Giant Metrewave Radio Telescope

National Centre for Radio Astrophysics

TATA INSTITUTE OF FUNDAMENTAL RESEARCH

Technical Report

"Testing of Brushless motor, Drive (large test setup) with Programmable Multi Axis Controller Configured as Position Loop and Velocity Loop"

By
Srinivasarao Beera
Engineer-C (FTA)
GMRT-TIFR
Pune, India

Under the guidance of

Dr. Bal Chandra Joshi
Mr. Suresh Sabapathy
Servo Group
GMRT-NCRA-TIFR
Pune, India

Table of Contents
Brushless Drives and Motors

A) Testing Brushless motor (Large test setup) with PMAC configured in Position Loop

B) Testing Brushless motor (Large test Setup) with PMAC configured in Velocity Loop

Date: - 10/03/2010

Testing Brushless Motor (large test setup) with PMAC configured in Position Loop

1) **AIM: -** To Measure the position accuracy with PMAC configured in Position Loop. Driving externally commutated Brushless motor.

2) **Block diagram of Test Setup Arrangement:-**

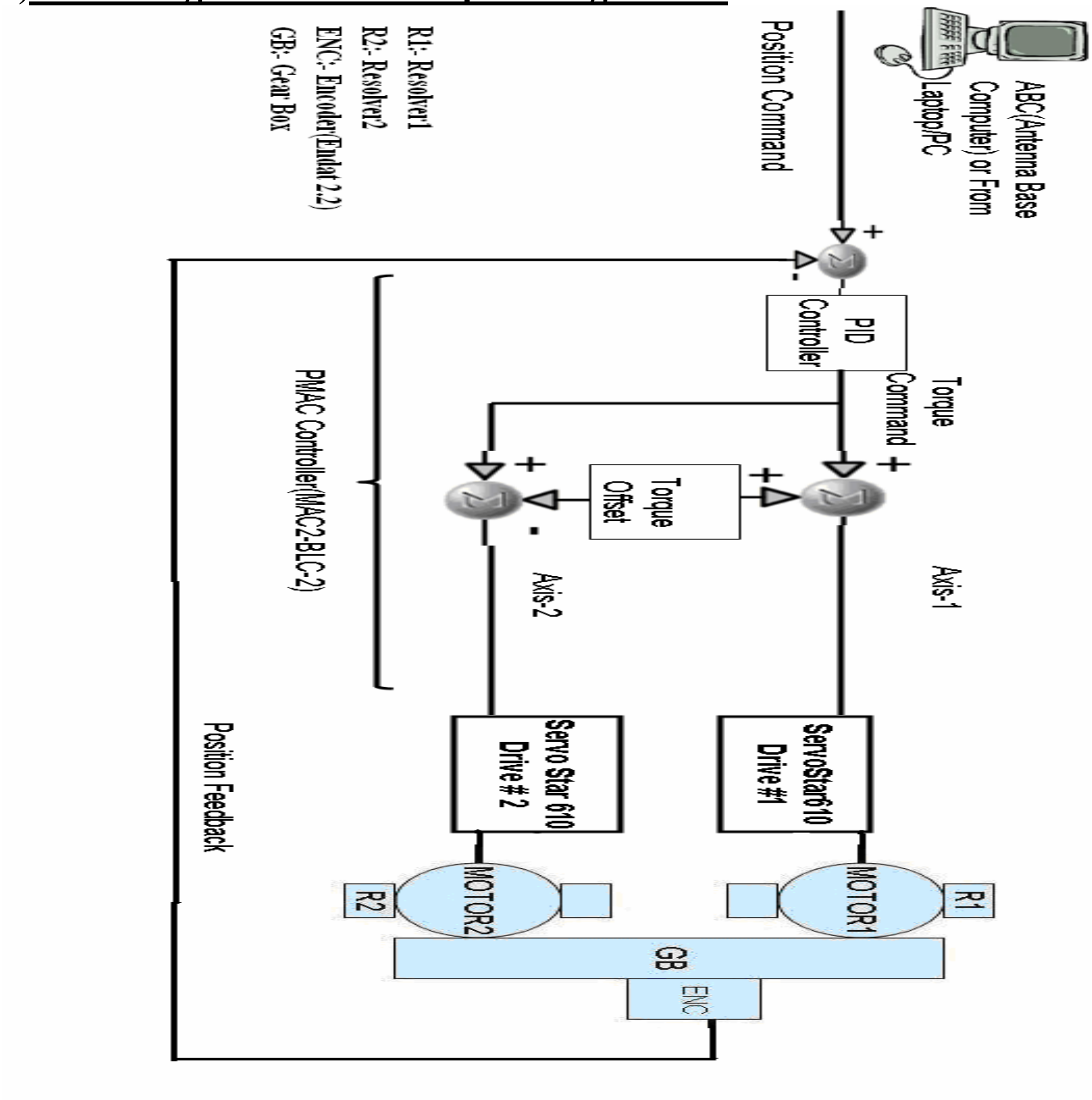


Fig-1

3) **Procedure for tuning of Brushless motor:-**

3.1) Connect Drives Servo Star 610 number-1 & 2 to PMAC Channels #1 and #2.

3.2) Connect Absolute Encoder ROC417 with interpolator (IBV102) to PMAC (2) Channel #5.

3.3) Tune the system for position loop and not as velocity loop (PMAC receives position command from PC/Laptop and the servo loop is configured for position).The diagram (fig-2) below shows the PID filter of PMAC position/velocity loop.

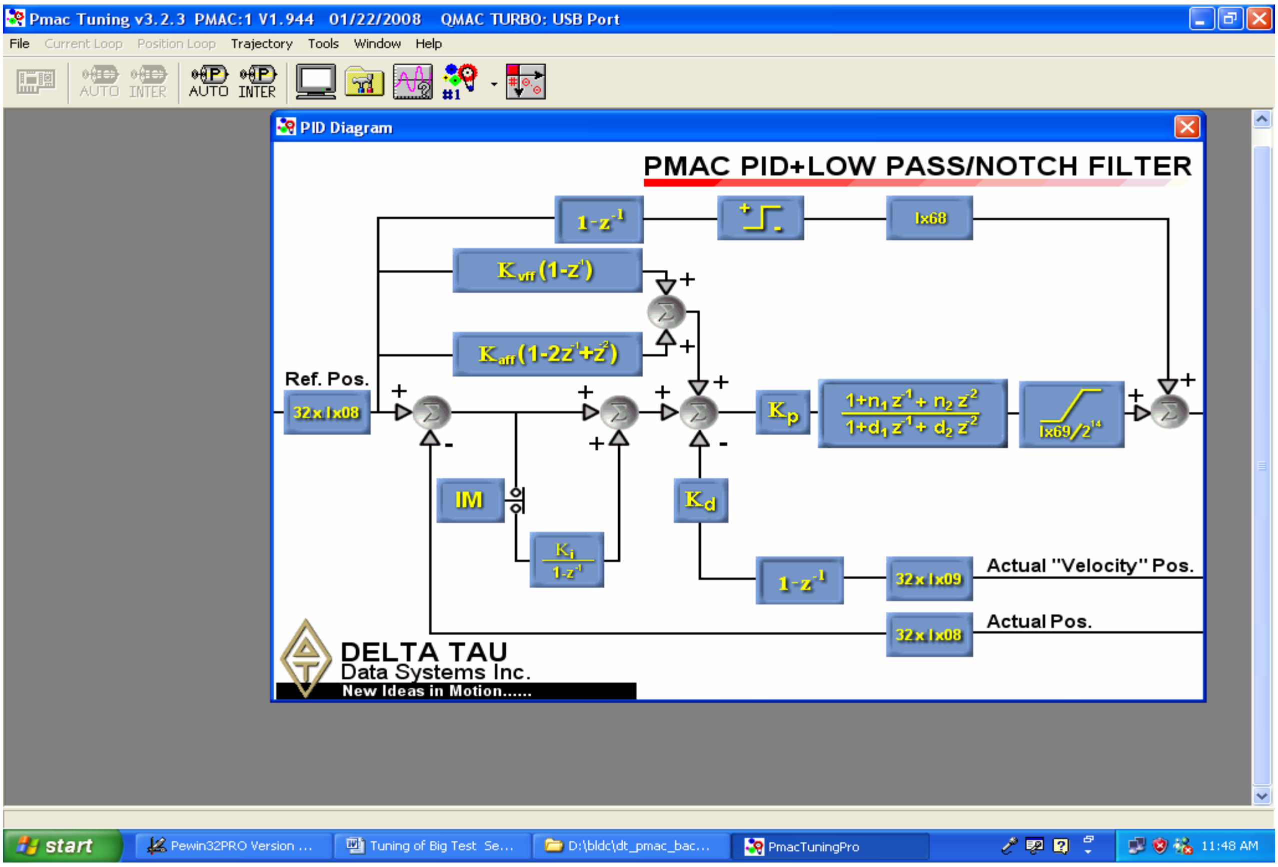


Fig-2

3.4) Move the test setup in open loop initially with only motor #1 (with motor #2 idling and load motor disconnected) for preliminary tuning. After tuning the values of various PID filter are

Proportional Gain (Kp)	= 5, 00,000
Differential Gain (Kd)	= 10,000
Velocity Feed Forward Gain (Kvff)	= 10,000
Integral Gain (Ki)	= 10,000

Similarly repeat the tuning for motor #2 (with motor #1 idling and load motor disconnected). The tuned values are almost the same as for motor #1.

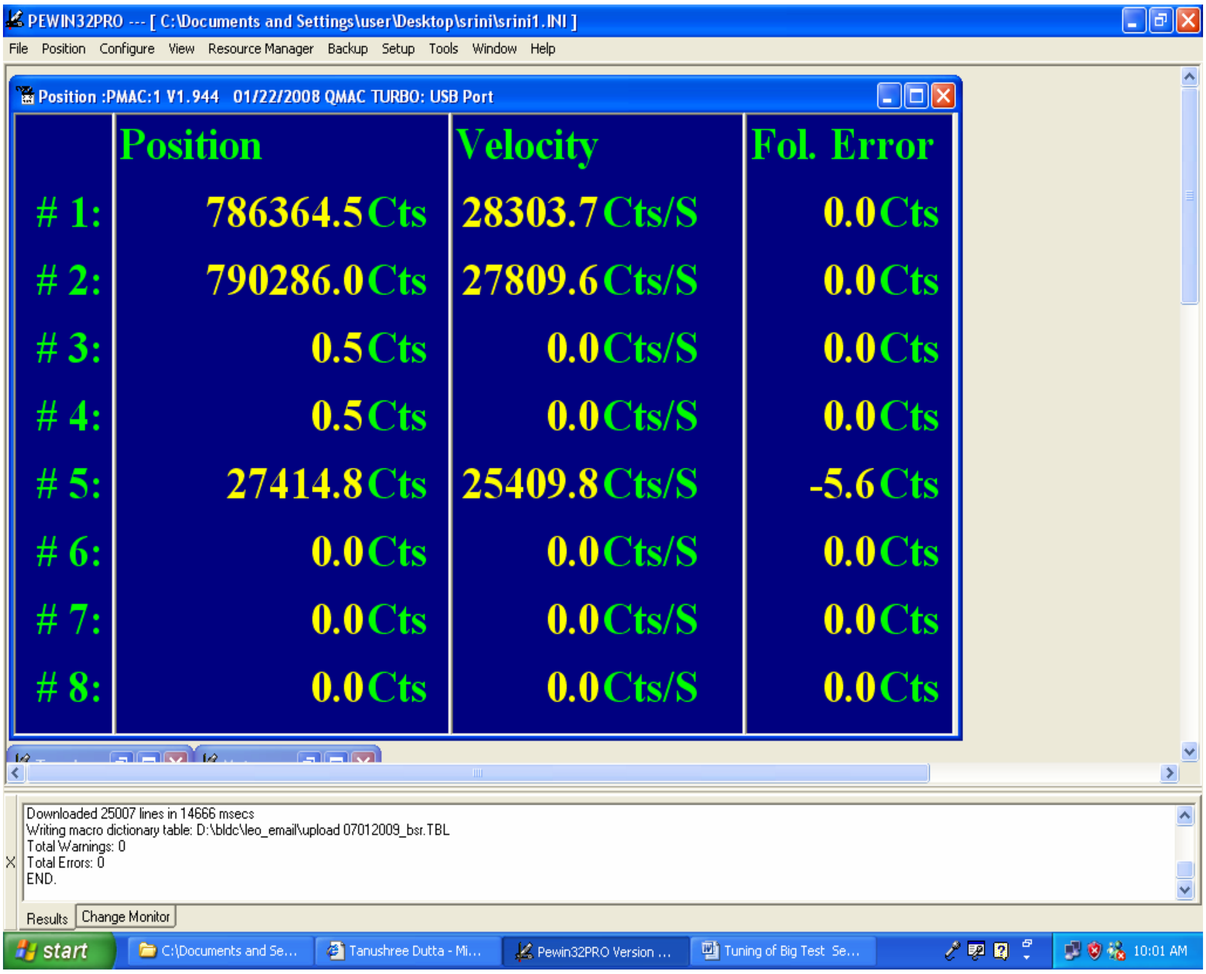
3.5) After completion of preliminary tuning of motors #1 and #2 individually, couple both motors to central gear (Ratio=73/19) and check the movement of motors #1 and #2 and load encoder.

3.5. A) Motor #1 to be moved in open loop by giving 7% torque command (#1o7).

3.5. B) Motor #2 to be kept idling (enabled by releasing only the brakes by #2o0).

3.5. C) Note the resolver counts of both motors and encoder counts of channel #5 in PMAC to check the polarity of position and velocity as shown in fig-3.

Fig-3



4. Procedure for Backlash Measurement:-

4.1) This is done by keeping motor #1 in closed loop "holding position" with command #1j/ and moving motor #2 in open loop by giving varying torque from 3% (#2o3)to 10% (#2o10) and Noting the counts in channels #2 and #5 of PMAC.Now repeat step 4.1) by giving varying torque in opposite direction by giving commands (#2o-3) to (#2o-10) and note down the counts in channels #2 and #5 of PMAC.

4.2) Repeat above tests with Motor # 2 in closed loop and motor #1 in open loop.

4.3) Tabulate the result as below and the backlash measured is 1110 counts which correspond to around 7% of maximum continuous torque.

Commands	Motor Encoder (Resolver)	Load Encoder
#1j/		
#1hmz#2hmz#5hmz		
#2o-10	#2: -36013	#5: -1456
#2o10	#2: 34318	#5: 1385
	70331	2841
#2j/		
#1hmz#2hmz#5hmz		
#1o10	#1: 39209	#5: 1022
#1o-10	#1: -33039	#5: -1172
	72248	2194
#1j/		
#1hmz#2hmz#5hmz		
#2o-5	#2: -10732	#5: -399
#2o5	#2: 27597	#5: 828
	38329	1227
#2j/		
#1hmz#2hmz#5hmz		
#1o5	#1: 36716	#5: 1389
#1o-5	#1: -981	#5: -139
	37697	1527

5. Moving the motor with Backlash algorithm

5.1) The measured backlash is implemented as an algorithm PLC0 which is given in the **Annexure - A**

5.2) Download algorithm 'plc0' along with suggested Variables of ' I 'and 'M', user defined variables of 'P'. All these values are edited in file name **"upload 07012009_bsr.CFG"**

5.3) Open PEWIN32PRO.exe; click Tab 'File' and ' 'download'. Wait for no errors.

5.4) When there are no errors after downloading, open TERMINAL Window in the 'view' pane and give commands as below :

```
Enable PLC0     ; plc0 is activated / enabled
#1o0            ; motor #1 brakes release and amplifier enabled in open loop
#2o0            ; motor #2 brakes release and amplifier enabled in open loop
#5o0            ; motor #5 brakes release and amplifier enabled in open loop
#5j+            ; move command to dummy axis #5 in closed loop.
```

This enables first the PLC and enables CH #1,#2,#5 of PMAC where both drives and encoder are connected and gives a move command to CH #5 to which the 17 bit encoder with interpolator is connected (shown in Block Diagram fig1)

6. Measuring Accuracy of position at various speeds of Motor

With both motors in pre-loaded condition move the system at various speeds and take plots for position accuracy at i522 = 0.1, 1 and 10 counts/m-sec which corresponds to rpm of motor 666, 66 and 6.6.

6.1) The plots are enclosed in the next several pages (Annexure **- B**) of this document and are self explanatory. The preliminary conclusion is, peak to peak error in position varies from a maximum of +/_ 25 cts (for 666 rpm) to +/_ 4 cts (for 6.6 rpm).

7. S-Curve Responses

In addition to the accuracy measurements and plots, a 'S-curve' trajectory is given as position command to CH #5 to check for accuracy of position and a good tuning for a gradual change in acceleration in position command. These plots are also enclosed with these documents at the end **(Annexure - C).**

18/03/10

8. Measurement of Bandwidth and Locked Rotor Frequency – LRF of Large test setup with PMAC configured as Position loop

1). Tested Brushless motor (Large test setup) with PMAC in position loop with Backlash algorithm. Observed Position Error for various speeds.

2). For finding Bandwidth of the System (Position loop of the PMAC+current loop of Servo star), have given sine move through PMAC TUNING PRO at various frequency i.e. 0.1hz,0.5 Hz ,1 hz,1.5hz ,2 hz,10hz,25hz and 100hz to PMAC(position loop) connected Servo star 610(Current loop) and this has connected Motor. Calculated Magnitude and phase from these plots .Drawn Bode plot by using magnitude and phase.

3) Steps 1) and 2) completed and the preliminary figures are

Position loop BW is = 24Hz (using Bode magnitude plot)
Locked rotor frequency - LRF = ?

The various response plots for **'Sine Wave'** input are as shown below.
Sine Wave Input to PMAC position loop is given through' ''pmac tuning pro' software and responses are gathered in a 'data gathering buffer' and plotted. The frequencies chosen are 0.1, 0.5, 1.0, 1.5, 2.0, 10, 25 Hz.

From each of these response plots the **magnitude** ratio and **phase** difference between Commanded wave (blue color) and actual wave (magenta color) are taken for each frequency and the BODE plot is plotted. **The 3dB bandwidth** is calculated from the magnitude BODE plot and is around 24Hz.

Sine @ 0.1Hz:-

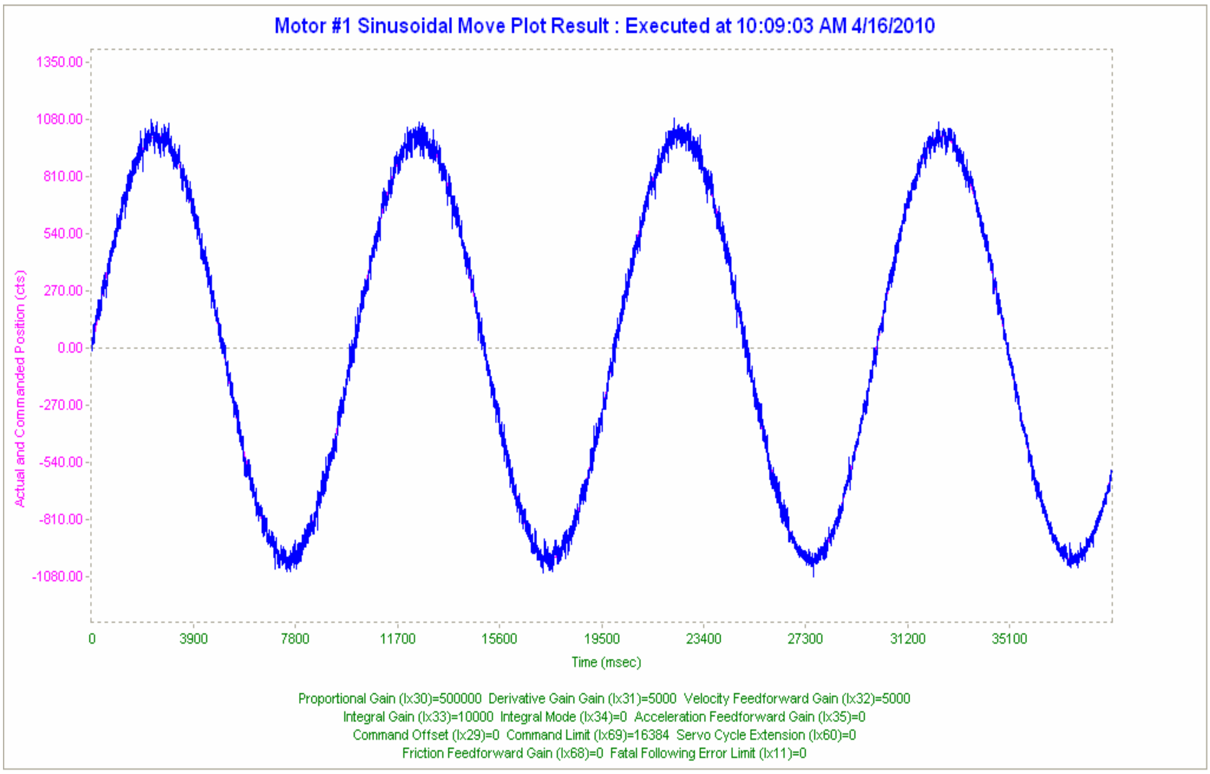


Figure 1 - Response to a Sine Wave input freq = 0.1 Hz given through pmac tuning pro

Calculations of Magnitude and phase from each frequency response plots **f = 0.1 Hz**

Magnitude ratio = Output/Input = Actual Pos. /Commanded Pos. = 1031/1031=1
Magnitude in dB = 20 log 1 = 0dB
Phase difference = 0 ms;
10000 ms - 360deg
0 ms - 0 x 360/10000 ms deg
0 ms **= - 0 deg.**

Sine_1Hz:-

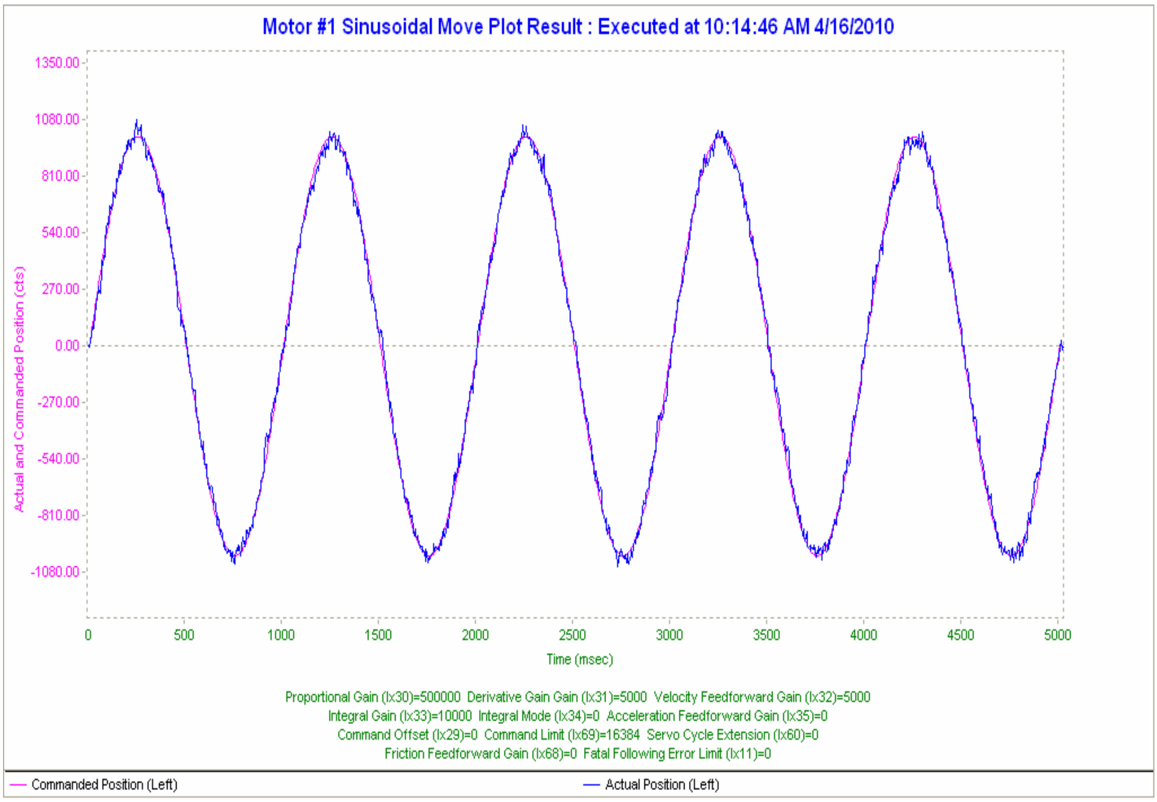


Figure 2 - Response to a Sine Wave input freq = 1 Hz given through pmac tuning pro

Calculations of Magnitude and phase from each frequency response plots **f = 1 Hz**

Magnitude ratio = Output/Input = Actual Pos. /Commanded Pos. = 994/994=1
Magnitude in dB = 20 log 1 = 0dB
Phase difference = 0 ms;
1000 ms - 360deg
0 ms - 0 x 360/1000 ms deg
0 ms = **- 0 deg.**

Sine_10Hz:-

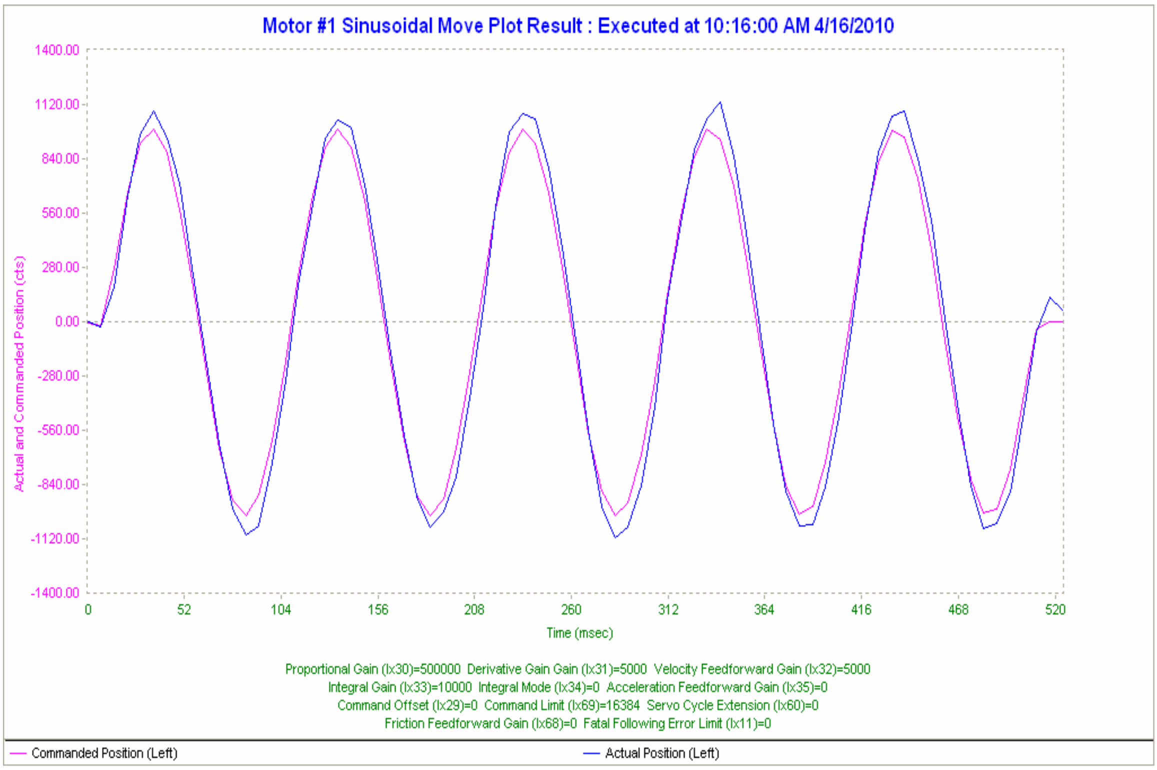


Figure 2 - Response to a Sine Wave input freq = 10 Hz given through pmac tuning pro

Calculations of Magnitude and phase from each frequency response plots **f = 10 Hz**

Magnitude ratio = Output/Input = Actual Pos. /Commanded Pos. = 1074/998=1.076
Magnitude in dB = 20 log 1.076 = 0.637
Phase difference = 0 ms;
100 ms - 360deg
0 ms - 0 x 360/100 ms deg
0 ms = **- 0 deg.**

Sine_20Hz

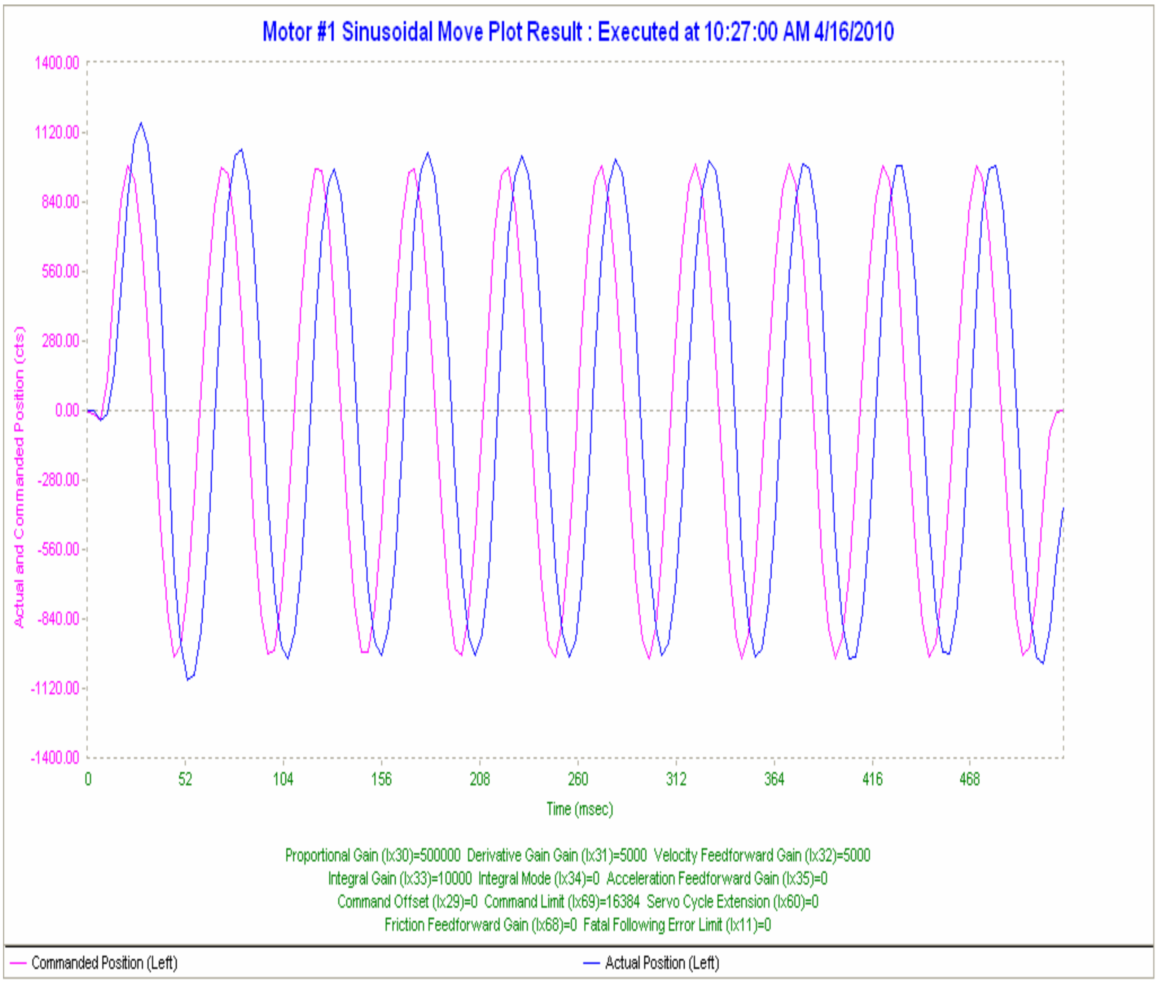


Figure 3 - Response to a Sine Wave input freq = 20 Hz given through pmac tuning pro

Calculations of Magnitude and phase from each frequency response plots **f = 20 Hz**

Magnitude ratio = Output/Input = Actual Pos. /Commanded Pos. = 1017/992=1.025
Magnitude in dB = 20 log 1.025 = 0.216
Phase difference = 7 ms;
49 ms - 360deg
7 ms - 7ms x 360/49 ms deg
0 ms = **- 51 deg.**

Sine_25Hz

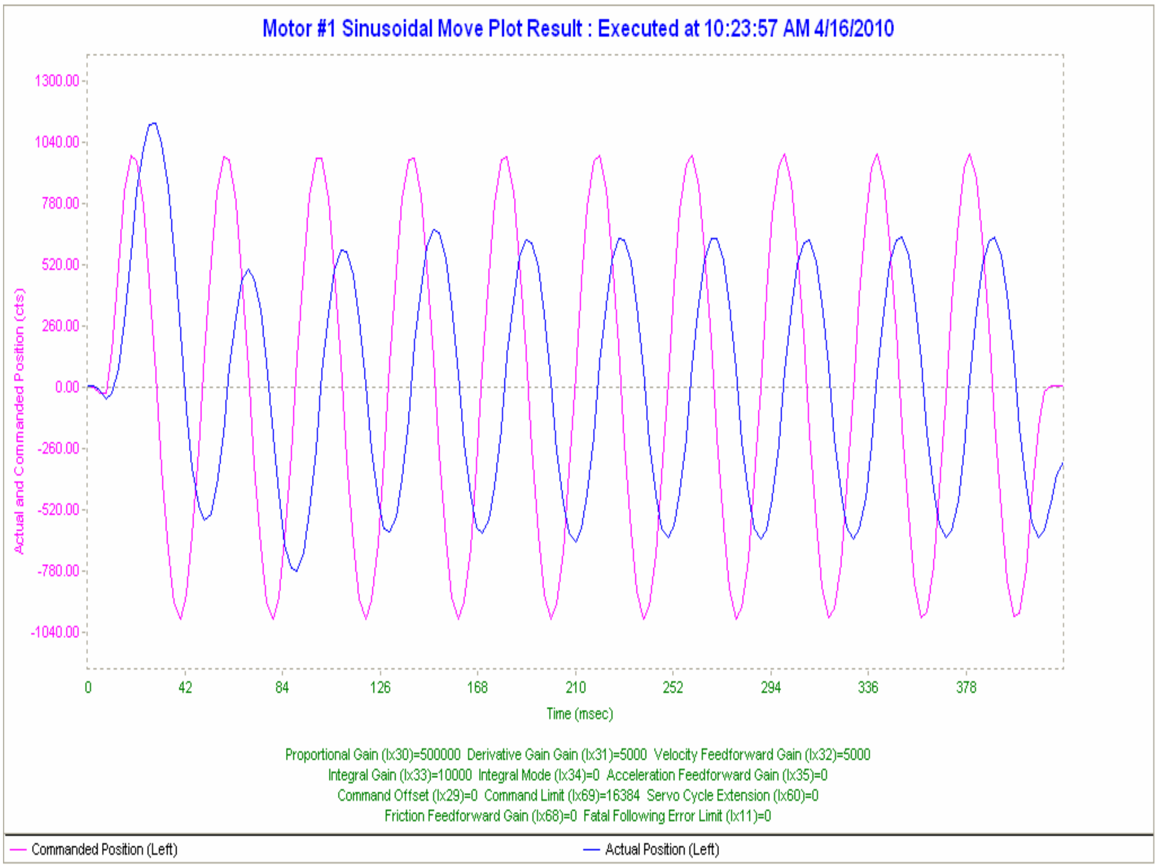


Figure 4 - Response to a Sine Wave input freq = 25 Hz given through pmac tuning pro

Calculations of Magnitude and phase from each frequency response plots **f = 25 Hz**

Magnitude ratio = Output/Input = Actual Pos. /Commanded Pos. = 635/980=0.68
Magnitude in dB = 20 log 0.68 = -3.31
Phase difference = 11 ms;
40 ms - 360deg
11 ms - 11 ms x 360/40 ms deg
0 ms = **- 99 deg.**

Sine_100Hz

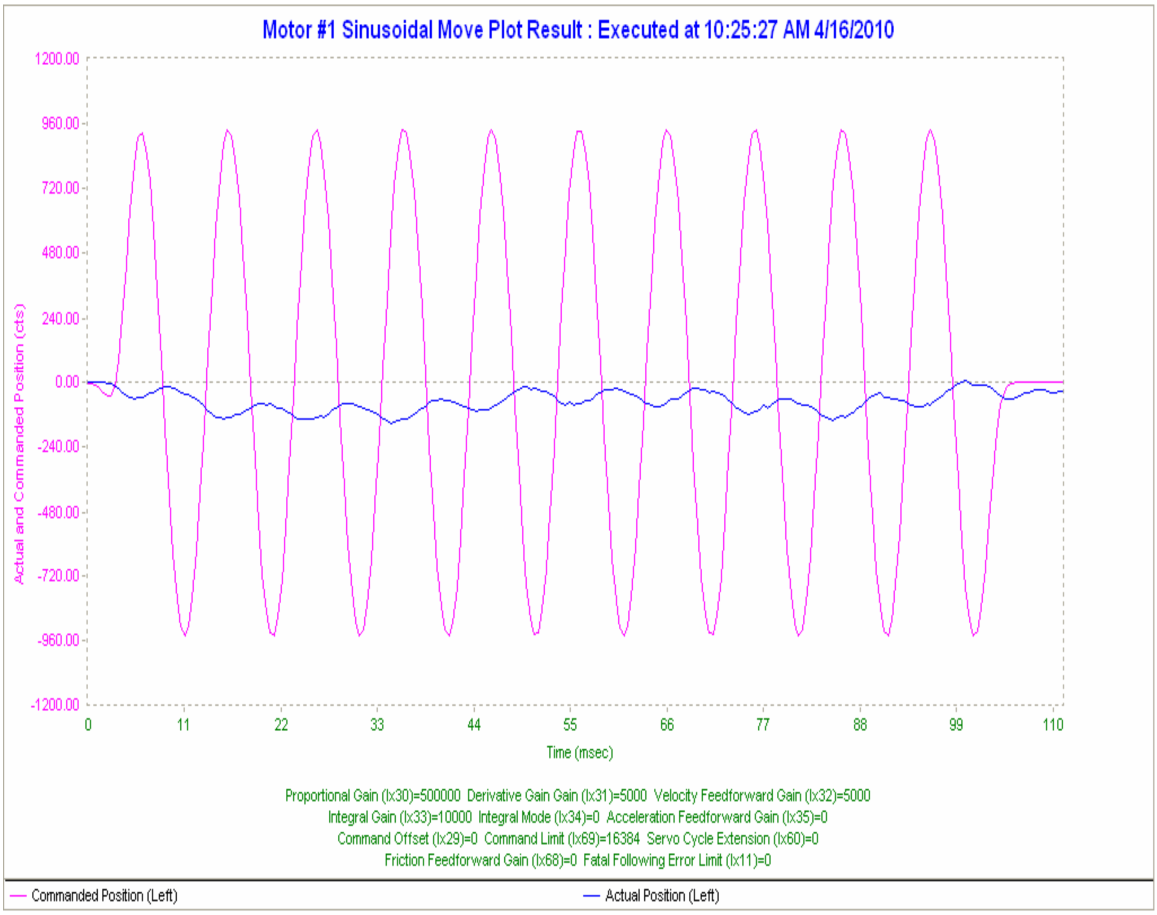
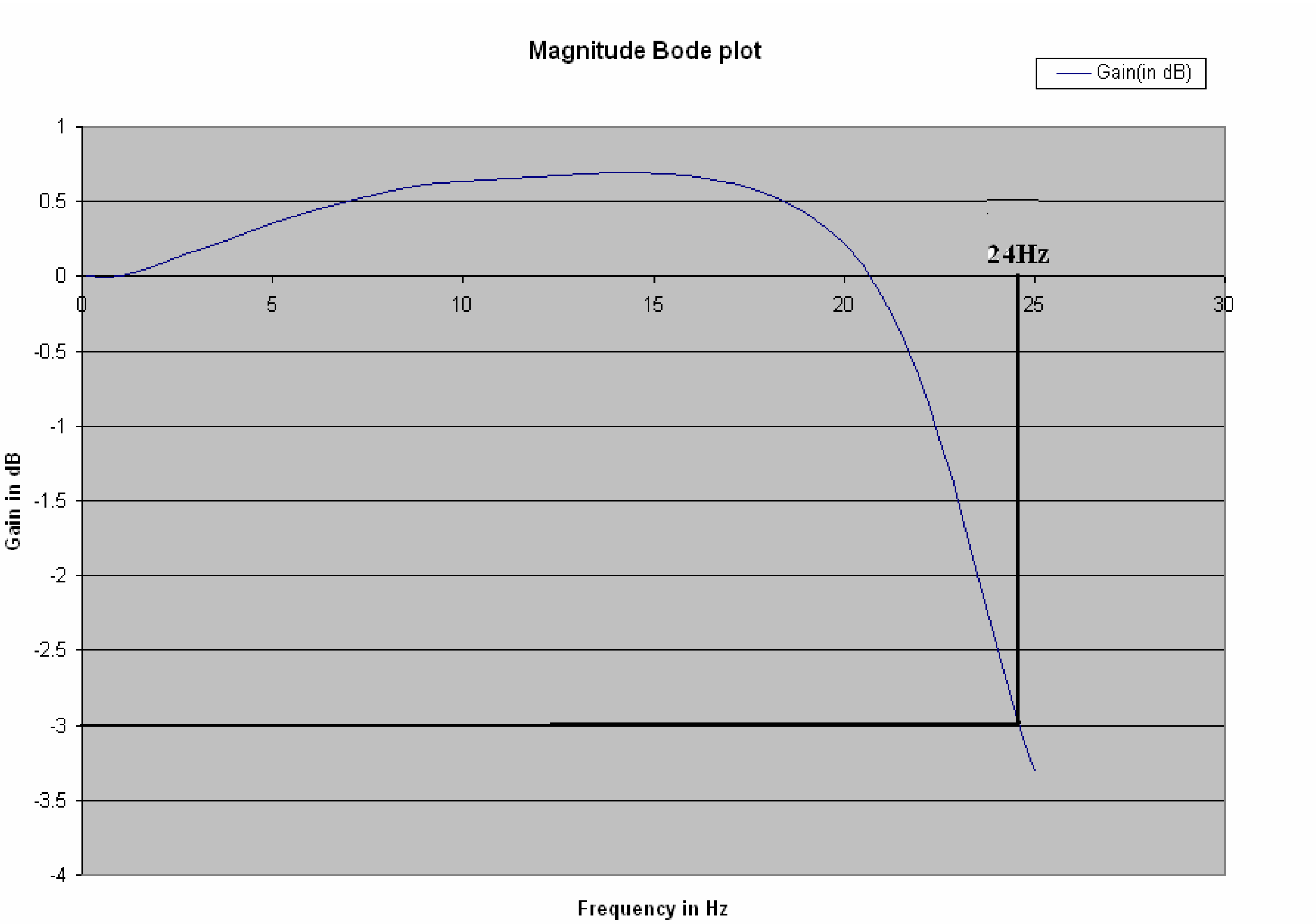


Figure 5 - Response to a Sine Wave input freq = 100 Hz given through pmac tuning pro

<u>The Following table shows Frequency Vs Magnitude:-</u>

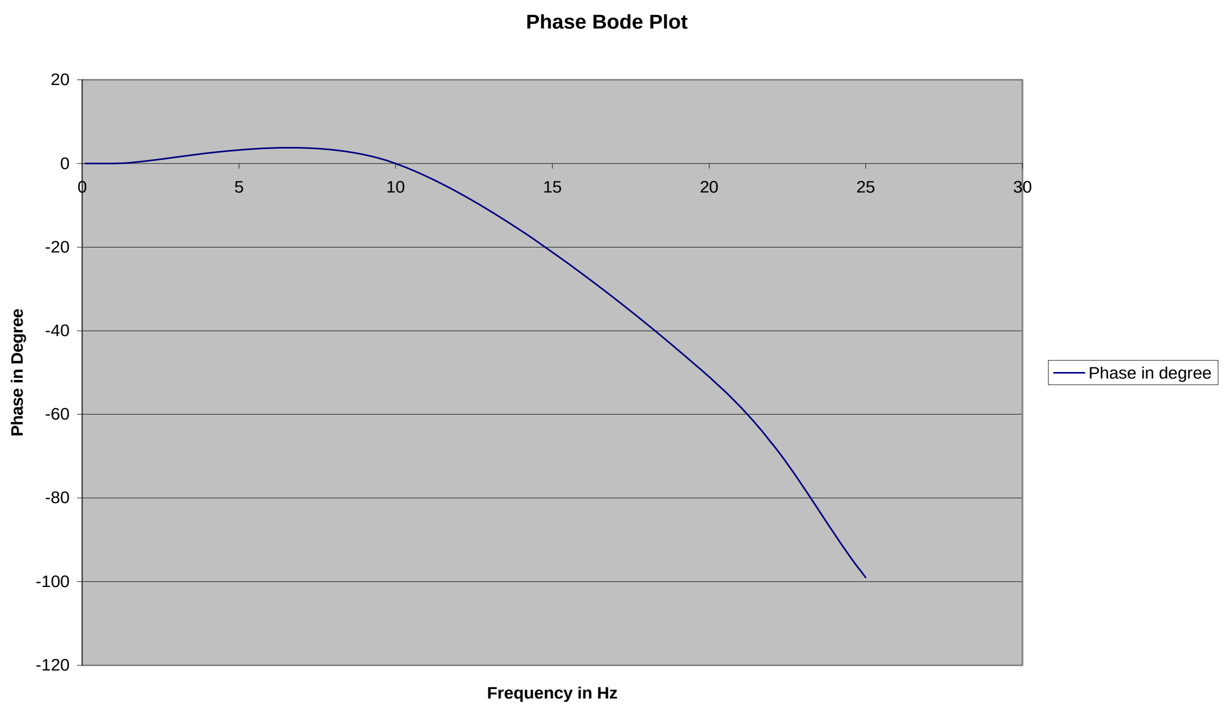
Frequency(Hz)	Gain(in dB)
0.1	0
1	0
10	0.635
20	0.216
25	-3.31



The Bandwidth of Position Loop in PMAC according to above plot is 24Hz.
Resonant peak is 0.7dB and resonant frequency is 15Hz.

