the mathematical equations to convert a tracking problem into a regulator problem.

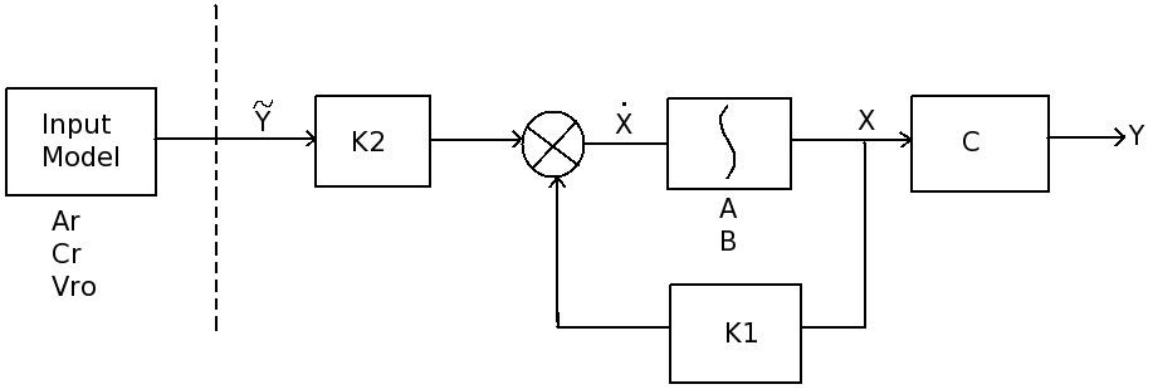

Figure 3.4: System with input reference model

3.4.1 Note on equations for LQTR

The PI for a tracking problem should minimize the difference between commanded trajectory and actual trajectory followed by the system. The PI should also include term to penalize the input to the plant. Hence the PI for the tracking problem can be defined as,

$$PI = \int_0^\infty (u'Ru + (y - \tilde{y})'Q(y - \tilde{y})dt \tag{3.29}$$

Where $\tilde{y}$ is desired trajectory and y is output trajectory.
Now we convert tracking problem into regulator problem by defining input reference model and auxiliary matrices. For GMRT the trajectory that the system needs to follow is a ramp signal. Its reference model is given by,

$$\begin{aligned} \dot{V}_r &= A_r v_r \\ \tilde{Y} &= C_r V_r \end{aligned}$$

And the initial state is given by V_{r0},

$$A_r = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{3.30}$$

$$C_r = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3.31}$$

$$V_{r0} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.32}$$

The auxiliary matrices to convert a tracking problem to a regulator problem are defined as,

$$L = C'(C.C')^{-1} \tag{3.33}$$

$$\overline{C} = I - L.C \tag{3.34}$$

Where C is plant output matrix. If C has rank p then we now define a fictitious output vector which is a linear combination of remaining n-p states. Where n is system order.

$$\overline{y}(t) = \overline{C}x(t) \tag{3.35}$$

And a fictitious desired state trajectory given by,

$$\tilde{x}(t) = L\tilde{y} \tag{3.36}$$

The modified PI is given as

$$PI = \int_0^\infty (u'Ru + \overline{y}'Q_1'\overline{y} + (y - \tilde{y})'Q_2(y - \tilde{y}))dt \tag{3.37}$$

Defining,

$$Q = \overline{C}'.Q_1.\overline{C} + C'Q_2.C \tag{3.38}$$

The PI becomes,

$$PI = \int_0^\infty (u'Ru + (x - \tilde{x})'Q(x - \tilde{x}))dt \tag{3.39}$$

Also $\tilde{y} = C_r v_r$ The augmented plant an reference model matrices are given as,

$$\hat{x} = \begin{bmatrix} x(t) \\ v_r(t) \end{bmatrix} \tag{3.40}$$

$$A_{TR} = \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix} \tag{3.41}$$

$$B_{TR} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tag{3.42}$$

$$PI = \int_0^\infty u'Ru + \hat{x}'Q_{TR}\hat{x}dt \tag{3.43}$$

Then the weight matrix Q becomes,

$$Q_{TR} = \begin{bmatrix} Q & -QLC_r \\ -C_r'L'Q & C_r'L'QLC_r \end{bmatrix} \tag{3.44}$$

Subject to augmented system,

$$\dot{\hat{x}} = A_{TR}\hat{x} + B_{TR}u \tag{3.45}$$

The resulting Matrix Riccati equation is solved by partitioning of state vector. This results in three steady state Riccati equations as follows

$$PA + A'P - PBR^{-1}B'P + Q = 0 \tag{3.46}$$

$$P_{12}A_r + A'P_{12} - PBR^{-1}B'P_{12} - Q.L.C_r = 0 \tag{3.47}$$

$$P_{22}A_r + A_r'P_{22} - P_{12}'BR^{-1}B'P_{12} + C_r'L'QLC_r = 0 \tag{3.48}$$

The input for optimizing the PI is given as,

$$u^* = K_1 x(t) + K_2 v_r(t) \tag{3.49}$$

where, K_1 is feedback gain matrix and K_2 is feed forward gain matrix given as,

$$K_1 = -R^{-1}B'P \tag{3.50}$$

$$K_2 = -R^{-1}B'P_{12} \tag{3.51}$$

3.4.2 Simulation results for LQTR

Appendix A gives the m-code for simulation of linear quadratic tracker. Equations 3.52, 3.53 and 3.54 give the weight matrices Q1, Q2 and R usd for simulations. These weight matrices give the least value of PI without violating system constraints. Equations 3.55 and 3.56 give gain matrices K1 and K2 calculated by simulation for given weight matrices. Figures 3.5 and 3.6 give the simulated motor current and dish position for the optimal tracker. Figure 3.8 gives the bode plot for the optimal tracker.

$$Q1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.52}$$

$$Q2 = [10000000] \tag{3.53}$$

$$R = [.001] \tag{3.54}$$

$$K1 = \Big[-3.5547 \quad -0.0799 \quad -24.2686 \quad -0.0199 \quad 2.9418 \times 10^4 \quad -1.1340 \times 10^3 \quad -6.1705 \times 10^4 \quad -3.6515 \times 10^3 \quad -0.3729 \quad 4.7306 \times 10^3\Big] \tag{3.55}$$

$$K2 = \begin{bmatrix} 1.0000 \times 10^5 & 7.7503 \times 10^3 \end{bmatrix} \tag{3.56}$$

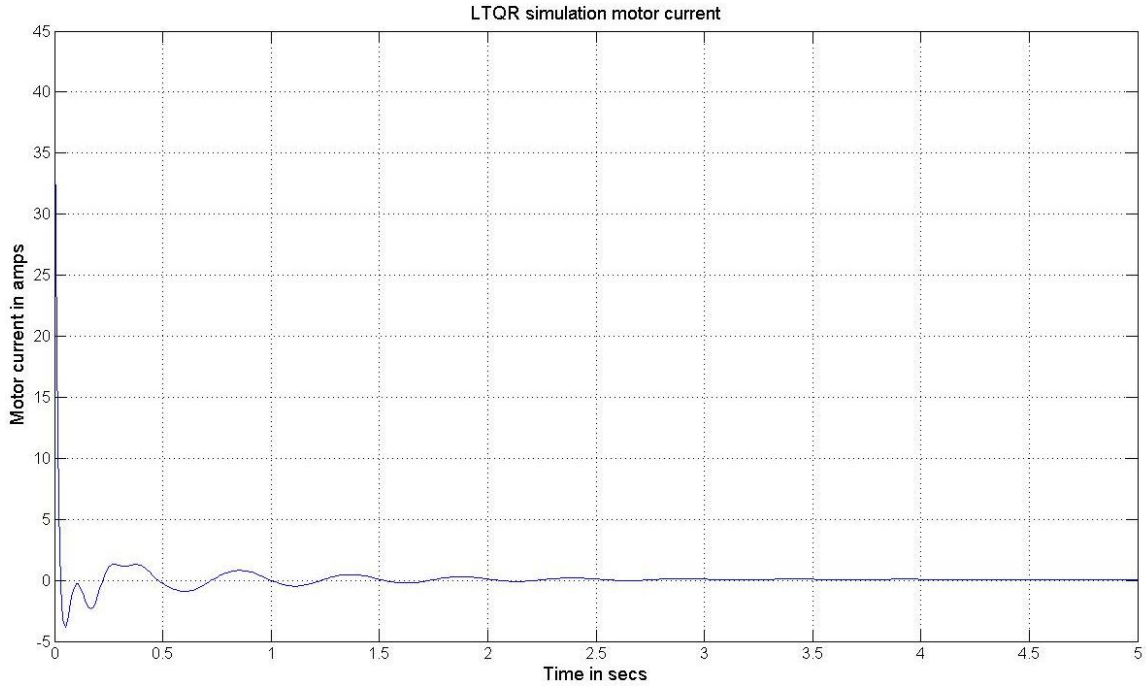

Figure 3.5: Response of LQTR: Motor current

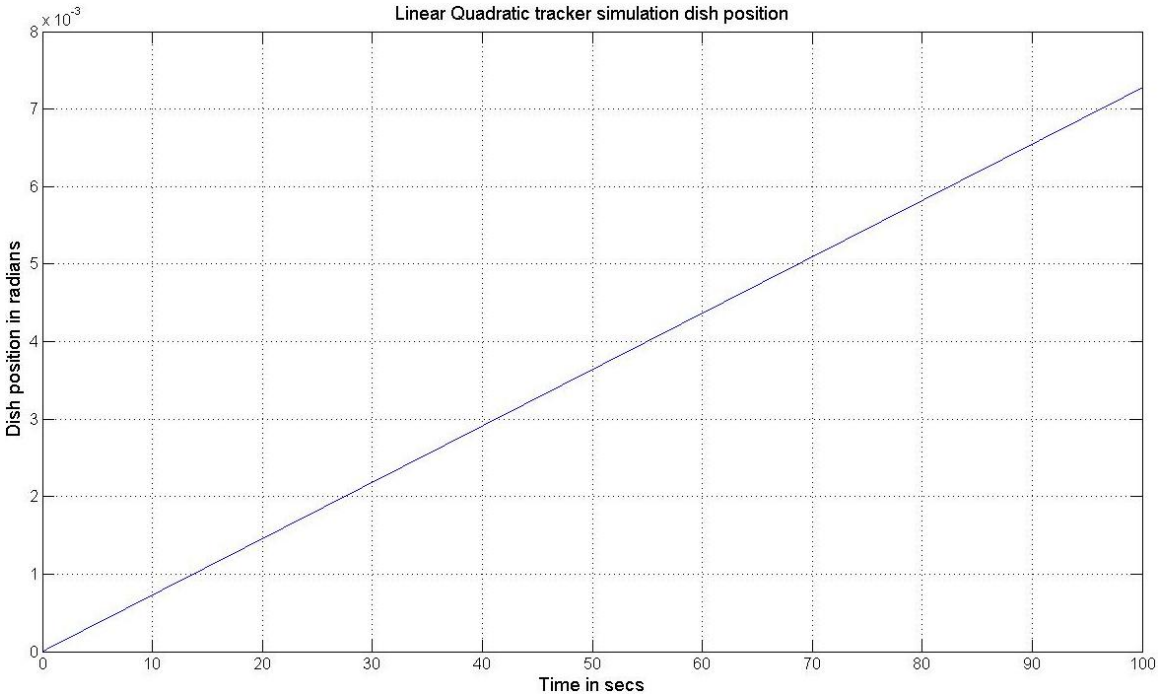

Figure 3.6: Response of LQTR: Dish Position

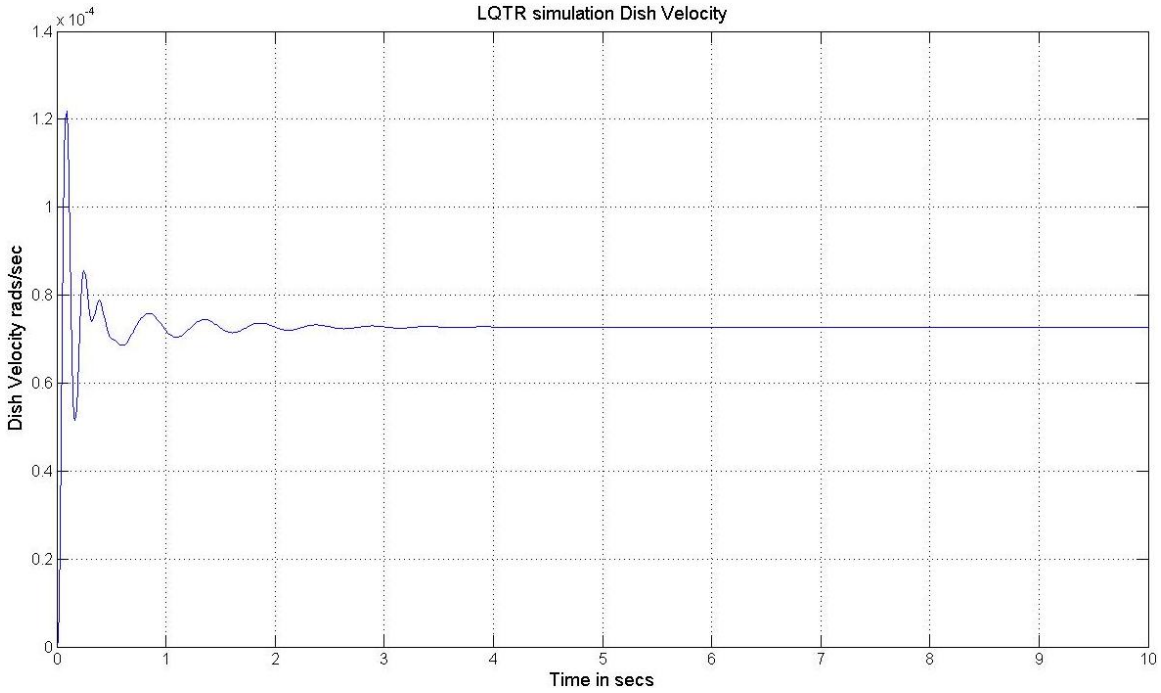

Figure 3.7: Response of LQTR: Dish velocity

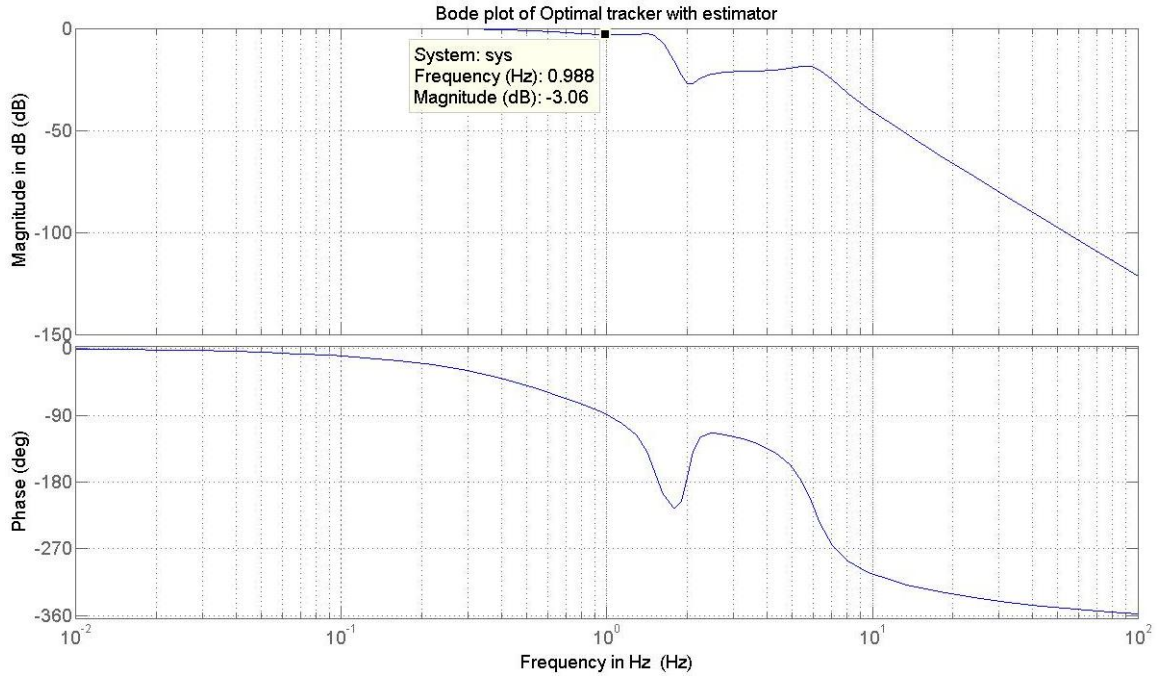

Figure 3.8: Response of LQTR: Bode plot

3.5 Linear Quadratic Tracker with Estimator

For the design of LQR and LQTR is was assumed that the complete state vector is available for feedback. But in reality only 3 variables of state vector are available. The motor current, motor velocity x(2) and controlled inertia position x(5) are available for measurement. The rest of the system states can be estimated by measuring the system input and output. For this it is necessary that the system is observable. By calculating the system observability matrix it is seen that the system is indeed observable hence all the system states can be estimated. This section discuss the design of the estimator. The estimated system states can then be used for feedback in LQTR controller.