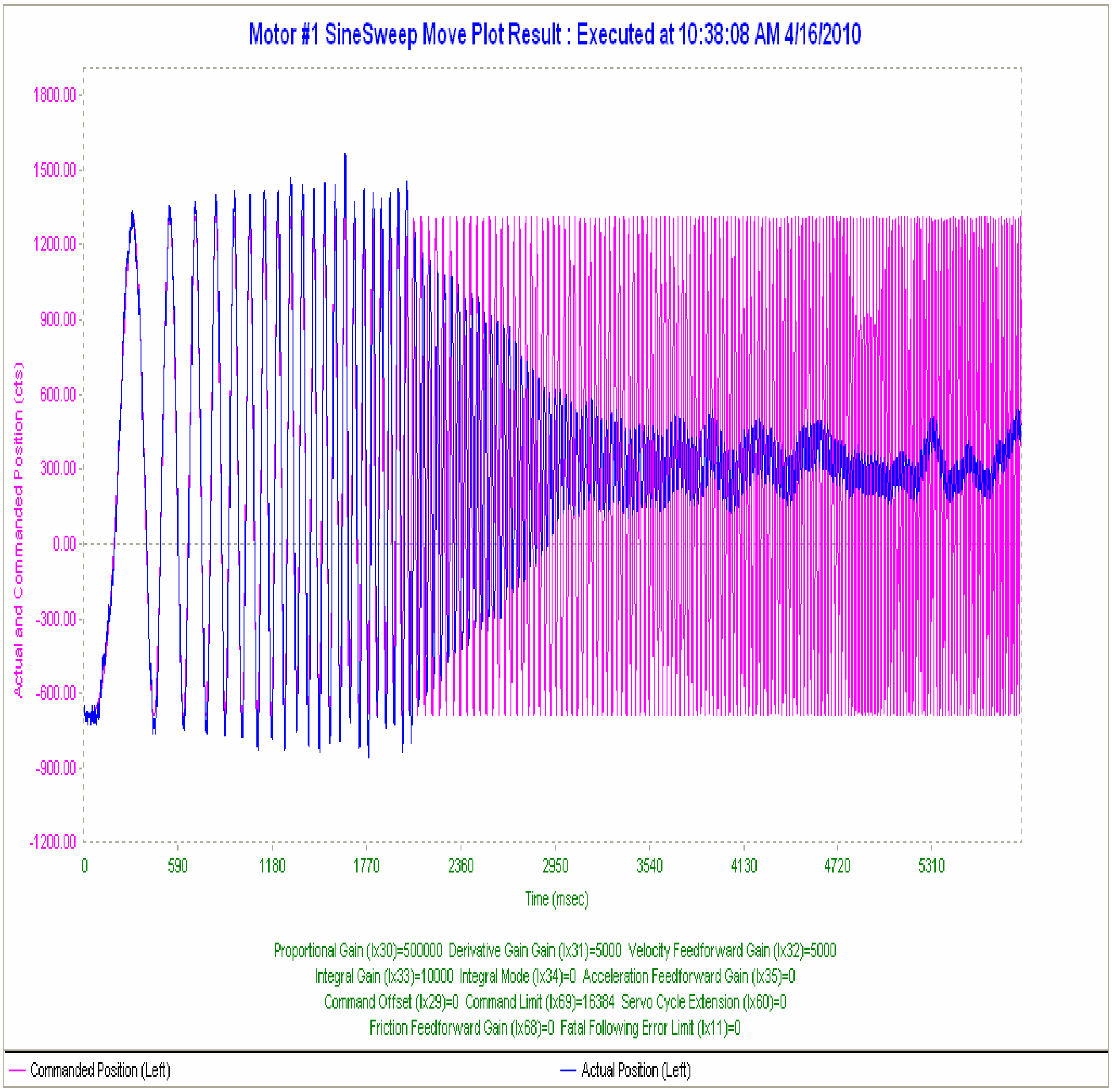
The Following Table shows' Frequency ' Vs ' Phase':-

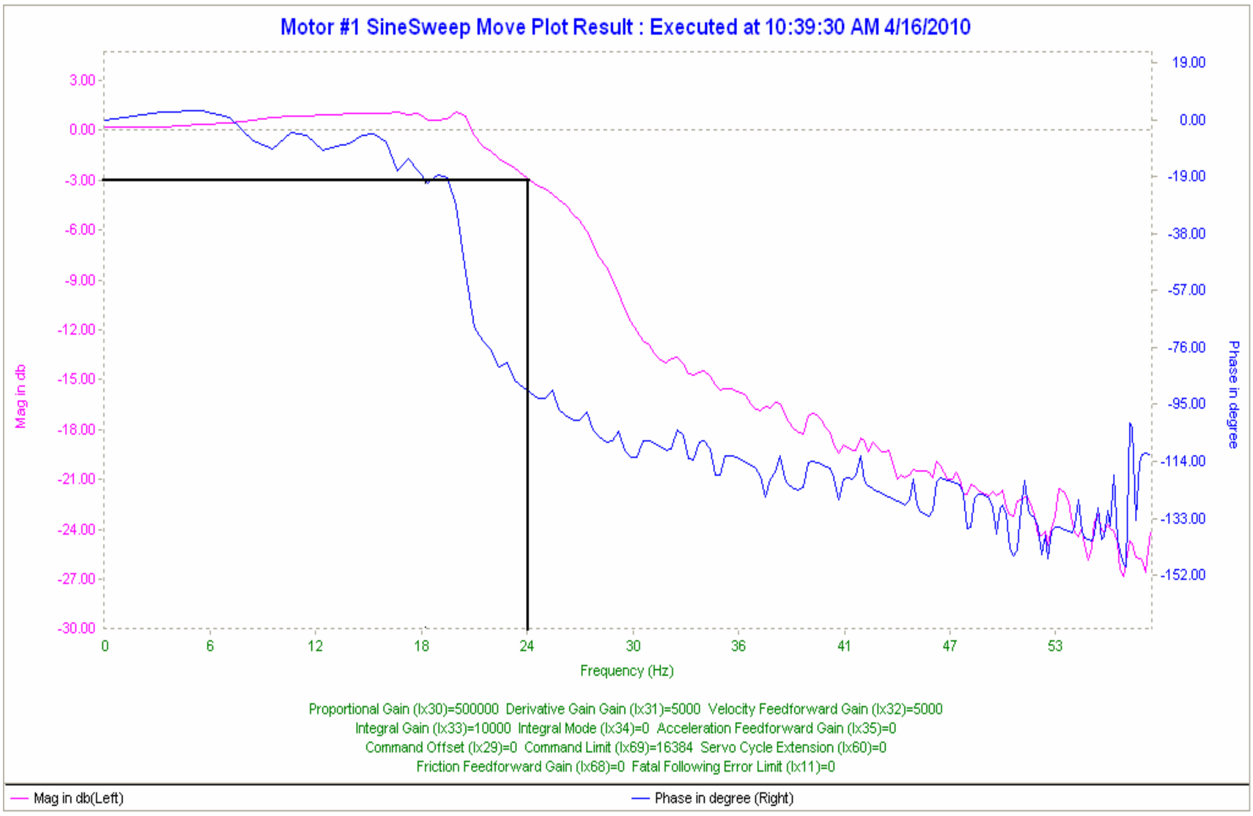
Frequency(Hz)	Phase in degree
0.1	0
1	0
10	0
20	-51
25	-99


Phase Bode Plot
20
0
-20
-40
-60
-80
-100
-120
0
5
10
15
20
25
30
Phase in Degree
Frequency in Hz
Phase in degree

9. Sine Sweep response in Position Loop

Sine Sweep (0.1Hz to 100 Hz and Sweep time=10s):-





The Bandwidth of Position Loop according above Sine Sweep Bode Plot is 24 Hz
The Resonant peak is approximately 1 dB and resonant frequency is 20Hz.

Conclusion: The bandwidth measured from Bode plot plotted directly from PMAC is same as our measurements and calculations from Sine responses plotted for various frequency ranges from 0.1 Hz to 100Hz fed externally from an "Arbitrary Waveform generator"

Date: - 19/03/2010
SS/BSR.

Testing Brushless motor(large test setup)with PMAC configured in Velocity Loop

10. Linearity check of ADC of MAC2-BLC-2:-

1) +/- 10 Volt analog voltage is connected through D type female connector to X11 (ANA CH #7) of back lash compensator MAC2-BLC-2.
   Pin2 of connector will go to +ve of the10V Supply.
   Pin6 of connector will go to –ve of the 10V supply.

2) Go to 'PEWIN32 Pro' and Download file "**bsr_adc_blc_19032010.pmc**" which has PLC30 program as given below.

```
i5=3                    ; PLC program ON for enabling in Terminal window
i7106=$1FFFFF           ; ADC strobe word default value for A/D conversion.
M5063->Y: $78115, 8, 16, S        ; analog i/p connected to ch# 7 of BLC
M5064->Y: $7811D, 8, 16, S        ; analog i/p connected to ch# 8 of BLC

Open plc30 clear

If (m5063 > 16383)
P0= (m5063-32768)*10/16383
Else
P0=m5063*10/16383
EndIf

Close;

; Memory location that store the 16 bit digital data proportional to analog
; Input signed form, not in unsigned form. Signed form will give digital
; Value from -32,678 to +32,678. Unsigned form will give only 0 to 65,536
; Digital values. Since we have used -/+ 10 analog voltage operation, so we
; have to take 'S'.
```

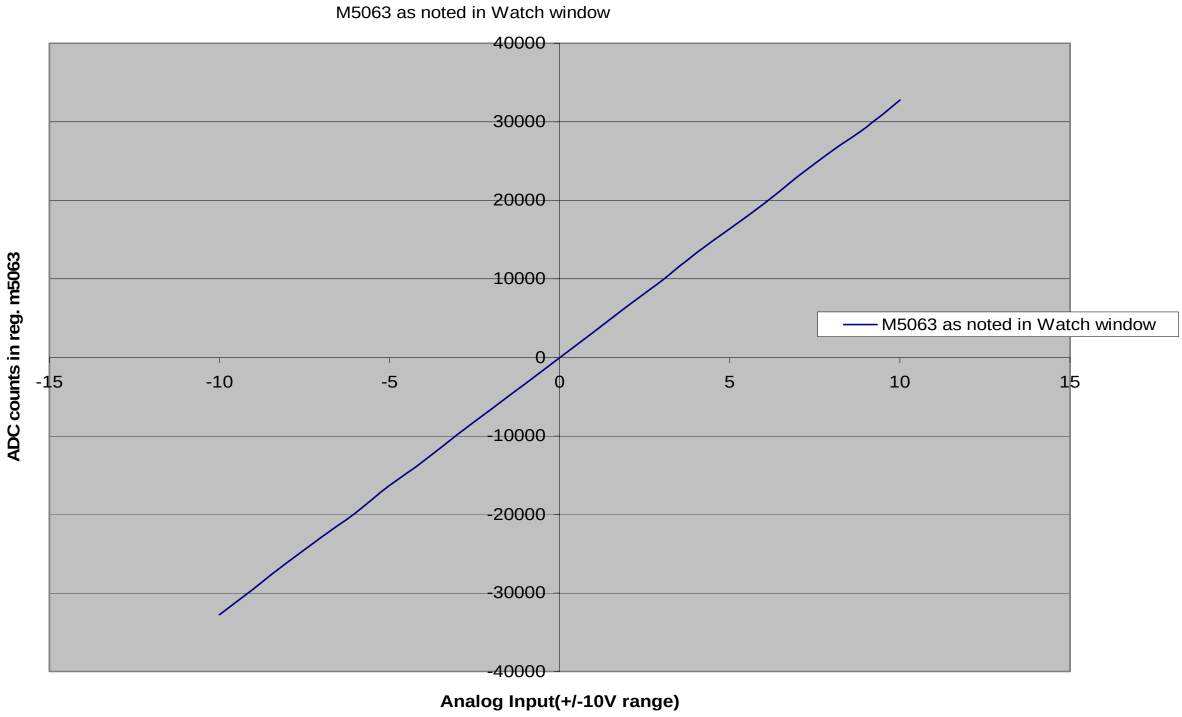
3) After downloading the file , go to Terminal Window and run the PLC30 by the Command ENABLE PLC30.

4) In VIEW menu open Watch Window add the M5063 which will show the digital value proportional to analog input.

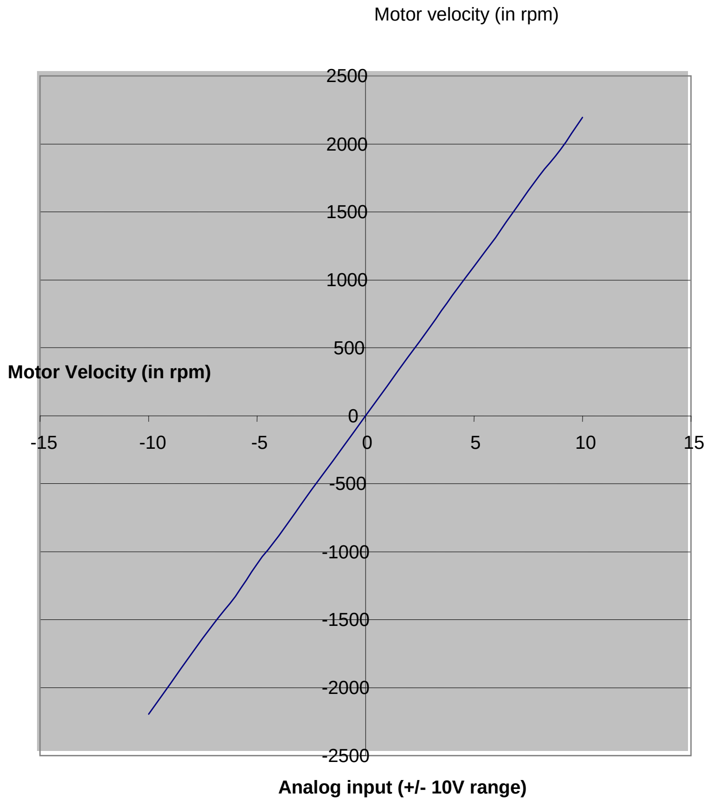
Observation: - M5063 values are noted in the Watch Window for various input values ranging from -10V to 0V and 0V to +10V analog voltage given from an external power source to the ANA CH #7.The values are tabulated in counts and plotted in counts and motor RPM below.

Analog Input	M5063 as noted in Watch window
-10 volts	-32768 counts
-9	-29441
-8	-26092
-7	-22890
-6	-19828
-5	-16311
-4	-13202
-3	-9802
-2	-6531
-1	-3333
0	-24
1	3258
2	6530
3	9781
4	13223
5	16405
6	19572
7	23020
8	26300
9	29253
10	32767

Plot between M5063 (ADC value in counts) and Analog input (+/-10V)



Plot between Motor Velocity (in rpm) and Analog input (+/-10V)



Conclusion: -

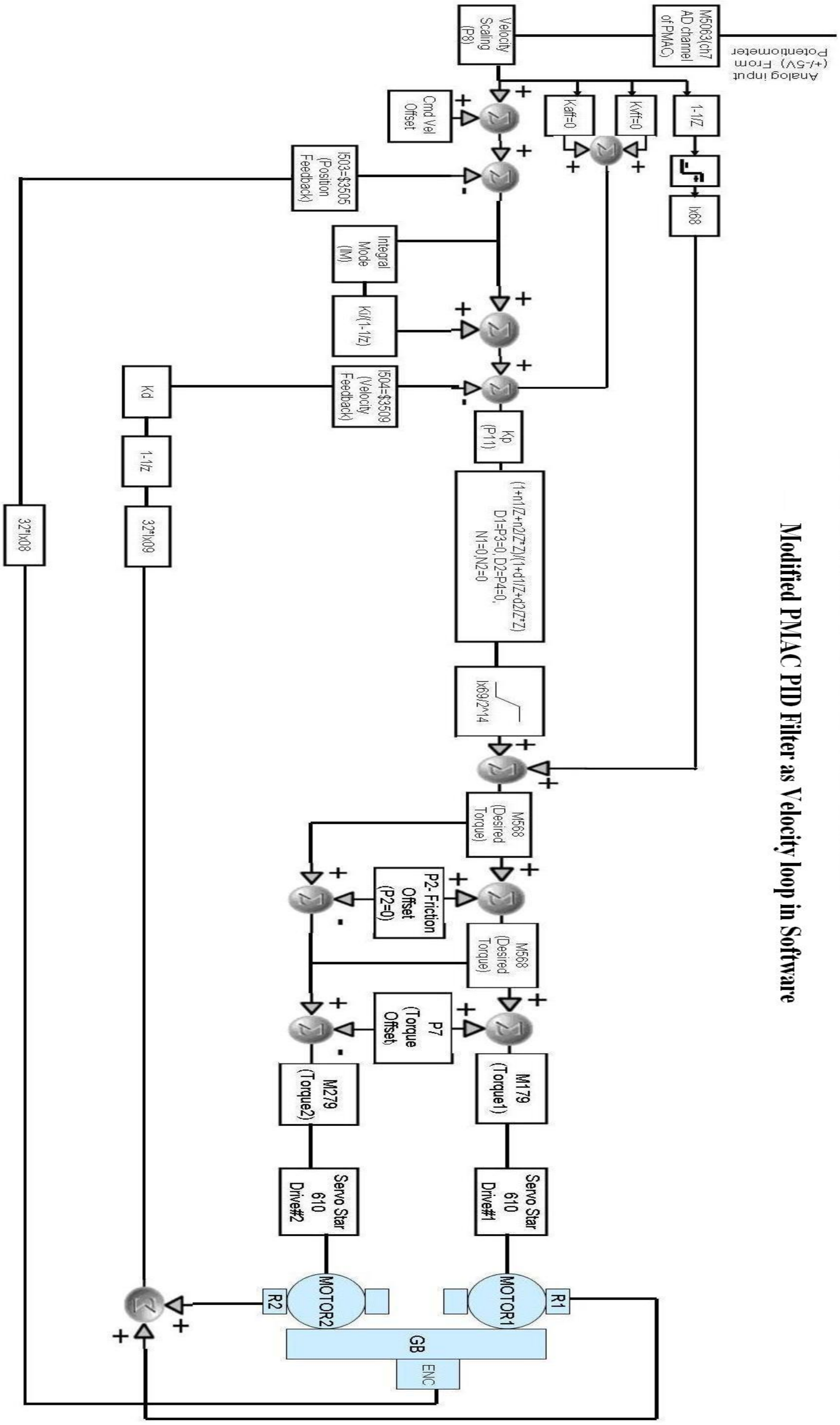
1) ADC check from the above plots shows good linearity.

2) + 10V analog input is programmed in PMAC to 32767 counts which Corresponds to full rated rpm of motor (2200 rpm).

3) Similarly for - 10V analog input equivalent digital counts is -32767 Which corresponds to 2200 rpm in reverse direction.

11. PMAC Configured as software velocity loop:-

2.1) Block Diagram of Large Test Setup arrangement(PMAC-velocity loop)



Modified PMAC PID Filter as Velocity loop in Software

Description of abbreviations used in block diagram above:

R1=Resolver #1
R2=Resolver #2
Enc= Roc417 Absolute Encode Version Endat2.2
GB=Gear Box
Ix68=Friction Feed Forward
Ix69=Command output limit
I503=Position feedback address
I504=Velocity feedback address
I508=position Scaling factor=96
I509=Velocity Scaling factor=4
Kp=Proportional Gain=1
Ki=Intergral Gain=0
Kd=Differential Gain=0
Kvff=Velocity Feed Forward=0
Kaff=Acceleration Feed Forward=0
IM(Intergral Mode)=0

Test Procedure:

2.2) Connect Drives Servo Star 610 number-1 & 2 to PMAC Channels #1 and #2.

2.3) Connect Absolute Encoder ROC417 with interpolator(IBV102) to PMAC(2) Channel #5.

2.4) Tune the system for Velocity loop and not as Position loop (PMAC receives velocity command by Analog input).

2.5) Restart PMAC with $$$ command in Terminal Window.

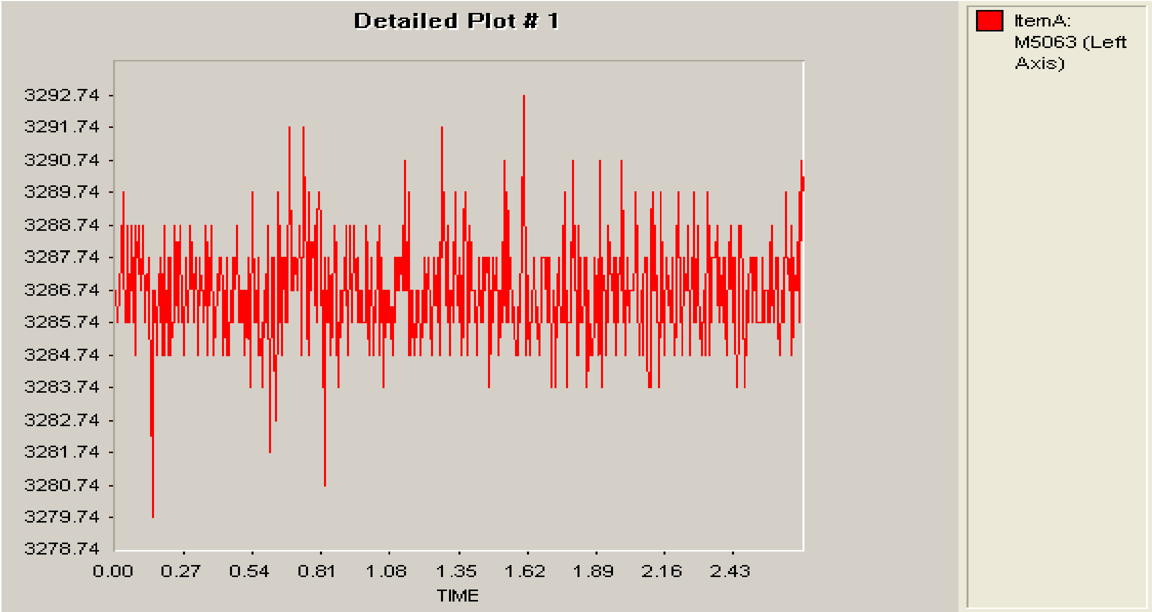
2.6) Go to 'Pewin32 pro' and in **file** menu→ **Download fi**le named **"upload 09012009_bsr.CFG"** which has PLC0 program for torque offset as given in **Annexure-D**

2.7) After downloading file, see for zero Errors or Warnings.

2.8) Now enable PLC0, motor #1, #2 and dummy ch #5 by commands
Enable PLC0, #1o0, #2o0 and #5o0 respectively.

2.9) Now give Analog input 1V from external voltage source and see the digital counts stored in memory location M5063 of PMAC.

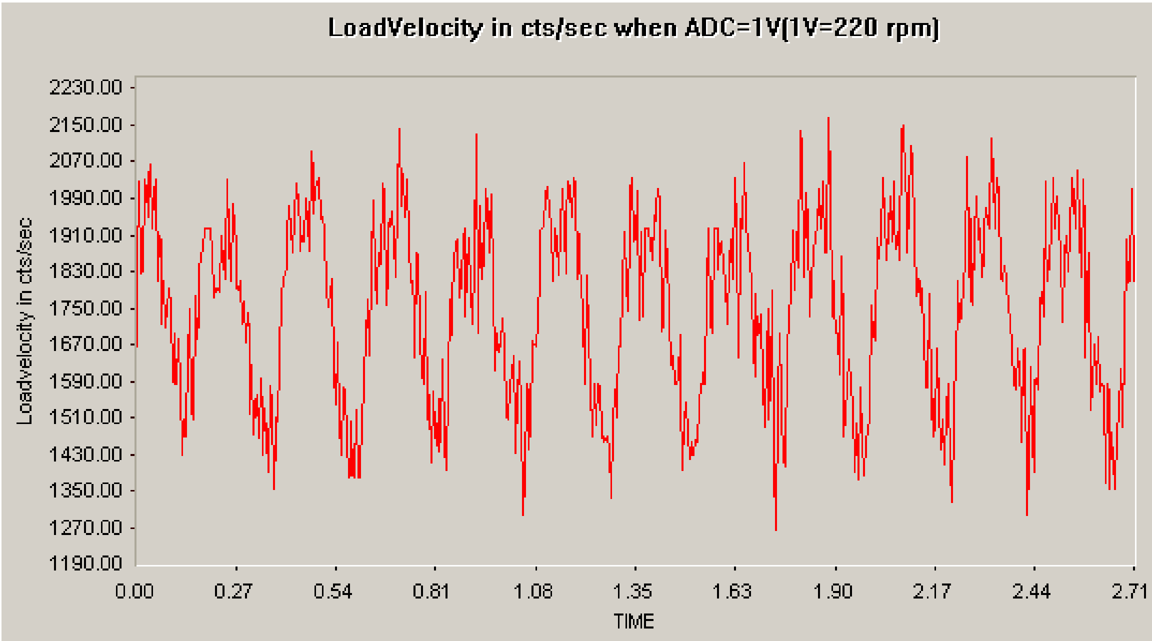
2.10) The plot looks as shown below. The digits vary by 4 counts - 3284.74 to 3288.74. The mean value of 3286 counts has been considered for all calculations for 1V analog i/p.



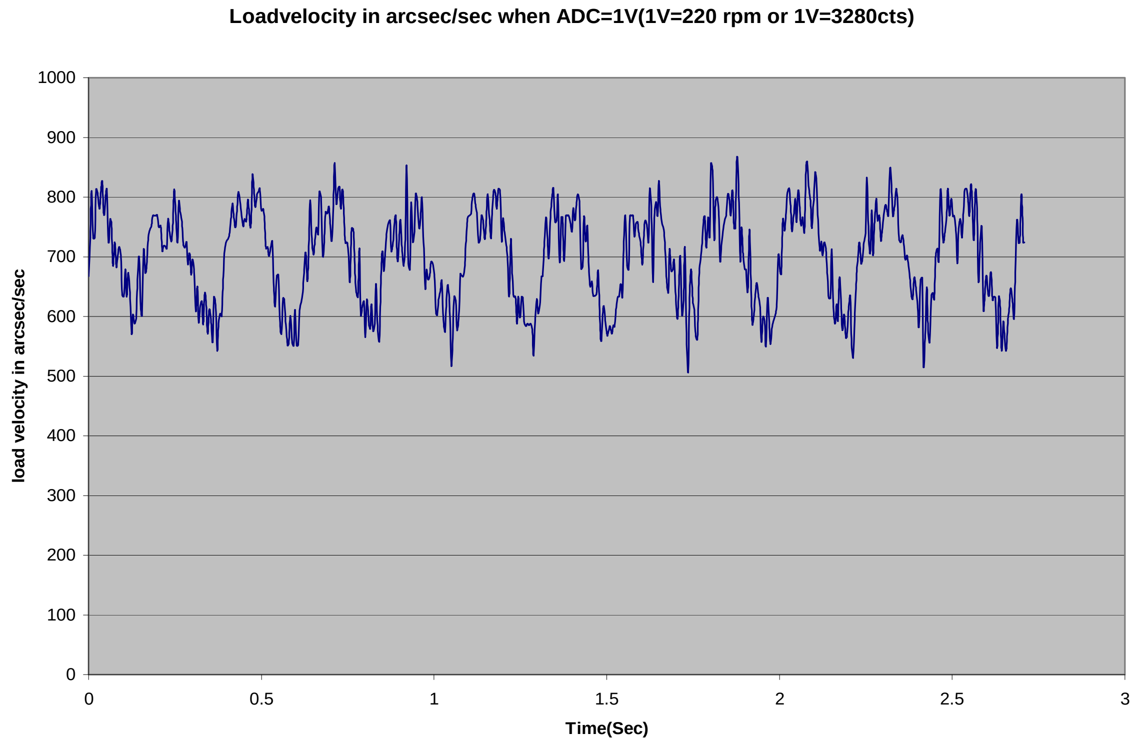
So for 1V analog input the average value of M5063 is 3286 cts.

The following plot shows the load or the central gear velocity as seen from encoder for 1V analog I/p.

Analog I/p in volts	M5063 in counts	Load velocity in counts / sec
1.0	3286	1750



Load Velocity scaled and plotted in arc second / second for the same analog I/p of 1V



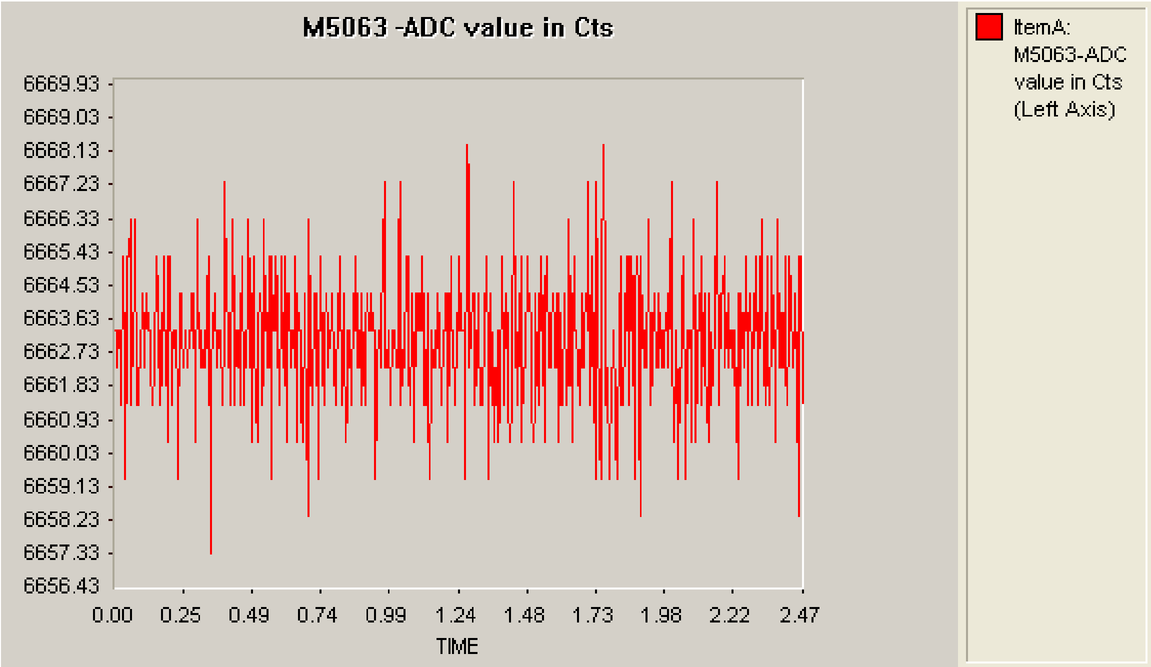
Observation: -

1) So if 1V analog input is giving to BLC, will get average M5063 (ADC Value in Cts) value is 3286Cts and average LoadVelocity is 1700Cts/sec Or 700 arcsec/sec.

2) Also the velocity varies by almost 200 arc seconds / second.

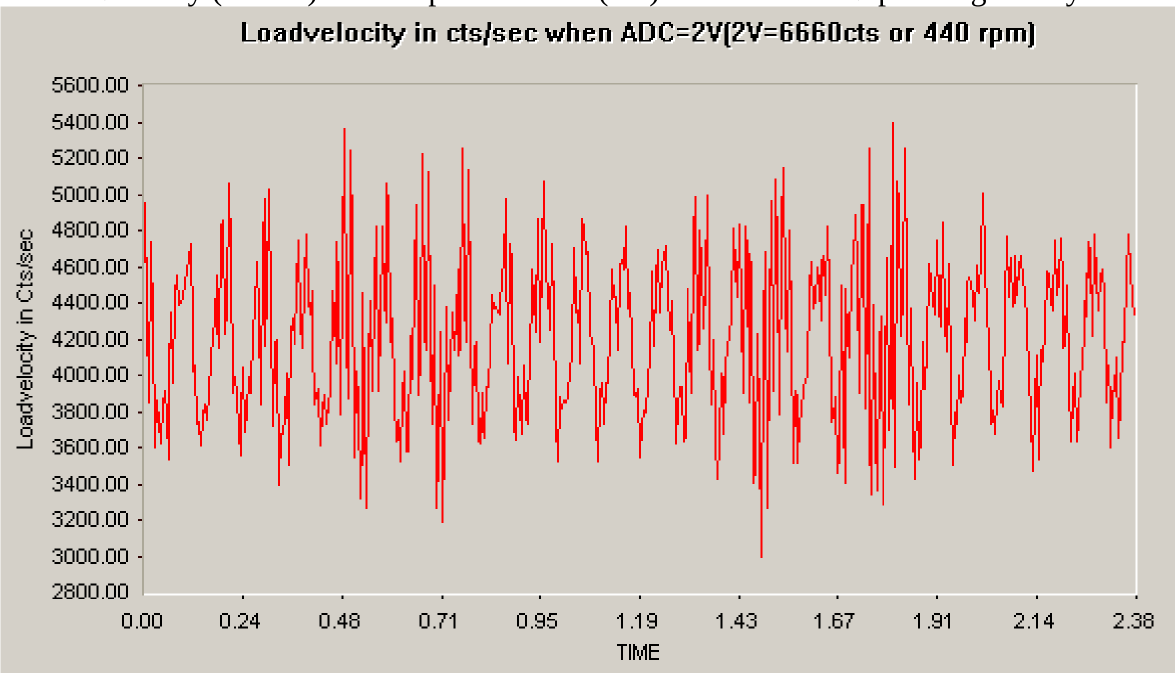
2.11) Now this procedure as above is repeated for one more analog input of 2 volts and the readings and plots are as below

:.

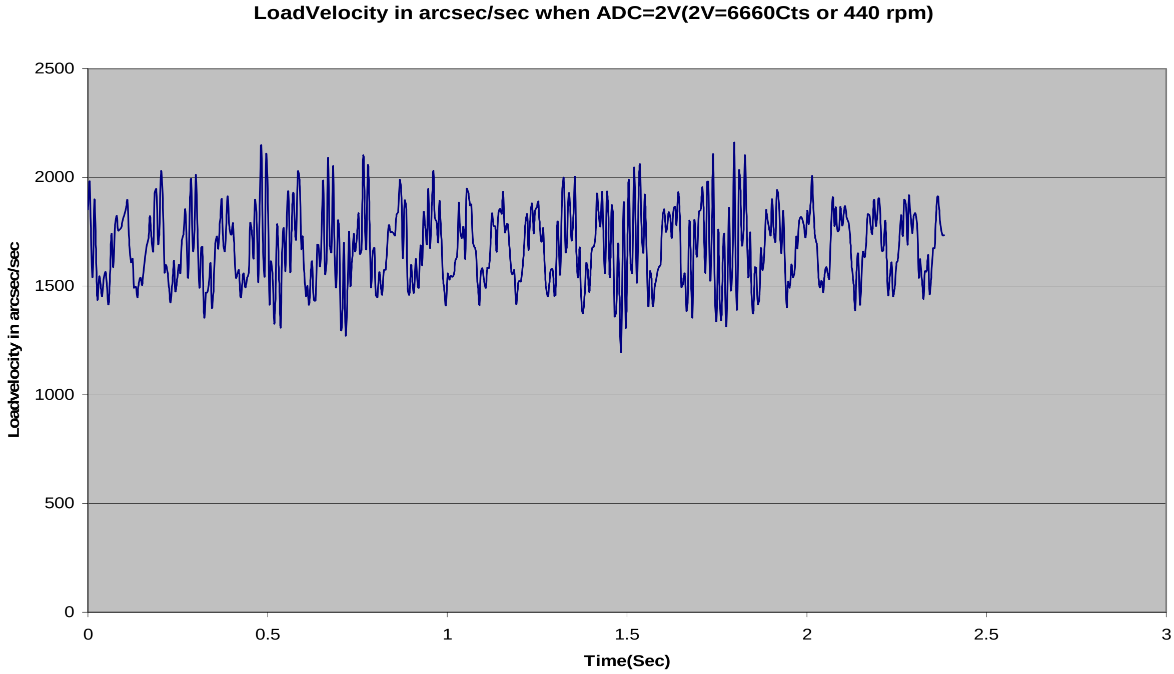


So for 2V analog input the average value of M5063 is 6663 cts.

LoadVelocity (cts/sec) with respect to Time (sec) when ADC=2V plot as given by



Load Velocity (arcsec/sec) with respect to Time (sec) when ADC=2V plot as given by



Observation: -

1) So if 2V analog input is given to BLC, then average M5063 value is 6663 Cts and average LoadVelocity is 4300Cts/sec or 1700 arcsec/sec.

Example:

1 rotation of big wheel = 8192 * 1488 * 73/19 rotation of motor in counts
= 8192 * 400 rotation of load encoder in counts

Therefore resolution ratio of Motor to load = 14.29

Velocity of Load velocity = 4300 cts / sec
= 4300 * 14.29 = 61447 cts/sec of motor
Which is 61447 * 60 / 8192 rpm of motor
= 450.23 rpm of motor

Which is roughly the same observed in Servo star - SS610 of 440rpm as shown in row 3 of table below:

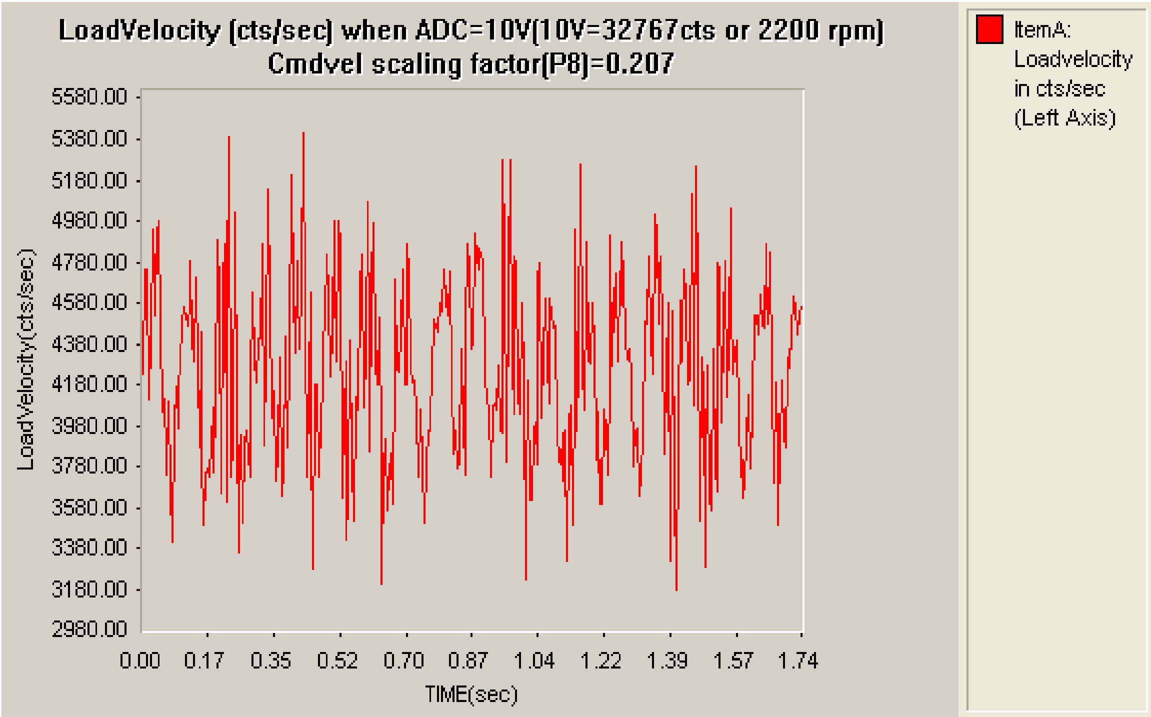
2.12) The table below summarizes the velocity of motor and load for various analog I/p voltages from 1 V to 10 V. The motor velocity is as seen in drive Servo star SS-610

Analog Input	M5063 (ADC value In Cts)	Velocity Load encoder In Cts/sec	Velocity Load Encoder in Arcsec/sec	Velocity Load enc. in deg/min	Motor Velocity In RPM in SS-610
1V	3286	1700	700	11	178-202
2V	6663	4300	1700	28	420-460
3V	9677	6250	2500	41	630-770
4V	13112	8750	3500	58	885-930
5V	16310	11350	4540	75	1127-1170
6V	19530	13193	5277	87	1380-1424
7V	22793	15244	6100	101	1570-1630
8V	26091	17624	7050	117	1790-1900
9V	29371	19175	7600	126	2070-2110
10V	32767	21725	8690	144	2180-2200

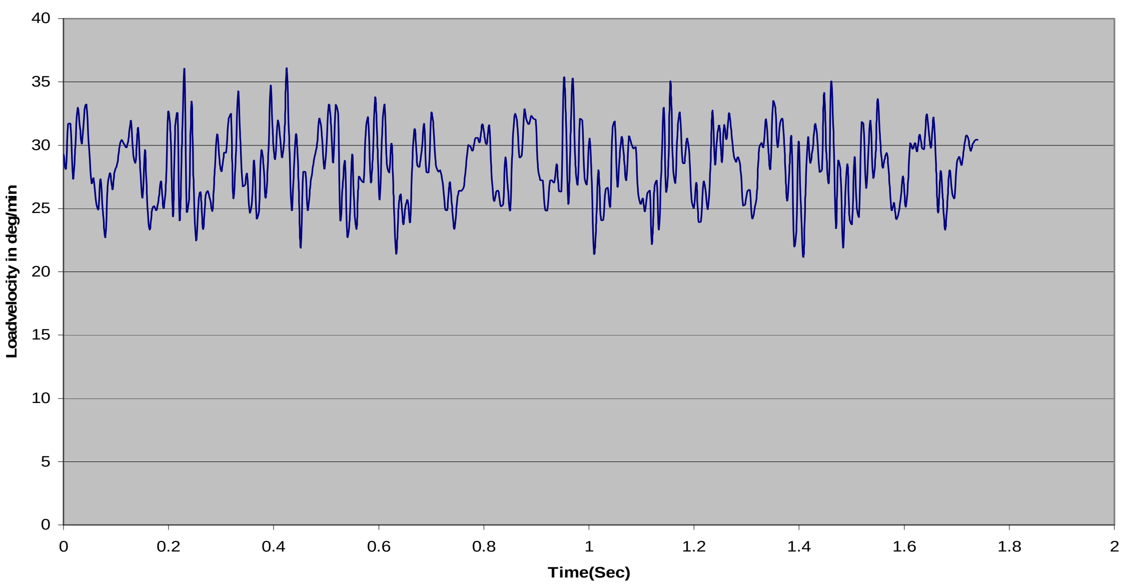
Note: - In GMRT the maximum velocity is 30 deg/Min and Minimum velocity is 15arcsec/sec.