3.5.1 Note on equations for tracker with estimator

The observer for the system defined in equations 3.19 and 3.20 is given as [7]

$$\dot{x}_e = (A + K_e C)x_e(t) + Bu(t) - K_e y(t) \tag{3.57}$$

The derivative of error between the actual states and the estimated states is given as,

$$\frac{d}{dt}(x - x_e) = (A + K_e C)(x - x_e) \tag{3.58}$$

Thus by proper pole placement the error can be made to quickly approach zero.
The measurement of actual system input and output will be corrupted by noise. This noise can be considered as Gaussian in nature. The system can thus be represented as,

$$\dot{x} = Ax(t) + Bu(t) + \nu(t) \tag{3.59}$$
$$y(t) = Cx(t) + \eta(t) \tag{3.60}$$

Where ν and η are input and output measurement noise. The noise variables are random variable with Gaussian distribution and zero mean. They can be described as statistical processes given by,

$$E[\nu(t)\nu'(t)] = Q\delta(t - \tau) \tag{3.61}$$
$$E[\nu(t)] = 0 \tag{3.62}$$
$$E[\eta(t)\eta'(t)] = R\delta(t - \tau) \tag{3.63}$$
$$E[\eta(t)] = 0 \tag{3.64}$$

Where Q is positive semidefinite matrix and R is positive definite matrix.
The performance index will require minimizing variance of error between actual and estimated states. The PI will thus become,

$$PI = E([x(t_1) - x_e(t_1)]'[x(t_1) - x_e(t_1)]) \tag{3.65}$$

Solving the Hamiltonian Jacobi of the PI gives a steady state matrix riccati equation,

$$PA' + AP - PC'R^{-1}CP + Q = 0 \tag{3.66}$$

The feedback gain K_e for optimal pole placement of observer poles is given by,

$$K_e = -PC'R^{-1} \tag{3.67}$$

The block diagram of complete design consisting of optimal tracker with estimator is shown in figure 3.9

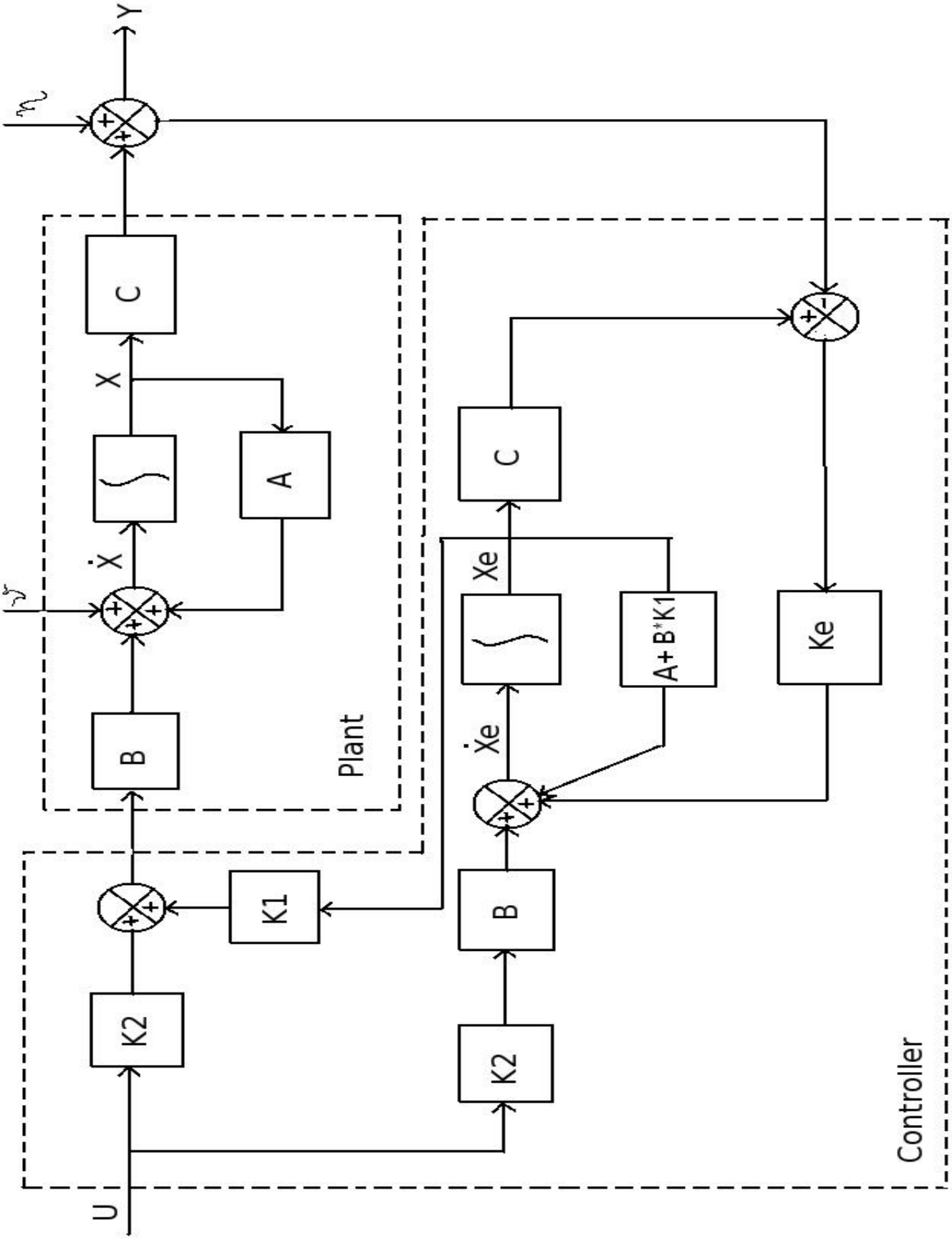

Figure 3.9: Block diagram of tracker plus estimator [1]

3.5.2 Simulations results of Tracker with estimator

The m-code for tracker with estimator is given in Appendix A. For the optimal tracker designed in section 3.4 the estimator feedback gain Ke calculated by simulation is given by equation 3.68. The weight matrices Q and R used for calculating Ke is given in equations 3.69 and 3.70 respectively. Table 3.5.2 gives the steady state error between plant state variable values and estimated state variable values.

$$Ke = \begin{bmatrix} 1.07106288 & -9797895.87647351 & 0.00094548 \\ 0.00663667 & 0.00006471 & 0.00046285 \\ 0.00003749 & 0.00055409 & -178.73408017 \\ 40.94071224 & -31588.23520335 & 0.06470993 \\ -26.20085944 & -23.52758256 & -2.20879454 \\ -2.44347976 & -2.57993737 & -3.92371986 \\ 0.90455371 & 24.79576052 & 147.65373854 \\ -1787.34080173 & 0.10998122 & 0.43789868 \\ 0.00904554 & 0.02619027 & 0.01008977 \\ 0.03965760 & -27726.46239188 & 3133.51609506 \end{bmatrix} \tag{3.68}$$

$$Q = \begin{bmatrix} 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 96 \times 10^{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^5 \end{bmatrix} \tag{3.69}$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{3.70}$$

State Variable	Errror
θ_m	.079 rads
θ_g	4.7414×10^{-5} rads
θ_c	3.7711×10^{-6} rads
θ_{uc}	3.7935×10^{-6} rads
x(9)	1.6719×10^{-6} Volts
x(10)	3.8639×10^{-5} Volts

Table 3.1: Error between plant state variables and estimator state variables at steady state

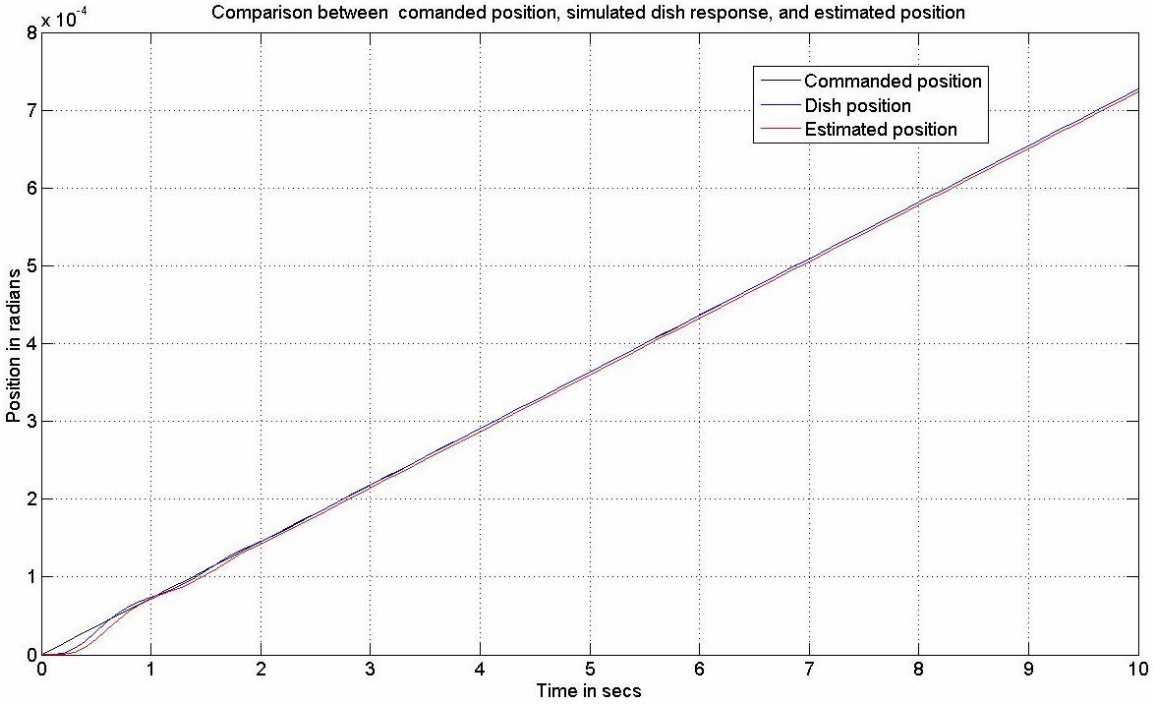

Figure 3.10: Response of Tracker with Estimator: Flexible inertia position

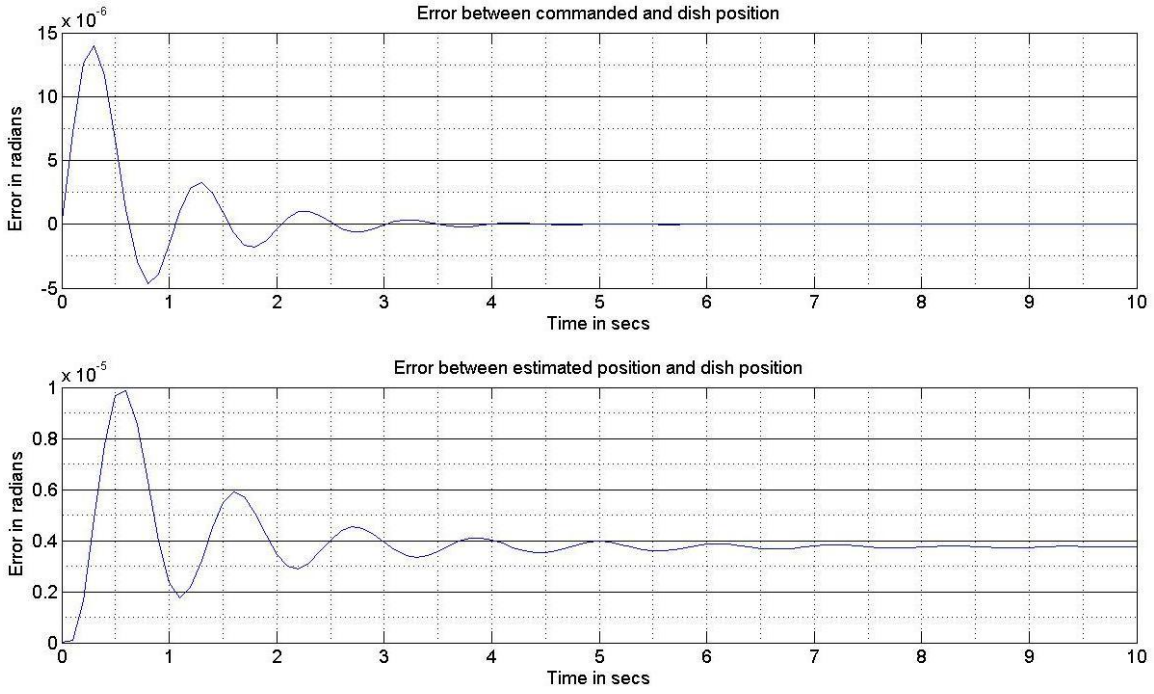

Figure 3.11: Response of Tracker with Estimator: error between commanded position, actual position and estimated position

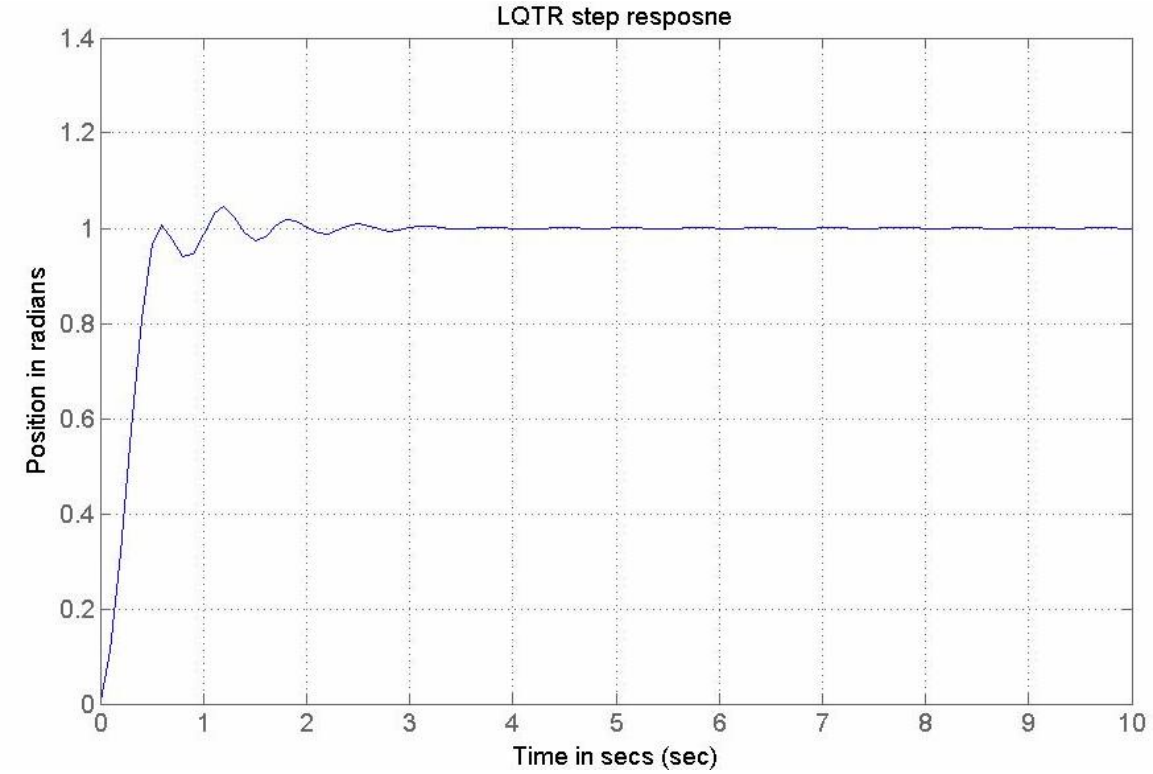

Figure 3.12: Response of Tracker with Estimator: step response

3.6 Conclusions

This chapter discussed the simulation of optimal controllers for GMRT servo. Optimal controller achieves the allowable system bandwidth of 1 Hz. This assures fastest transient response with acceptable overshoot and zero steady state error. Also none of the system limits were violated. As compared classical design this system performance was achieved by fewer iterations of controller parameters Q and R. This makes optimal controllers more attractive for tuning purpose. The LQTR controller discussed in this chapter is a memory less controller of type 0. Though we get a zero steady state error in absence of disturbance, there will be a steady state error in presence of disturbances like wind. Hence as discussed in [1] and [8] response of tracker with a feedforward integral gain needs to be simulated. The integral gain will lead to near zero steady state error even in case of wind disturbances.

Chapter 4

Conclusions and notes on further work plan

4.1 What more remains in Simulation?

The simulation response of brushed DC motor was found to be a close match with that of the measured system response. Differences in transient response can be speculated to be because of inaccuracy of mechanical model. Hence more work needs to be done on modeling of mechanical system. This issue is discussed in detail later in this section. For a large radio telescope like GMRT the azimuth dynamics are expected to change with elevation angle. A detailed LRF test of azimuth axis for various elevation angles was carried. No significant change was found in first resonance frequency as a function of elevation. This is because for a GMRT servo-mechanical system the motor inertia is much higher than reflected inertia of the antenna. Thus the changing antenna dynamics with change in elevation has little effect on overall system dynamics. In the simulation of GMRT servo system, so far we have considered only the simulation of azimuth axis. The simulation of elevation axis was carried out in similar fashion as for azimuth axis with appropriate change in parameter values. As the motor inertia dominates over the antenna dish inertia the response of elevation axis is similar to that of azimuth axis. All results discussed for azimuth axis hence apply even for elevation axis.

The simulation of optimal controller shows the best possible response that the servo system can give without violating current or speed limits. But these results are obtained in absence of any disturbance. The results will be more meaningful if simulations show that optimal controllers give better accuracy even in presence of external disturbances. Simulations in this regard are already in progress. The biggest external disturbance to the system is wind load. Simulation of wind effects on tracking accuracy of antenna are needed to be carried out for the present servo system. This can be done either by the approach as given used in [1] or as given in [9]. Simultaneously it is necessary to experimentally measure wind power spectrum and verify the simulated and measured wind loads. The wind meters used at GMRT are cup type anemometers. These wind anemometers are have mechanical moving parts and hence slower response time. This makes them inefficient to measure fast varying wind gusts. A more precise wind meter should be used to measure wind gusts at GMRT and consequently its effect on antenna tracking accuracy. A project in this direction has already been initiated.

For effectiveness of optimal controller depends on accuracy of the estimator used in the controller. The equations of the estimator are same as plant equations. Hence it is necessary to have an accurate plant model. For this a detailed system identification of the plant model needs to be carried out. So far simple least square estimation techniques and frequency domain techniques like sine sweeps have been tried. But these methods are are not effective for a complex system like GMRT. By using input which is rich in its frequency content so that it excites all the modes of the system, a better estimation of higher order resonance frequencies of the system can be achieved. One such input is Pseudo Random Binary signals(PRBS) [10]. These signals can be generated using programmable arbitrary waveform generator or using Simulink [11] and National instruments Data acquisition card. Similarly, antenna can be excited using white noise which can also be generated either by MATLAB or arbitrary waveform generator.