2.12) From the above table it is apparent that at full analog i/p of 10 V the load speed is 144 deg. / min, where as GMRT maximum speed required at SLEW is only 30deg./min

2.13) So for getting LoadVelocity 4500cts/sec or 1800 arcsec/sec or 30 deg/min velocity, CmdVel Scaling factor (P8) need to change from **1 to 0.207** in PLC0 program which is given in Annexure-D.After changing the "command. Velocity scale "factor the load encoder plot is taken and is observed to be 30deg/min. Both plots counts / sec and deg / min are plotted below.

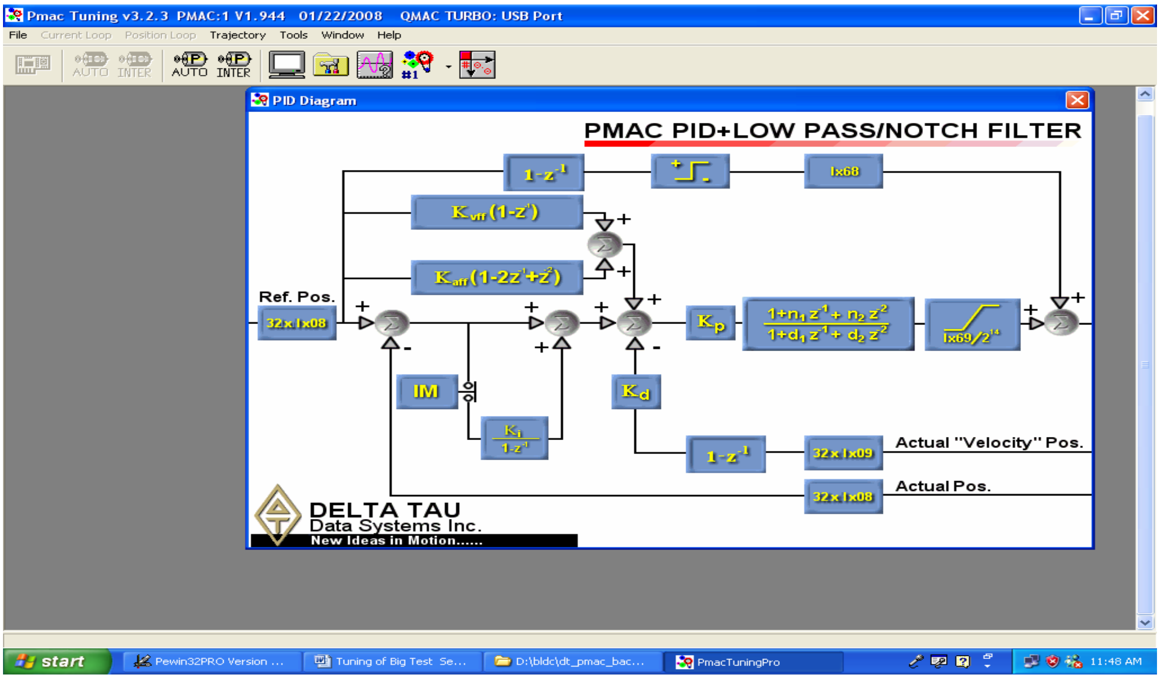


Loadvelocity in deg/min (Left Axis) when ADC=10V(10V=32767 or 2200 rpm) with Cmdvel Scaling factor(p8)=0.207



12.PMAC Configured as a velocity loop with standard PMAC PID filter:-

3.1) Block Diagram of PMAC PID filter as given below:-



Kp= Proportional gain
Ki= Integral gain
Kd=Differential gain
IM=Integral Mode
Kvff=Velocity Feed Forward
Kaff=Acceleration feedFoward
n1, n2, d1, d2= Notch filter coefficients
Ix08=Actual Position feedback
Ix09=Actual velocity feedback.

Test procedure:

3.2) For making this PMAC PID filter as velocity loop, we have to make position Feedback needs to be zero. For that we are taking load encoder feedback to #7 Channel, so while closing velocity loop, Position feedback will remain open at Channel #5 and position Feedback will be zero. At Ch #7 we can get encoder Data, which we can used for plotting data between Actual Velocity (arcsec/sec) and Time (sec).

3.3) Here Analog input is given to Ch#7 ADC channel of PMAC which address is given By M5063-> y: $78115, 8, 16, S.Means Analog value will be stored in Signed 16 Bits Form. This value will scaled and stored in Hand-wheel Position register (M567).So M567 is given by

M567= 32*96*M5063* AzcmdvelScaling factor/ 32767.
Azcmdvelscaling factor can change depend on our speed Requirement (for GMRT speed is 15”/sec to 30deg/min).

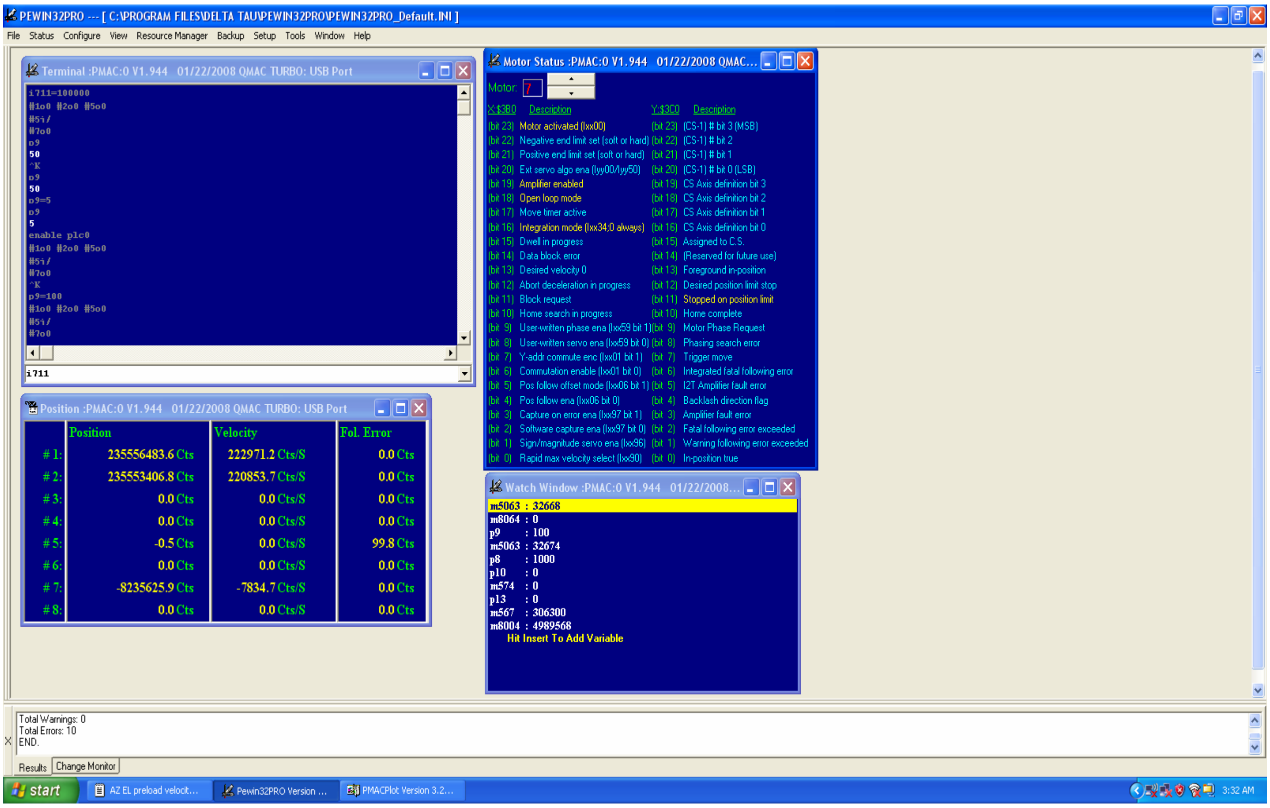
3.3) Go to Pewin32 pro and File → download file "AZ EL preload velocity 1_bsr.pmc" .This contains PLC0 program for torque offset. The PLC0 Program as given in **Annexure-E**.

3.3) After downloading above file see for 0 errors and warning.

3.4) Enable PLC0 , motor #1,#2,#5 and #7(for position feedback) by commands

Enable PLC0
#1o0
#2o0
#5o0
#7o0

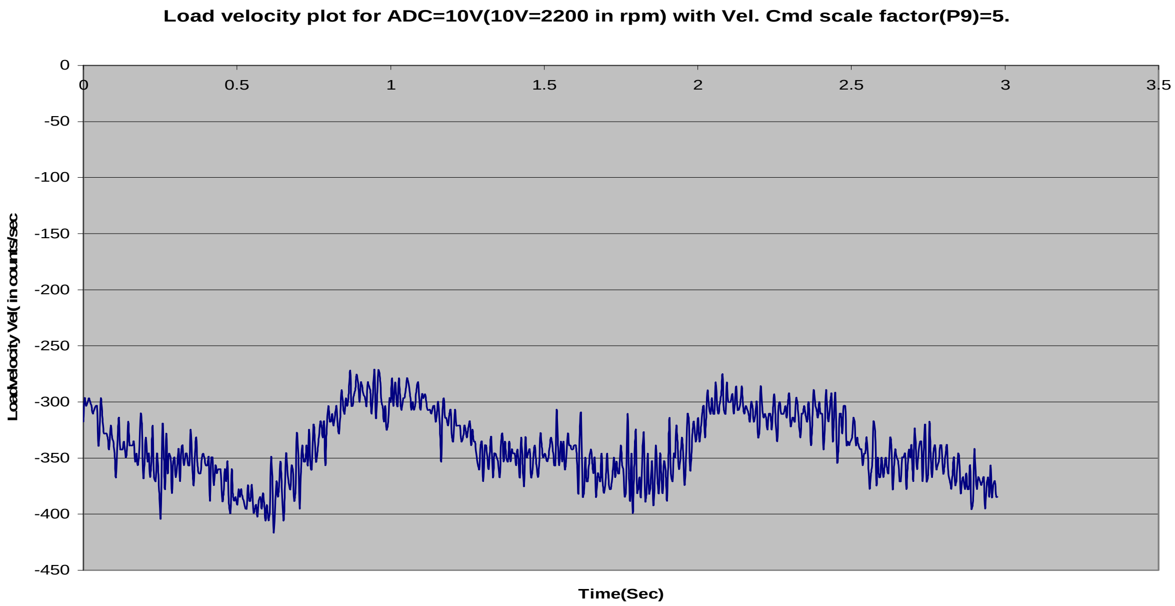
3.5) Close the loop by #5j/. Give 1V analog input. Observe channel #1, #2 and #7 readings. The following shows window regarding position, velocity of #1, #2, and #7, terminal window, watch window and motor status.

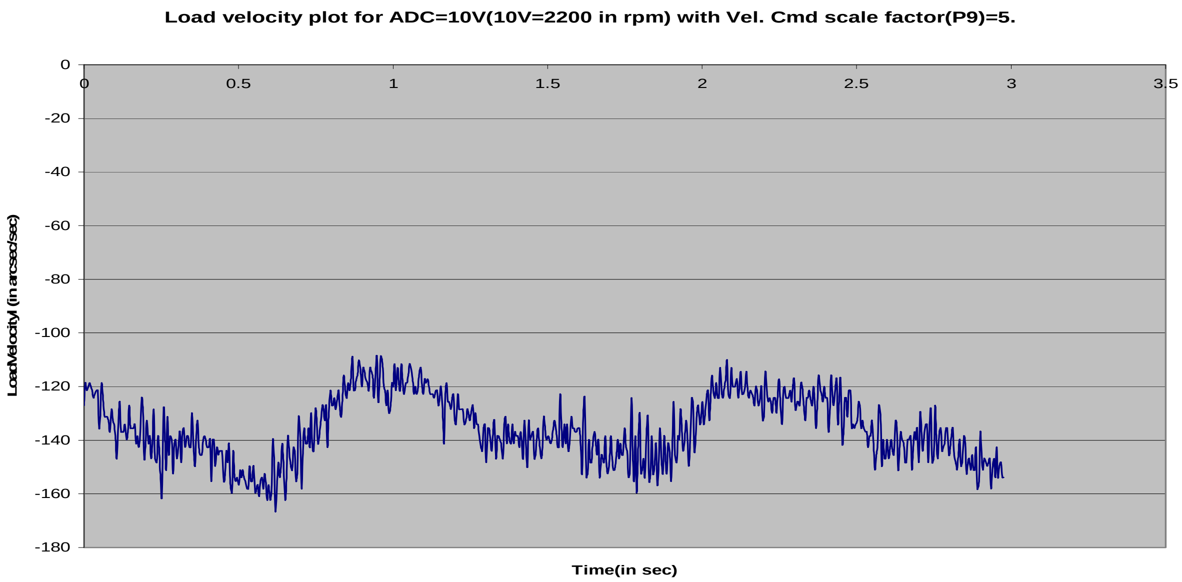


3.6) when increasing Analog input from 1V to 10V, can observe increase in Velocities.

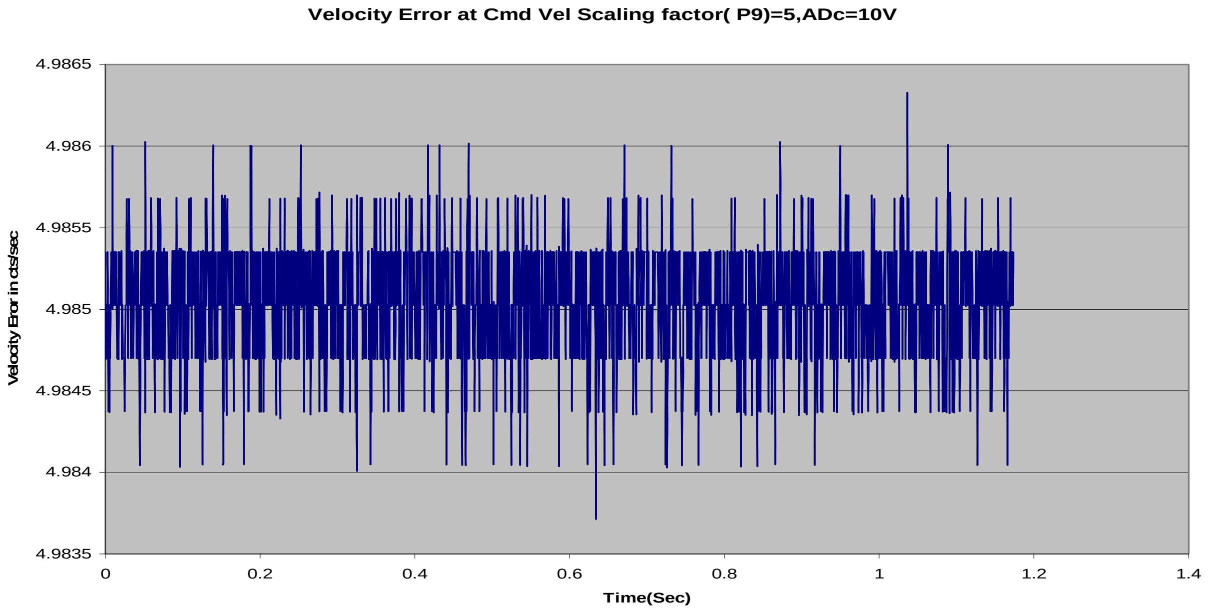
3.7) Now give Analog input 10V and set CmdVel Scaling factor (P9) is 5 then Plot the LoadVelocity (cts/sec) , LoadVelocity (arcsec/sec) , Velocity Error(cts/sec) and Velocity Error(arcsec/sec) with respect to Time (Sec).The related plots are given below.

<u>Load Velocity when P9 (CmdVel Scaling factor=5) and Analog input is 10V</u>

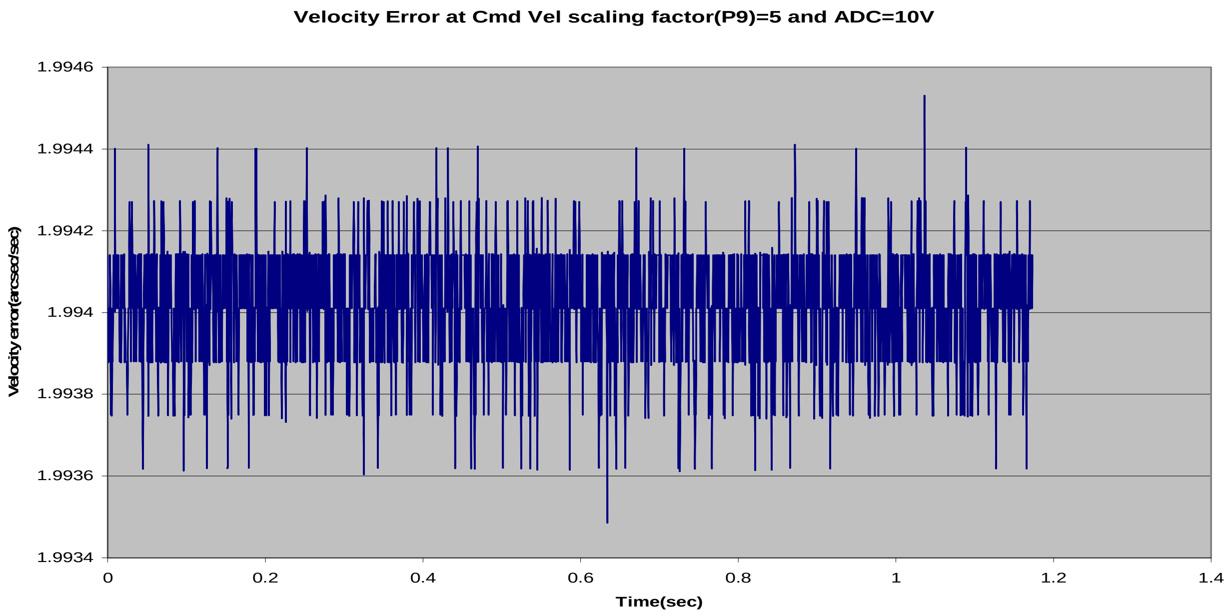




<u>Velocity Error (cts/sec) at Cmd Vel Scaling factor (P9) =5 and ADC=10V</u>



<u>Velocity Error (arcsec/sec) at Cmd Vel Scaling factor (P9) =5 and ADC=10V</u>



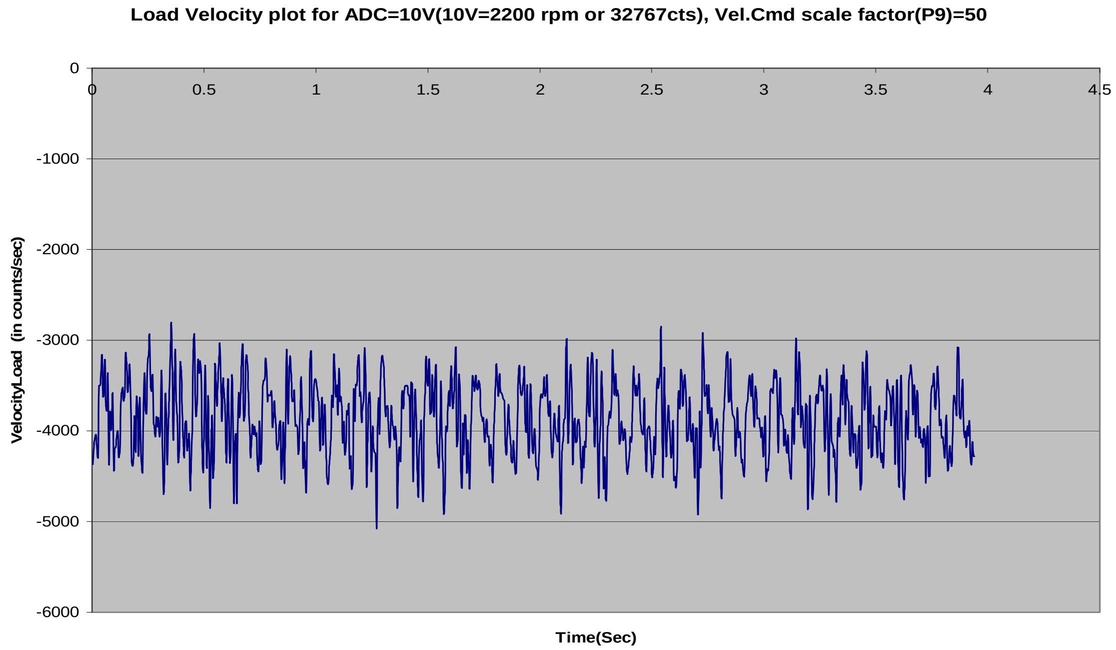
Observation: -

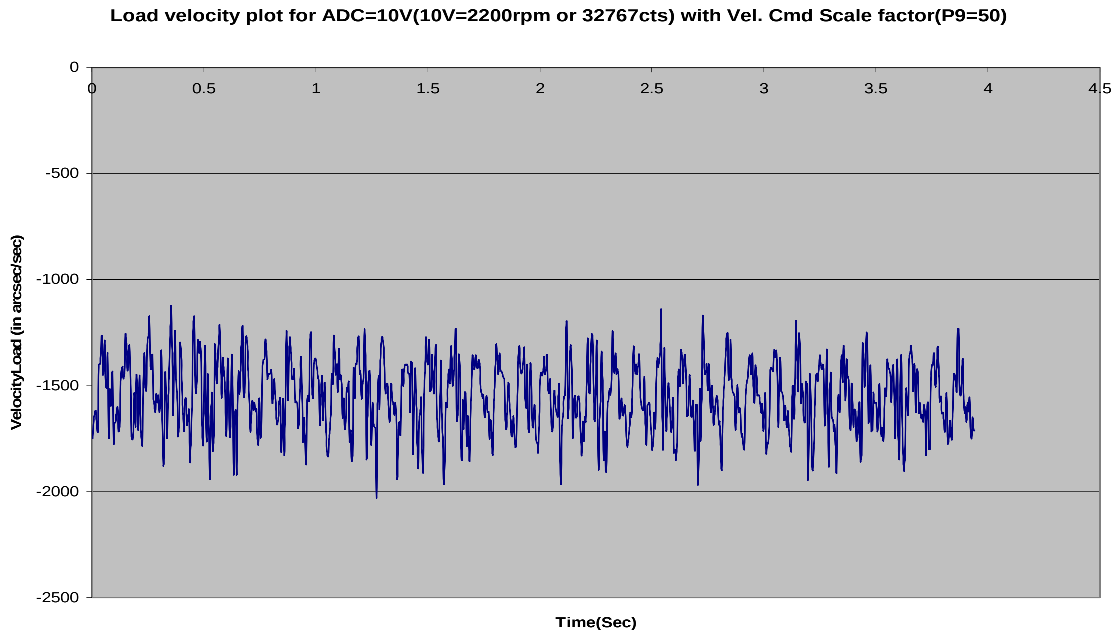
From the above plots, if 10V analog input is given to BLC with 'CmdVel scaling factor' – P9 =5 then we get an average Load Velocity of 350Cts**/sec** or **140 Arcsec/sec or 2 deg/min**. Velocity Error between command actual works out to **4.985** Cts/sec or **1.994** arcsec/sec.

Repeat the above Test Procedure for CmdVel Scaling factor P9 = 50

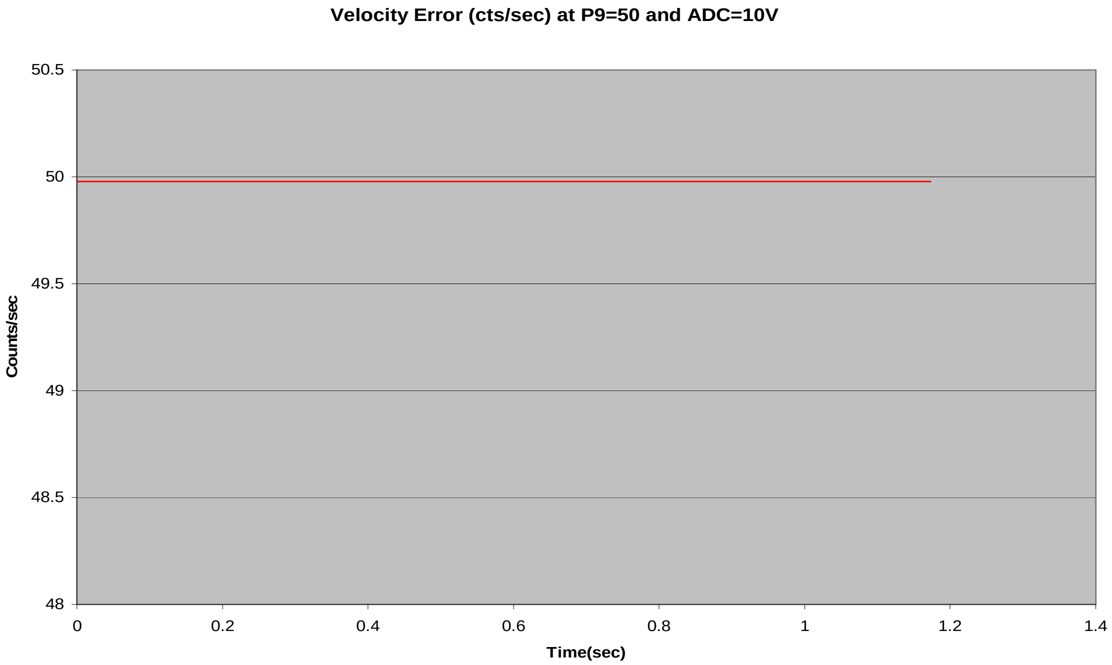
Now give Analog input 10V and set Cmdvel Scaling factor (P9) is 50 then Plot the LoadVelocity (cts/sec), LoadVelocity (arcsec/sec),Velocity Error(cts/sec) and velocity Error(arcsec/sec) with respect to Time (Sec).The related plots are given below.

<u>Load Velocity when P9 (CmdVel Scaling factor=50) and Analog input is 10V</u>

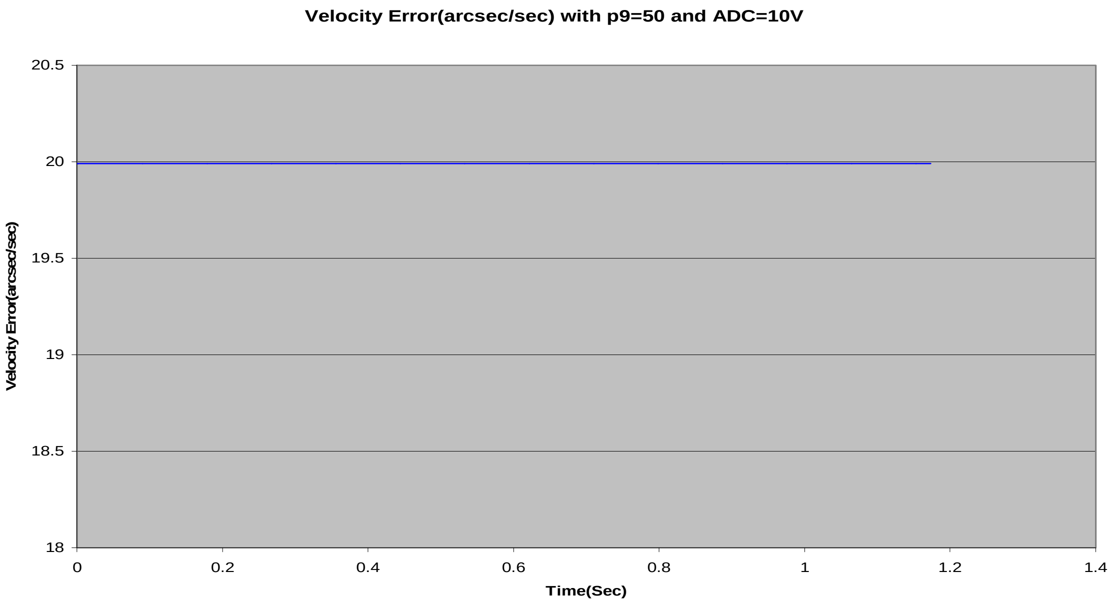




Velocity Error (cts/sec) at Cmd Vel Scaling factor (P9) =50 and ADC=10V



Velocity Error (arcsec/sec) at Cmd Vel Scaling factor (P9) =50 and ADC=10V



Observation: -

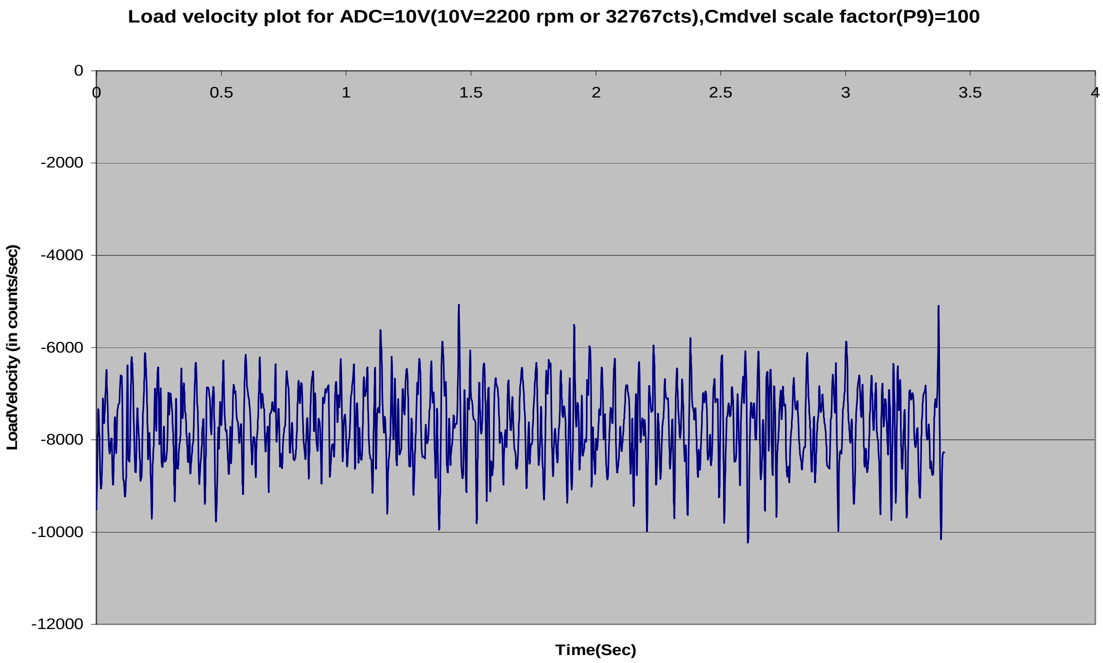
1) So if 10V analog input is giving to BLC and CmdVel scaling factor (P9) =50 then will get average LoadVelocity is **3500Cts/sec** or **1400 Arcsec/sec or 23 deg/min**. Velocity Error is **49.9** Cts/sec or **19.9** arcsec/sec.

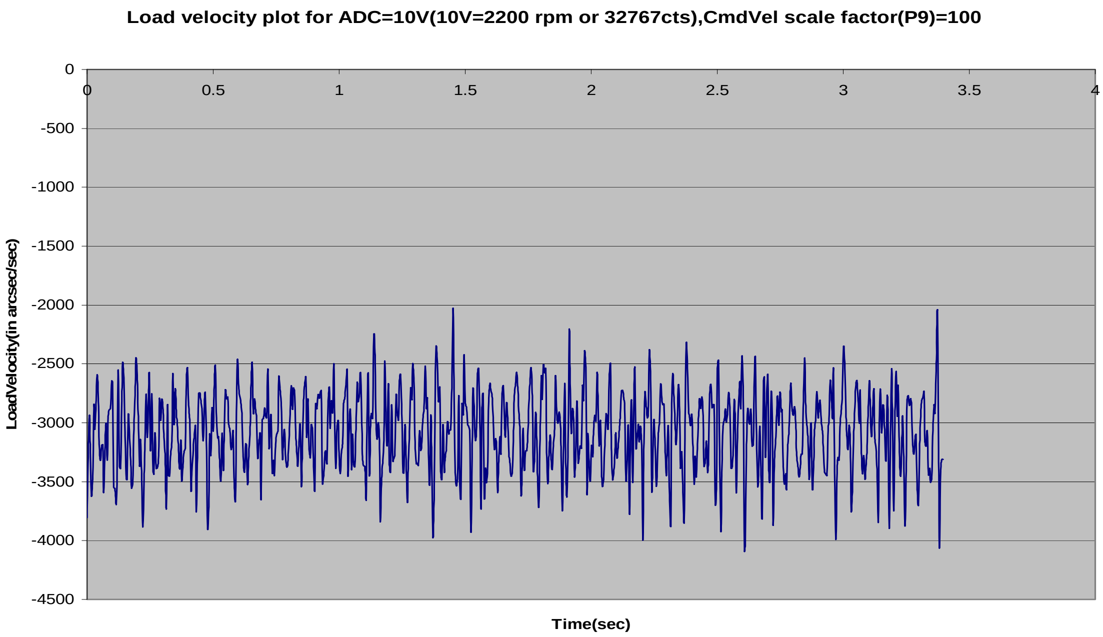
2) The error in velocity has increased ten fold as scale factor is increased by 10

Repeat the above Test Procedure for CmdVel scaling factor P9 = 100

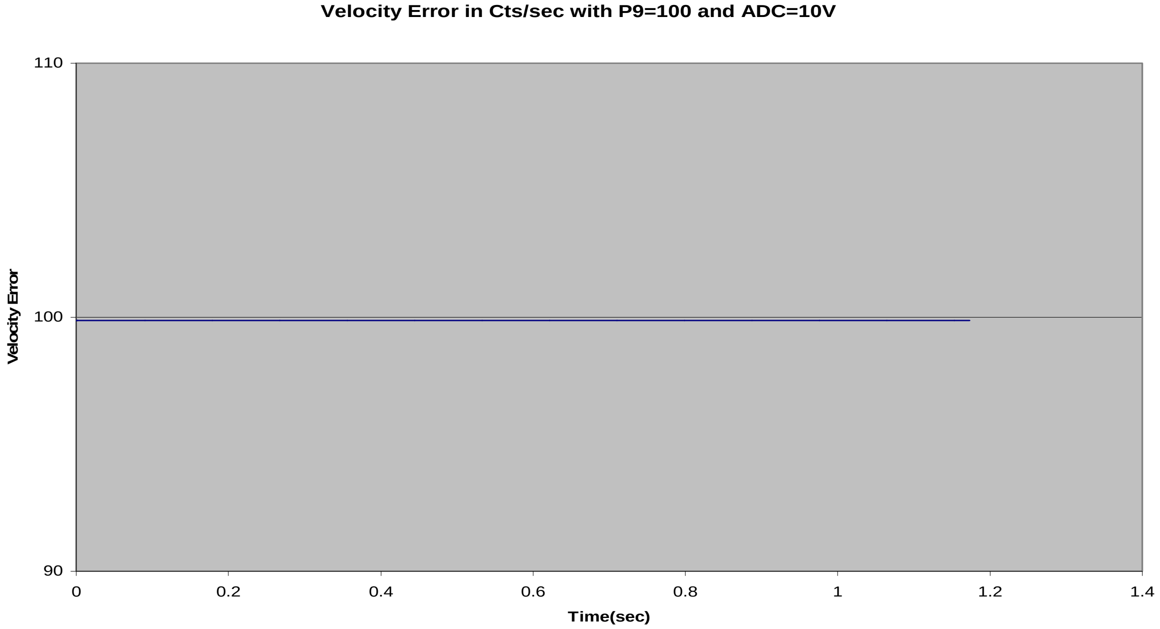
Now give Analog input 10V and set CmdVel Scaling factor (P9) is 100 then Plot the LoadVelocity (cts/sec), LoadVelocity (arcsec/sec),Velocity Error(cnts/sec) and Velocity Error(arcsec/sec) with respect to Time (Sec).The related plots are given below.

LoadVelocity when P9 (CmdVel Scaling factor=100) and Analog input is 10V

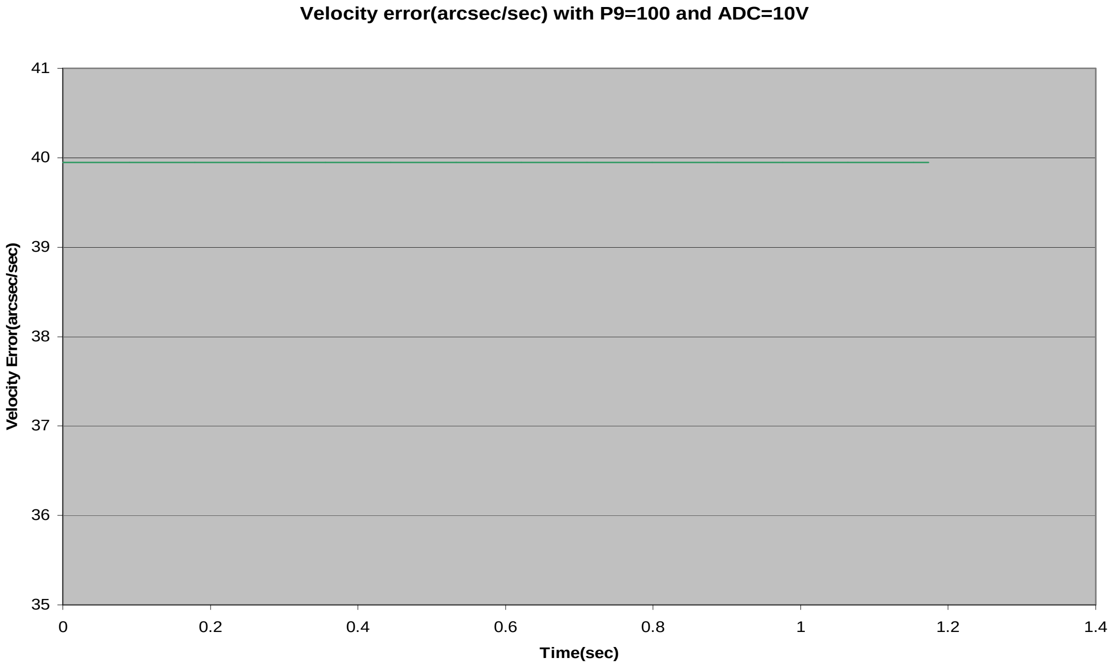




Velocity Error (cts/sec) at Cmd Vel Scaling factor (P9) =100 and ADC=10V



Velocity Error (arcsec/sec) at Cmd Vel Scaling factor (P9) =100 and ADC=10V



Observation: -

1) So if 10V analog input is giving to BLC and CmdVel scaling factor (P9) =100 then will get average LoadVelocity is **7500Cts/sec** or **3000 Arcsec/sec or 50 deg/min**. Velocity Error is **99.8** Cts/sec or **39.9** arcsec/sec.

2) Now the error in velocity has increased 20 fold for an increase of P9 20 times which is consistent with last result.

Summary of load velocities for different scale factors and for a max. I/p of 10V analog

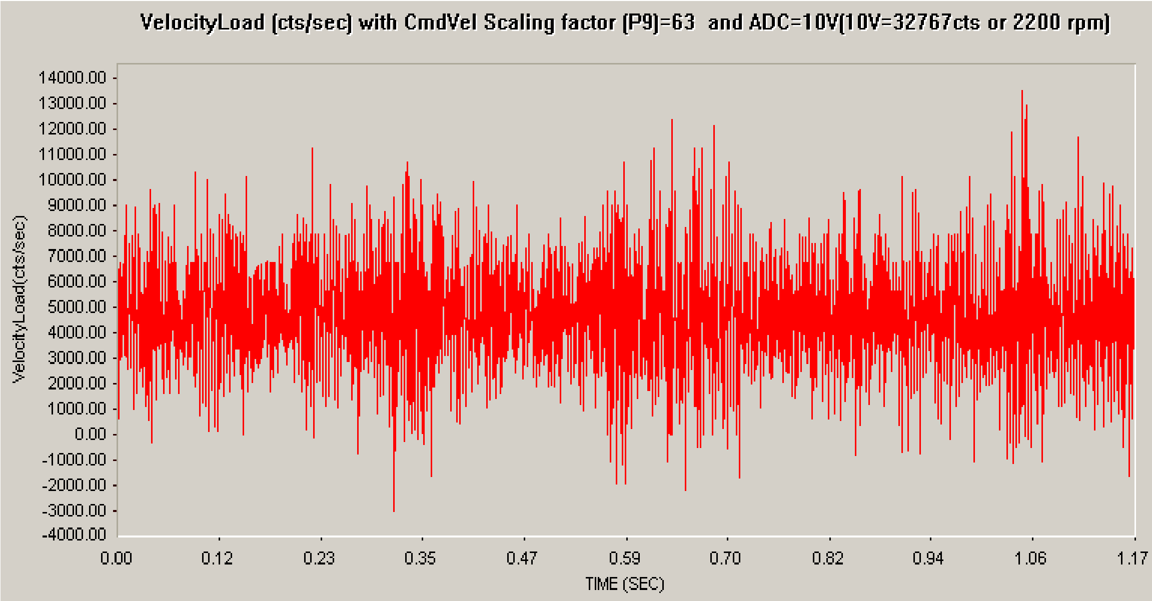
Analog input	CmdVel Scale Factor(P9)	LoadVelocity In Cts/Sec	LoadVelocity In arcsec/sec	LoadVelocity In deg/min
10V	5	350	140	2
10V	50	3500	1400	23
10V	100	7500	3000	50

Speed specifications of GMRT Azimuth axis

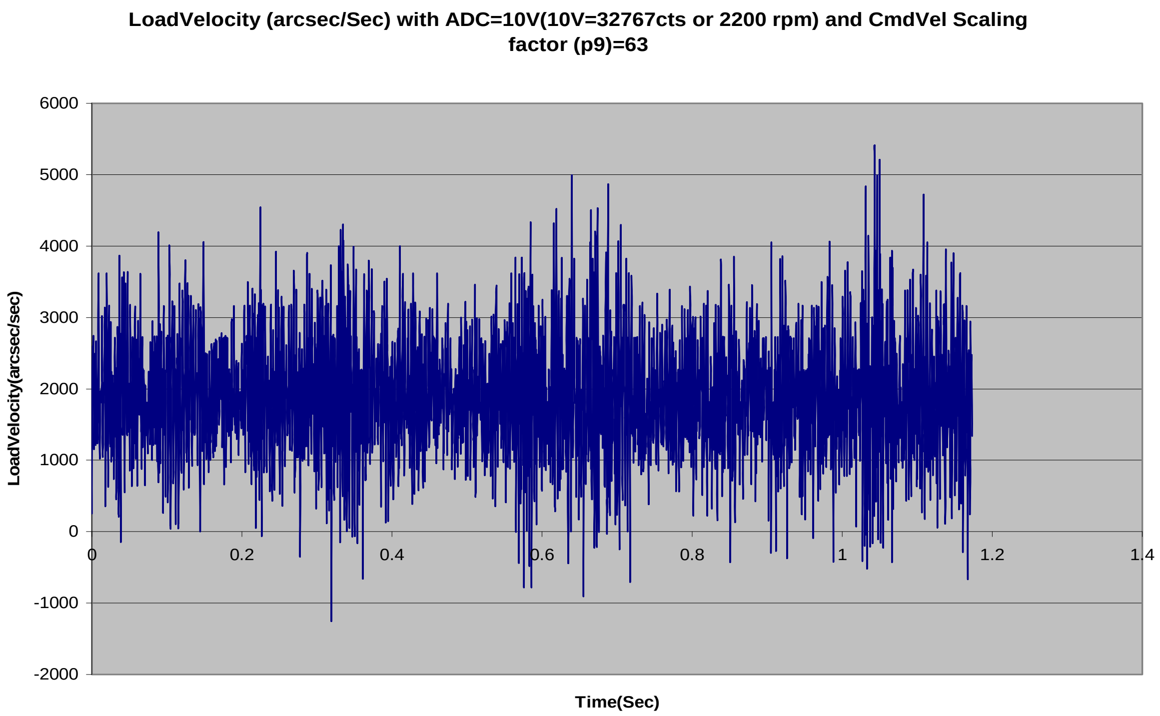
Since the GMRT maximum velocity requirement is **30 deg / Min** and Minimum velocity is **15arcsec/sec** we have to programme the Cmd. Velocity Scale factor accordingly so that for a max. I/p of analog 10Volt we get 30 deg / min load speed.