It is found that the transition matrix A of plant model of the system is ill conditioned. Hence state matrix conditioning method like Hankel singular value decomposition and model reduction methods like balanced truncation method can be tried [12]. This step of conditioning of state matrices and model reduction will play an important role in implementation of optimal controllers in servo computer. If the system matrix is badly conditioned it will lead to accumulation of computational errors. This will result in undesirable response or even unstable response of the system. Model reduction will lead to a model of much lesser model order. Simulation results show that there is scope to reduce present model order of 12 to as low as 5^{th} order or 6^{th} order. This will make the task of on line parameter estimation much easier. Also the computational load on servo computer to implement optimal controller will decrease by a large factor.

The problem of system identification and model reduction of the system are coupled to each other. Both these problems can be tackled simultaneously if the coordinate system is changed from θ and $\dot{\theta}$ to the modes of the system. This approach has been used in implementing optimal controllers for NASA's DSN network telescopes and Large Millimeter Telescope [13]. Once the coordinates of the system are changes to the modes, model reduction can be obtained by discarding non-dominant modes of the system. By carrying modal testing of the system the modal model of the system can be identified using Eigensystem Realization Algorithm as explained in [14]. To have a modal model of the system it is necessary to carry out FEM analysis of the system. FEM model along with system identification will help in removing inaccuracies of mechanical model. For GMRT a detailed FEM analysis is being presently carried out by Tata Consulting Engineers. Once the FEM model of the system is ready both problems of model reduction and system identification can be tackled simultaneously.

4.2 Hardware in the loop simulator

The successful implementation of optimal controller can be done only after an accurate plant model has been obtained. This will be possible only after successful system identification of the system. The other aspect in implementation of optimal controllers is its integration with present servo system. This task will typically include writing an appropriate code of optimal controller to run on servo computer. The issues that will be faced are maintaining computational load of

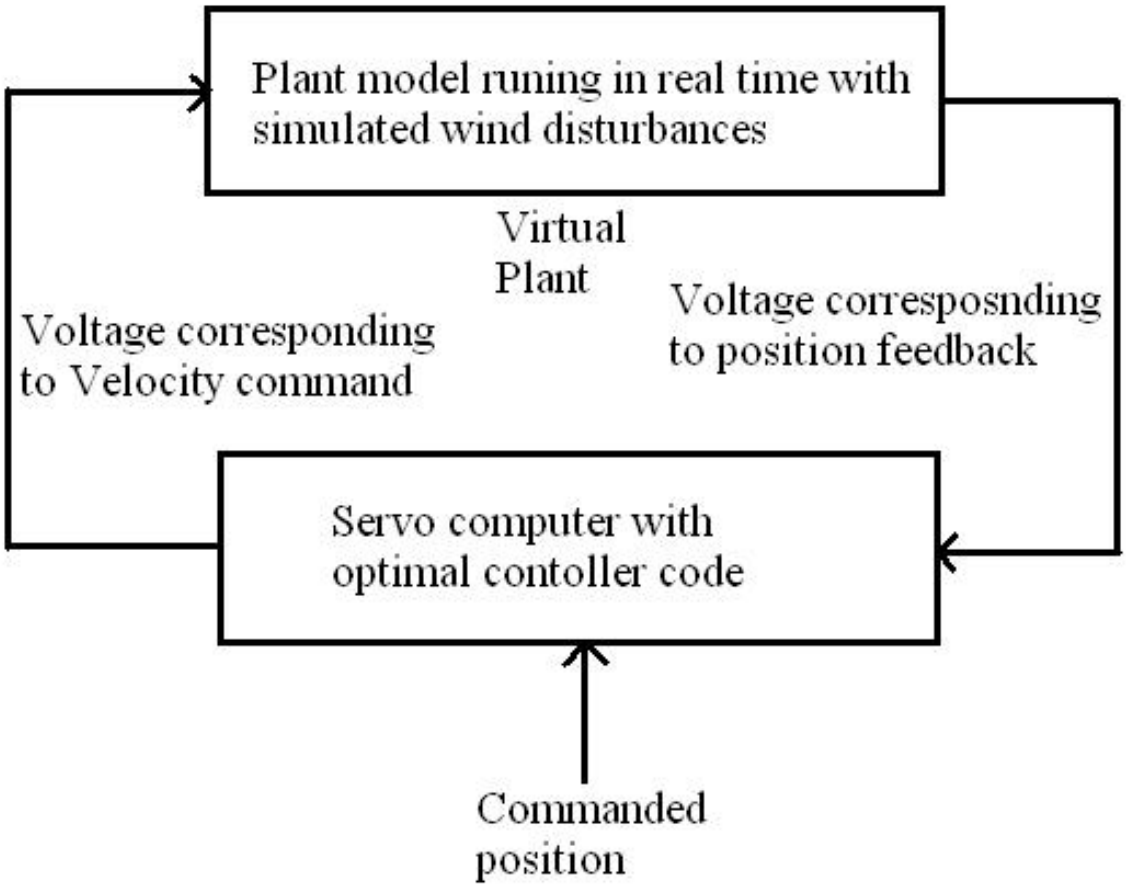

Figure 4.1: HILS schematic

servo computer and getting required computational accuracy. The more detailed is the model of estimator better will be the accuracy of estimated states but this will increase the load on servo computer. A trade-off between accuracy and computational load need to be achieved such that the final design meets the pointing challenges even in presence of wind. The present simulation gives the initial guess of the state feedback gains and and plant model that will be needed for optimal control implementation. Using Simulink's Real Time workshop toolbox one can quickly generate the code needed to implement optimal controller. The plant model can also be run in real time so that it acts as a virtual antenna. The virtual antenna can then be used to test the optimal controller code. Such a setup which is a standard practice in aerospace and automobile industry is called Hardware in the loop simulator (HILS). Below is the outline of steps that can be followed for forming HILS setup using simulink. Figure 4.1 shows a schematic representation for HILS setup.

1. Setup up a real-time model of antenna to be run on a PC called Target PC. Matlab provides toolboxes to create executable to run in real time on real-time windows kernel, xPC kernel or real time linux kernel among other. Chose the a suitable kernel depending on available input/output interface options and ease of implementation. The real time model will act as a virtual antenna generating antenna position, velocity, current and will communicate with the controller.

2. Setup a communication between the RT antenna model and the hardware controller. The is done by accessing the position output of the RT model in a suitable form as a real world signal (Voltage). This signal can be the 17 bit serial output mimicking the actual encoder. The signal can then straight away be given to hardware controller which has the facility to read the encoder. But the actual encoder works on 100 Khz clock. Matlab does not have a facility to simulate such high frequency clock in RT. The highest possible sampling rate in real time approaches 50 KHz only as given in key features of xPC target toolbox [15]. The solution for this is to generate a quantized (mimicking the quantization of encoder output) analog signal corresponding to the position output of RT model. The hardware controller can be easily configured to read this signal for position feedback instead of encoder signal and convert it to appropriate units. According to the position loop requirements this signal will require a sampling frequency of only 20Hz which is easily achievable in matlab generated RT model.

3. Use real time workshop to generate optimal controller code which will work on servo computer having RT Linux OS. Test the controller code for various performance parameters.

By following the above steps a laboratory setup to test optimal controller design can be established. In this case the plant to be controlled (RT model of antenna) is exactly same as the estimator plant model in controller. Hence, the inaccuracies and instability of optimal controller due to inaccurate plant model can be avoided for initial testing phase. This will help in understanding the computational load of the controller hardware for optimal control design. Some part of safety interlock logic can also be implemented in RT model of plant in matlab. Thus the behaviour of optimal controller can be tested when it hits various limits. If the antenna shows limit cycling on hitting the limits a command pre-processor can be developed as discussed in [16]. An optimal controller can be tried on antenna only after through testing, which is possible only after stetting up a laboratory setup. The work of estimating the actual parameters of the plant can be carried in parallel to setting up of a laboratory setup for testing of optimal control design. As and how the plant model is improved by successful parameter estimation experiments it can be accordingly modified in controller hardware and the RT model of plant. This parallel approach to solve the problem will greatly expedite the implementation and optimal controllers on actual antenna.

Appendix A

Simulation Programs

All simulations for are carried in Matlab and Simulink software. For more information about programing in matlab and simulink the reader can refer to [17] and [18] respectively. The simulation of brushed DC motor and brushless DC motor servo system are done in Simulink. But response of these simulation is plotted in matlab workspace. These simulink files will play an important role for converting these simulations in real-time. Models generated in simulink alone can be converted in real time executable. This facility is not available for programs written in m-file. The experimental data is imported in matlab workspace and is then compared with simulated response. The programs for optimal control and frequency test analysis are written in m-files which are executed in matlab workspace. To run the programs properly the programs must be preferably in matlab's current directory.

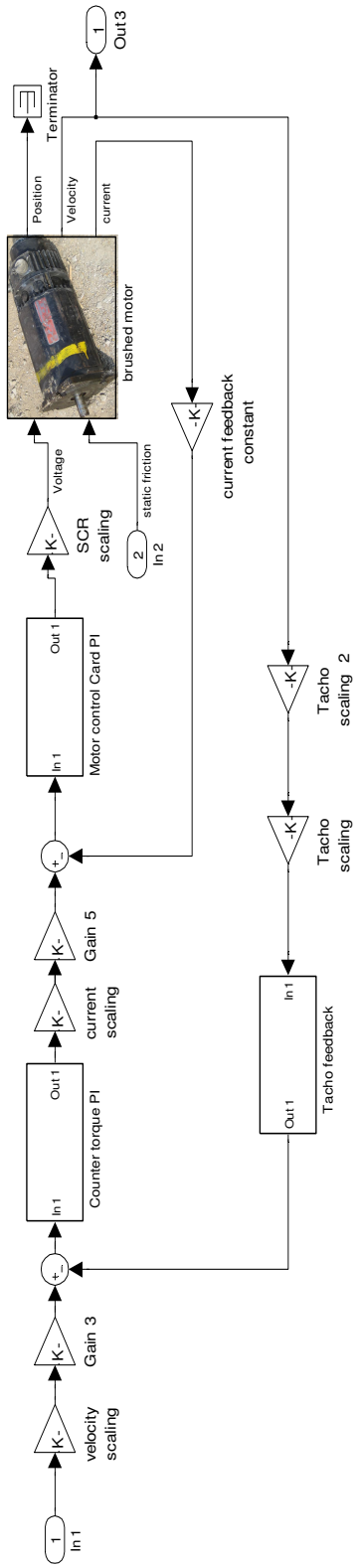

Figure A.1: Simulink model of Brushed DC motor and its servo system

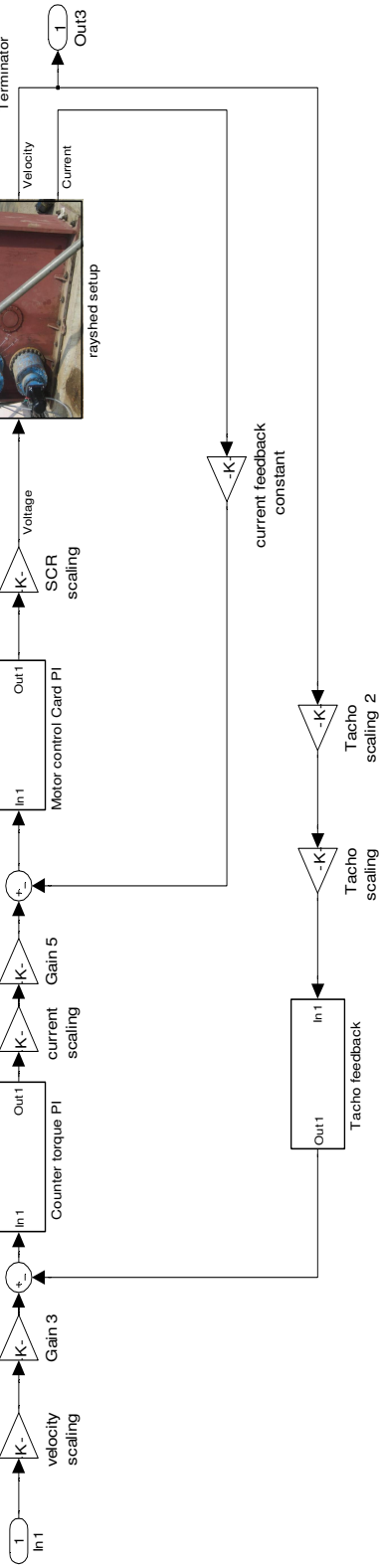

Figure A.2: Simulink model of Brushed DC motor with rayshed test setup

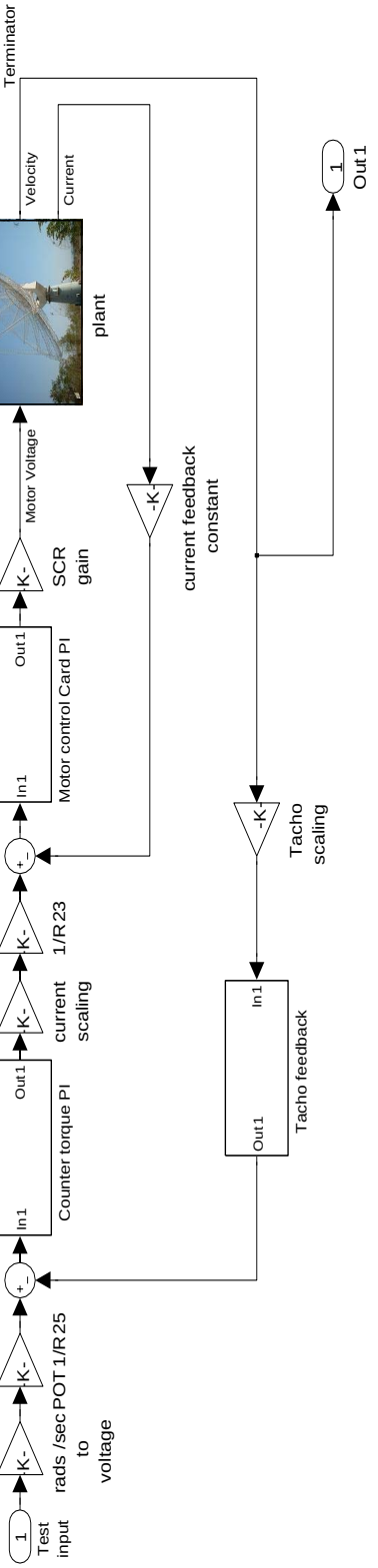

Figure A.3: Simulink model of GMRT antenna with brushed DC servo system

```
%************************************************************************
%Program for generating bode plot from detailed frequency tests on antenna.
%The program assumes that data is collected using NI DAQ card using signal
%express software. Store all files of one axis in one folder in tab
%delimited format with .txt extension.
%************************************************************************
clf
clear
clc
Pathname = input('Pathname of data folder: ','s');
extension = '\*.txt';
path = strcat(Pathname,extension);
lrf_data = dir(path);
dt = input('Sampling period for data Aquisition: ');
input_col = input('Column number of input signal: ');
output_col = input('Column number of output signal: ');
Hline = input('Specify the line number of last header text: ');
[a,b]=size(lrf_data);
for i=1:1:a
    lrf_data(i).name;
    filename = strcat(Pathname,'\',lrf_data(i).name);
    A = importdata(filename,'\t',Hline);
    x = A.data;
    [n,m]=size(x);
    sa1=fft(x(:,input_col));
    sa2=fft(-x(:,output_col));
    [mag1, mag1_index] = max(abs(sa1));
    freq(i) = (mag1_index - 1)/(n-1)/dt;
    phase1 = angle(sa1(mag1_index));
    [mag2, mag2_index] = max(abs(sa2));
    phase2 = angle(sa2(mag2_index));
    phase(i)= (phase2-phase1)*180/pi;
    mag(i) = 10*log(mag2/mag1);
end
subplot(2,1,1)
plot(freq , mag,'.')
title('Magnitude plot')
xlabel('Frequency in Hz')
ylabel('Magnitude in dB')
grid on
grid minor

subplot(2,1,2)
plot(freq,phase,'.')
title('Phase plot')
xlabel('Frequency in Hz')
ylabel('Phase in degrees')
grid on
grid minor
%-------------------------------------------------------------------------
% End of program
%-------------------------------------------------------------------------
```

```
%************************************************************************
%Program to simulate linear quadratic  regulator for GMRT servo system.
%The plant model and velocity controller are as per present system.
%************************************************************************
%-------------------------------------------------------------------------
%Parameters for Mechanical model of GMRT
%-------------------------------------------------------------------------
jm=.066768;
jg=3.5;
jc=2*0.995e6;
juc=10.9e6;
bm=.005424;
bg=4379.2;
bc=7.8336e6;
buc=1.34368e7;
kmg=1e7;
kgp=5.0864e9;
kad=1.428e9;
n=1505;
ns=12.6;
kf=.0625;
R1=24e3;
R2=24e3;
R3=24e3;
R4=20e3;
R28=180e3;
R66=100e3;
R25=27.4e3;
R26=13.7e3;
C4=0.22e-6;
C5=0.68e-6;
kt=0.56;
K=2*.1614/R25;
N=1/R26;
%-------------------------------------------------------------------------
%Velocity controller trasfer function coefficients.
%-------------------------------------------------------------------------
Zw0=R4*(R1+R2)/[R1*R2+(R1+R2)*(R3+R4)];
Zw1=R4/((R1*R2+(R1+R2)*(R3+R4))*C4);
Pw1=(R1+R3+R4)/((R1*R2+(R1+R2)*(R3+R4))*C4);
Pz1=1/((R28+R66)*C5);
Zz1=R66/((R66+R28)*C5);
Zz0=R28*R66/(R28+R66);
%-------------------------------------------------------------------------
%State Space representation of GMRT mechanical + motor model + velocity
%controller
%-------------------------------------------------------------------------
A=[0 1 0 0 0 0 0 0 0 0;
   -kmg/(jm*n^2) (-bm/jm-Zz0*Zw0*K*kt/(kf*jm)) kmg/(jm*n) 0 0 0 0 0
   kt*1/(jm*kf) -kt*1*Zz0/(jm*kf);
   0 0 0 1 0 0 0 0 0 0 ;
   kmg/(jg*n) 0 -(kmg+kgp/ns^2)/jg -bg/jg kgp/(jg*ns) 0 0 0 0 0 ;
   0 0 0 0 0 1 0 0 0 0;
   0 0 kgp/(jc*ns) 0 -(kgp+kad)/jc -(bc+buc)/jc kad/jc buc/jc 0 0;
   0 0 0 0 0 0 0 1 0 0;
   0 0 0 0 kad/juc buc/juc -kad/juc -buc/juc 0 0;
```

```
    0 -Zw0*K*(Zz1 -Zz0*Pz1) 0 0 0 0 0 0 -Pz1 -(Zz1-Zz0*Pz1);
    0 K*(Zw1-Zw0*Pw1) 0 0 0 0 0 0 0 -Pw1
    ];
B=[0; kt*Zz0*N/(jm*kf);0;0;0;0;0;0;N*(Zz1 - Zz0*Pz1);0];
C=[0 1 0 0 0 0 0 0 0 0 ; -Zz0*Zw0*[0 1 0 0 0 0 0 0]*K 1 -Zz0*1];
D=[0;Zz0*N];
sys=ss(A,B,C,D);
step(sys)
pause
%-------------------------------------------------------------------------
% Defining weight matrices Q and R for linear quadratic regulator and
% finding state feedback gain matrix k.
%-------------------------------------------------------------------------
q=[0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 20e3 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0
    ];
r=1;
[k,s]=lqr(A,B,q,r);
%-------------------------------------------------------------------------
%Closing loop with lqr controller and finding system response to 'zero
%input (reference position), non-zero initial states'.
%-------------------------------------------------------------------------
a=A-B*k;
b=B;
c=C;
d=D;
ang=0.4167;
x0=[1505*12.5*ang*2*pi/360;
    0;
    12.5*ang*2*pi/360;
    0;
    ang*2*pi/360;
    0;
    ang*2*pi/360;
    0;
    0;
    0
    ];
t=0:0.01:50;
u=zeros(length(t),1);
sys1=ss(a,b,c,d);
[y,t,x]=lsim(sys1,u,t,x0);
for I=1:10;
    var_num = num2str(I);
    var = strcat('x',var_num);
    plot(t,x(:,I));
    title(var);
    grid on;
```

```
    pause
end
clear u
u=-k*x';
%-------------------------------------------------------------------------------
%Plotting system states, velocity loop input voltage and motor current
%-------------------------------------------------------------------------------
plot(t,u);
title('input');
grid;
pause;
veler=u;
curr=(y(:,2) + Zz0*N*u')/kf;
plot(t,curr);
title('motor current')
grid;
pause;
%-------------------------------------------------------------------------------
%end of program
%-------------------------------------------------------------------------------
```

```
%************************************************************************
%Program to simulate linear quadratic tracking regulator for GMRT servo
%system. The plant model and velocity controller are as per present system.
%************************************************************************
%-------------------------------------------------------------------------
%Parameters for Mechanical model of GMRT
%-------------------------------------------------------------------------
jm=.066768;
jg=3.5;
jc=2*0.995e6;
juc=8.9e6;
bm=.005424;
bg=4379.2;
bc=7.8336e6;
buc=1.34368e7;
kmg=1e7;
kgp=5.0864e9;
kad=1.428e9;
n=1505;
ns=12.6;
kf=.0625;
R1=24e3;
R2=24e3;
R3=24e3;
R4=20e3;
R28=180e3;
R66=100e3;
R25=27.4e3;
R26=13.7e3;
C4=0.22e-6;
C5=0.68e-6;
kt=0.56;
K=2*.1614/R25;
N=1/R26;
%-------------------------------------------------------------------------
%Velocity controller trasfer function coefficients.
%-------------------------------------------------------------------------
Zw0=R4*(R1+R2)/[R1*R2+(R1+R2)*(R3+R4)];
Zw1=R4/((R1*R2+(R1+R2)*(R3+R4))*C4);
Pw1=(R1+R3+R4)/((R1*R2+(R1+R2)*(R3+R4))*C4);
Pz1=1/((R28+R66)*C5);
Zz1=R66/((R66+R28)*C5);
Zz0=R28*R66/(R28+R66);
%-------------------------------------------------------------------------
%State Space representation of GMRT mechanical + motor model + velocity
%controller
%-------------------------------------------------------------------------
a=[0 1 0 0 0 0 0 0 0 0;
   -kmg/(jm*n^2) (-bm/jm-Zz0*Zw0*K*kt/(kf*jm)) kmg/(jm*n) 0 0 0 0 0
   kt*1/(jm*kf) -kt*1*Zz0/(jm*kf);
   0 0 0 1 0 0 0 0 0 0 ;
   kmg/(jg*n) 0 -(kmg+kgp/ns^2)/jg -bg/jg kgp/(jg*ns) 0 0 0 0 0 ;
   0 0 0 0 0 1 0 0 0 0;
   0 0 kgp/(jc*ns) 0 -(kgp+kad)/jc -(bc+buc)/jc kad/jc buc/jc 0 0;
   0 0 0 0 0 0 0 1 0 0;
   0 0 0 0 kad/juc buc/juc -kad/juc -buc/juc 0 0;
```

```
   0 -Zw0*K*(Zz1 -Zz0*Pz1) 0 0 0 0 0 0 -Pz1 -(Zz1-Zz0*Pz1);
   0 K*(Zw1-Zw0*Pw1) 0 0 0 0 0 0 0 -Pw1
   ];
b=[0; kt*Zz0*N/(jm*kf);0;0;0;0;0;0;N*(Zz1 - Zz0*Pz1);0];
c=[zeros(1,4) 1 zeros(1,5)];
d=[0];
sys=ss(a,b,c,d);
step(sys)
pause
%-------------------------------------------------------------------------
%State space model for refrence input. In this case it is a ramp input with
%a slope of 7.2e-5 rad/sec
%-------------------------------------------------------------------------
ar=[0 1; 0 0 ];
cr=[1 0];
zr0=[0; ((2*pi)/(360*60*60)*15)];
%-------------------------------------------------------------------------
%define auxilary matrices to convert linear quadratic tracking regulator
%problem to linear quadratic regulator problem.