Command velocity scale factor - P9 = 63

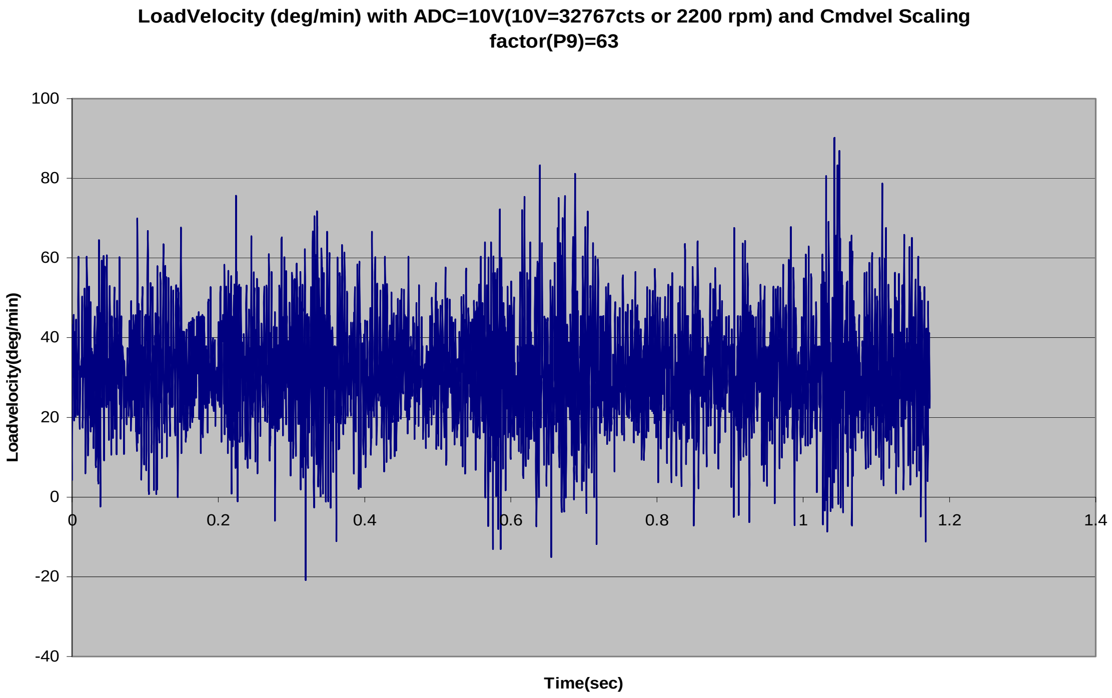
4.1) So for getting LoadVelocity **4500cts/sec** or **1800 arcsec/sec** or **30 Deg/min** velocity, Cmdvel Scaling factor (P9) need to change to **63** in PLC0 program which is given in Annexure-E. The related plots are given Below.



Average Velocity Load is **4500cts/sec**.

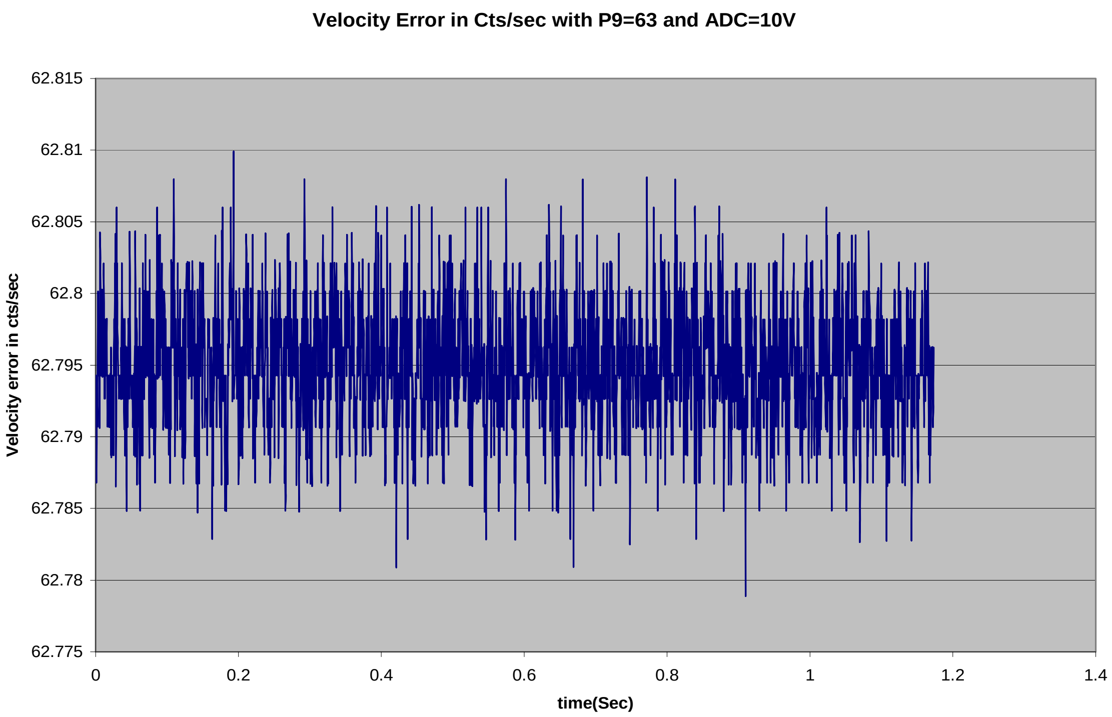


Average Velocity Load is **1800arcsec/sec**.

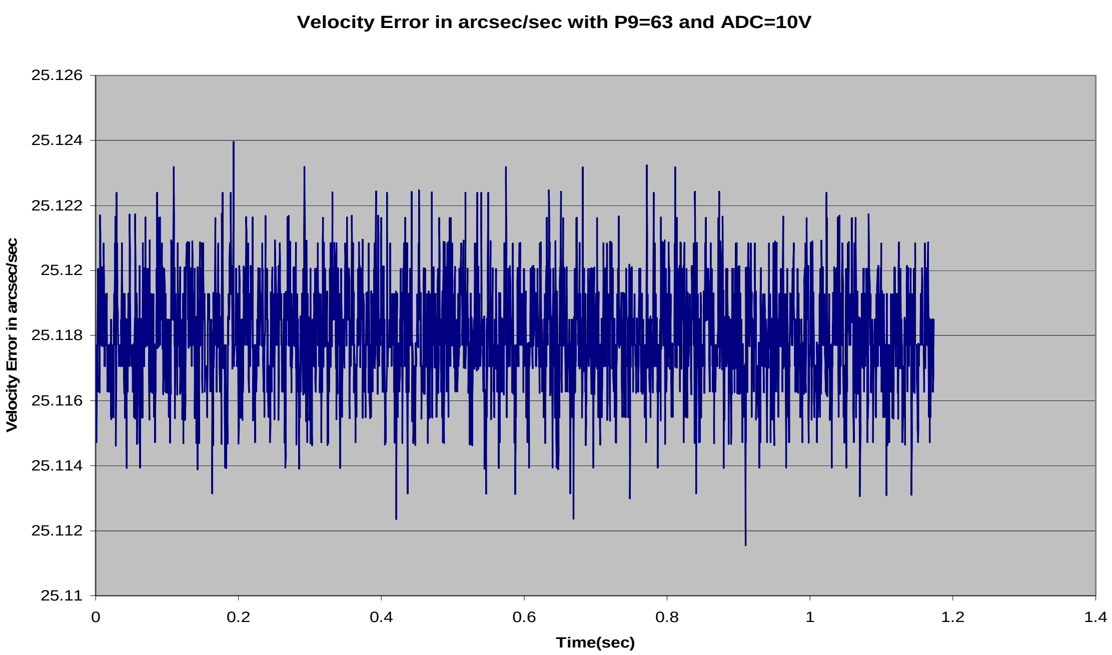


Average Velocity Load is 30 deg/min.

Velocity Error (cts/sec) at Cmd Vel Scaling factor (P9) =63 and ADC=10V



Velocity Error (arcsec/sec) at Cmd Vel Scaling factor (P9) =63 and ADC=10V

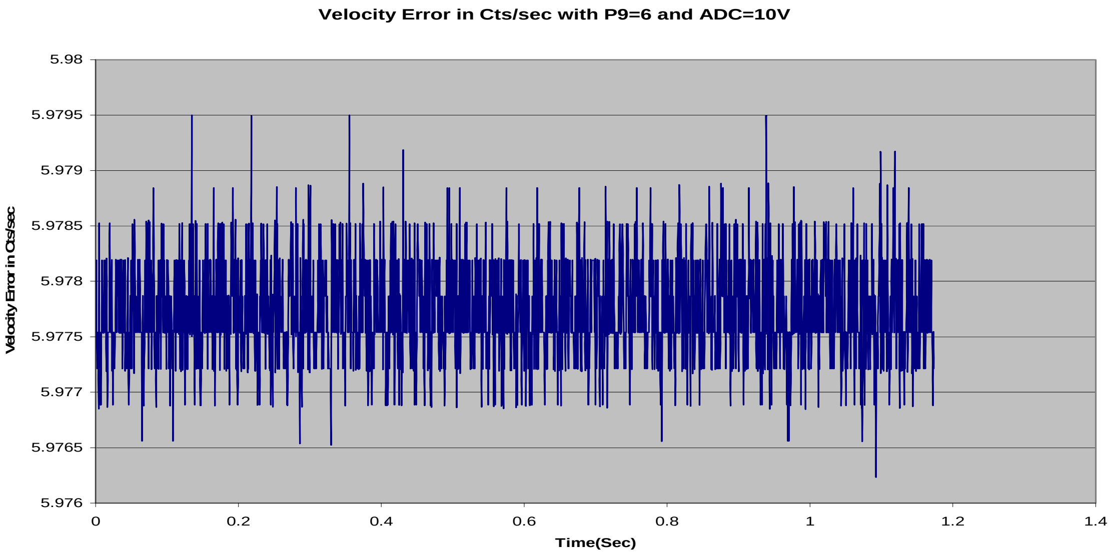


Velocity Error with P9 (Cmd Vel Scaling Factor) =63 and ADC=10V is **62.7** Cts/sec or **25.11** arcsec/sec.

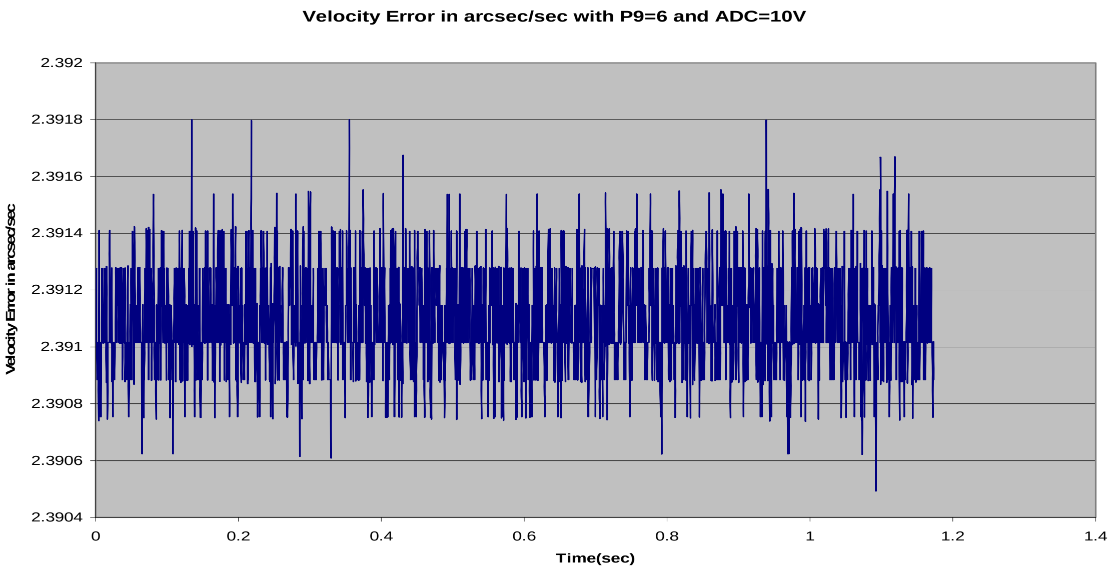
Command velocity scale factor - P9 = 5.35

4.1) For getting Load Velocity **375cts/sec** or **150 arcsec/sec** velocity, CmdVel Scaling factor (P9) need to change to **5.35** in PLC0 Program which is given in Annexure-E. The Velocity Error (Cts/sec) and Velocity Error (arcsec/sec) with respect to Time (sec) Plotted below.

Velocity Error (cts/sec) at Cmd Vel Scaling factor (P9) =6 and ADC=10V



Velocity Error (arcsec/sec) at Cmd Vel Scaling factor (P9) =6 and ADC=10V



Velocity Error with P9 (Cmd Vel Scaling Factor) =6 and ADC=10V is **5.9** Cts/sec or **2.39** arcsec/sec.

4.2) **Summarization of Velocity Error for Various Velocities.**

Velocities	Cmd Vel Scaling Factor(P9)	Velocity error In Cts/sec	Velocity error In arcsec/sec
15 arcsec/sec	**0.53**	*****	*****
150arcsec/sec	**6**	**5.9**	**2.39**
30 deg/min	**63**	**62.7**	**25.11**

Note: - 1) ***** Large test Setup at Rayshed is not responding for CmdVel Scaling factor (p9) =0.53 for getting Load Velocity 15 arcsec/sec or 37.5 cts/sec with Analog input 10V.

2) For getting Load Velocity =15 arcsec/sec, keep CmdVel Scaling factor = 6 and change Analog Input to 1V.

13. Step response plots for PMAC configured as Velocity loop

1) Keep the Cmd.Velocity Scaling factor – P9 =50 in the PLC 0 algorithm and
2) Keep analog input voltage at 0V from Power supply (put the SW in OFF) before beginning the test.

Procedure:

First open 'Pmacplot pro' from toolbar for plotting the response

Click detail plot,
|
Keep gather period = 10
|
Go to
Item to gather,
|
Scaling and Processing,
|
Items to plot ---- define gather buffer
|
Begin gathering
|
Adjust analog i/p to ADC = 5 Volt and switch ON the ADC I/p supply
|
End gather after 10000ms
|
Upload data
|
And plot data.

In order to get multiple step response switch the power supply on / off several times after 'begin gather' and before 'end gather'.

The following Commands are to be given in TERMINAL window before giving STEP Input

```
Enable plc0,
#1o0 #2o0 #5o0 #7o0        ; enable channels 1, 2, 5 and 7,
 #5j/                      ;   close Channel 5,
```

The plot is as shown below.

Example illustrating the procedure and calculation of damping ratio and damped natural frequency

1. Keep P9 (CmdVel Scaling factor) =63 and keep analog input 5V
   From Power supply.

2. Procedure for Step response:-

2.1) Go to **PMAC Plot Pro**

2.2) Detail Plot

2.3) Press →Item to Gather→ Take Gather Period as 10
→Take Source1 for M5063 (Command Vel Signal)
→Take Source2 for M762 (Actual Position) → OK

2.4) Press → Scaling and Processing → Edit Source1 → Change Item Name as M5063 → Scale factor as 0.00055→ differentiate as none→OK

(Here the default scale of 0.000326 is changed to 0.00055 to increase the Response to match to the Step input.)

2.4) Edit Source2→ Change Item Name as M762 → Scale factor as 0.000326 differentiate as Once(Velocity)→OK→ OK

(M762 is Actual Position of Load Encoder, but we need Actual Velocity. So we are Differentiating once to Actual Position for getting Actual Velocity)

2.6) Press → Items to plot → Edit → Horizontal Axis as Time
→ Left Vertical Axis as M5063 and M762

→ OK

2.7) Press **Define Gather Buffer**

2.8) Press **Begin Gathering**

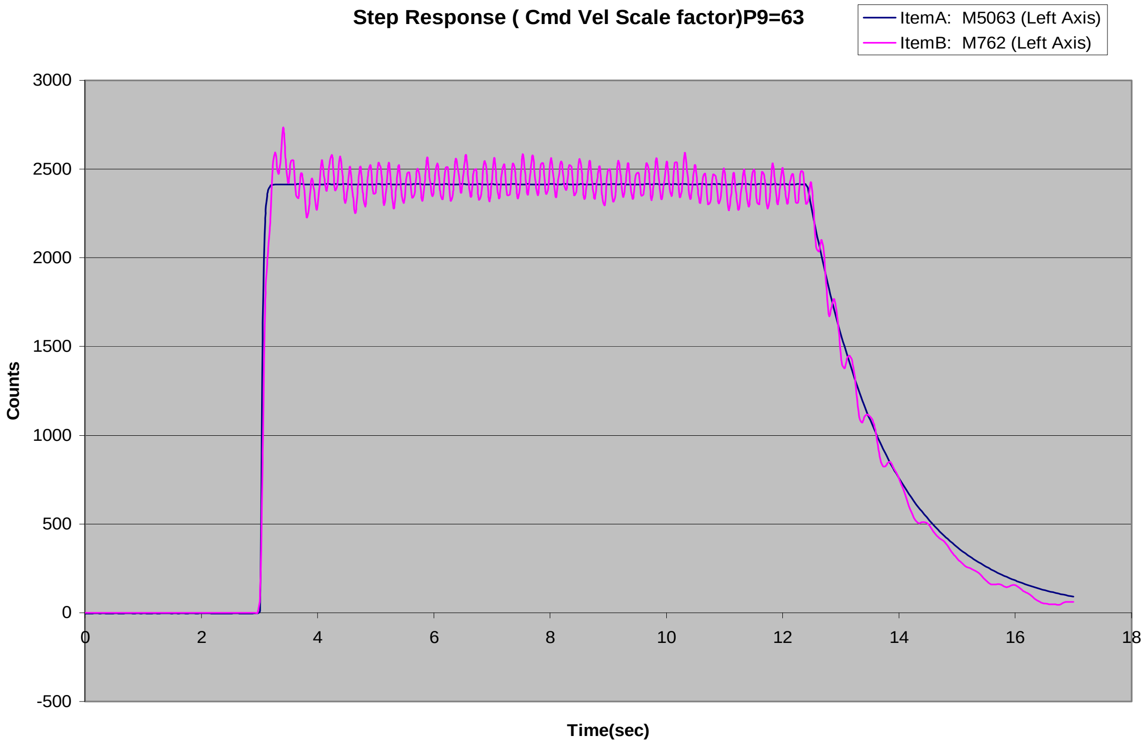
2.9) After 2000ms switch on 5V analog Input Power supply.

2.10) After 15000ms Press **End Gathering**.

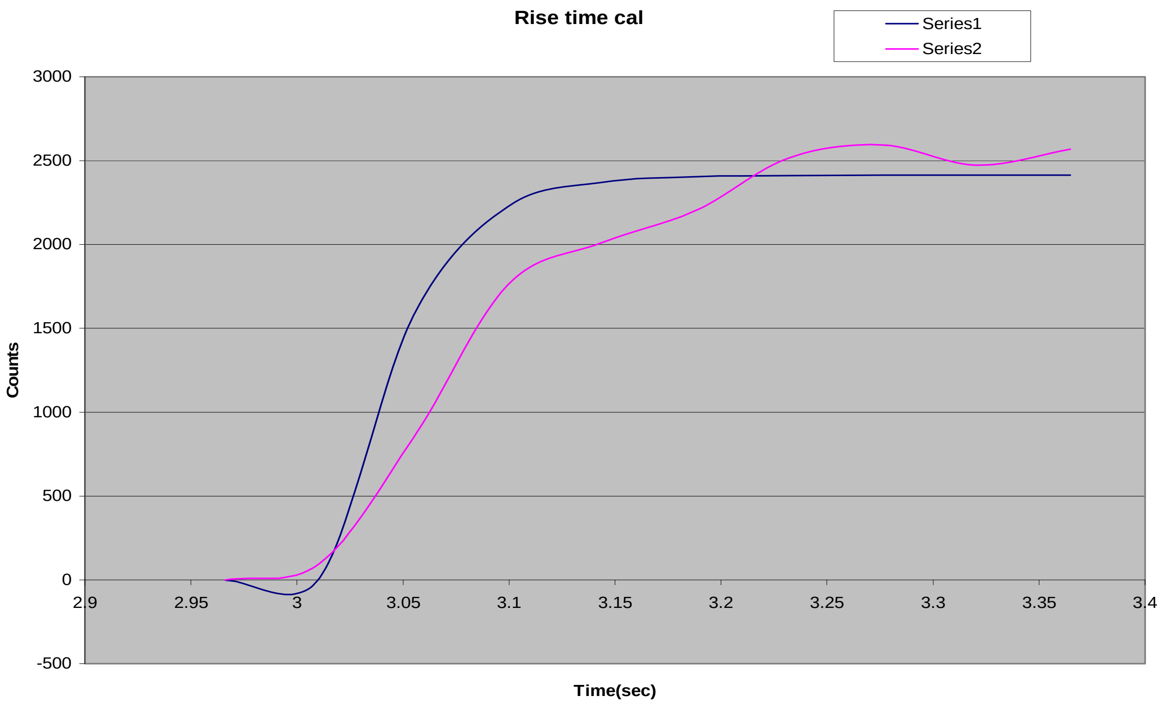
2.11) Press **Upload Data**.

2.12) Press **Plot Data**

Step Response (Cmd Vel Scale Factor) P9=63. And 0V to 5V Step has given



Rise time Cal:-



Rise Time:

Maximum values of the Counts are: - 2414 cnts
10 % of the Max counts are: - 241 cnts
90 % of the Max counts are: - 2172 cnts

At 10 % of Max counts the time is 3.05 sec
At 90 % of Max counts the time is 3.18 sec

Rise time = 2.44-2.35= **0.13sec.**

Max Overshoot: -

Maximum Overshoot = 2735 Cnts

Max Step Value in counts = 2414 cnts.

Overshoot in % = ((2735-2414)/2414)*100

= 13%

Settling Time:-

Maximum Step Counts are 2414cnts
+/- 5% of Max. Counts are = 2414+120=2534(max)
2414-120 = 2294(min)

So Act Vel (M762) when it is settling in between 5% of Max. Step Counts will give you the Settling Time.

5% Settling Time = 4.64Sec

Conclusion:

Zeta – 'Damping ratio' is calculated from the % max. Overshoot is **0.55**

The **natural frequency 'wn'** calculated from settling time and Zeta of 0.55 comes to **1.16 rads / sec**

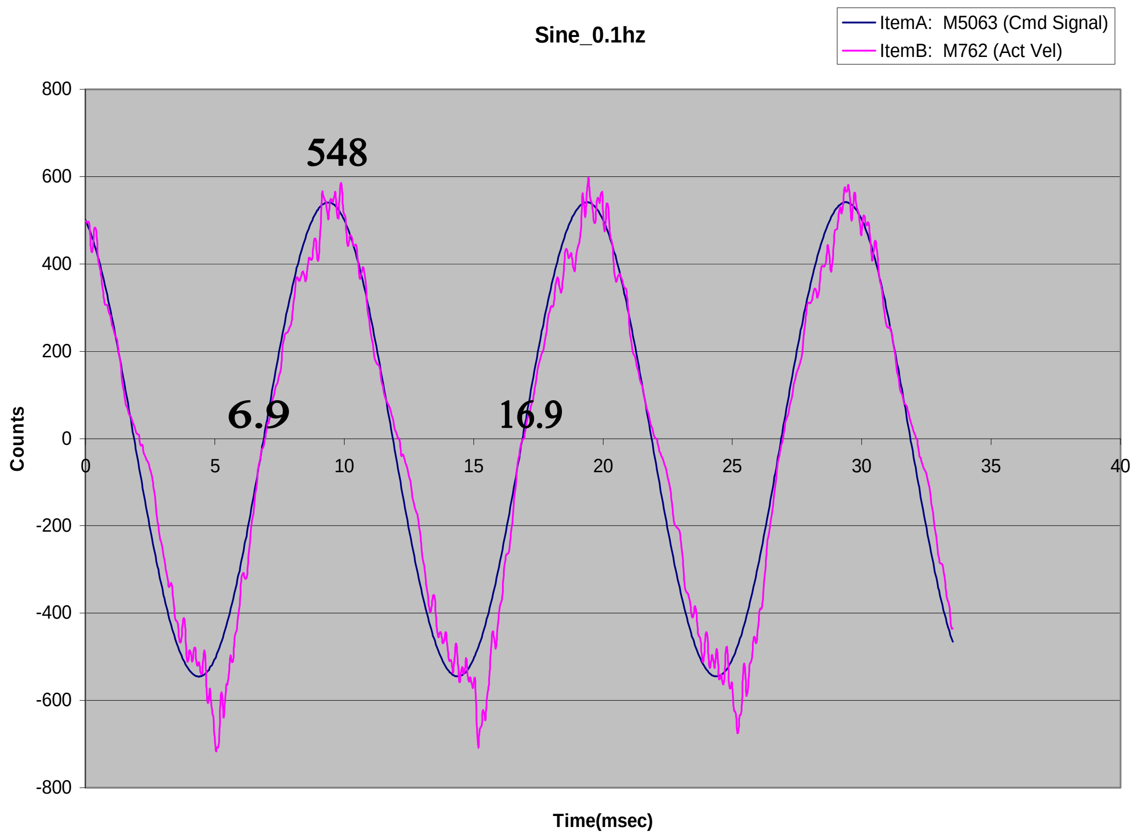
The **damped frequency** of oscillation **'wd'** is **0.96 rads / sec**

The **resonant frequency** of oscillation of the large test setup **'wr'** is **0.72 rads / sec**

14. Sine Response in Velocity Loop:-

Sine wave of frequencies from 0.1Hz to 25 Hz has given to Ch #7 ADC of PMAC instead of analog input. The following plots shows sine responses For frequencies 0.1hz, 0.25hz, 0.5hz, 1hz, 2hz, 3hz, 5hz, 7hz, 10hz, 15hz, 20 Hz, 25 Hz, 30 Hz and 100 Hz.

14.1) **Sine_0.1hz**



Calculations of Magnitude and phase from each frequency response plots **f = 0.1Hz**

Magnitude ratio = 548/548=1

Magnitude in dB = 20 log 1 = 0 **dB**

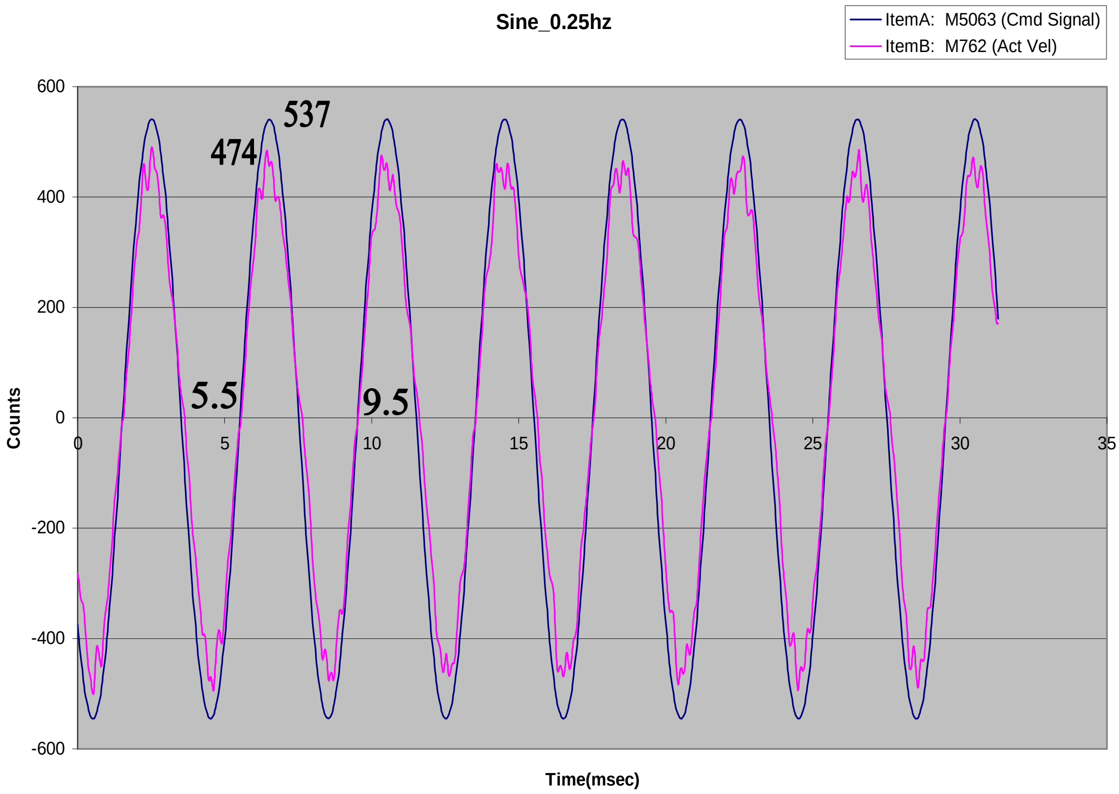
Phase difference = 0-msec;

10m sec = 360deg

0 msec = -0 x 360/10 deg

0 msec **= -0deg**

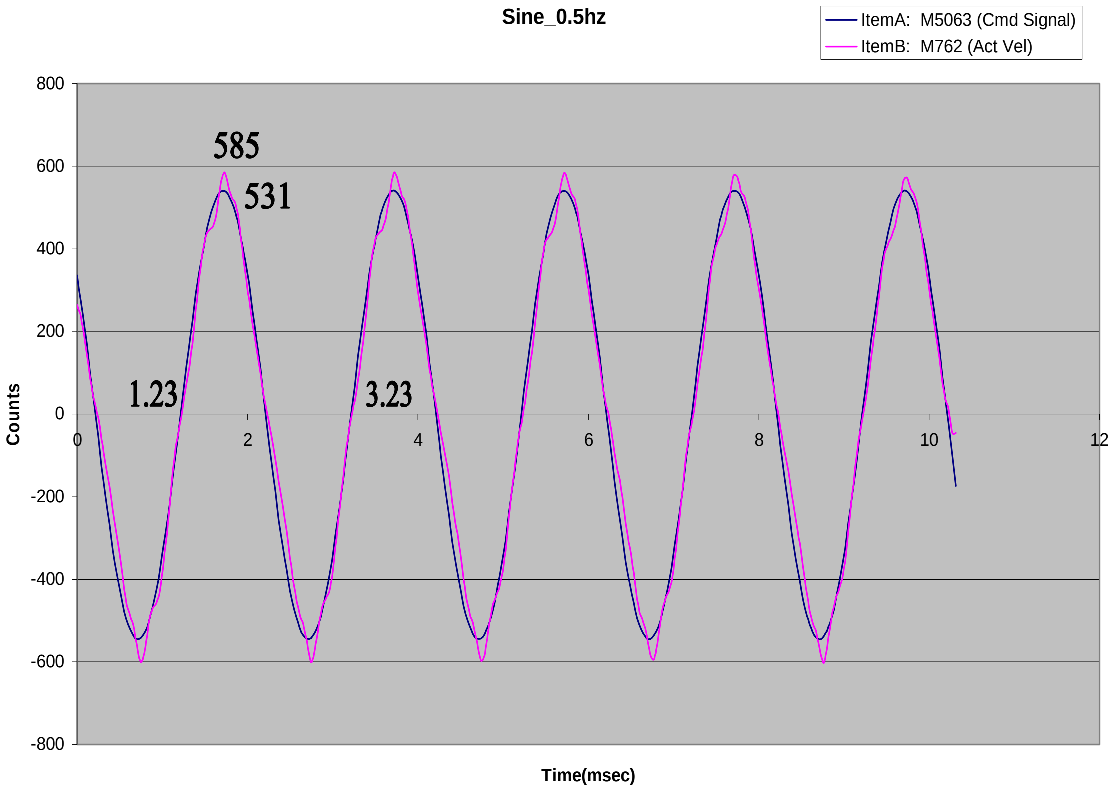
14.2) Sine_0.25hz



Calculations of Magnitude and phase from each frequency response plots **f = 0.25Hz**

Magnitude ratio = 474/537=0.88
Magnitude in dB = 20 log 0.88 = -1.08**dB**
Phase difference = 0-msec;
4m sec - 360deg
0 msec = -0 x 360/4 deg
0 msec = **-0deg**

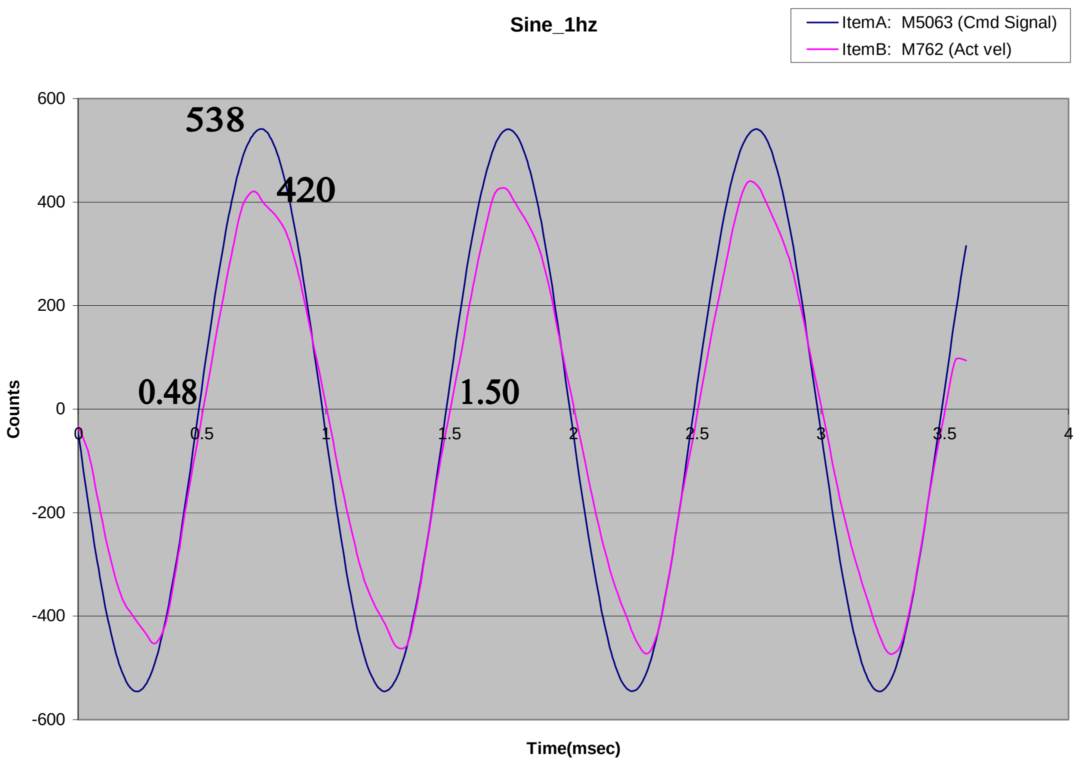
14.3) Sine_0.5hz



Calculations of Magnitude and phase from each frequency response plots **f = 0.5Hz**

Magnitude ratio = 585/531=1.108
Magnitude in dB = 20 log 1.108 = 0.84
Phase difference = 0-msec;
2m sec - 360deg
0 msec = -0 x 360/2 deg
0 msec = **-0deg**

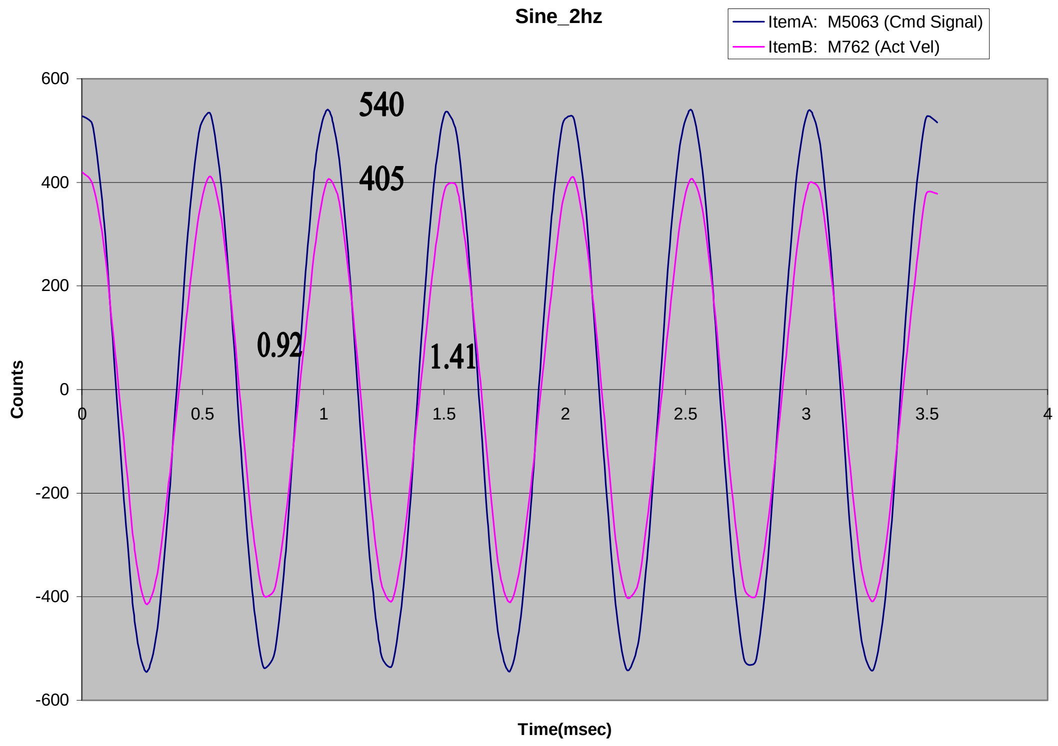
14.4) Sine_1hz



Calculations of Magnitude and phase from each frequency response plots **f = 1Hz**

Magnitude ratio = 420/538=0.78
Magnitude in dB = 20 log 0.78 = -2.15
Phase difference = 0-msec;
1.02m sec - 360deg
0 msec = -0 x 360/1.02 deg
0 msec = **-0deg**

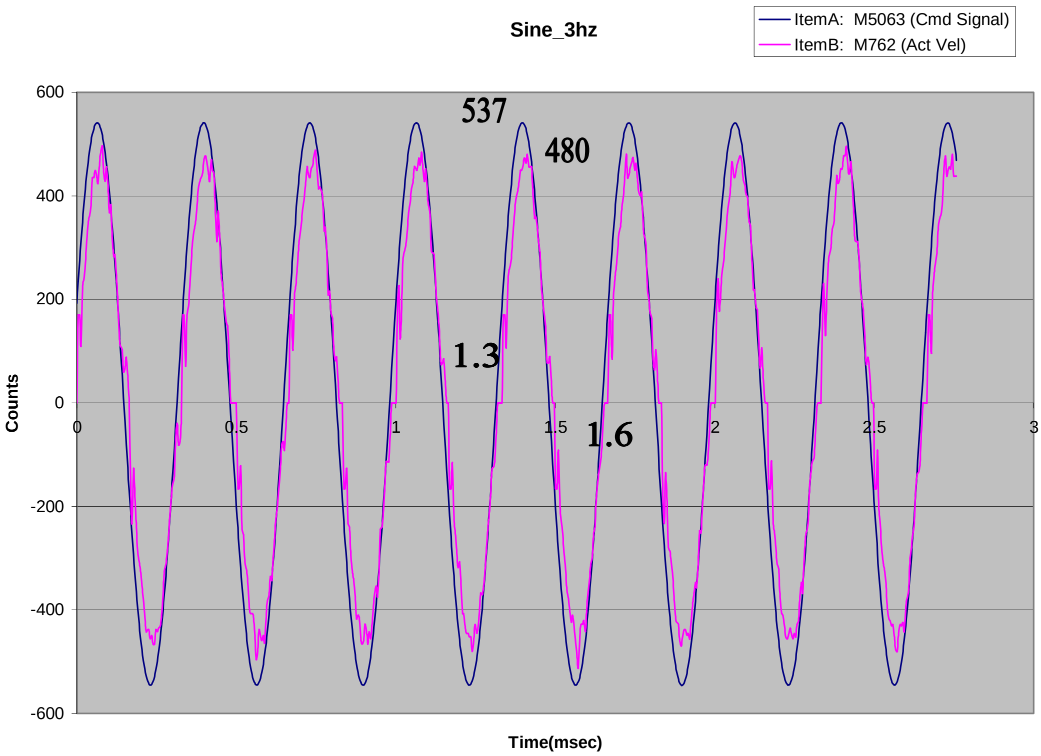
14.5) Sine_2hz



Calculations of Magnitude and phase from each frequency response plots **f = 2 Hz**

Magnitude ratio = 405/540=0.75
Magnitude in dB = 20 log 0.75= -2.4
Phase difference = 0-msec;
0.5m sec - 360deg
0 msec = -0 x 360/1.02 deg
0 msec = **-0deg**

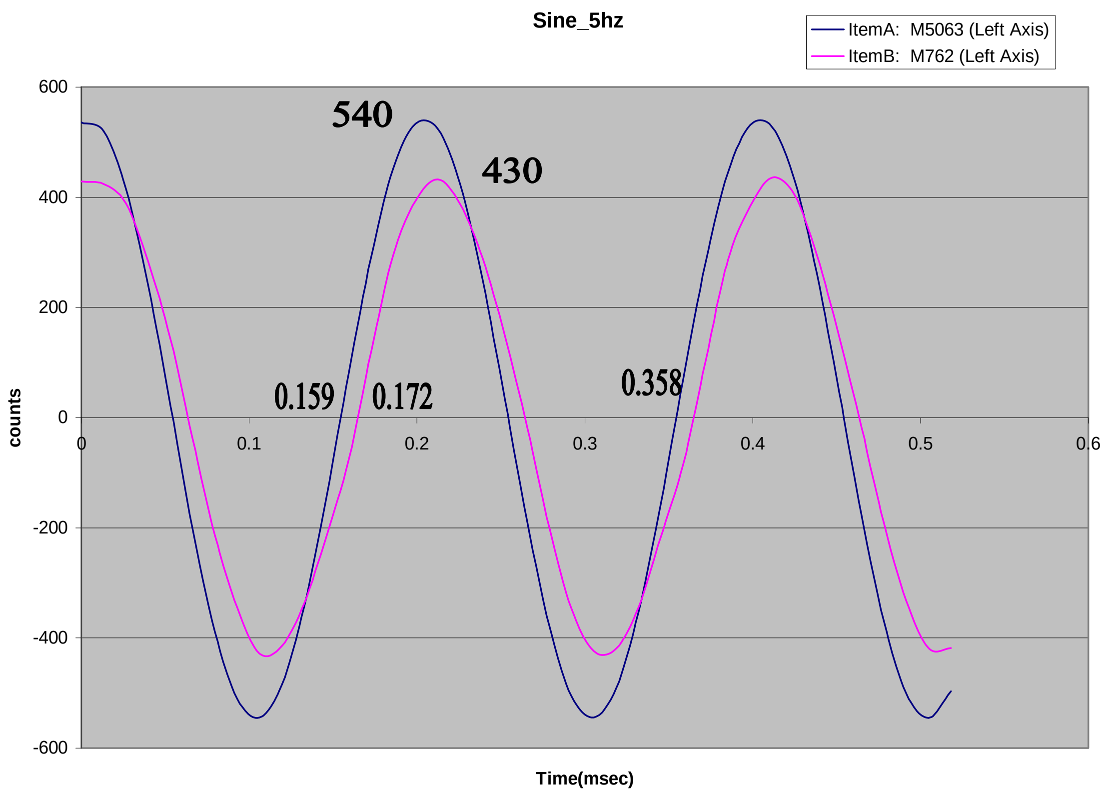
14.6) Sine_3hz



Calculations of Magnitude and phase from each frequency response plots **f = 3 Hz**

Magnitude ratio = 480/537=0.89
Magnitude in dB = 20 log 0.89= -0.97
Phase difference = 0-msec;
0.3m sec - 360deg
0 msec = -0 x 360/0.3 deg
0 msec = **-0deg**

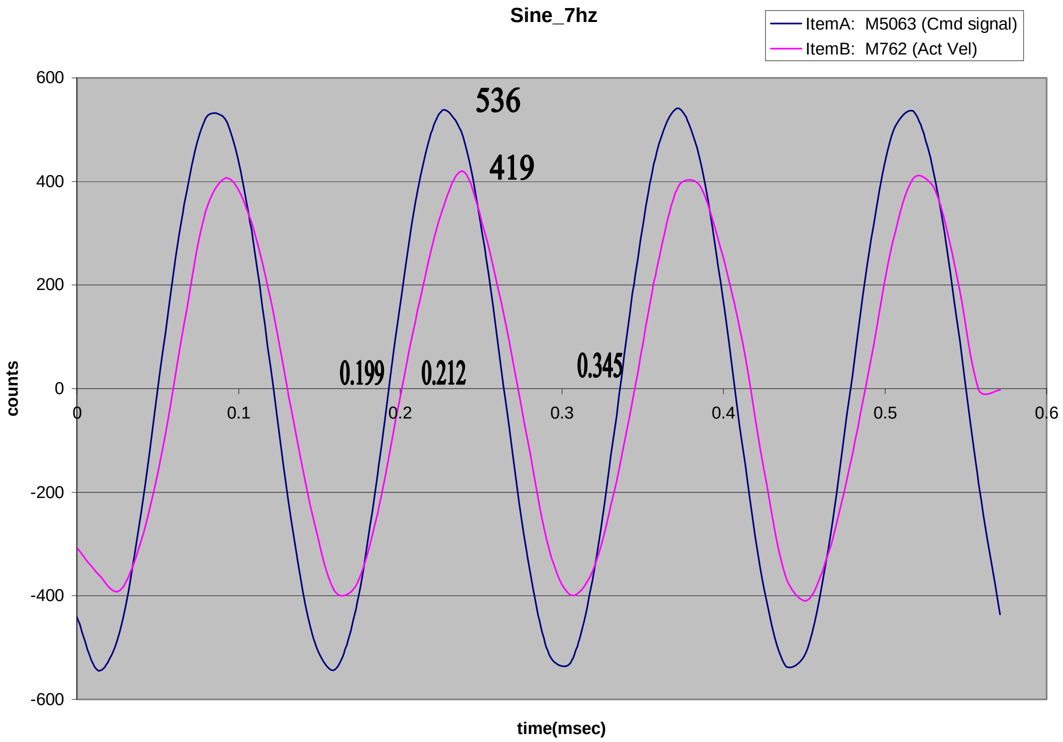
14.7) Sine_5hz



Calculations of Magnitude and phase from each frequency response plots f = 5 Hz

Magnitude ratio = 430/540=0.796
Magnitude in dB = 20 log 0.796= -1.97
Phase difference = 0-msec;
0.2m sec - 360deg
0.013 msec = -0.013 x 360/0.2 deg
0 msec = **-23deg**

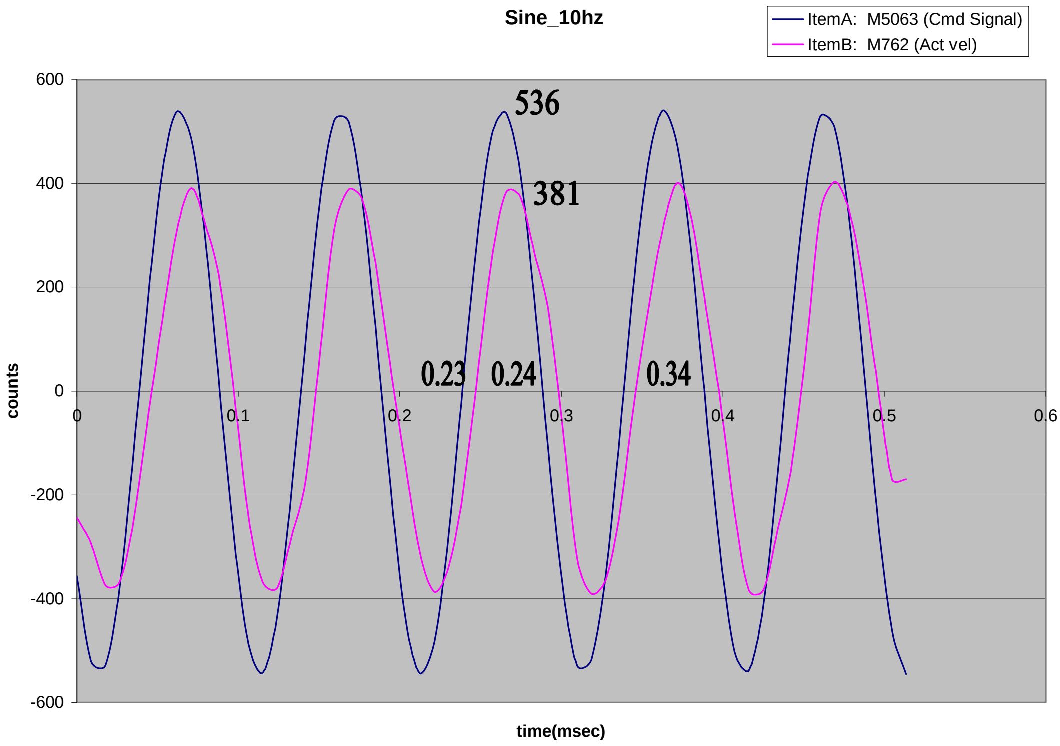
14.8) Sine_7hz



Calculations of Magnitude and phase from each frequency response plots **f = 7 Hz**

Magnitude ratio = 419/536=0.781
Magnitude in dB = 20 log 0.781= -2.13
Phase difference = 0.013-msec;
0.146m sec - 360deg
0.013 msec = -0.013 x 360/0.146 deg
0.013 msec **= -32deg**

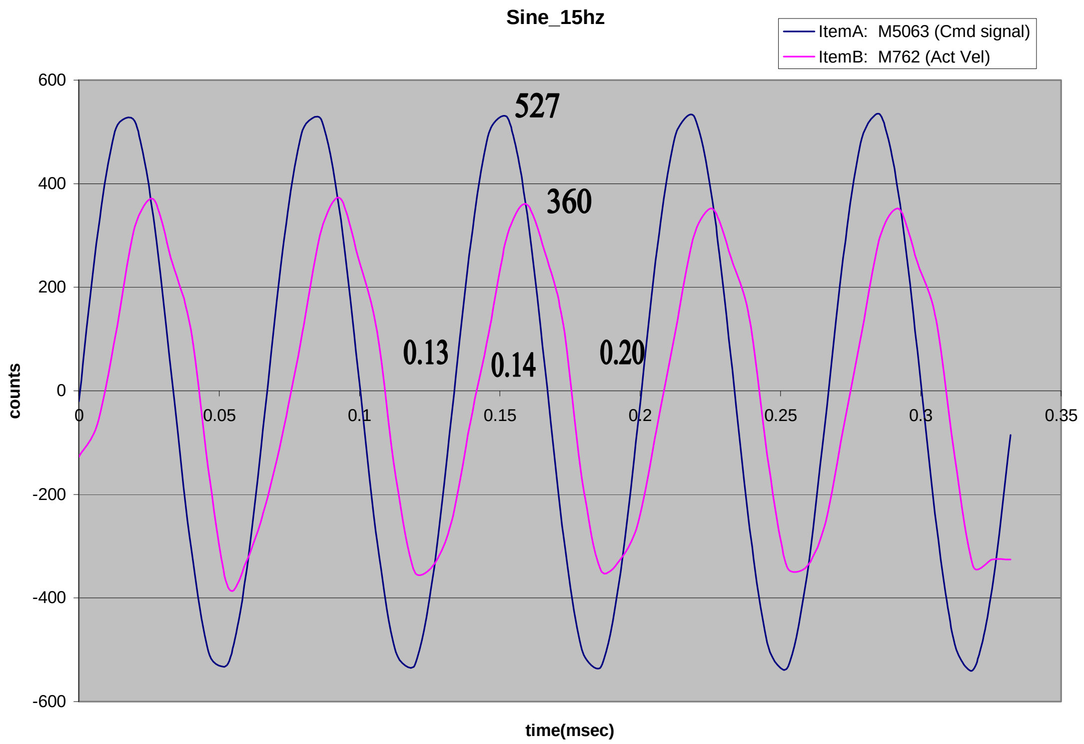
14.9) Sine_10Hz



Calculations of Magnitude and phase from each frequency response plots **f = 10 Hz**

Magnitude ratio = 381/536=0.71
Magnitude in dB = 20 log 0.71= -2.96
Phase difference = 0.01-msec;
0.11m sec - 360deg
0.01 msec = -0.01 x 360/0.11 deg
0.01 msec = **-33deg**

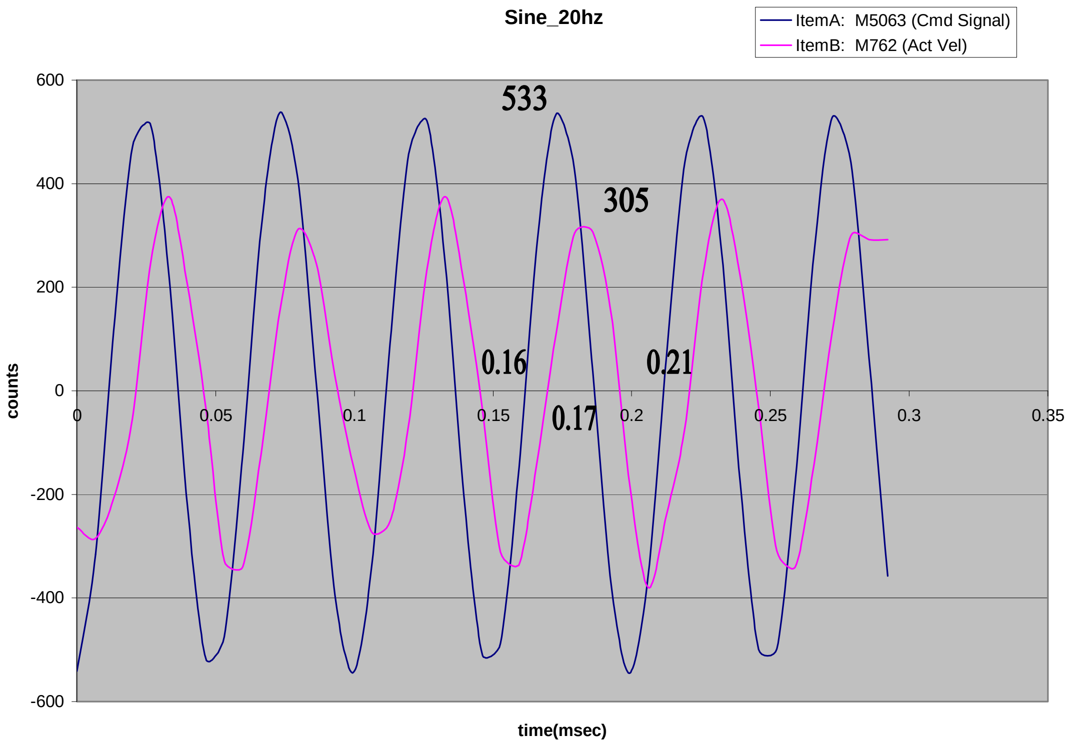
14.10) Sine_15hz



Calculations of Magnitude and phase from each frequency response plots **f = 15 Hz**

Magnitude ratio = 360/527=0.68
Magnitude in dB = 20 log 0.68= -3.3
Phase difference = 0.01-msec;
0.07 m sec - 360deg
0.01 msec = -0.01 x 360/0.07 deg
0.01 msec = **-51deg**

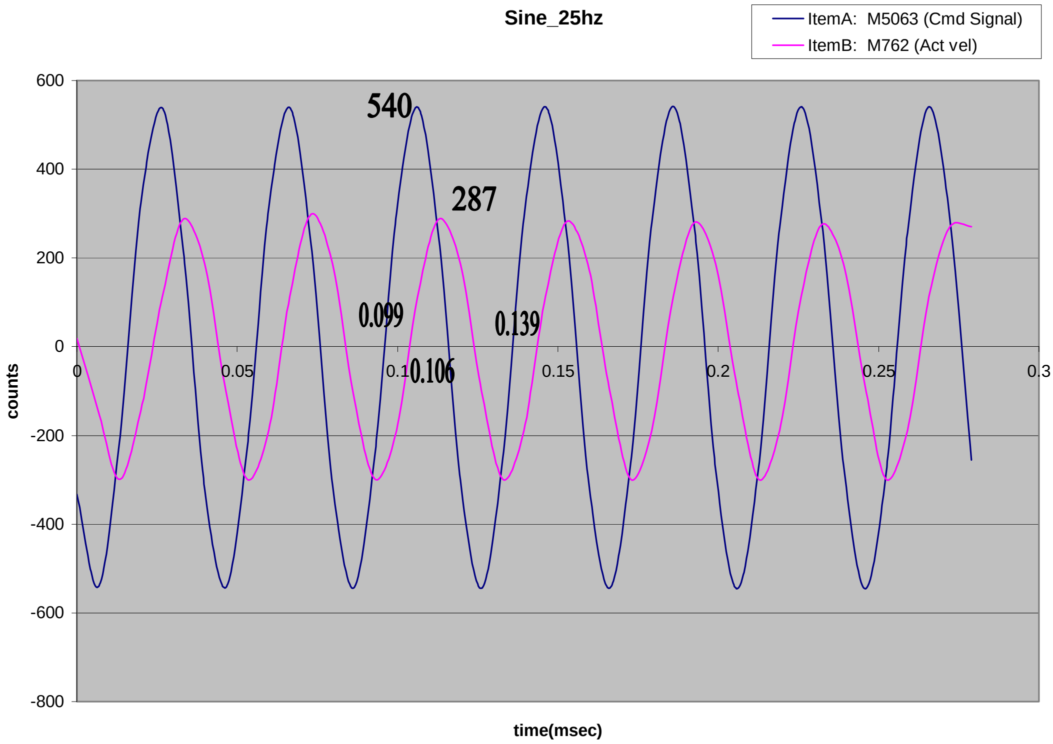
14.11) Sine_20hz



Calculations of Magnitude and phase from each frequency response plots **f = 20 Hz**

Magnitude ratio = 305/533=0.57
Magnitude in dB = 20 log 0.68= -4.8
Phase difference = 0.01-msec;
0.05 m sec - 360deg
0.01 msec = -0.01 x 360/0.05 deg
0.01 msec = **-72deg**

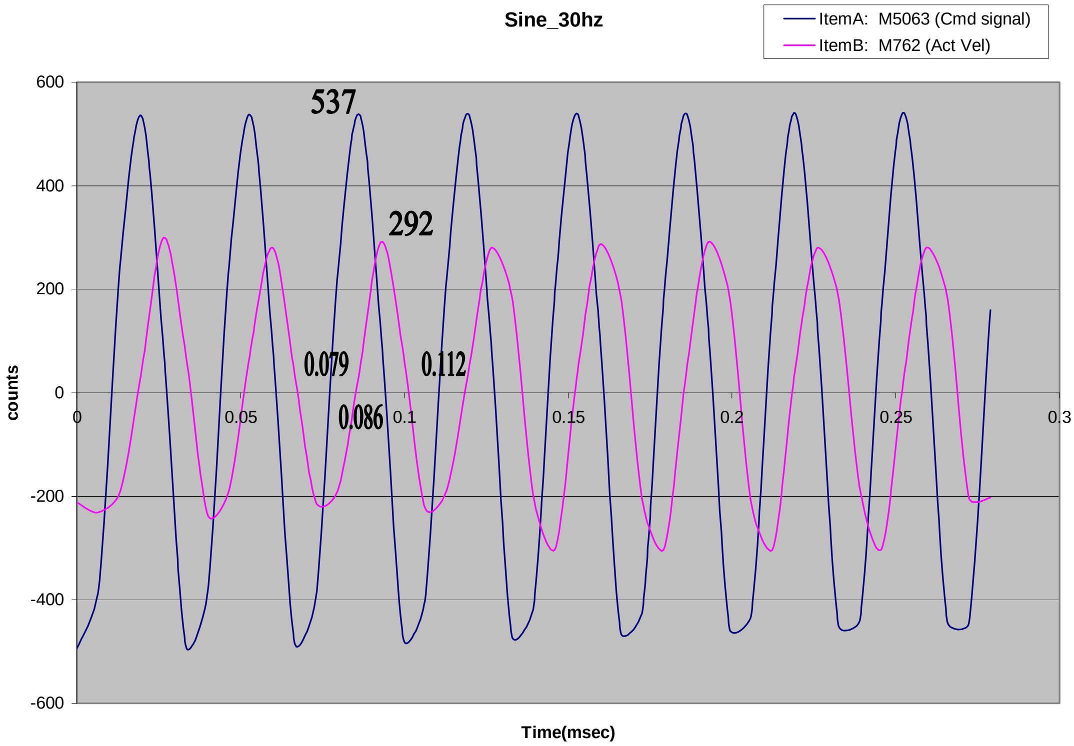
14.12) Sine_25hz



Calculations of Magnitude and phase from each frequency response plots **f = 25 Hz**

Magnitude ratio = 287/540=0.53
Magnitude in dB = 20 log 0.68= -5.4
Phase difference = 0.007-msec;
0.04 m sec - 360deg
0.007 msec = -0.007 x 360/0.04 deg
0.007 msec **= -63deg**

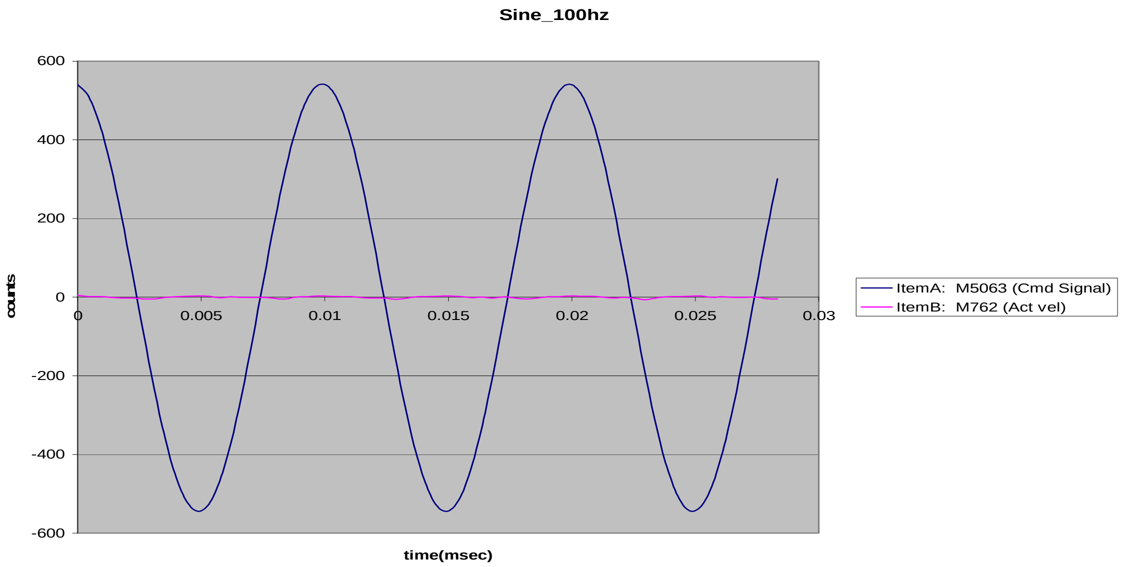
14.13) Sine_30hz



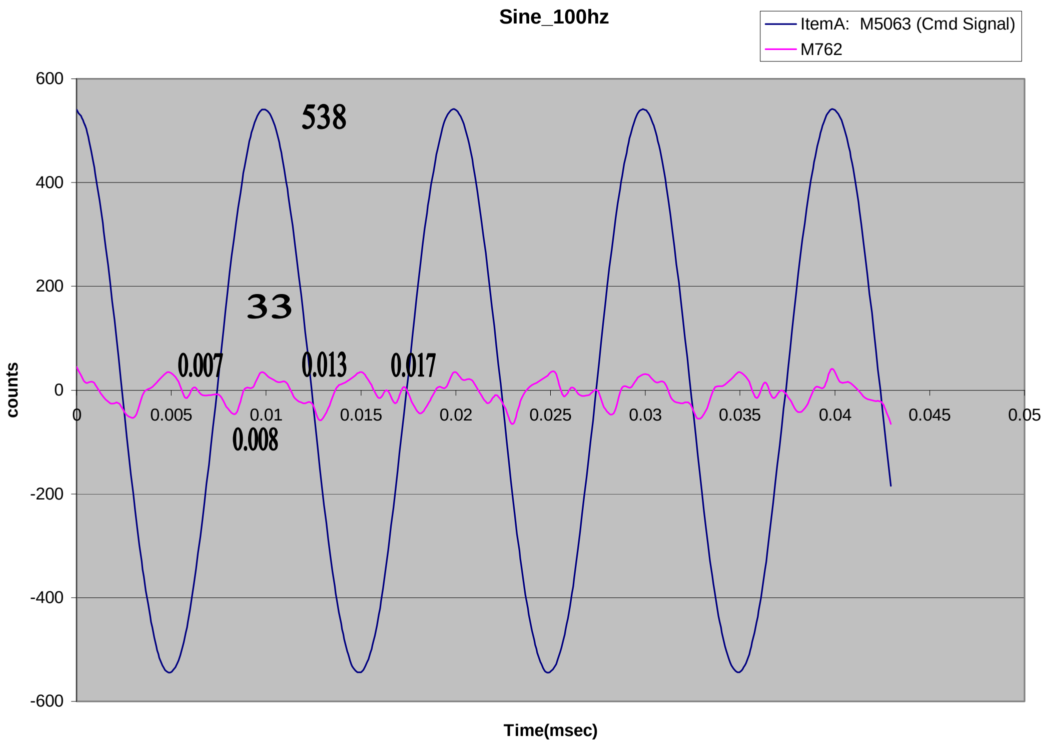
Calculations of Magnitude and phase from each frequency response plots **f = 25 Hz**

Magnitude ratio = 292/537=0.54
Magnitude in dB = 20 log 0.68= -5.2
Phase difference = 0.007-msec;
0.033 m sec - 360deg
0.007 msec = -0.007 x 360/0.033 deg
0.007 msec = **-76deg**

14.14) Sine_100hz



Here Act Vel (M762) is not visible, so I am multiplying M762 with 10, calculating gain and dividing with 10.
After multiplying with 10, the above plot is shown below

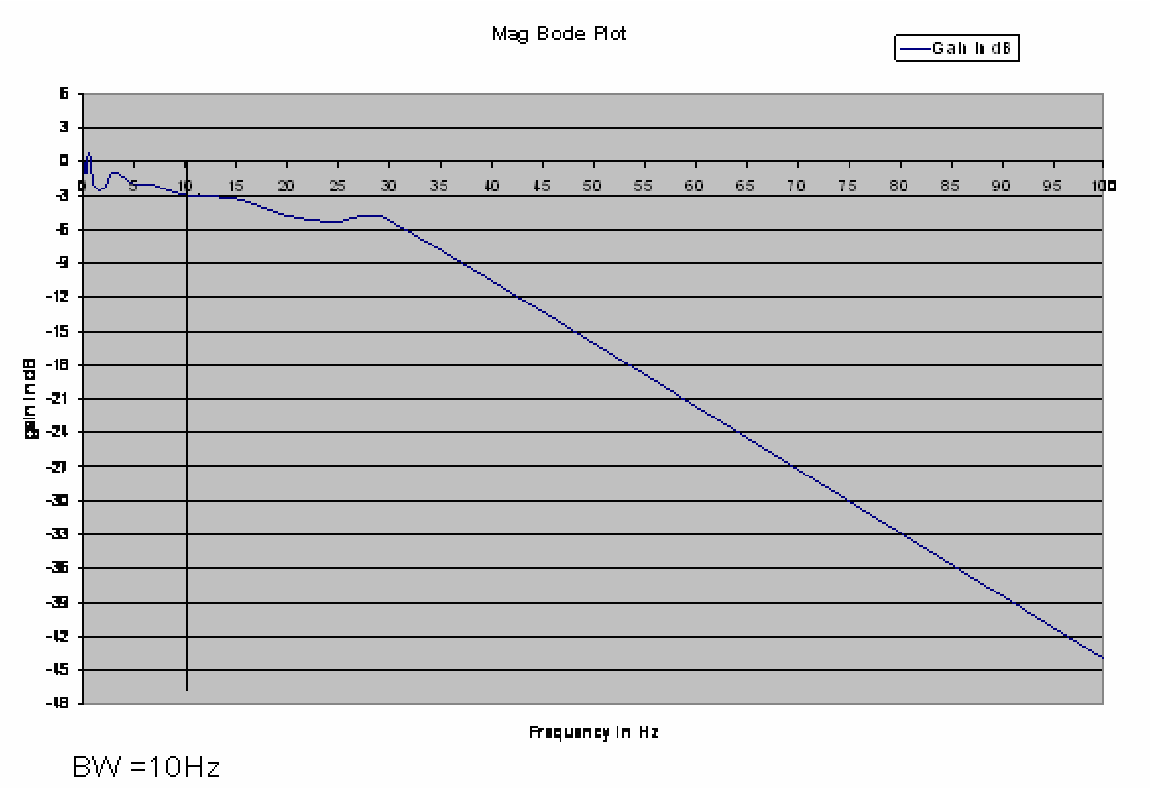


Calculations of Magnitude and phase from each frequency response plots **f =100 Hz**

Magnitude ratio = 33/538=0.061 (but we multiplied with 10 to M762), so actual
Magnitude ratio = 0.061/10=0.0061
Magnitude in dB = 20 log 0.0061= -44dB
Phase difference = (0.001+0.004)/2 = 0.0025
0.01 m sec - 360deg
0.0025 msec = -0.0025 x 360/0.01 deg
0.001 msec **= -90deg**

Sine wave frequencies and related magnitude as given below in table

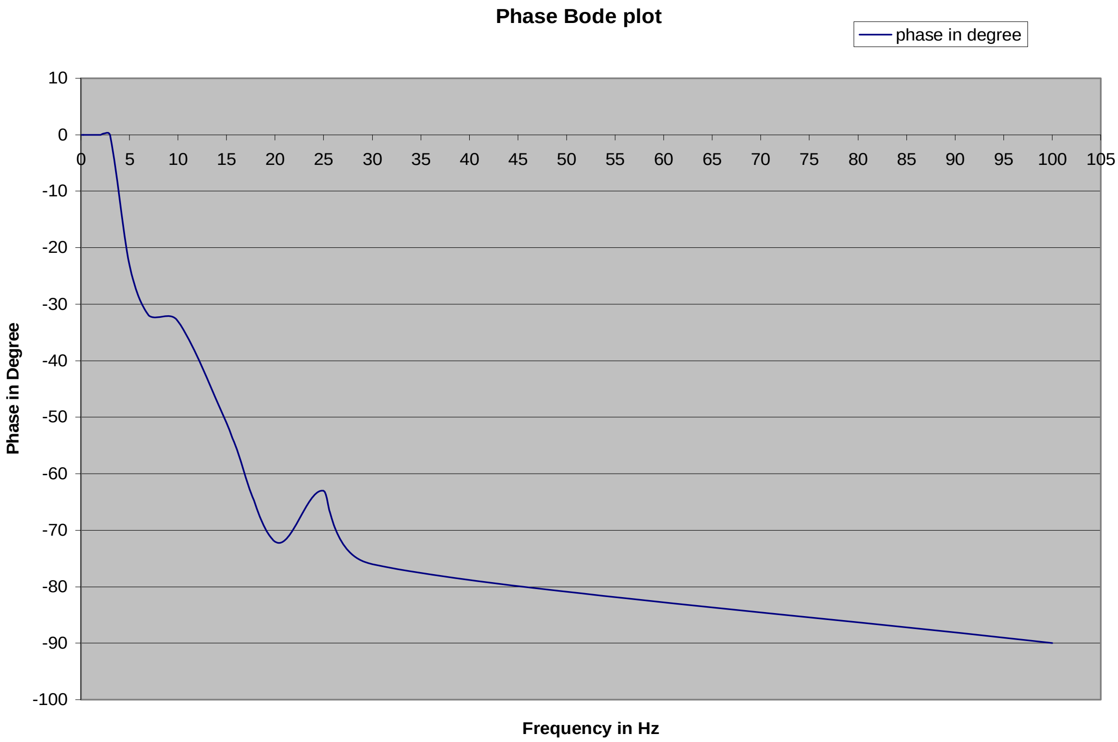
frequency in Hz	Gain in dB
0.1	0
0.25	-1.08
0.5	0.84
1	-2.15
2	-2.4
3	-0.97
5	-1.97
7	-2.13
10	-2.96
15	-3.3
20	-4.8
25	-5.4
30	-5.2
100	-44



3 dB bandwidth of Velocity loop is given by **10Hz** (according to above fig.)

Sine wave frequencies and related Phases as given below in table

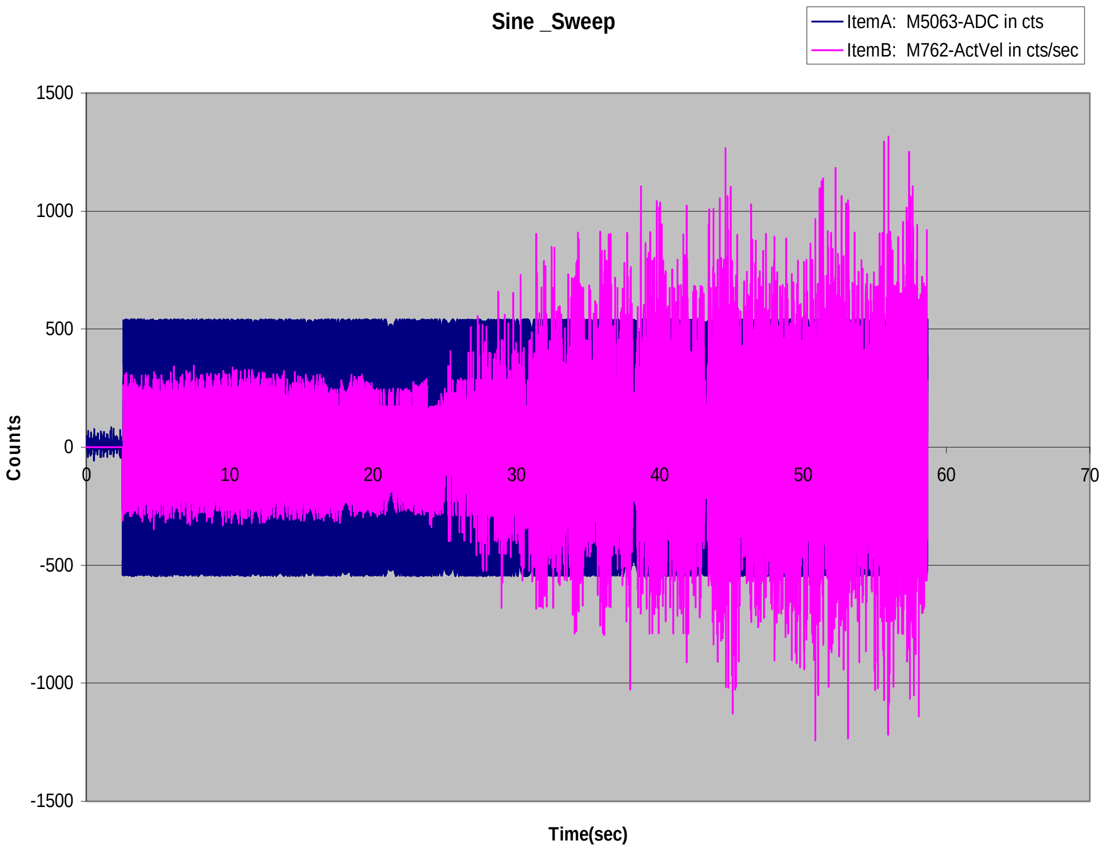
frequency in Hz	phase in degree
0.1	0
0.25	0
0.5	0
1	0
2	0
3	0
5	-23
7	-32
10	-33
15	-51
20	-72
25	-63
30	-76
100	-90



15. Sine Sweep in Velocity Loop:-

Sine sweep from 0.1 Hz to 100 Hz with 100sec
Sine Sweep has given to Ch #7 ADC of PMAC instead of analog Input. The following shows sine sweep responses.

Sine sweep from 0.1Hz to 100 Hz, Sweep time =100sec and ActVel and CmdVel in cts/sec



16. Conclusion of the tests on Large test setup:-

The plots taken by us are similar to the plots by Mr. Leopold of MACCON GmbH, Germany in his visit in Jan 2009 and April 2009, both the procedure and results are validated.

17. Future Plans:-

5.1) Testing with present SSC (Station Servo Computer) or PC104 with Velocity Loop (in PMAC).
5.2) Testing Counter Torque Card and Servo star 610 drives without PMAC.

Leopold's test report of tests performed in large test setup and C04 antenna

Large Test setup report:-

05.01.2009

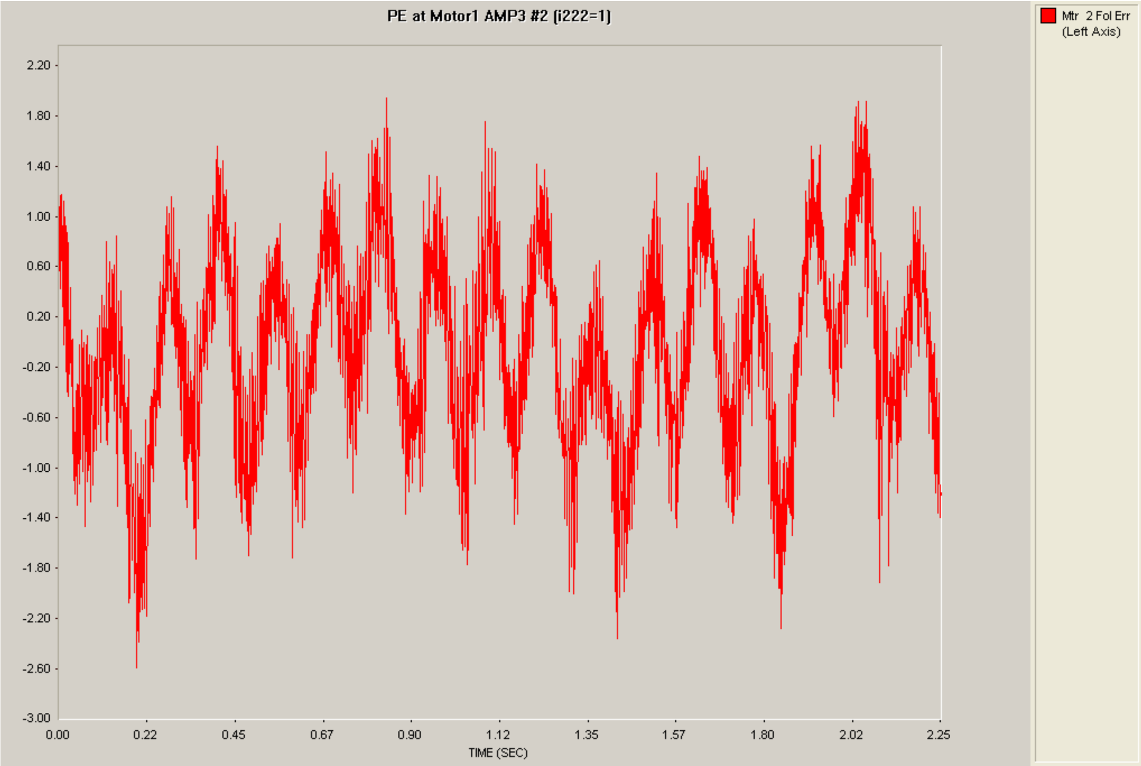
Ad 1) Tests with small test set-up system

Motor 1 with Amp 3 and #2 of PMAC
Motor 2 with Amp 4 and #3 of PMAC
Load enc at ENC 1 of PMAC

New tuning of motor 1, Results:

Ixx30 = 250.000	P-Gain
Ixx31 = 7.000	D-Gain
Ixx32 = 7.000	Vel FF
Ixx33 = 10.000	I-Gain
Ixx34 = 0	I Mode Switch

Results see below, during the move we have a position error of about +/- 2 counts:



Checking the play

#2j/ (oscillating)
#3o5 (not oscillating)

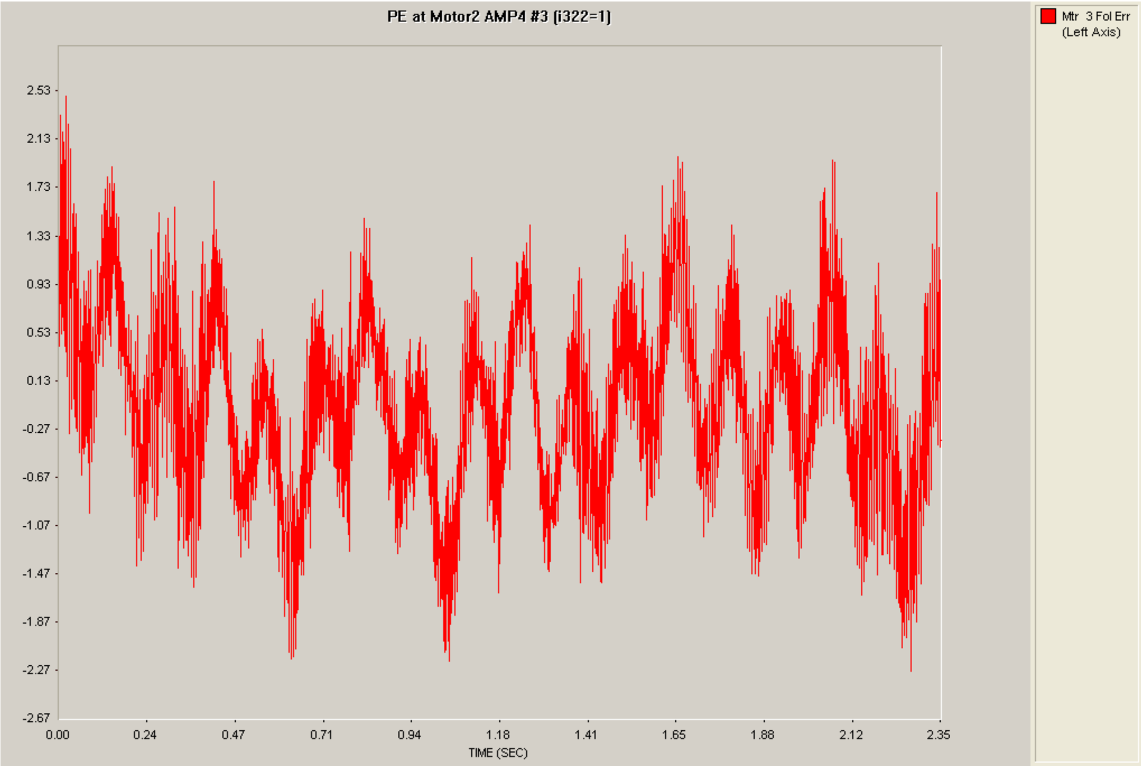
#1hmz#2hmz#3hmz

#3o-5 gives:

#1: -3 counts
#2: 0 counts
#3: -49 counts

Now copying the PID values to motor 2

Results are very similar; during the move we have and position error of about +/- 2 counts:



Checking the play

#3j/ (oscillating)
#2o5 (not oscillating)

#1hmz#2hmz#3hmz

#2o-5 gives:

#1: -4 counts
#2: -48 counts
#3: 0 counts

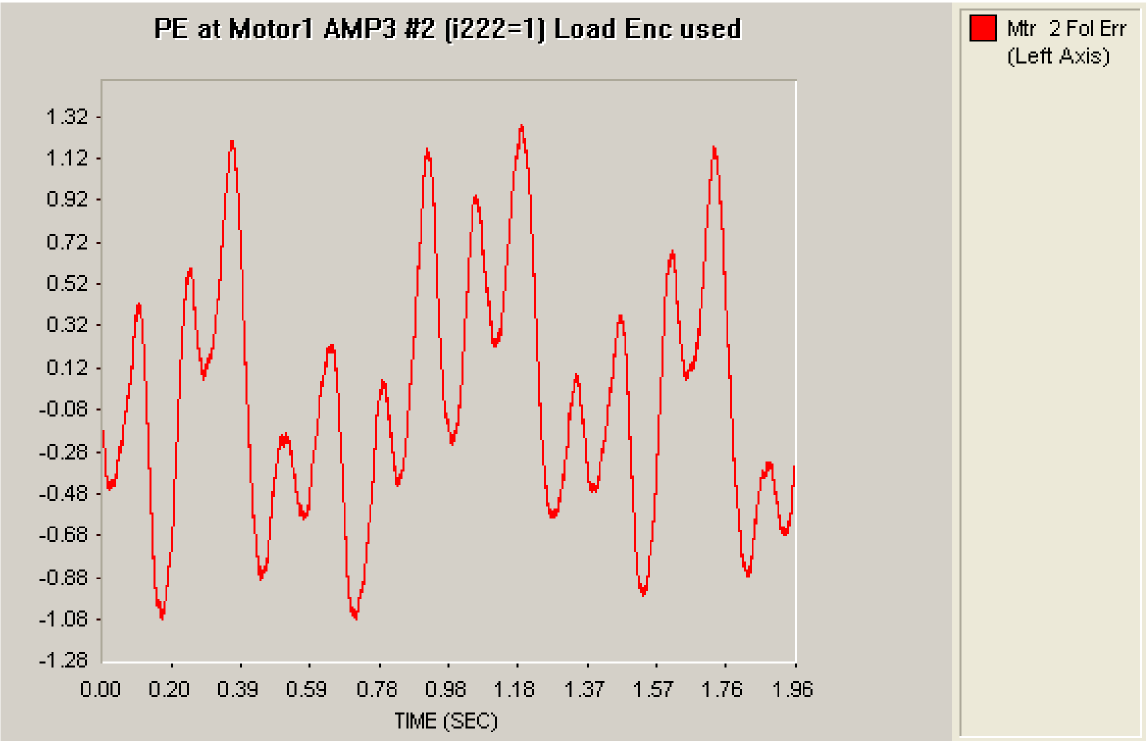
→ is ok

Now test with load encoder and simplified backlash compensation
(by adding a torque bias to the other motor by open loop command)

Tuning:

Ixx30 = 300.000	P-Gain
Ixx31 = 7.000	D-Gain
Ixx32 = 7.000	Vel FF
Ixx33 = 10.000	I-Gain
Ixx34 = 0	I Mode Switch

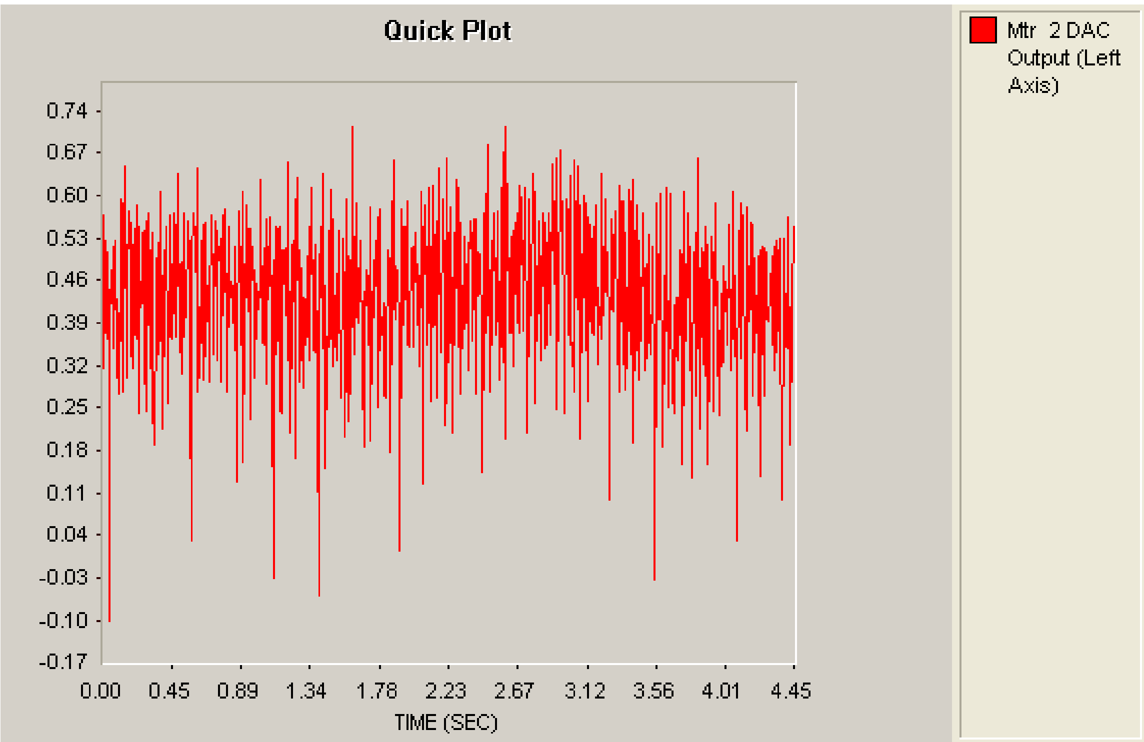
Results see below, during the move we have a position error of about +/- 1 count:



06.01.2009
Ad 2) Tests with big test set-up system

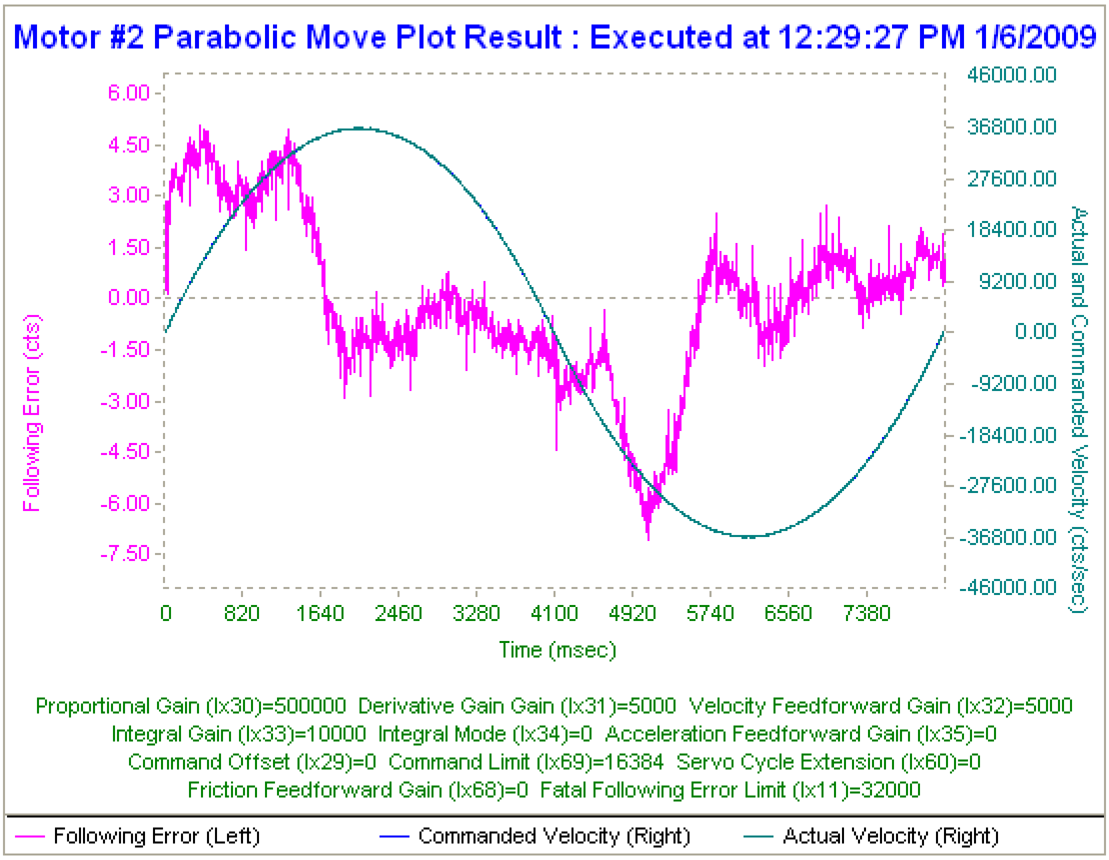
Motor 1 with Amp 3 and #2 of PMAC
Motor 2 with Amp 4 and #3 of PMAC
Load enc at ENC 1 of PMAC

Running at 1000 rpm with Motor 1, Motor 2 is in open loop, Braking motor decoupled, but the gear box is still mounted

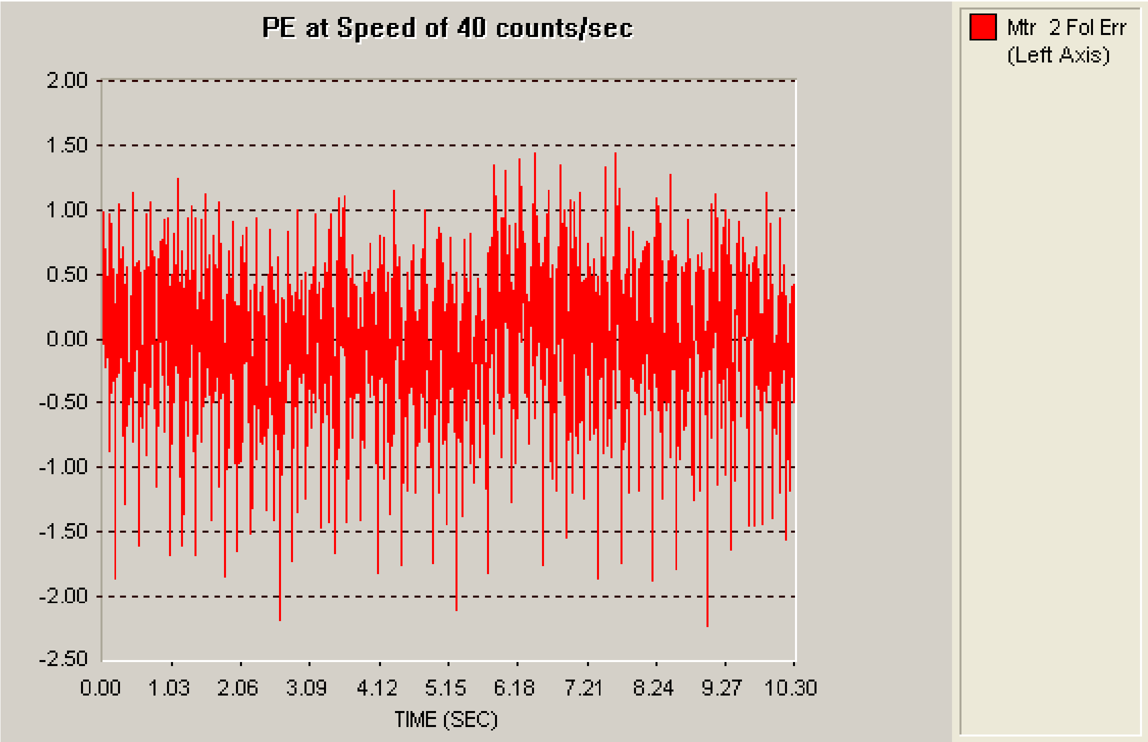


M268 is in the range of about 1000 digits, Measurements with the current probe is about 1.8 to 2A, see plot 1 of the FLUKE scope, used with MACCON current probe.