%-------------------------------------------------------------------------
l=c'*inv(c*c');
[m,n]=size(a);
cuo=eye(m,n)-l*c;
%-------------------------------------------------------------------------
%Define weight matrices q1, q2 and r and find state feedback gains k1 and
%k2
%-------------------------------------------------------------------------
q1=[zeros(8,10);
    zeros(1,8) 0.01 zeros(1,1);
    zeros(1,10)];
q2=[1e7];
r=[0.001];
q=cuo'*q1*cuo+c'*q2*c;
qcap=[q -q*l*cr;
    -cr'*l'*q cr'*l'*q*l*cr];

Aaug=[a zeros(m,2);
    zeros(2,n) ar];
baug=[b;0;0];

[k,P]=lqr(a,b,q,r);
P12=lyap((a'-P*b*inv(r)*b'),ar,(-q*l*cr));
%-------------------------------------------------------------------------
%Form closed loop matrices with Linear quadratic tracking regulator state
%feedback gains. Also  calculate system reponse to ramp input of 7.2e-5
%rad/sec. Here the actual input 'u' to plant is zero. But as input model is
%agumented with system matrices, the response of system is to non zero
%initial state of ramp input model which generates the necessary reference
%position.
%-------------------------------------------------------------------------
ac=a-b*inv(r)*b'*P;
a=[ac -b*inv(r)*b'*P12;
    zeros(2,n) ar];
k1=-inv(r)*b'*P;
k2=-inv(r)*b'*P12;
bt=b;
```

```
b=baug;
c=[c 0 0];
d=[0];
x0=[zeros(10,1);
    zr0];
u=zeros(10001,1);
t=0:0.01:100;
[y,x]=lsim(a,b,c,d,u,t,x0);
%-------------------------------------------------------------------------
%Plotting system states, velocity loop input voltage and motor current
%-------------------------------------------------------------------------
for I=1:12;
    var_num = num2str(I);
    var = strcat('x',var_num);
    plot(t,x(:,I));
    title(var);
    grid on;
    pause
end
u=[-inv(r)*bt'*P  -inv(r)*bt'*P12]*x';
plot(t,u);
title('input');
grid
pause;
cur=([-Zz0*Zw0*[0 1 0 0 0 0 0 0]*K 1 -Zz0*1 0 0]*x'+ Zz0*N*u)/kf;
plot(t,cur);
pause
c=[0 0 0 0 1 0 0 0 0 0 0 0];
d=[1];
sys=ss(a,b,c,d);
bode(sys)
%-------------------------------------------------------------------------
% end of program
%-------------------------------------------------------------------------
```

```
%************************************************************************
%Program to simulate linear quadratic tracking regulator with estimator
%for GMRT servo system. The plant model and velocity controller are as per
%present system.
%************************************************************************
%-------------------------------------------------------------------------
%Parameters for Mechanical model of GMRT
%-------------------------------------------------------------------------
jm=.066768;
jg=3.5;
jc=2*0.995e6;
juc=8.9e6;
bm=.005424;
bg=4379.2;
bc=7.8336e6;
buc=1.34368e7;
kmg=1e7;
kgp=5.0864e9;
kad=1.428e9;
n=1505;
ns=12.6;
kf=.0625;
R1=24e3;
R2=24e3;
R3=24e3;
R4=20e3;
R28=180e3;
R66=100e3;
R25=27.4e3;
R26=13.7e3;
C4=0.22e-6;
C5=0.68e-6;
kt=0.56;
K=2*.1614/R25;
N=1/R26;
%-------------------------------------------------------------------------
%Velocity controller trasfer function coefficients.
%-------------------------------------------------------------------------
Zw0=R4*(R1+R2)/[R1*R2+(R1+R2)*(R3+R4)];
Zw1=R4/((R1*R2+(R1+R2)*(R3+R4))*C4);
Pw1=(R1+R3+R4)/((R1*R2+(R1+R2)*(R3+R4))*C4);
Pz1=1/((R28+R66)*C5);
Zz1=R66/((R66+R28)*C5);
Zz0=R28*R66/(R28+R66);
%-------------------------------------------------------------------------
%State Space representation of GMRT mechanical + motor model + velocity
%controller
%-------------------------------------------------------------------------
a=[0 1 0 0 0 0 0 0 0 0;
   -kmg/(jm*n^2) (-bm/jm-Zz0*Zw0*K*kt/(kf*jm)) kmg/(jm*n) 0 0 0 0 0
   kt*1/(jm*kf) -kt*1*Zz0/(jm*kf);
   0 0 0 1 0 0 0 0 0 0 ;
   kmg/(jg*n) 0 -(kmg+kgp/ns^2)/jg -bg/jg kgp/(jg*ns) 0 0 0 0 0 ;
   0 0 0 0 0 0 1 0 0 0 0;
   0 0 kgp/(jc*ns) 0 -(kgp+kad)/jc -(bc+buc)/jc kad/jc buc/jc 0 0;
   0 0 0 0 0 0 0 0 1 0 0;
```

```
   0 0 0 0 kad/juc buc/juc -kad/juc -buc/juc 0 0;
   0 -Zw0*K*(Zz1 -Zz0*Pz1) 0 0 0 0 0 0 -Pz1 -(Zz1-Zz0*Pz1);
   0 K*(Zw1-Zw0*Pw1) 0 0 0 0 0 0 0 -Pw1
   ];
b=[0; kt*Zz0*N/(jm*kf);0;0;0;0;0;0;N*(Zz1 - Zz0*Pz1);0];
c=[zeros(1,4) 1 zeros(1,5)];
d=[0];
sys=ss(a,b,c,d);
step(sys)
pause
%-------------------------------------------------------------------------
%Linear quadratic tracking regulator
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%State space model for refrence input. In this case it is a ramp input with
%a slope of 7.2e-5 rad/sec
%-------------------------------------------------------------------------
ar=[0 1; 0 0 ];
cr=[1 0];
zr0=[0; ((2*pi)/(360*60*60)*15)];
%-------------------------------------------------------------------------
%define auxilary matrices to convert linear quadratic tracking regulator
%problem to linear quadratic regulator problem.
%-------------------------------------------------------------------------
l=c'*inv(c*c');
[m,n]=size(a);
cuo=eye(m,n)-l*c;
%-------------------------------------------------------------------------
%Define weight matrices q1, q2 and r and find state feedback gains k1 and
%k2
%-------------------------------------------------------------------------
q1=[zeros(8,10);
    zeros(1,8) 0.01 zeros(1,1);
    zeros(1,10)];
q2=[1e7];
r=[0.001];
q=cuo'*q1*cuo+c'*q2*c;
clear cuo q1 q2 m n
qr=q;rr=r;

[k,P]=lqr(a,b,q,r);
P12=lyap((a'-P*b*inv(r)*b'),ar,(-q*l*cr));
%-------------------------------------------------------------------------
%Form closed loop matrices with Linear quadratic tracking regulator state
%feedback gains. Also  calculate system reponse to ramp input of 7.2e-5
%rad/sec. Here the actual input 'u' to plant is zero. But as input model is
%agumented with system matrices, the response of system is to non zero
%initial state of ramp input model which generates the necessary reference
%position.
%-------------------------------------------------------------------------
ac=a-b*inv(r)*b'*P;
atr=[ac -b*inv(r)*b'*P12;
    zeros(2,10) ar];
k1=-inv(r)*b'*P;
k2=-inv(r)*b'*P12;
btr=[b;0; 0];
```

```
ctr=[c 0 0];
dtr=[1];
%-------------------------------------------------------------------------
%Linear Quadratic Estimtor
%-------------------------------------------------------------------------
clear l k P P12 r q
c=[0 1 zeros(1,8);
    zeros(1,4) 1 zeros(1,5);
    zeros(1,8) 1 0];
d=zeros(3,1);
%-------------------------------------------------------------------------
%Defining weight matices
%-------------------------------------------------------------------------
r=[1 0 0;
    0 0.001 0;
    0 0 0.1];
q=[10e5 zeros(1,9);
    0 96e12 zeros(1,8);
    0 0 0 zeros(1,7);
    zeros(1,3) 0 zeros(1,6);
    zeros(1,4) 0 zeros(1,5);
    zeros(1,5) 0 zeros(1,4);
    zeros(1,6) 0 zeros(1,3);
    zeros(1,7) 0 0 0;
    zeros(1,8) 20 0;
    zeros(1,9) 10e5];
%-------------------------------------------------------------------------
%Calculating Estimator gains
%-------------------------------------------------------------------------
[k,P]=lqr((a+b*k1)',c',q,r);
ke=-P*c'*inv(r);
clear k P
qe=q;re=r;
%-------------------------------------------------------------------------
%Forming system matices for agumented system of tracker plus estimator
%-------------------------------------------------------------------------
ae = a+ke*c+b*k1;
bt=-ke;
be=[b*k2 bt];
ce=eye(10,10);
de=zeros(10,5);
clear bt;
%-------------------------------------------------------------------------
%Defining matices to checking system response
%-------------------------------------------------------------------------
x0=[zeros(10,1);
    zr0];
ctr=[c zeros(3,2)];
dtr=[0;0;0];
u=zeros(1001,1);
t=0:.1:100;
[y,x]=lsim(atr,btr,ctr,dtr,u,t,x0);                %Actual system response
ye=lsim(ae,be,ce,de,[x(:,11:12) y],t);           %Estimated System Response
%-------------------------------------------------------------------------
%Plotting system response
%-------------------------------------------------------------------------
```

```
for i=1:10
    var_num = num2str(i);
    t1 = strcat('error ','x',var_num);
    t2 = strcat('x',var_num);
    plot(t,(x(:,i)-ye(:,i)));
    title(t1);
    grid on
    pause
    plot(t,x(:,i))
    hold on
    plot(t,ye(:,i),'r')
    title(t2);
    pause
    clf
end
%-------------------------------------------------------------------------------
%End of program
%-------------------------------------------------------------------------------
```