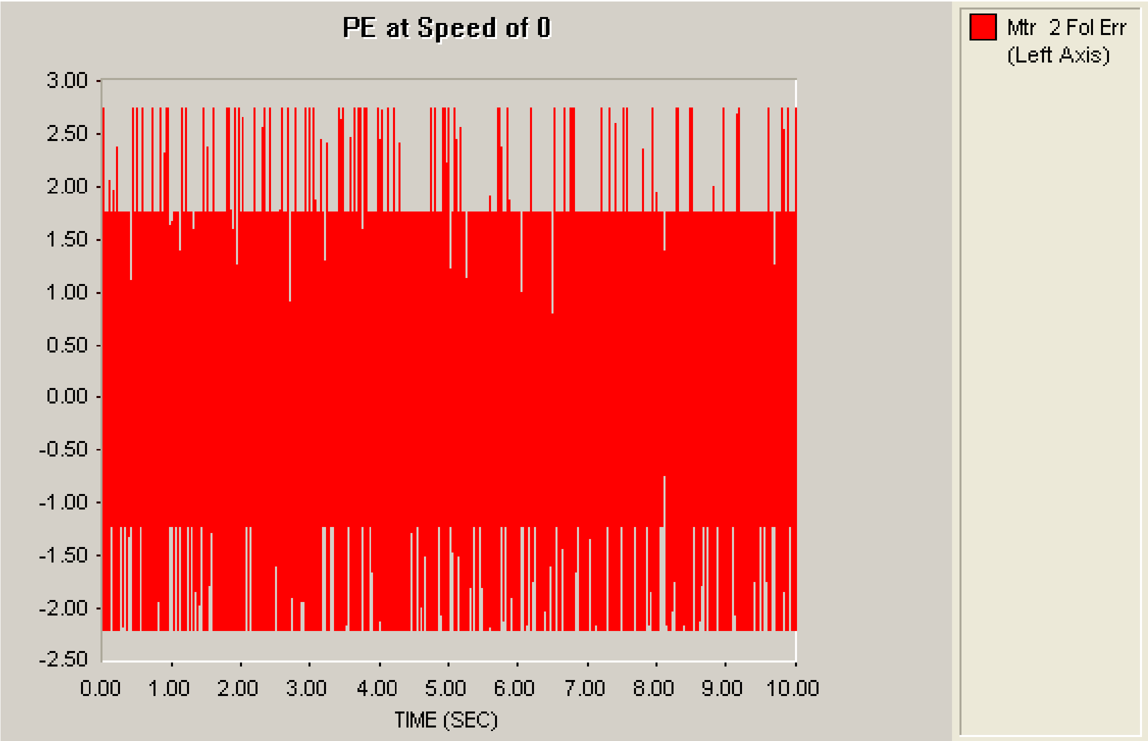
Now tuning motor 1



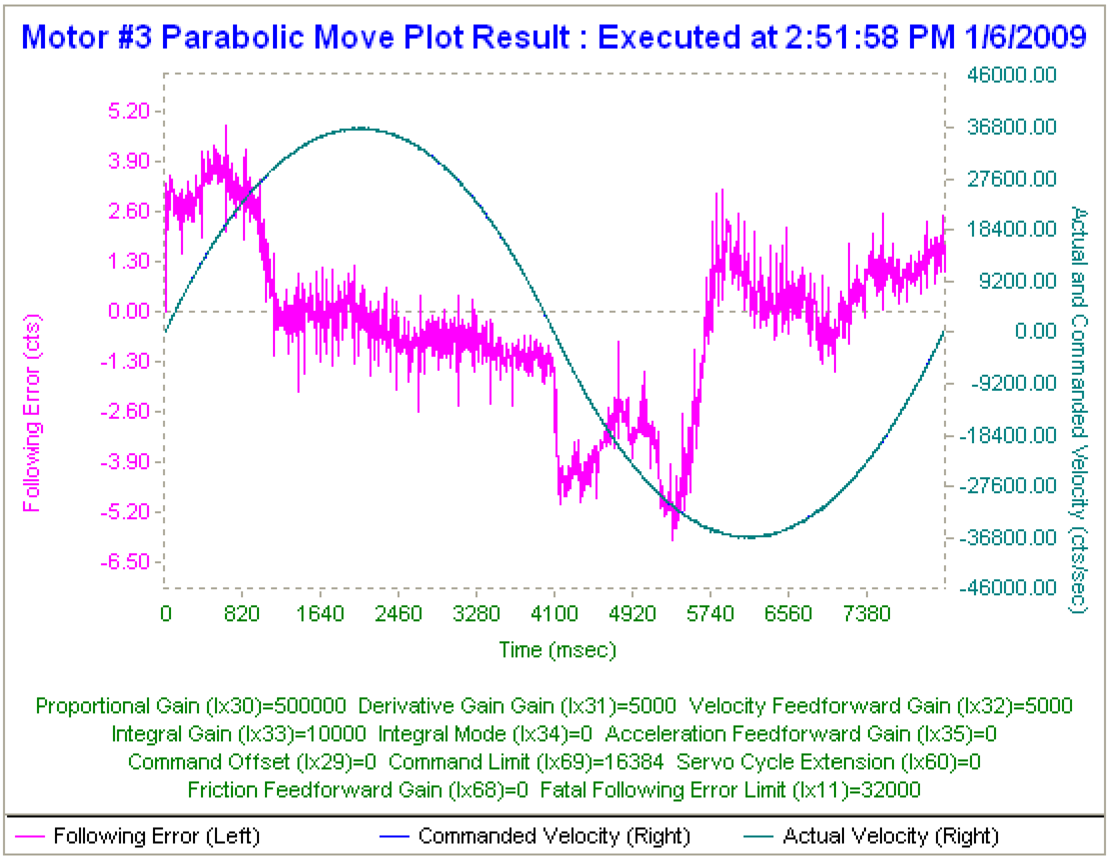
Move with constant speed (40 counts/msec):



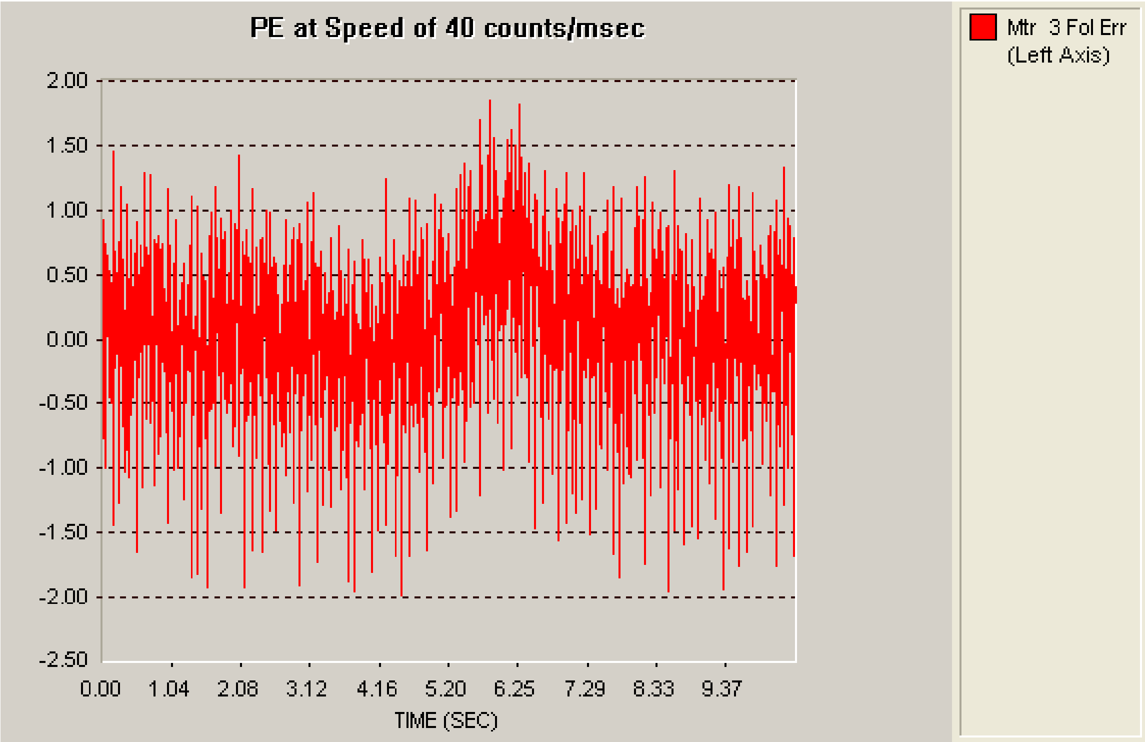
In Position (speed = 0):



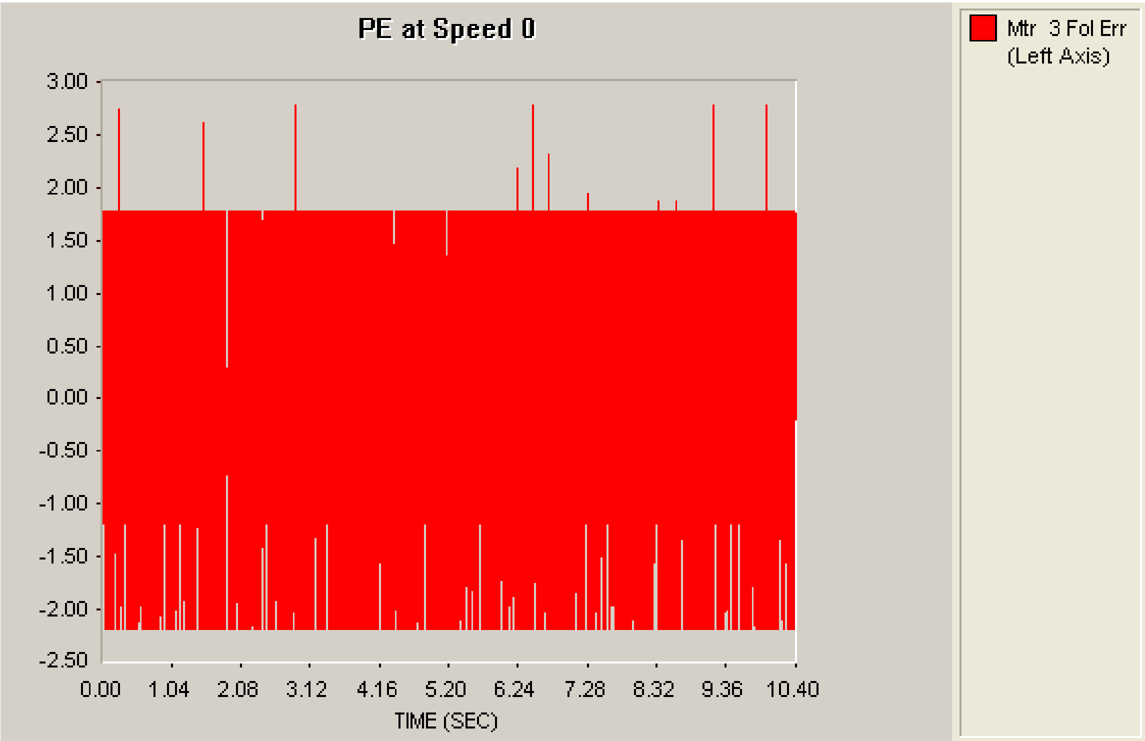
Now with motor 2



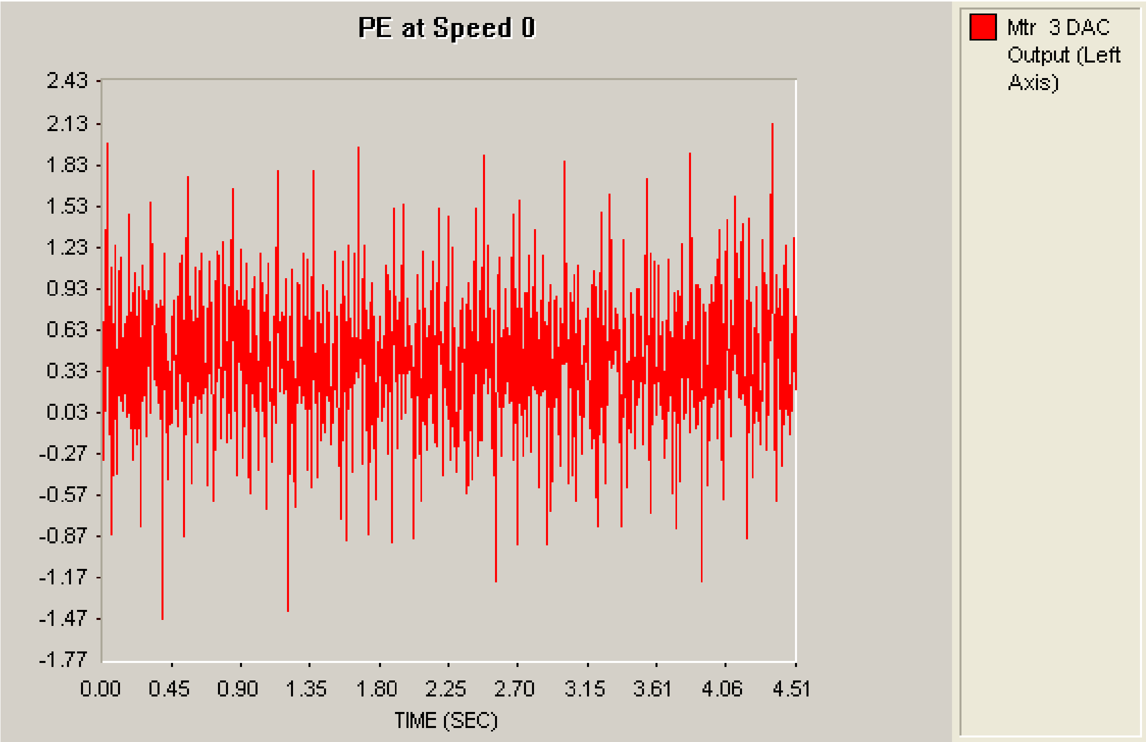
Move with constant speed (40 counts/msec):



In Position (speed = 0):



Running at 1000 rpm with Motor 2, Motor 1 is in open loop, Braking motor decoupled, but the gear box is still mounted

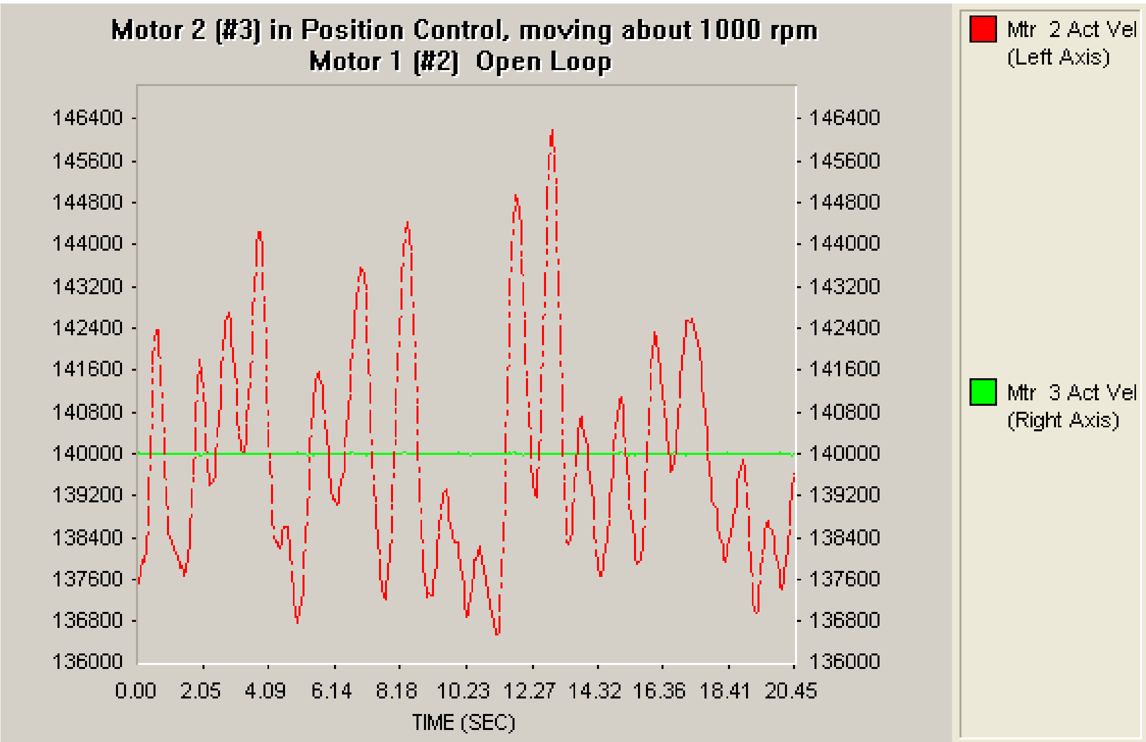


M368 is in the range of about 1500 digits

Measurements with the current probe is about 2.3A, see plot 1 of the FLUKE scope, used with MACCON current probe.

Values are higher than motor 1, reason is the tuning (which was not made before!)

Open point: How many points can be saved in PMAC?
Answer: about 13500 values can be stored in PMAC data gathering



Absolute load encoder mounted, check of the resolution:

Move over one full revolution of the big wheel.

8192 (resolution of the resolver at the motor) * 1488.94 (gear at the motor) * 73 / 19 (gear) = 46863366
→ gives one rev of the wheel

Resolution at the load encoder: 8192 (sin/cos) * 50 (interpolation of IBV102) * 4 (quad)
→ gives 1638400 → ok!

07.01.2009
IBV resolution changed to 100* interpolation, unit used:
IBV102: Ident. no. 536 422-20, serial no. 21 616 432
Settings in the interpolation box (highest frequency setting → could be lowered)

S1	S2	S3	S4	S5	S6	S7	S8
off	on	on	on	off	off	off	on

PMAC changed:
old SN: C0004HSV (4 axis module), new SN: C00004JWX (8 axis module)

Open point:

- Servostar wiring of analog signals → picture?

Configuration changed back to the original state:
Motor 1 → Servostar 3 → Axis 1 of the PMAC
Motor 2 → Servostar 4 → Axis 2 of the PMAC
Load Encoder → ENC5 of PMAC

Setting the configuration in the new drive and checking all axes
→ ok

Current consumption at 24V DC:
PMAC + 2 servostars (brakes not active): 2,1A
PMAC + 2 servostars (1 brake active): 3,3A
PMAC + 2 servostars (2 brakes active): 4,5A

Measuring the backlash at the big test set-up system (carried out as before at the small test set-up; → o10 gives about 2A (measured via Servostar = effective current)

#1j/		
#1hmz#2hmz#5hmz		
#2o-10	#2: -36013	#5: -1456
#2o10	#2: 34318	#5: 1385
	70331	2841
#2j/		
#1hmz#2hmz#5hmz		
#1o10	#1: 39209	#5: 1022
#1o-10	#1: -33039	#5: -1172
	72248	2194
#1j/		
#1hmz#2hmz#5hmz		
#2o-5	#2: -10732	#5: -399
#2o5	#2: 27597	#5: 828
	38329	1227
#2j/		
#1hmz#2hmz#5hmz		
#1o5	#1: 36716	#5: 1389
#1o-5	#1: -981	#5: -139
	37697	1527

Test with dual feedback (motor 1 and load encoder = ENC 5, motor 2 in idle)

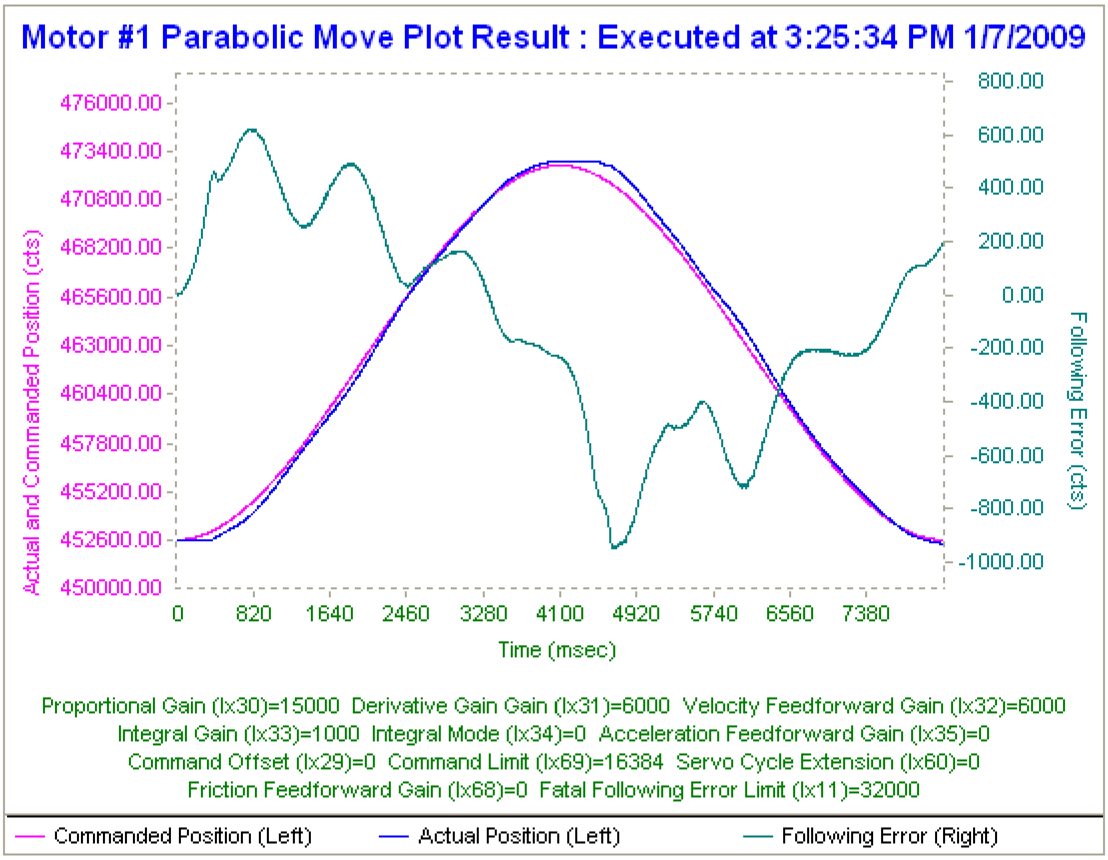
Therefore:

I103	Pos. Feedback	$3505
I104	Vel. Feedback	$3501

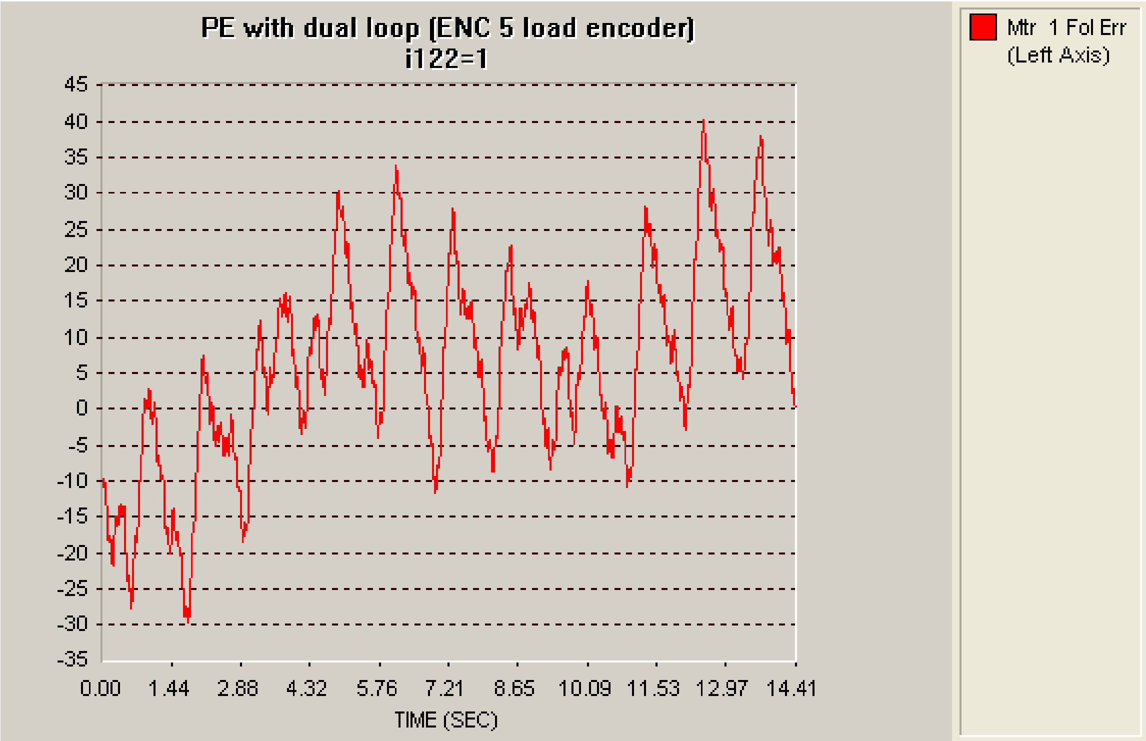
I108	Pos. Feedback Scaling	96 (default)
I109	Vel. Feedback Scaling	8 (exact: 6,72) since:

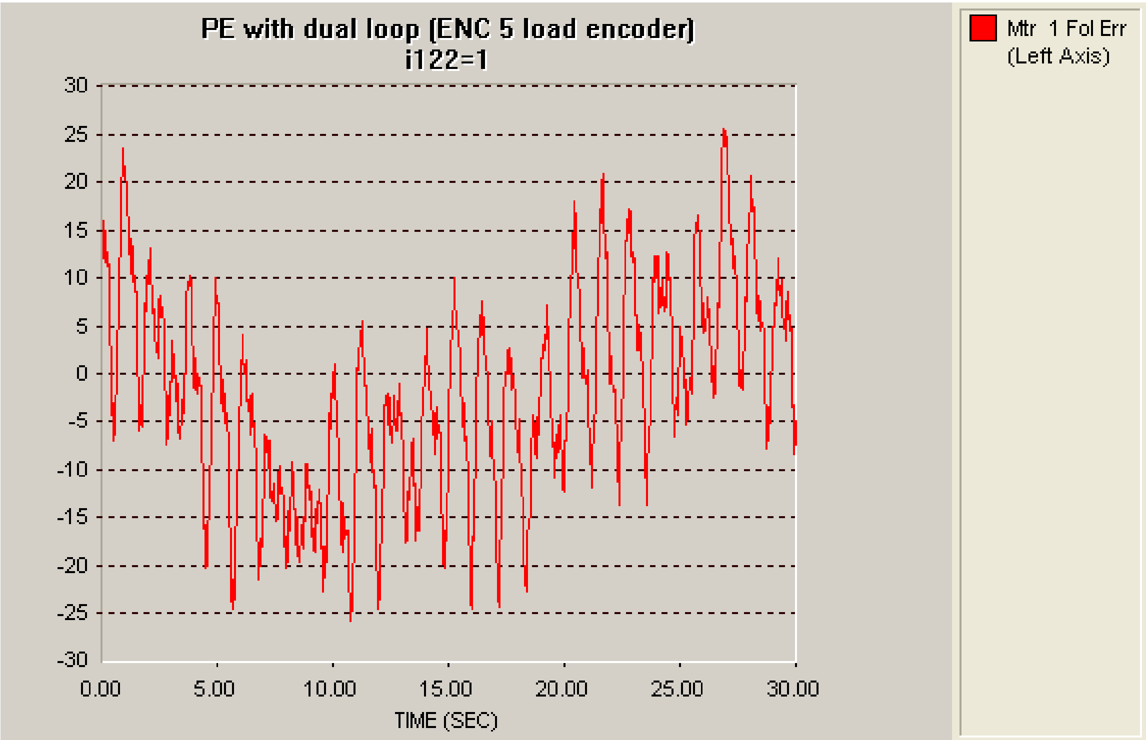
Resolver / Motor resolution for 1 rev at the big wheel:	8192 * 1488 * 3,84 = 46808432
Load Encoder resolution	8192 * 400 = 3276800
Motor / Load resolution:	14.28

After some tuning:

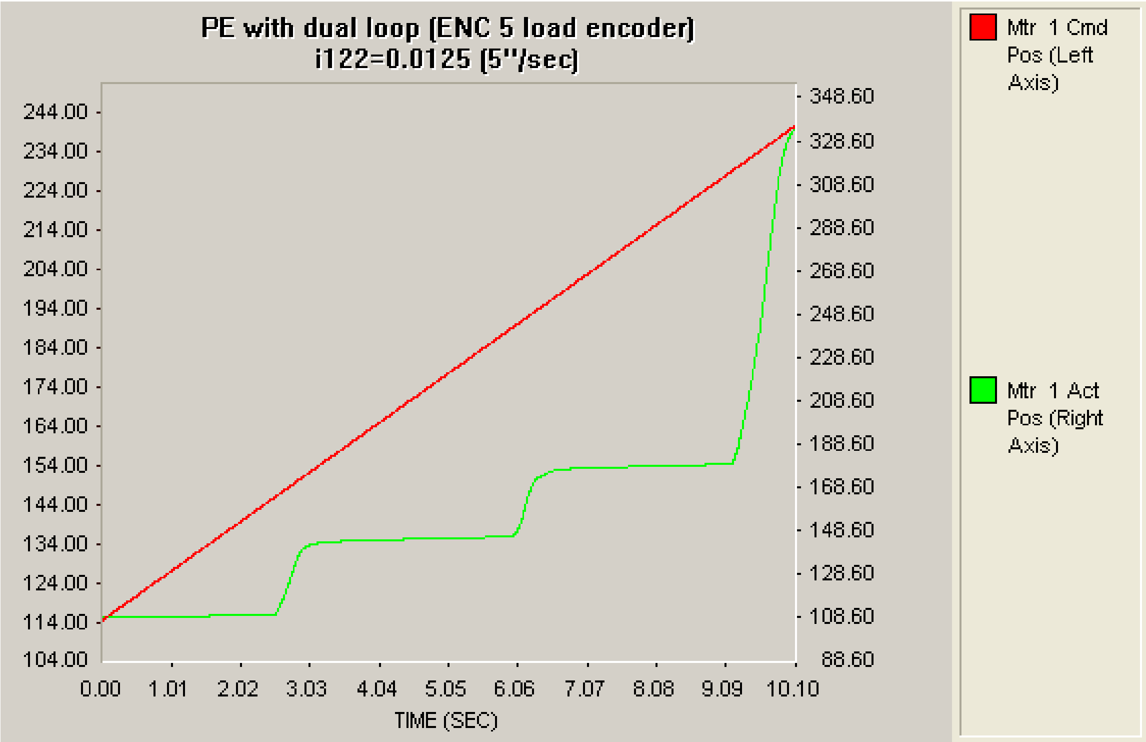


Constant speed





at min. tracking speed → stick-slip



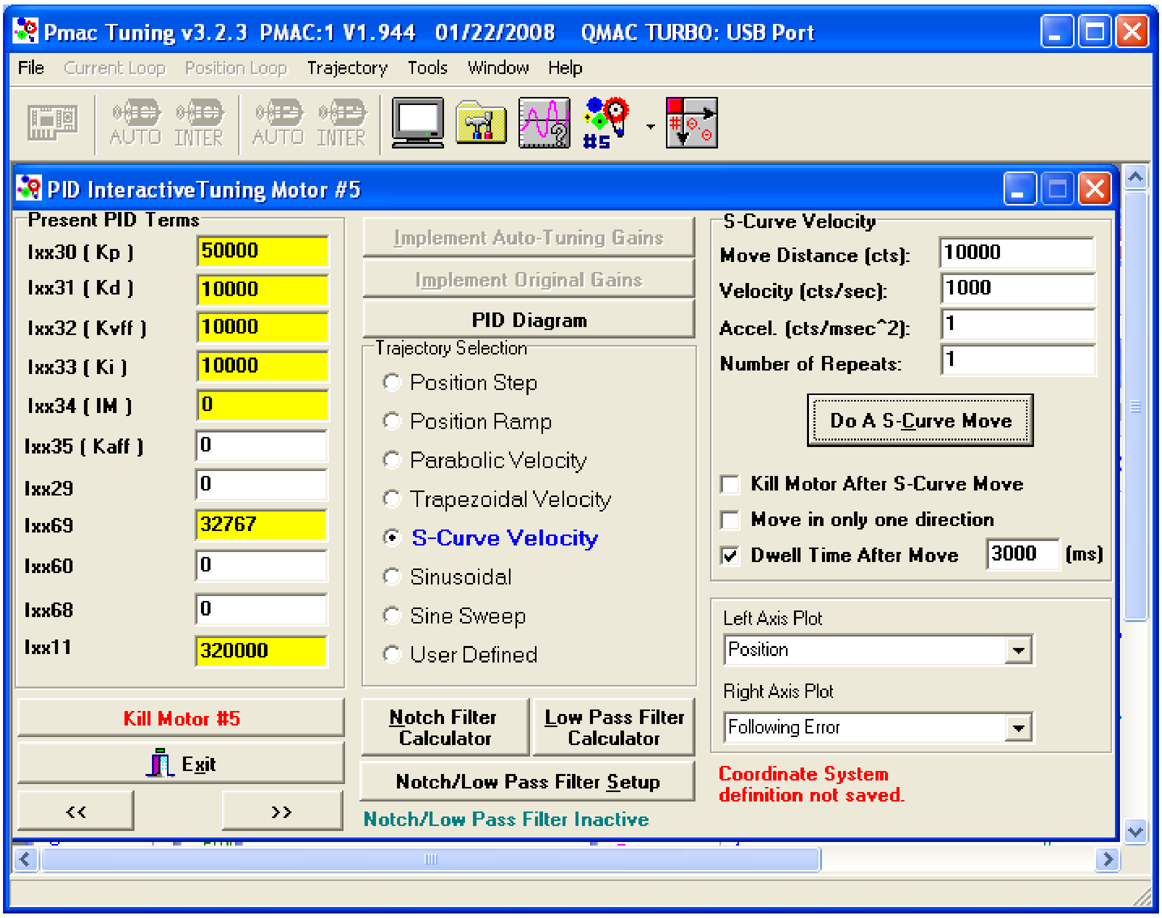
Test with backlash compensation algorithm

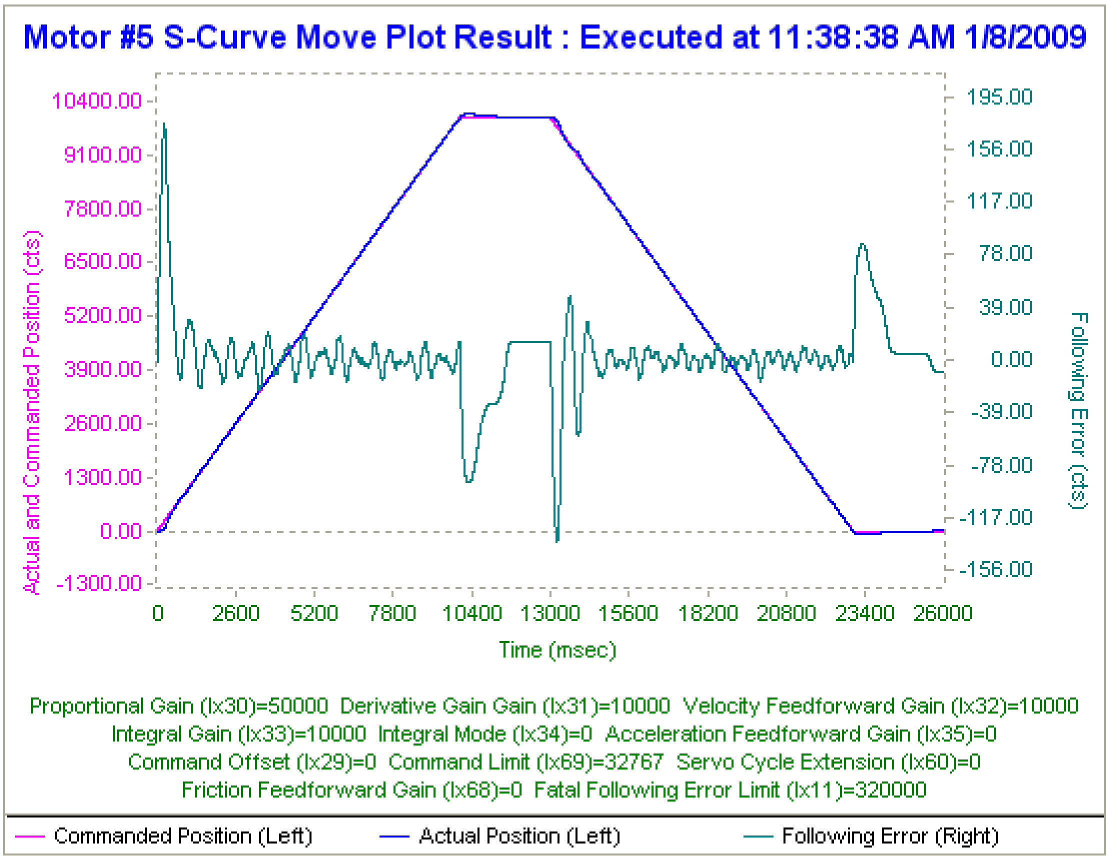
→ see preload13.pmc

Problem with “desired velocity 0” ???

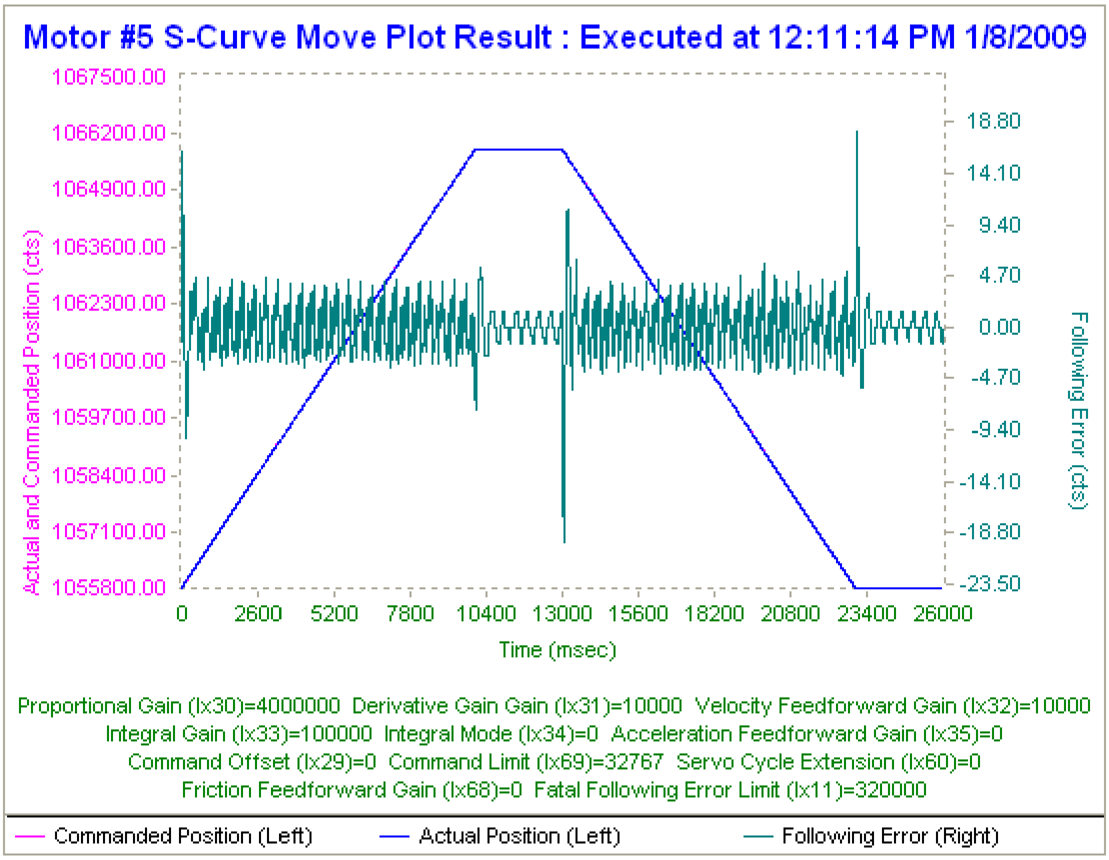
08.01.2009

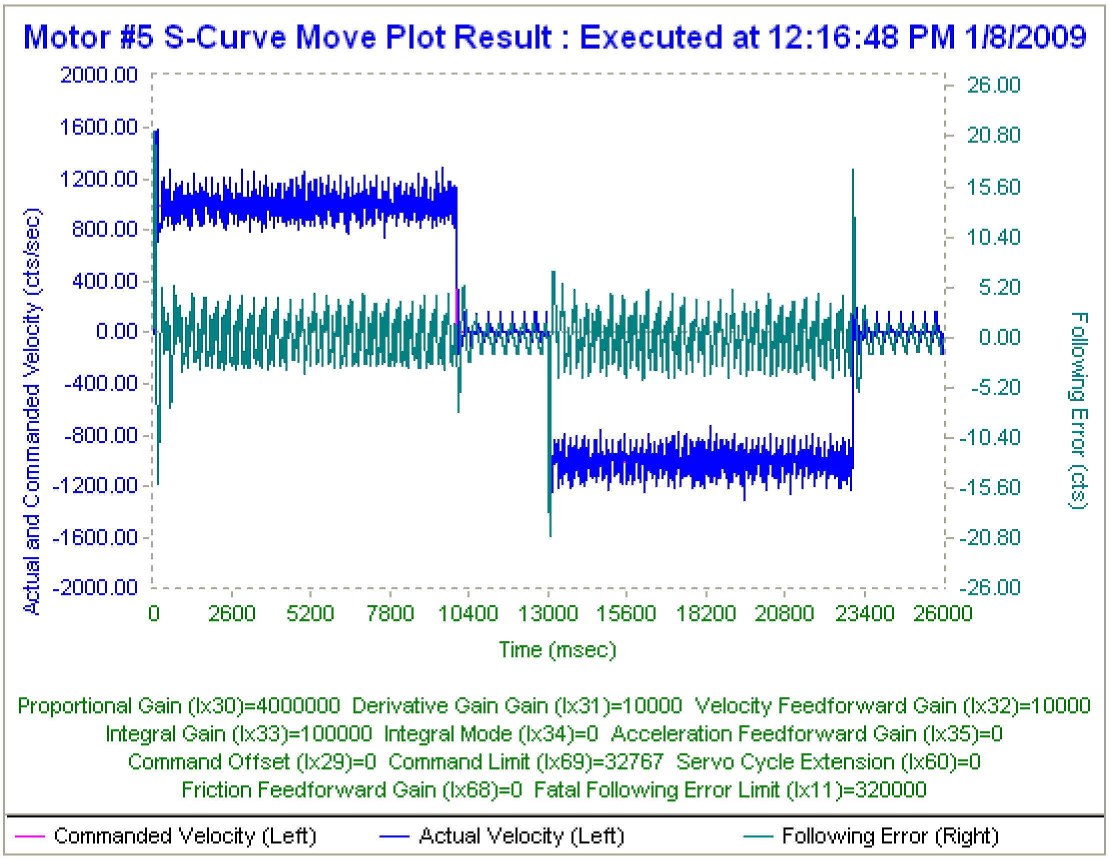
Backlash compensation now working properly



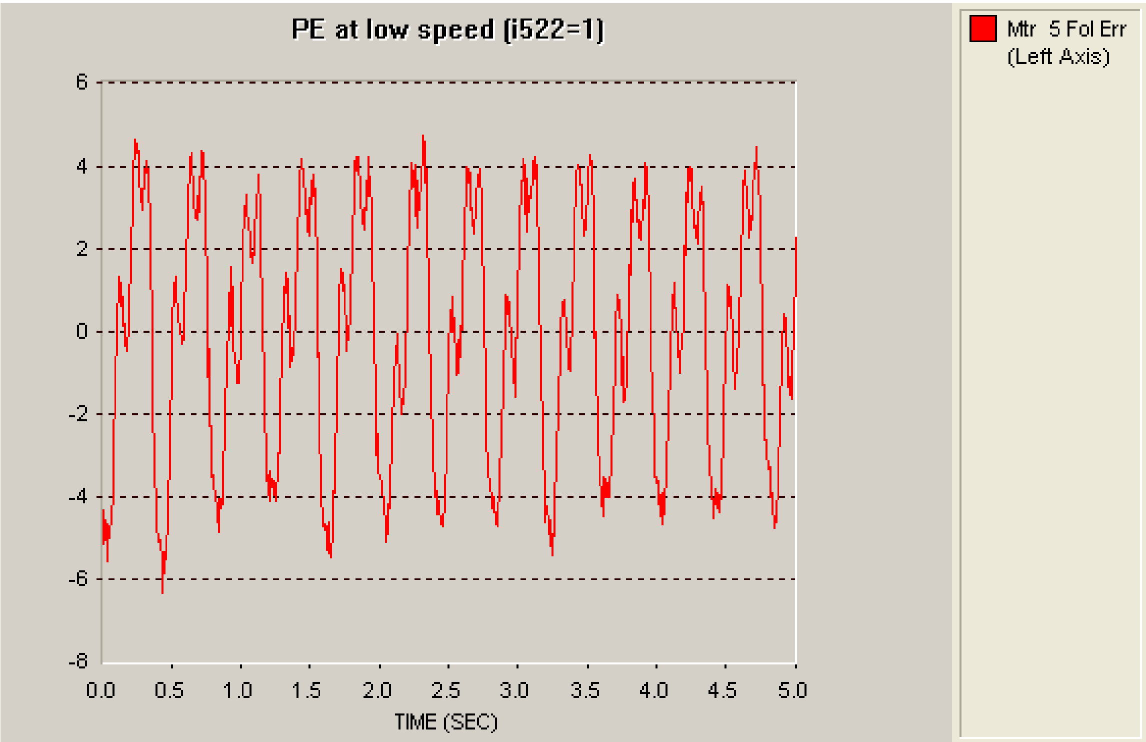


After new tuning



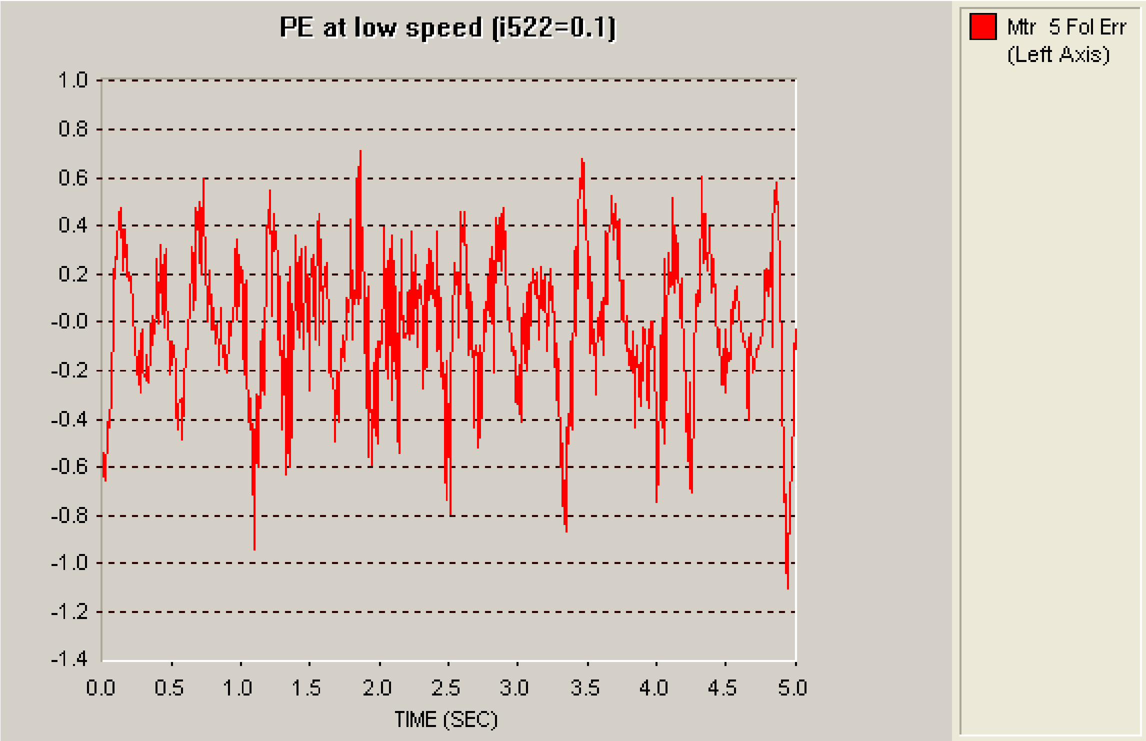


Motor #5 S-Curve Move Plot Result : Executed at 12:16:48 PM 1/8/2009
2000.00
1600.00
1200.00
800.00
400.00
0.00
-400.00
-800.00
-1200.00
-1600.00
-2000.00
Actual and Commanded Velocity (cts/sec)
26.00
20.80
15.60
10.40
5.20
0.00
-5.20
-10.40
-15.60
-20.80
-26.00
Following Error (cts)
0
2600
5200
7800
10400
13000
15600
18200
20800
23400
26000
Time (msec)
Proportional Gain (Ix30)=4000000 Derivative Gain Gain (Ix31)=10000 Velocity Feedforward Gain (Ix32)=10000
Integral Gain (Ix33)=100000 Integral Mode (Ix34)=0 Acceleration Feedforward Gain (Ix35)=0
Command Offset (Ix29)=0 Command Limit (Ix69)=32767 Servo Cycle Extension (Ix60)=0
Friction Feedforward Gain (Ix68)=0 Fatal Following Error Limit (Ix11)=320000
Commanded Velocity (Left)
Actual Velocity (Left)
Following Error (Right)

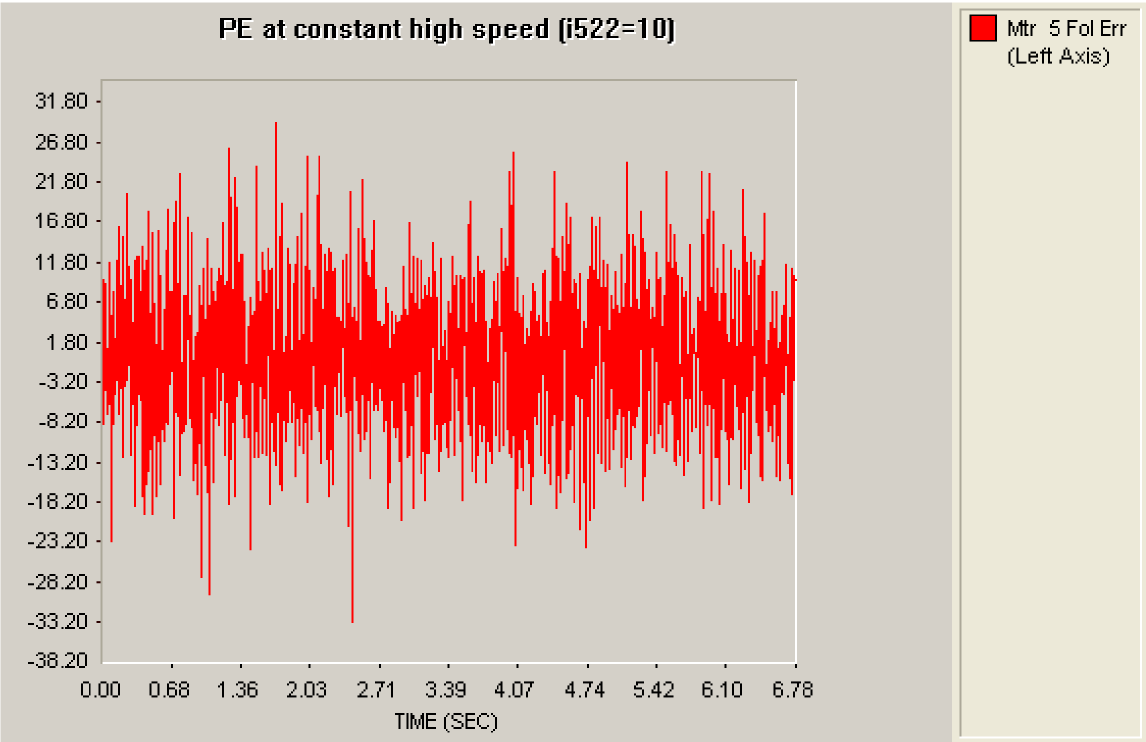


PE at low speed (i522=1)
6
4
2
0
-2
-4
-6
-8
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
TIME (SEC)
Mtr 5 Fol Err
(Left Axis)

Very low speed



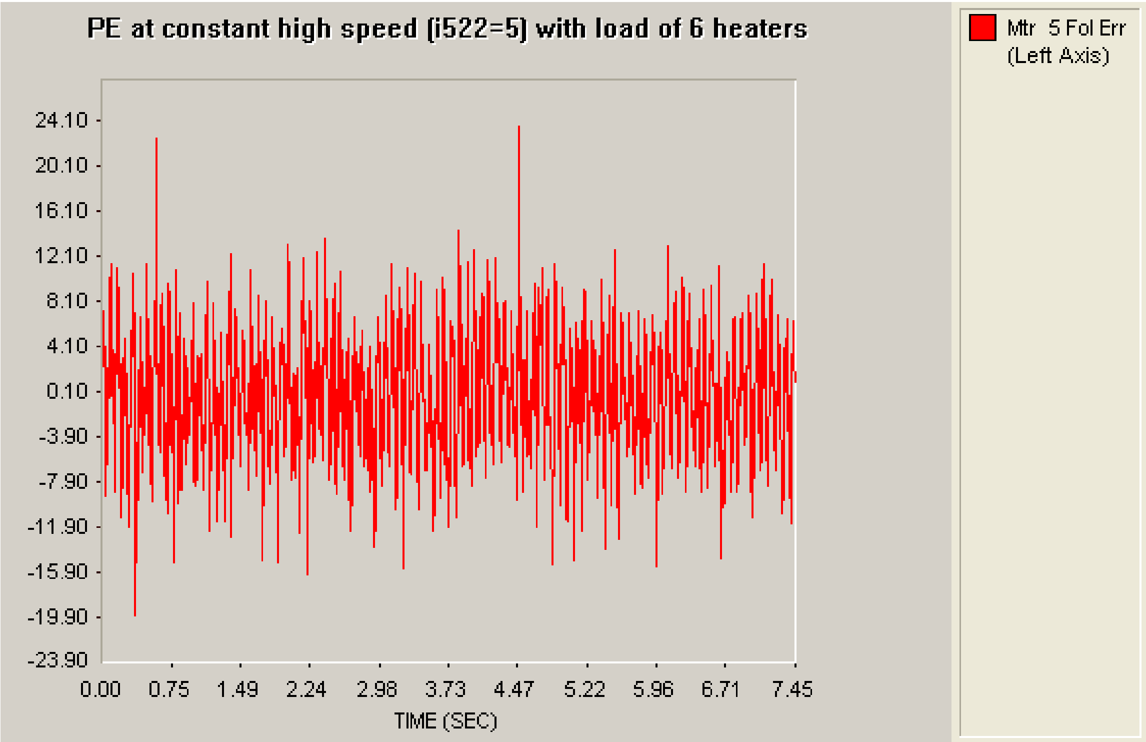
Following error with backlash compensation at high speed (i522=10)

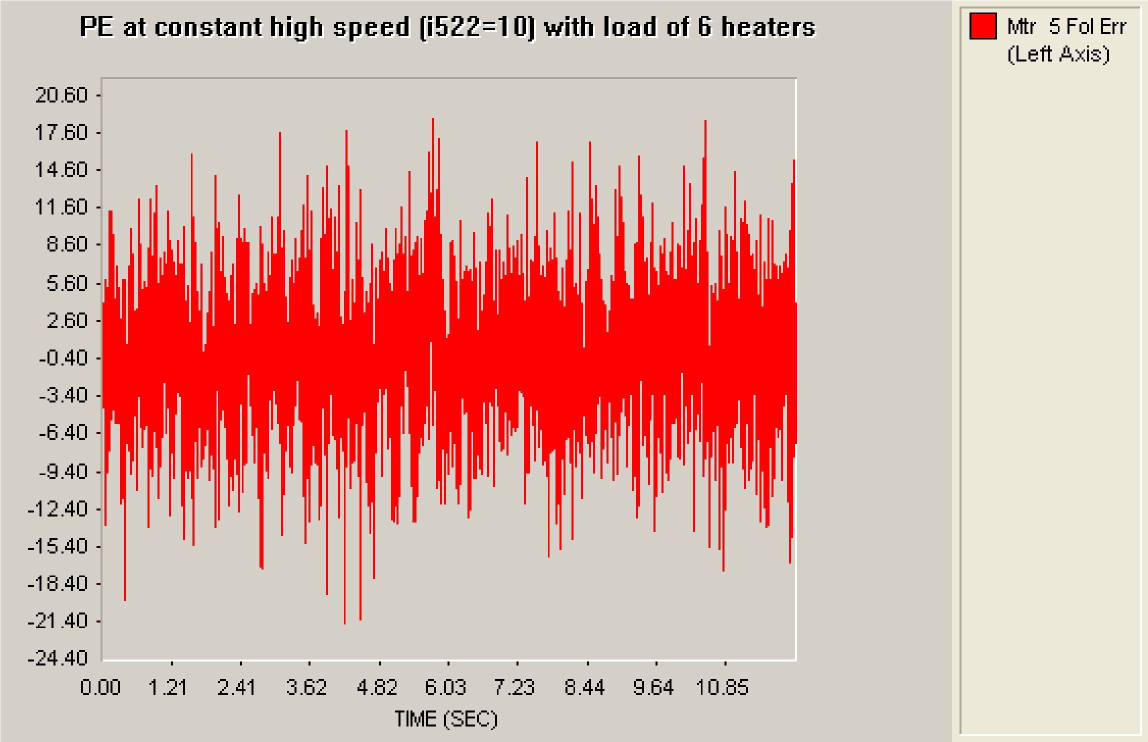


ad 3) Tests with big test set-up system and load (external DC motor)

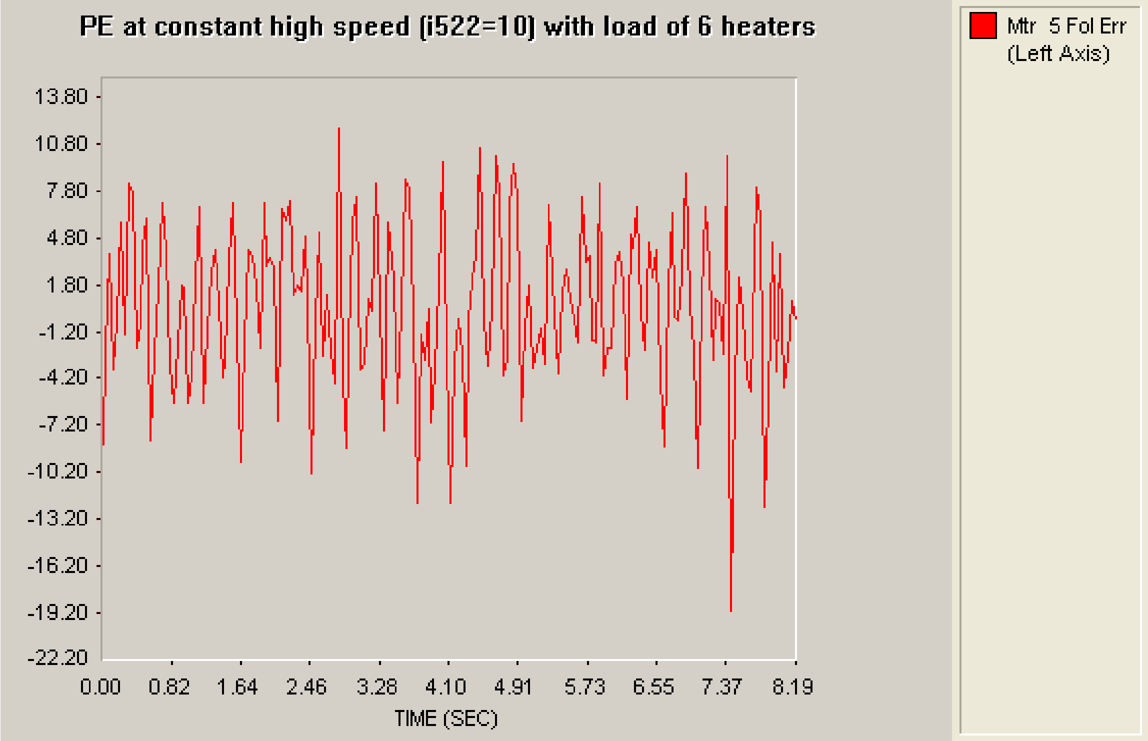
Load are 6 heaters in parallel (4.4 Ohms) and a DC motor with 59V/krpm and 0.5Nm/A, speed relation about: i522 = 5 → 500 rpm at the motor

I522	Speed at motor [rpm]	Measured voltage [V]	Measured current [A]	Torque at load motor [Nm]	Current at drive motor (SS value)
5	500	33	7.6	3.8	3
10	1000	66	22	11	5
15	1500	100	30	15	8

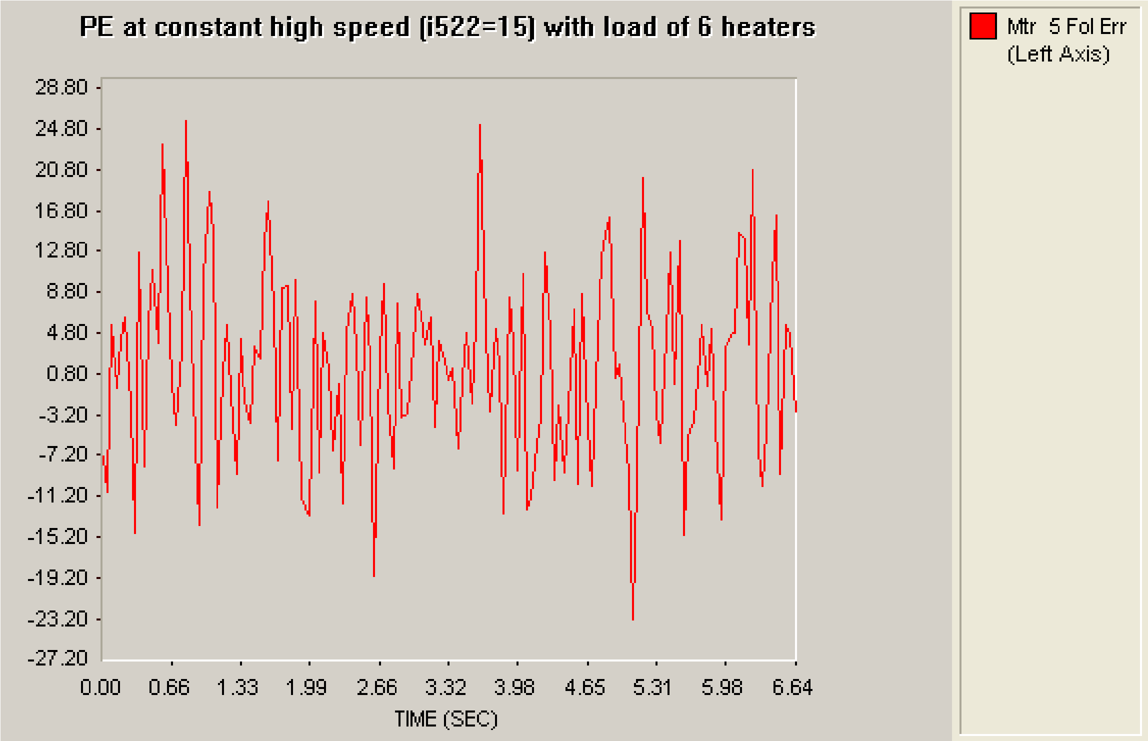




PE at constant high speed (i522=10) with load of 6 heaters
20.60
17.60
14.60
11.60
8.60
5.60
2.60
-0.40
-3.40
-6.40
-9.40
-12.40
-15.40
-18.40
-21.40
-24.40
0.00
1.21
2.41
3.62
4.82
6.03
7.23
8.44
9.64
10.85
TIME (SEC)
Mtr 5 Fol Err
(Left Axis)



PE at constant high speed (i522=10) with load of 6 heaters
13.80
10.80
7.80
4.80
1.80
-1.20
-4.20
-7.20
-10.20
-13.20
-16.20
-19.20
-22.20
0.00
0.82
1.64
2.46
3.28
4.10
4.91
5.73
6.55
7.37
8.19
TIME (SEC)
Mtr 5 Fol Err
(Left Axis)



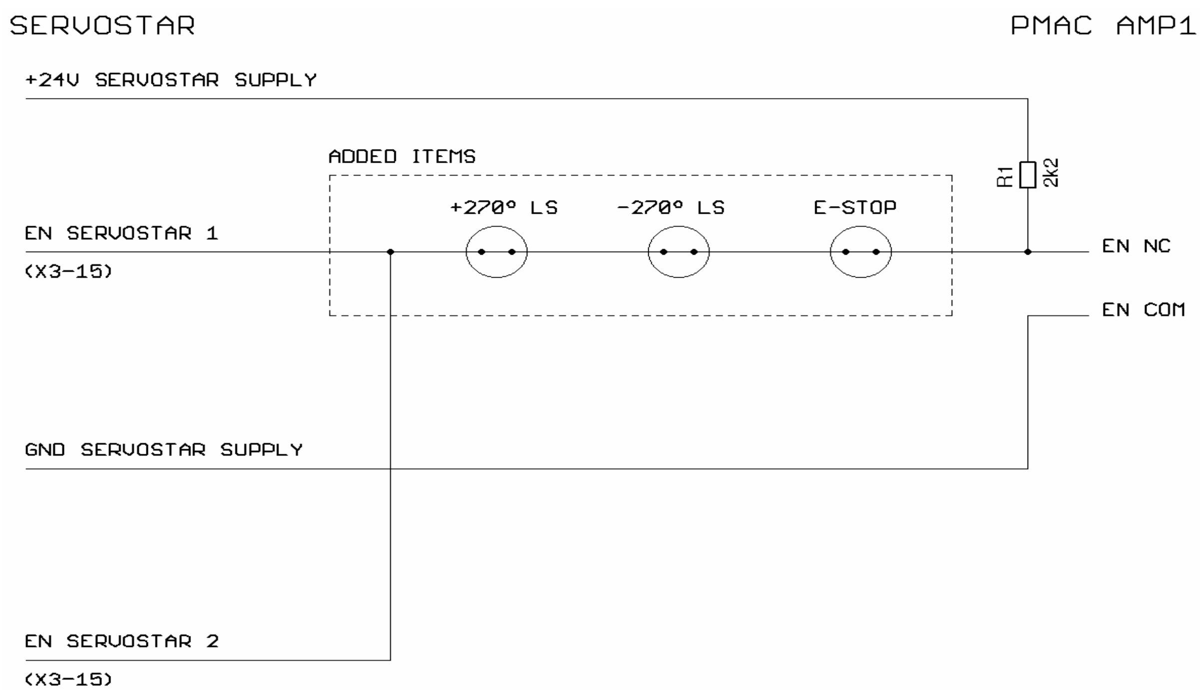
Then gear wheel broke

Report of Tests on C04 Antenna:-

Task 1 at AZ (16.04.2009)

Re-wiring the interlock system: The “+270°-limit” switch and the “+270°-limit” as well as the “safety switch” should go into the enable line of the Servostar (→ this will engage the motor brake incase of error as well!). Both enable lines for the Servo star must be connected in parallel because there is only one contact per limit switch (in case that the enable of the second Servostar is not wanted it needs to be disabled with Servostar software disable).

Circuitry of enable part (interconnection BLC and Servostar)



Final limit will not be used (as in the current system)

Cable wrap switch is used to de-power the whole system (as used in the current system)

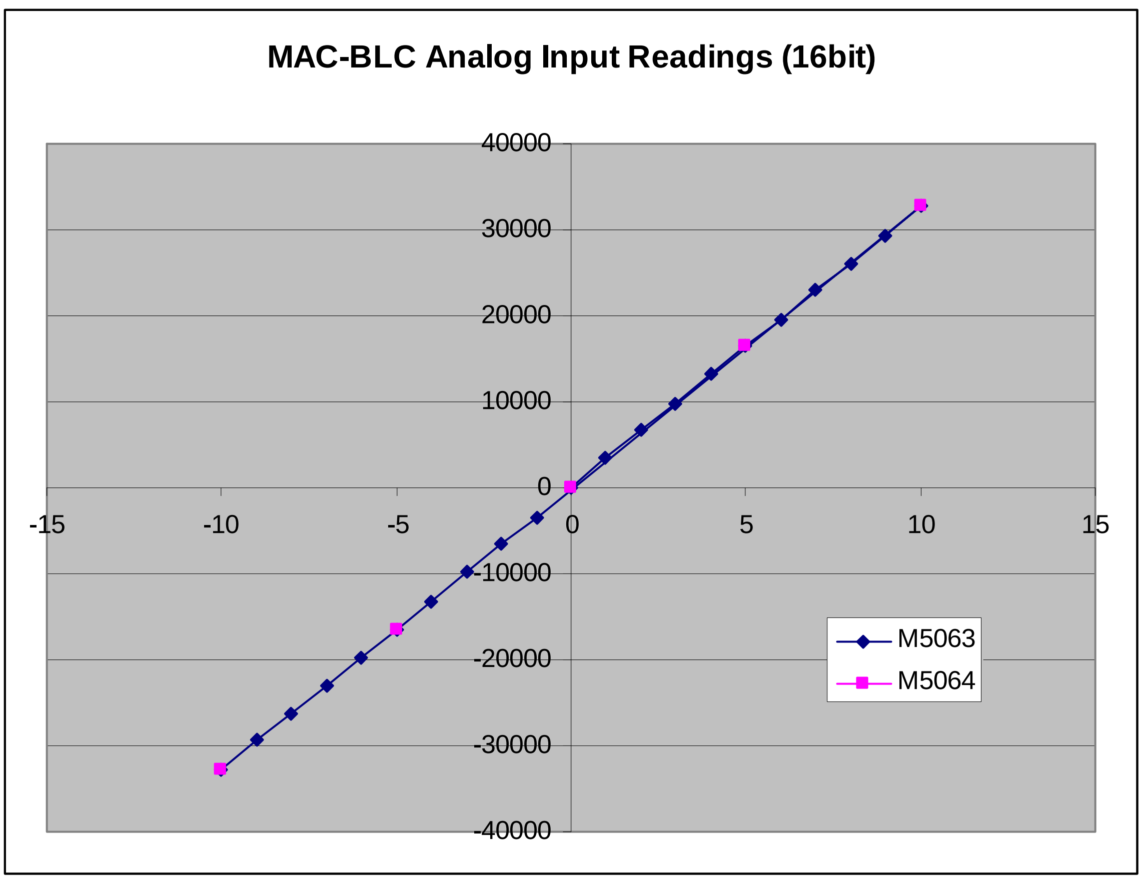
Because of the not detecting the 0° switch the travel is only possible between about -60° and +60° (on the way to -270° the axis will pass the +270° first!)

→ Check of the limit switches (-270° and +270°) successfully (motors stopped moving)
→ Check of the E-Stop Switch successfully (motors stopped moving)
→ Check of the Cable Wrap Switch successfully (whole rack was de-powered)

Check of analog reading with BLC ADC:
→ When leaving the default strobe word (I7106 = $FFFFFE) and using the plc30 (see “Measurements ADC at BLC.pmc”) for correction and using the differential input (+Input = pin 2 and -Input = pin 6) we have a linear behavior from -10V to +10V
→ Using the strobe word (I7106 = $1FFFFF) and the definition M5063->Y:$78115,8,16,s we are getting the correct readings:

Input Voltage [V]	Reading at M5063		Reading at M5064	
	Pos. Input	Neg. Input	Pos. Input	Neg. Input
0	-18	-18	-33	-33
1	3393	-3435		
2	6668	-6591		
3	9741	-9854		
4	13282	-13268		
5	16536	-16432	16496	-16496
6	19612	-19673		
7	22965	-22999		
8	26149	-26244		
9	29415	-29332		
10	32767	-32690	32767	-32678

(see file :” ADC Readings.xls”)



Check of the current loop parameters in Servostar:
Kt of the BL motor: 3.1Nm/A
Max. continuous torque (limited by the gearbox): 20Nm
Peak torque: 2 * continuous torque

→ Settings for Servostar:
ICONT: 6.5A
IPEAK: 13A

→ see file: “16042009 Adapted Current Levels”

Further tries were made for setting up a long RS232 cable in order to control the BLC from the tent or the platform → not successful, next day we will try with Ethernet connection

Task 2 at AZ

→ Both motor were driving the antenna in closed loop mode (position loop closed by resolver at motor)
→ Current about 1A (measured with Servostar, changing between 0.5 – 1.5A), when running 1000 rpm at motor → 16.67 rpm → 0.32°/s at load

17.04.2009

Truptis calculations for position feed back and velocity "position" feed back scales at antenna side:

Resolver (motor) count after one revolution of big gear = 4096*4*1488*12.6 = 307180339.2 (to be checked because of different possible gear factor!)

Load encoder reading (considering 17 bit absolute and interpolator) 8192*400 = 3276800

motor to load reading ratio = 307180339.2/3276800 = **97.744**

if position feedback scaling =96
then velocity "position" feedback scale is =1.018 = 1 approx.

Check the communication to BLC with Ethernet
BLC gets the IP address: 192.6.94.5
Laptop: IP address: 192.6.94.1 (no DHCP) → working!

Now we are working on the first platform, in order the have a better feeling on the system behavior.

Checking the resolution of the system and the scaling:
→ Load encoder opposite to the resolvers → change #5 feedback orientation: I7110=7

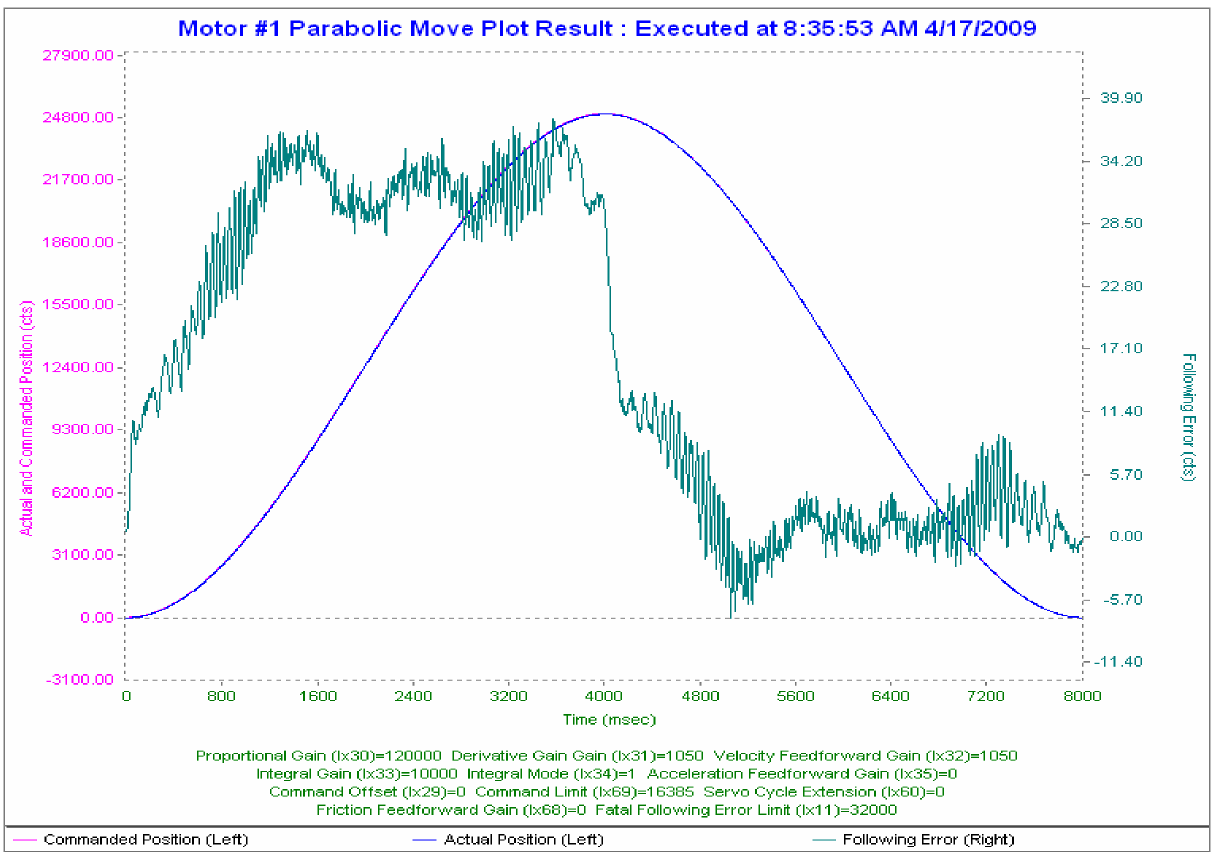
#1 moved for 1000000 cnts and #5 moved 10626 counts → calculated: 10230,8
#1 moved for 3000000 cnts and #5 moved 31890 counts → calculated: 30692,4
#1 moved for 7384808 cnts and #5 moved 78510 counts → calculated: 75552,5

Currently used parameter (without tuning):

I130=120000	;Motor 1 PID Proportional Gain
I131=1050	;Motor 1 PID Derivative Gain
I132=1050	;Motor 1 PID Velocity Feed Forward Gain
I133=10000	;Motor 1 PID Integral Gain
I134=1	;Motor 1 PID Integration Mode
I135=0	;Motor 1 PID Acceleration Feed Forward Gain
I136=0	;Motor 1 PID Notch Filter Coefficient N1
I137=0	;Motor 1 PID Notch Filter Coefficient N2
I138=0	;Motor 1 PID Notch Filter Coefficient D1
I139=0	;Motor 1 PID Notch Filter Coefficient D2

Checking the backlash (#1j/ / #2o-3 / #1hmz #2hmz #5hmh / #2o3)
#2 = 118222 cnts
#5 = 495 cnts → 0.055° = 200arcsec

Check Parabolic move with: motor1 with resolver1, #2 just #2o-3:



(see file: “parabolic vel motor1 resolver1 axis2 o-3.TXT”)

Checking the necessary torque offset:

#1j/
#2o2 / #2hmz / #5hmz
#2 = -106741 #5 = -469 (cnts)
#2o-2
#2 = -295 #5 = -3 (cnts)
#2o-3
#2 = -114636 #5 = -485 (cnts)
#2o3
#2 = -6831 #5 = -11 (cnts)

→ we start with torque bias of 2% → Mx68=330 (checked at Servostar → about 0.4A)

AZ LOAD ENCODER SCALING: 2.5 counts / 1'' → 0.4" per 1 count

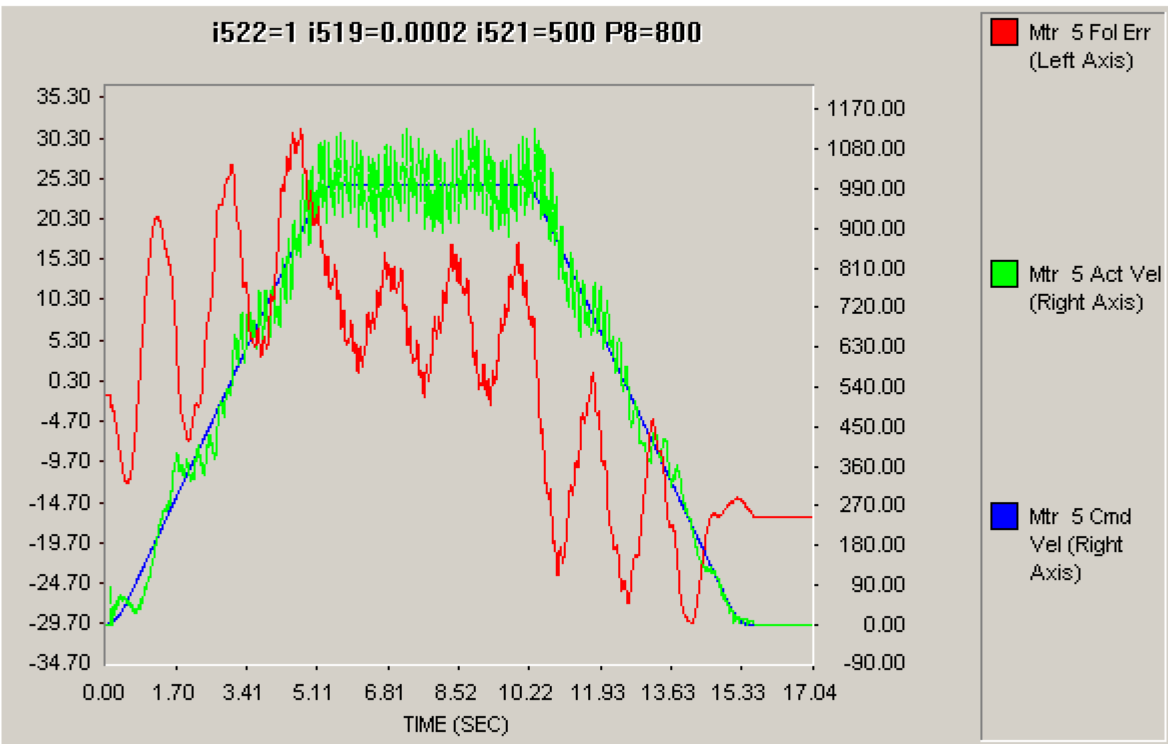
After some tuning:

I530=1000	;Motor 5 PID Proportional Gain
I531=1000	;Motor 5 PID Derivative Gain
I532=0	;Motor 5 PID Velocity Feed Forward Gain
I533=0	;Motor 5 PID Integral Gain
I534=0	;Motor 5 PID Integration Mode
I535=0	;Motor 5 PID Acceleration Feed Forward Gain
I536=0	;Motor 5 PID Notch Filter Coefficient N1
I537=0	;Motor 5 PID Notch Filter Coefficient N2
I538=-1.6273140907	;Motor 5 PID Notch Filter Coefficient D1
I539=0.6799309254	;Motor 5 PID Notch Filter Coefficient D2

(Low pass 2. order at 100Hz)

I568=400	; Friction FF

Now torque offset increased: P8=800

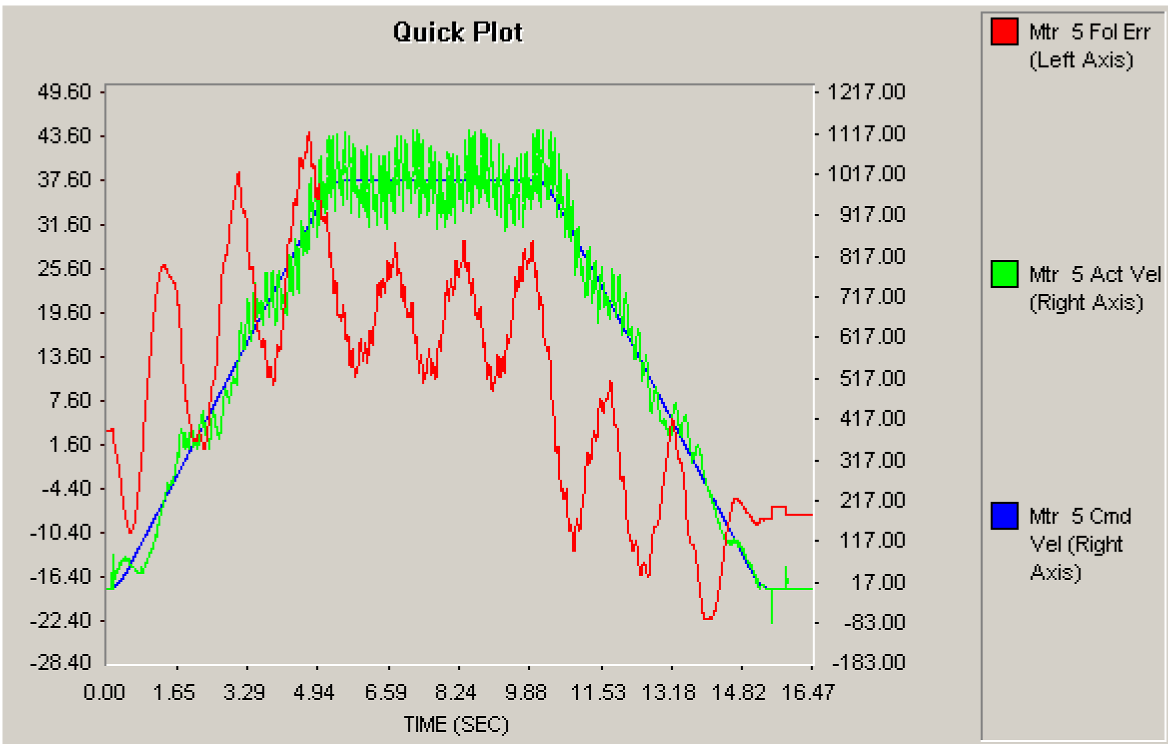


(see file “i522=1 i519=0.0002 i521=500 P8=800.txt”)

→ Accuracy already quite good, so we try to decrease the hearable noise of the system.