References

[1] B. C. Joshi, "Simulation study of servocontrol system of 45 meter giant metrewave radio telescope," Master's thesis, Indian Institute of Technology, Bombay, 1992.

[2] GMRT. Servo position loop principle. [Online]. Available: http://www.gmrt.ncra.tifr.res.in/gmrt_hpage/sub_system/servo/prin.html

[3] N. Nise, *Control Systems Engineering*. John Wiley, 2004.

[4] A. Dobos, A. Kader, D. Luong, and M. Piper. (2005) Systen identification of dc motor. [Online]. Available: http://www.sccs.swarthmore.edu/users/06/adem/engin/e58/lab2/

[5] W. Gawronski and K. Souccar, "Control systems of large millimeter telescope," *IEEE Antennas Propag. Mag.*, vol. 47, pp. 41–49, Aug. 2006.

[6] M. Gopal, *Modern Control System Theory*. New Age International Publishers, 1993.

[7] B. D. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Dover Publications, INC, 2007.

[8] W.Gawronski, "Control and pointing challenges of antennas and telescopes," in *American Control Conference*, Portland,OR,USA, Jun. 2005, pp. 3758–3769.

[9] W. Gawronski, "Modeling wind-gust disturbances for the analysis of antenna pointing accuracy," *IEEE Antennas Propag. Mag.*, vol. 46, pp. 50–58, Feb. 2004.

[10] L. Ljung, *System Identification: Theory for the User*. Prentice Hall PTR, 1998.

[11] Gilles. (2008) Simulink pseudo-random binary sequence generators. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/18434

[12] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995.

[13] W. Gawronski, *Modeling and Control of Antennas and Telescopes*. Springer US, 2008.

[14] ——, *Advanced Structural Dynamics and Active Control of Structures*. Springer US, 2004.

[15] Mathworks. (2008) xpc key features. [Online]. Available: http://www.mathworks.com/products/xpctarget/description1.html

[16] W. Gawronski and W. Almassy, "Command preprocessor for radio telescopes and microwave antennas," *IEEE Antennas Propag. Mag.*, vol. 44, pp. 30–37, Apr. 2002.

[17] Mathworks. (2009) Matlab getting started guide. [Online]. Available: http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf

[18] ——. (2009) Simulink getting started guide. [Online]. Available: http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/sl_gs.pdf