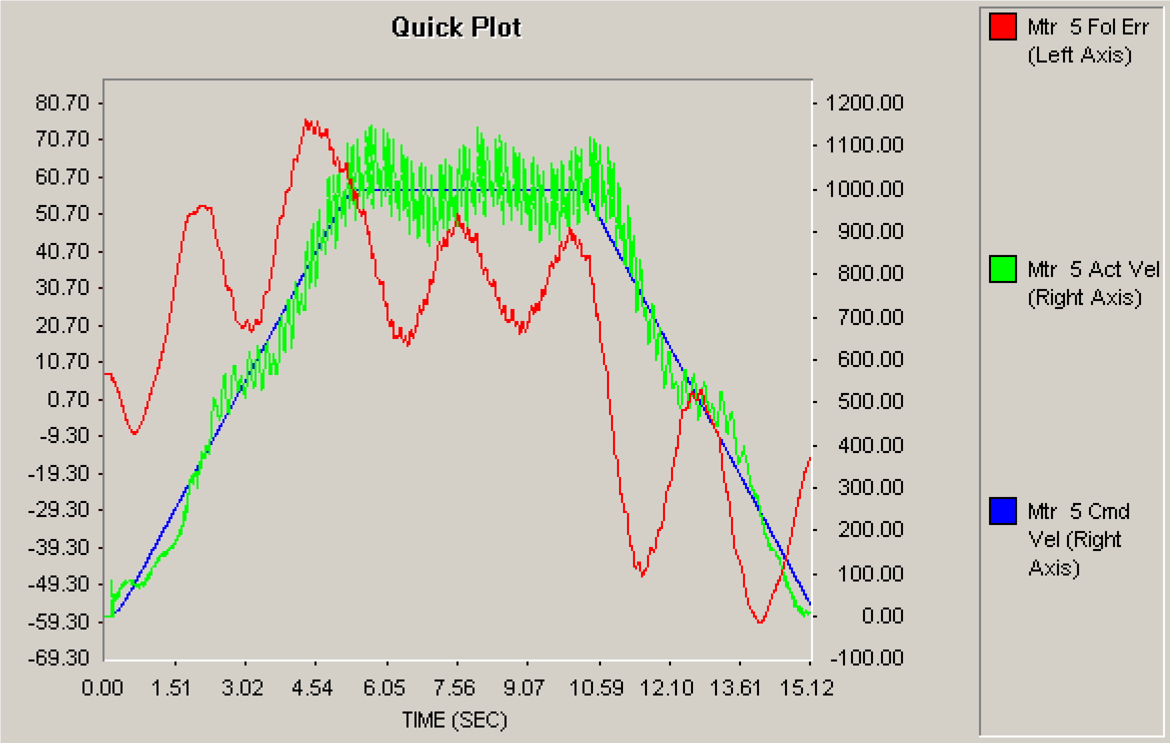
Change the low pass filter to 50Hz

```
I530=302              ;Motor 5 PID Proportional Gain
I531=1000             ;Motor 5 PID Derivative Gain
I532=0                ;Motor 5 PID Velocity Feed Forward Gain
I533=0                ;Motor 5 PID Integral Gain
I534=0                ;Motor 5 PID Integration Mode
I535=0                ;Motor 5 PID Acceleration Feed Forward Gain
I536=0                ;Motor 5 PID Notch Filter Coefficient N1
I537=0                ;Motor 5 PID Notch Filter Coefficient N2
I538=-1.8064441681    ;Motor 5 PID Notch Filter Coefficient D1
I539=0.8223538399     ;Motor 5 PID Notch Filter Coefficient D2
```



→ not successful!

→ taking out the low pass and setting it to 1000Hz:
I530=11489 ;Motor 5 PID Proportional Gain
I531=1000 ;Motor 5 PID Derivative Gain
I532=0 ;Motor 5 PID Velocity Feed Forward Gain
I533=0 ;Motor 5 PID Integral Gain
I534=0 ;Motor 5 PID Integration Mode
I535=0 ;Motor 5 PID Acceleration Feed Forward Gain
I536=0 ;Motor 5 PID Notch Filter Coefficient N1
I537=0 ;Motor 5 PID Notch Filter Coefficient N2
I538=-0.4682340622 ;Motor 5 PID Notch Filter Coefficient D1
I539=0.07891392708 ;Motor 5 PID Notch Filter Coefficient D2



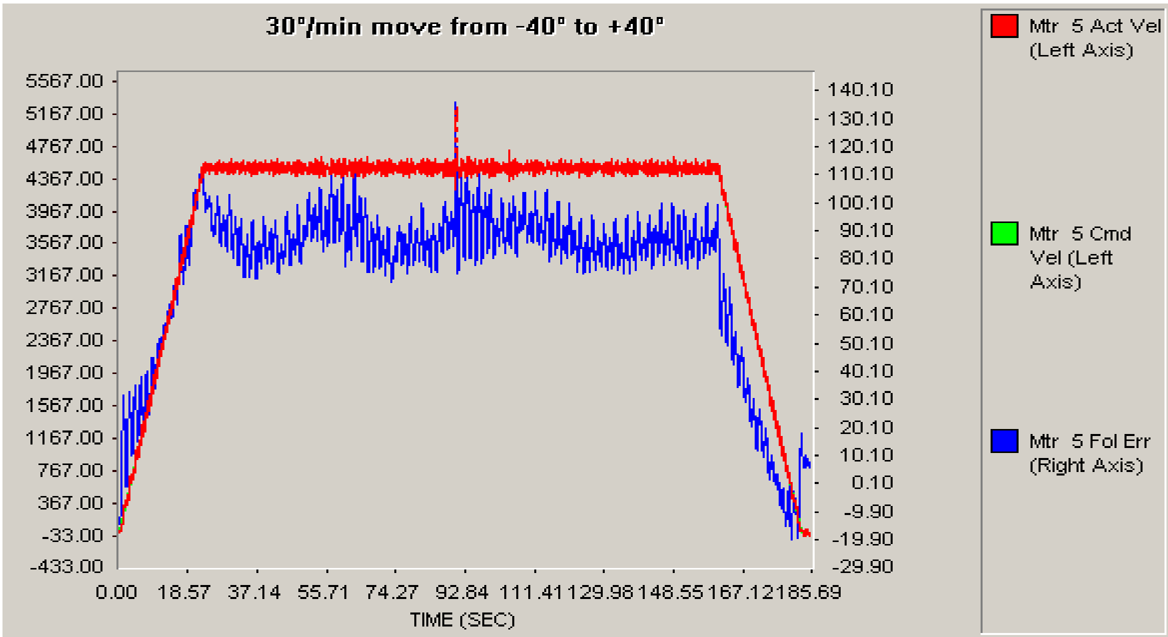
Visual inspection on top of the AZ platform:
→ Hearable noise is coming from the thunderstorm "slip" ring elements (they are vibrating because of loose parts), therefore the noise can not be measured at the motors neither the load encoder, checking the behavior at the motor side it is ok! (→ see picture below)

Now checking higher and lower speeds:
High speed is 30°/min → i522 = 4.551 counts/msec

→ running with 30°/min with j+ → antenna stopped at about 71° because of LS
→ overwriting the LS at Servostar
→ running with 30°/min with j- → antenna stopped at about -52.5° (so probably 0° position was not exact)

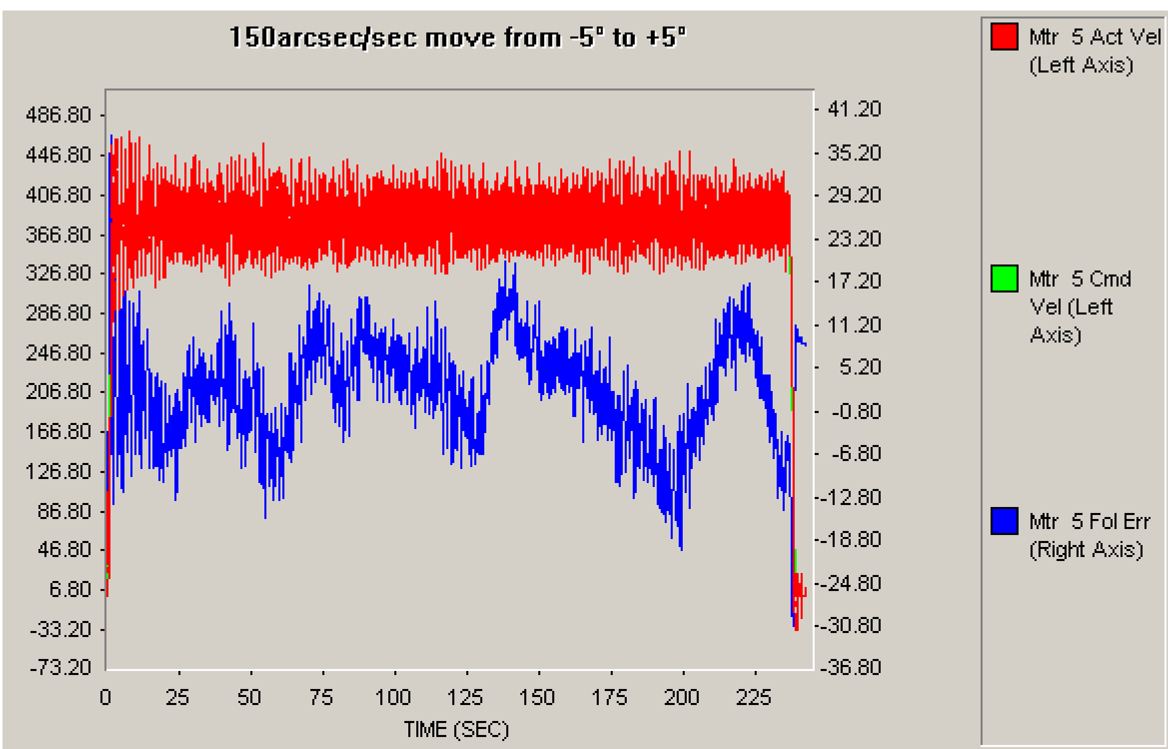
→ running with 30°/min is ok

Now doing data gathering:
30°/min move from -40° = -360000 counts to +40° = 360000 → about 3min → data gathering period = 50



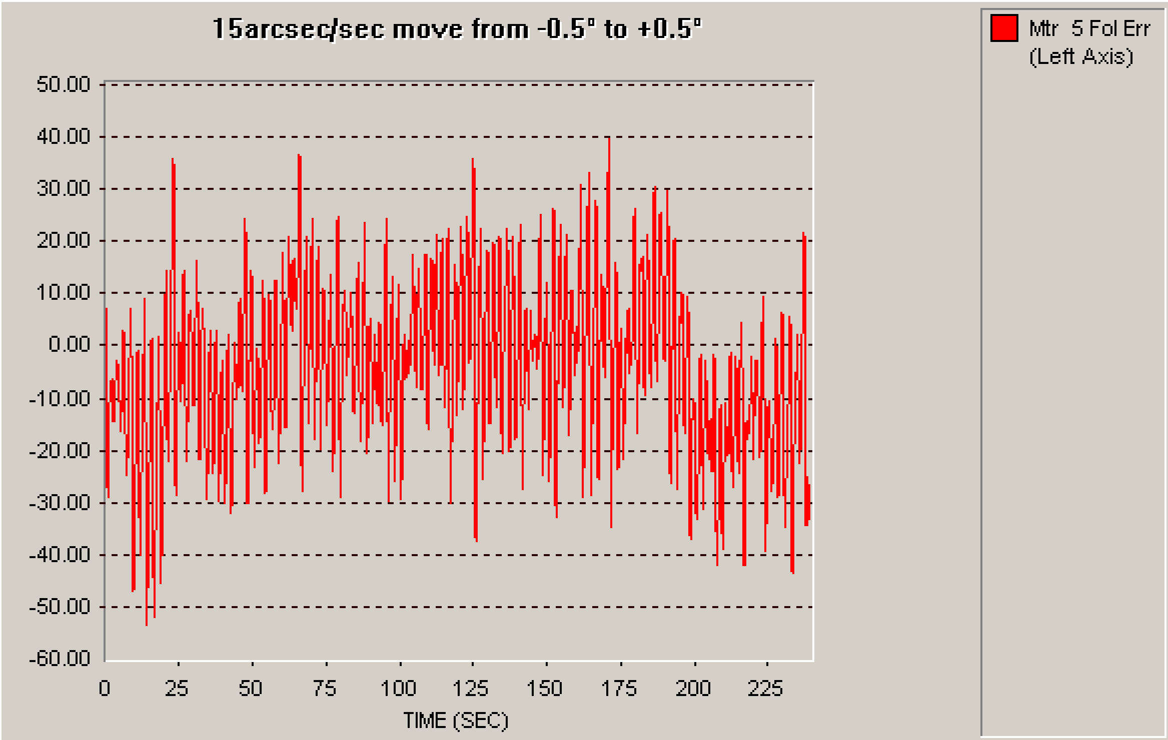
(see file “30° per min move from -40° to +40°.txt”)

150arcsec/sev move from -5° = -45000 counts to +5° = 45000 → about 4min → data gathering period = 50 / i522 = 0.38



(see file “150 arcsec per sec move from -5° to +5°.txt”)

15arcsec/sev move from -0.5° = -4500 counts to +0.5° = 4500 → about 4min →
data gathering period = 50 / i522 = 0.038



(see file "15arcsec per sec move from -0.5° to +0.5°.txt")

Short discussion with N.V. Nagarathnam concerning clock issues:
Timing accuracy for 3sec over about 7 hours → to be checked!

Task 3 at AZ 18.04.09

Include "Tp2mvar.pmc" in the program preload13_2.pmc
tested after "$$$***" → ok

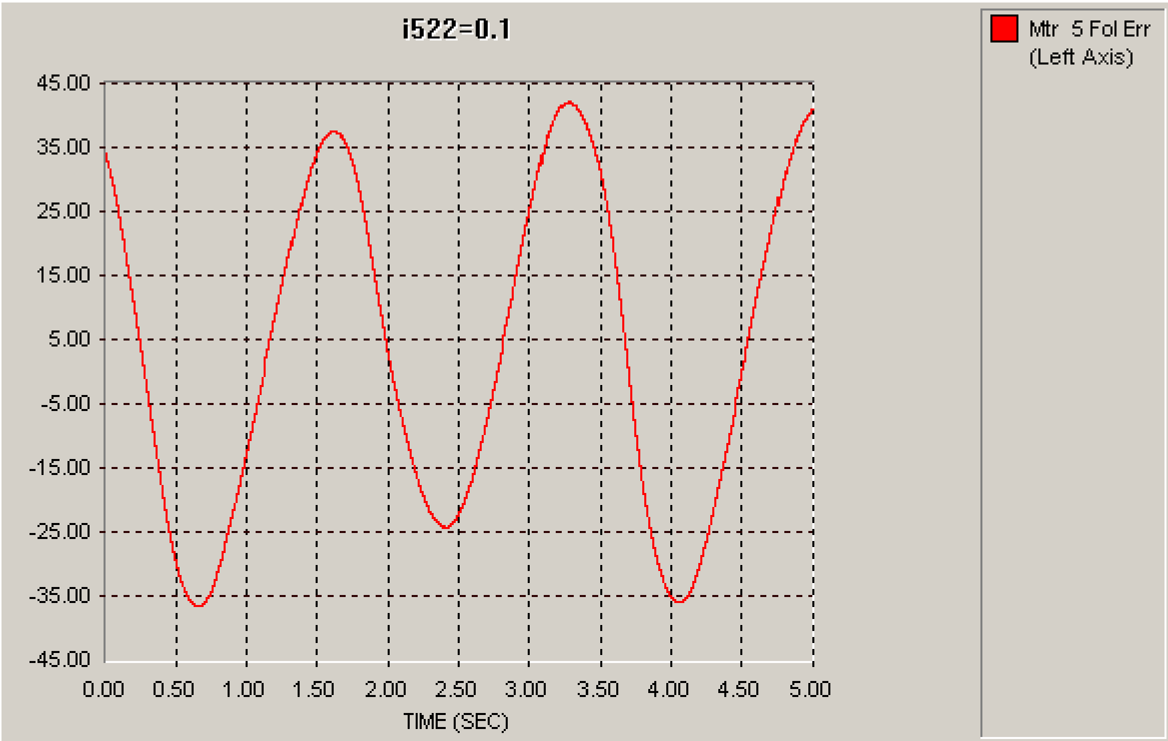
Check with preload14_plc0.pmc (i.e. the velocity loop calculation are done in the plc0 together with the calculations of the backlash algorithm)

Decreasing the PMAC max. DAC to 5000

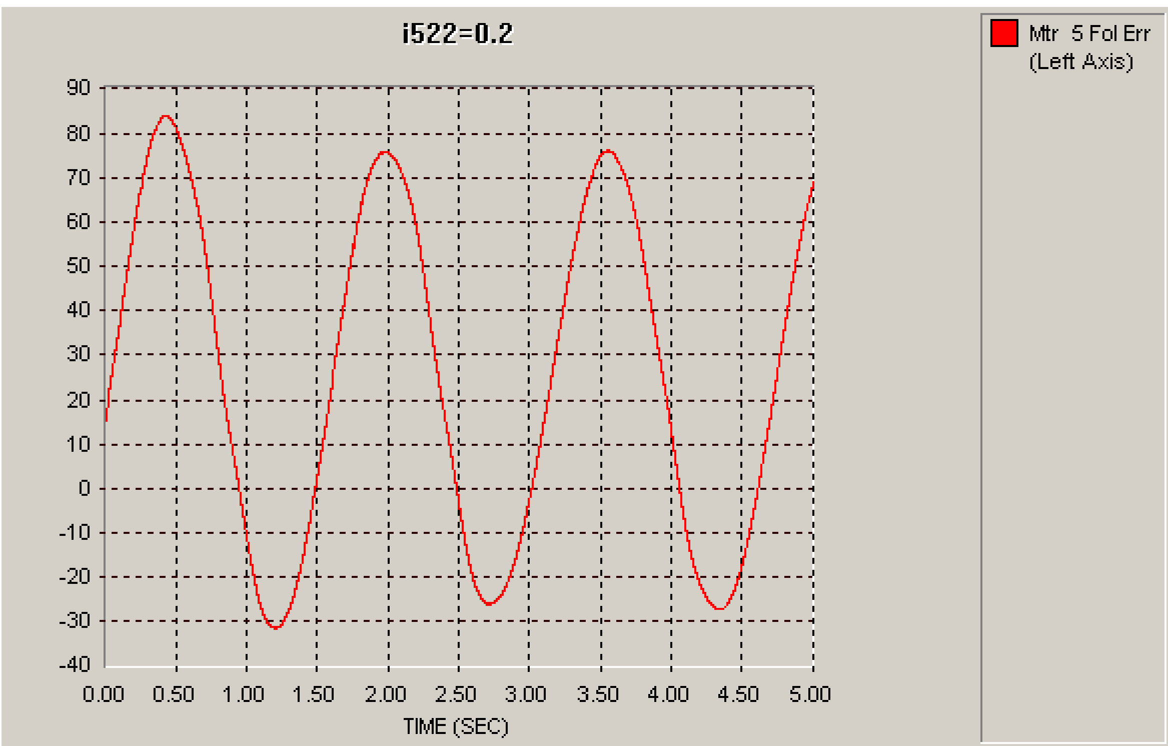
Checking the scaling of the variables:
Running with i129=550 in order to run the axis in open loop
Speed at motor 1 and 2 is about 130000cnts/s
P13 is -364000

After some tests this way of implementing the velocity loop is not considered further. We are making the velocity loop by standard PMAC functions.

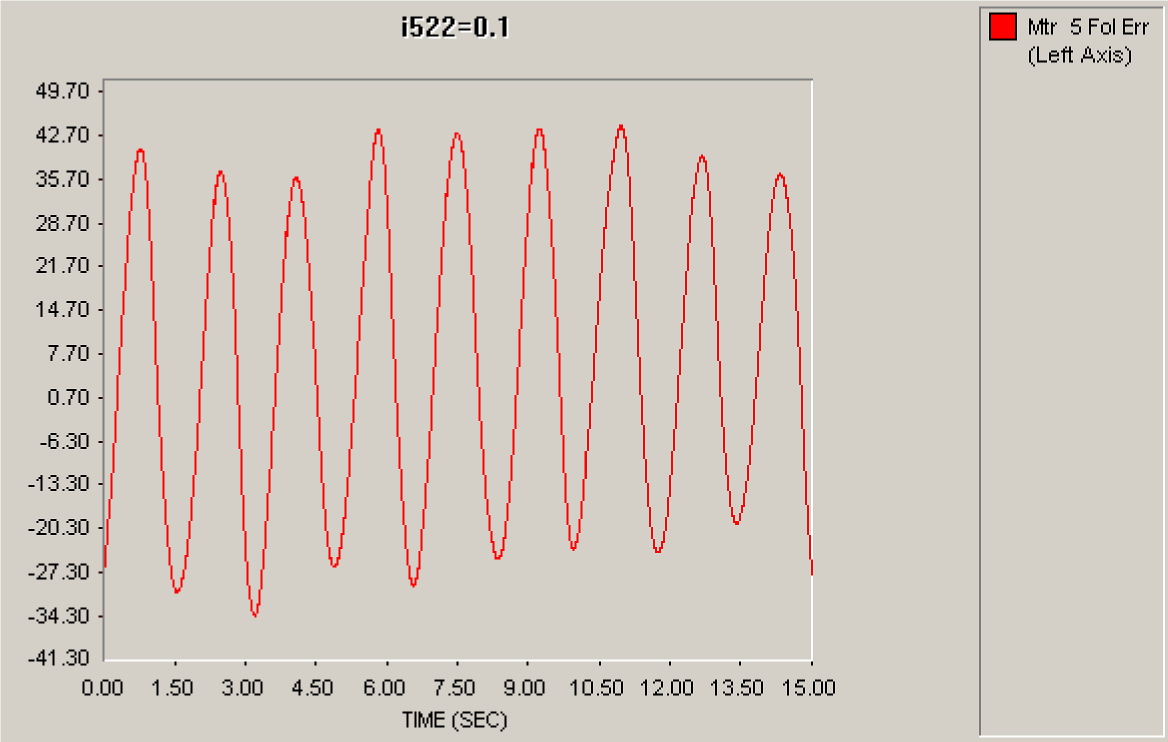
20.04.09:
Short check with full PMAC Position Loop (position loop with load encoder closed by PMAC, backlash compensation is active). Speed i522=0.1 means 40”/s; i522=0.2 means 80”/s; following error is in counts and scaled by 0.4” per count.



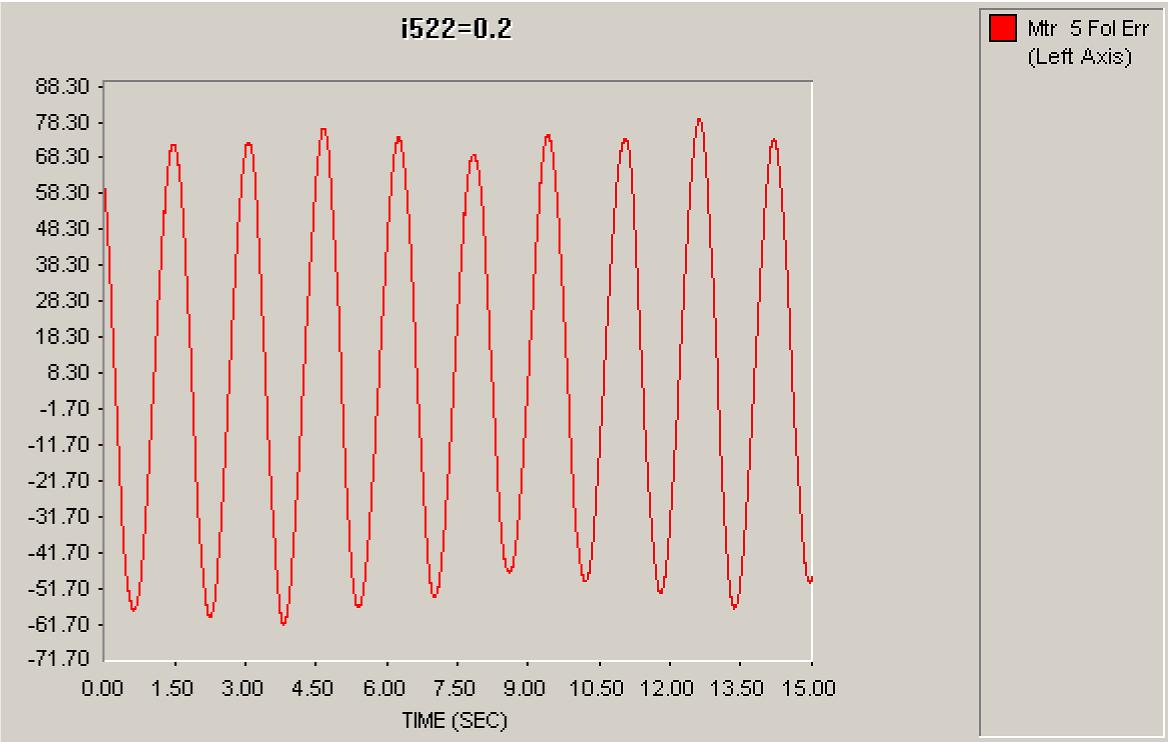
→ sinusoidal error with amplitude of +/- 14”



→ sinusoidal error with amplitude of +/- 24”



→ sinusoidal error with amplitude of +/- 15” (measured over a long time period)

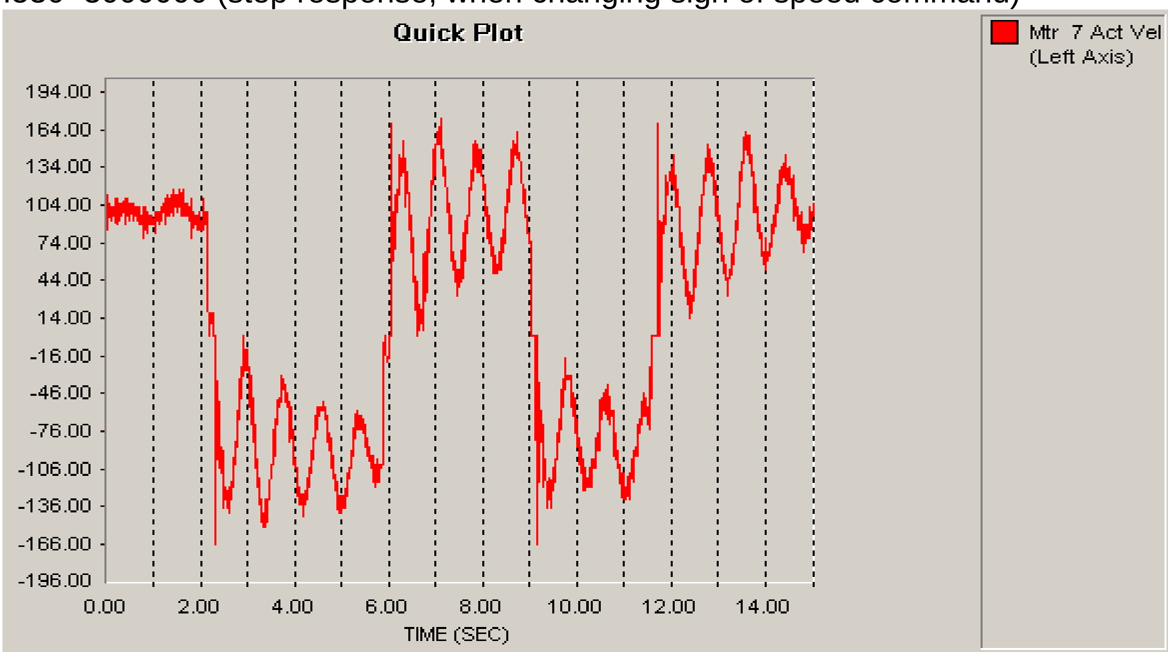


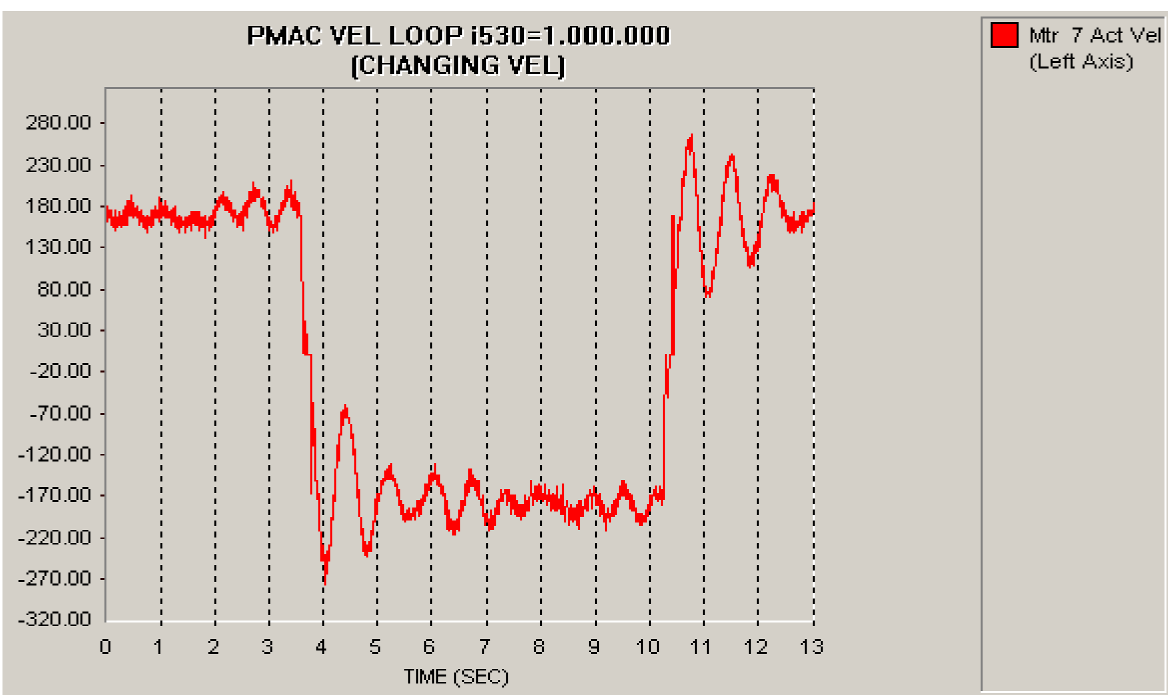
→ sinusoidal error with amplitude of +/- 26” (measured over a long time period)

Reason for this sinusoidal error should be investigated!

PMAC Velocity Loop
Now the standard PMAC PID is used to make the velocity loop and reading the analog input as the velocity command → see file “preload14_PMAC.pmc”
Implementation: position feedback needs to be zero (set to inactive axis)
Kd (ix31) becomes the scaling factor for vel feedback (→128)
Following error needs to be disabled
Analog input is gives scaled to hand-wheel register (M567)

I530=3000000 (step response, when changing sign of speed command)





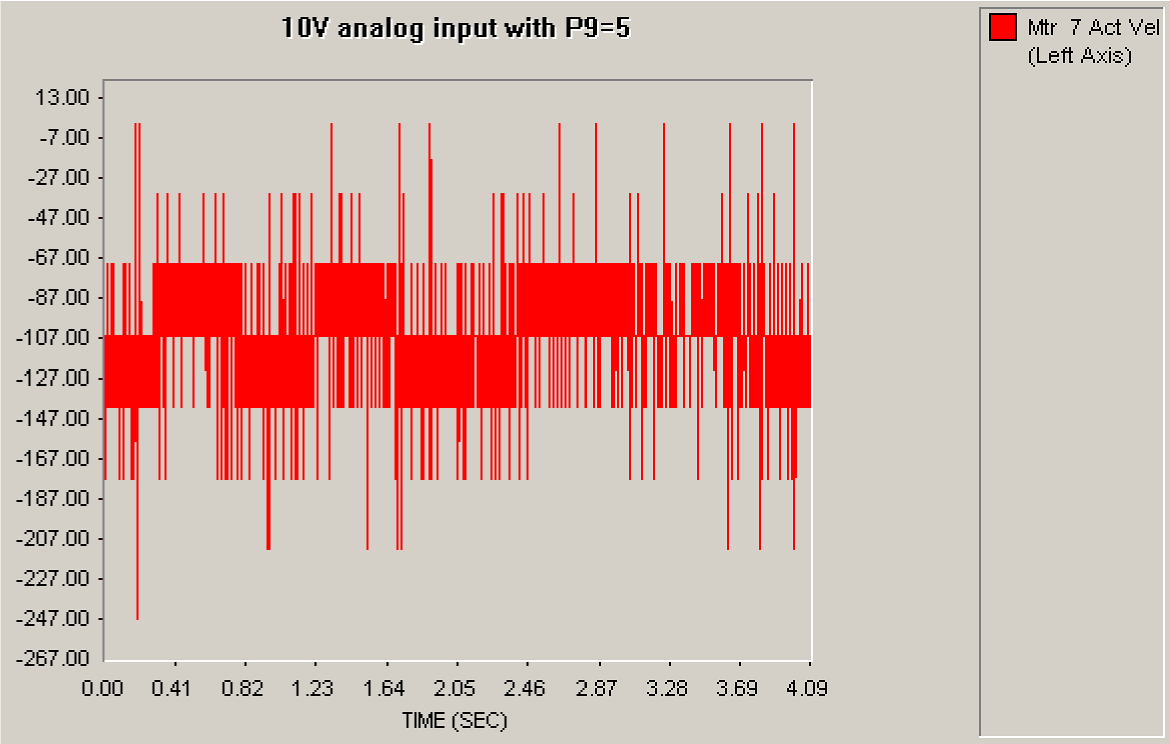
(see file “PMAC V Loop i530 10 lakh.txt”) → 1 lakh = 100000

Check with low pass filtering the PID: → 100 Hz and 10 Hz (low pass 2. order), no changes

Task 4 at AZ "Locked Rotor Frequency" Test (LRF)

With the old system a LRF was made. Basically it is a sine sweep of the velocity loop with the following parameters:
Start frequency: 0.2Hz
Stop frequency: 9 Hz
Sweep time: 100s
Amplitude (at motor): 35rpm
With our gear ratio of 1488 * 12.6 = 18748.8 this means about 40 arcsec / sec = 100 counts / sec
With the scaling of the analog input (P9) = 200 we are getting with 10V analog input about 4500 counts / sec
So we are changing the P9 = 5 and we are getting 112 counts / sec with 10V

Check:



Unstated behavior when input voltages pass zero line:

This unstated behavior might be in relation to the sinusoidal following error at low speeds.

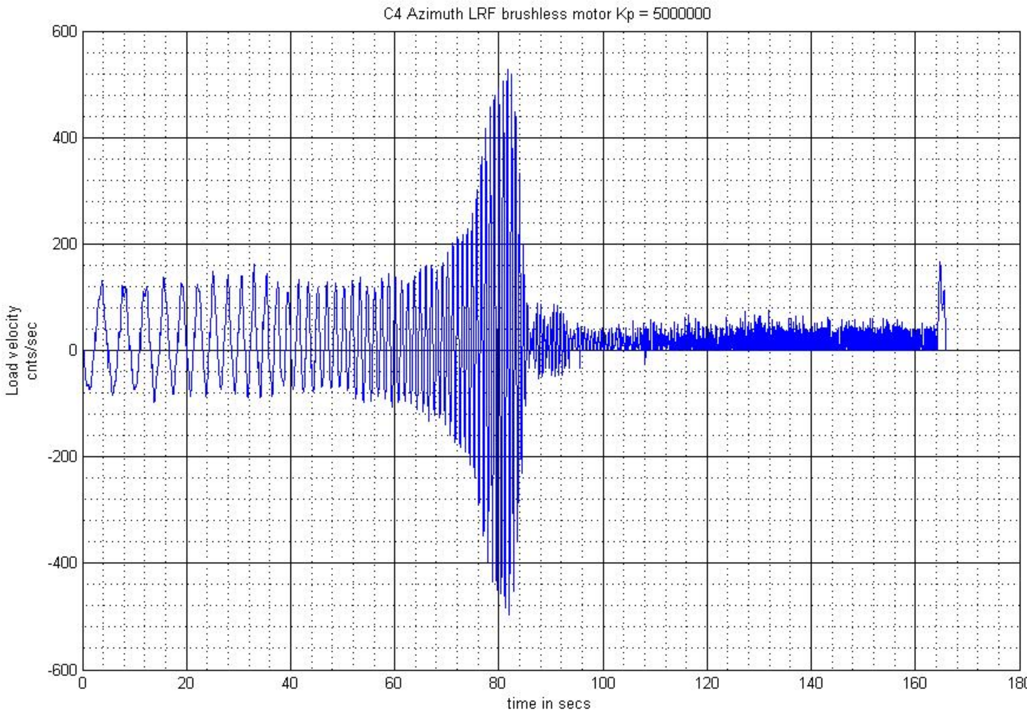
LRF test was carried out with the command given by a function generator to the analog input of the BLC. Analog input (M5063, scaled with 0.0039 in detailed plot) and actual position (M562) of the **load encoder (!)** was recorded by detailed plot of PMAC Plot Pro (measuring the load encoder is different to the old system, there only tachometer was measured).

Scaling of analog input P9 = cmdVelScaling = 50 in order to obtain with sweep amplitude of 2V (peak-peak) the requested velocities.

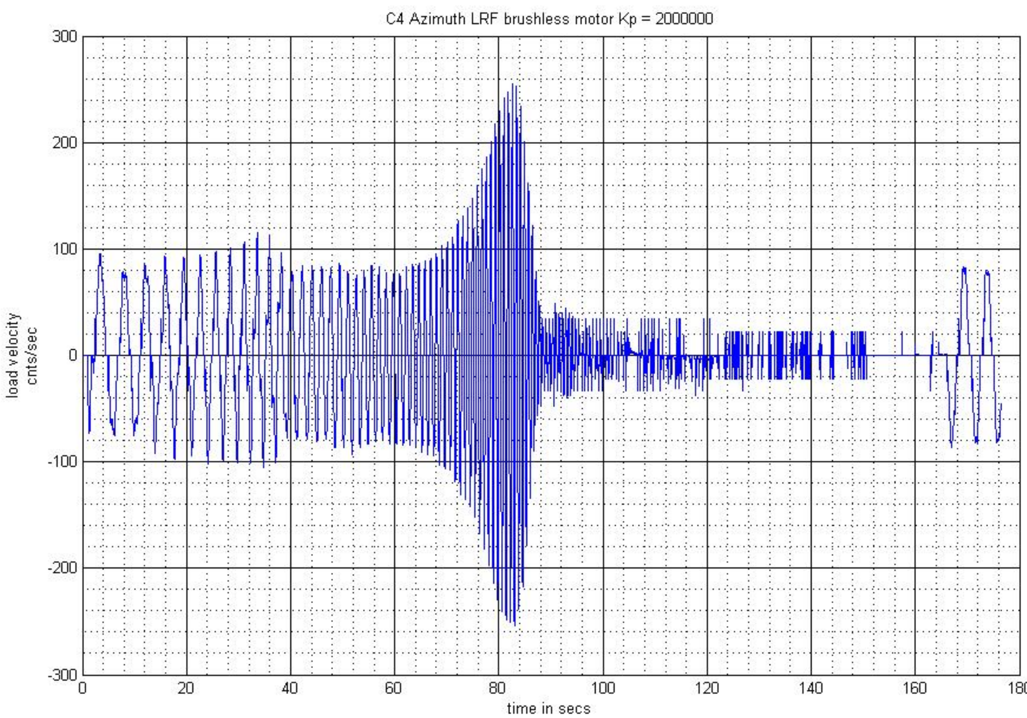
For higher accuracy the position was measured, velocity needs to be calculated offline.

Different P gains of PMAC velocity loop were checked (i530).

Measurements should be repeated with measuring the resolver feedback.



Resonance frequency: 1.288Hz, cut-off: about 1.45Hz



Resonance frequency: 1.33Hz, cut-off: about 1.5Hz

Task 1 to 4 at EL (21.04.2009)

Servostar configurations saved to the EL 1 und EL2 drives (“16042009 Adapted Current Levels”)

DAC offsets #3 (EL1): i329=-280
DAC offsets #4 (EL2): i429=-170 (later removed)

Axis 3 orientation has to be changed, since the motors are mounted opposite to each other, therefore:

- i7030=3 (all other axis are at 7)
- analog command changed at Servostar side X3/4 and X3/5 is swapped

Checking the backlash:
#4j/ #3o5 / #3hmz #4hmz
Position of #3 after change of o command: 136.592 counts → 10804640,63” (motor side) → 430” = 7 arcmin (load side)

Run motor 4 in closed loop (only resolver feedback / motor 3 in open loop o0) → ok
Run motor 3 in closed loop (only resolver feedback / motor 4 in open loop o0) → ok

Check of load encoder:
Feedback orientation needs to be changed: I7120 = 3

Check of the resolution:
#3j=1000000counts → #6 = 8000 counts

Resolver (motor) count after one revolution of big gear = 4096*4*821.976*30.55 = 411424633.7 (to be checked because of different possible gear factor!)

Load encoder reading (considering 17 bit absolute and interpolator) 8192*400 = 3276800

motor to load reading ratio = 411424633.7/3276800 = **125.5568**

if position feedback scaling = 96
then velocity "position" feedback scale is = 1 approx.

New Encoder Table Definitions.
Entry Address Y-Word Conversion Method

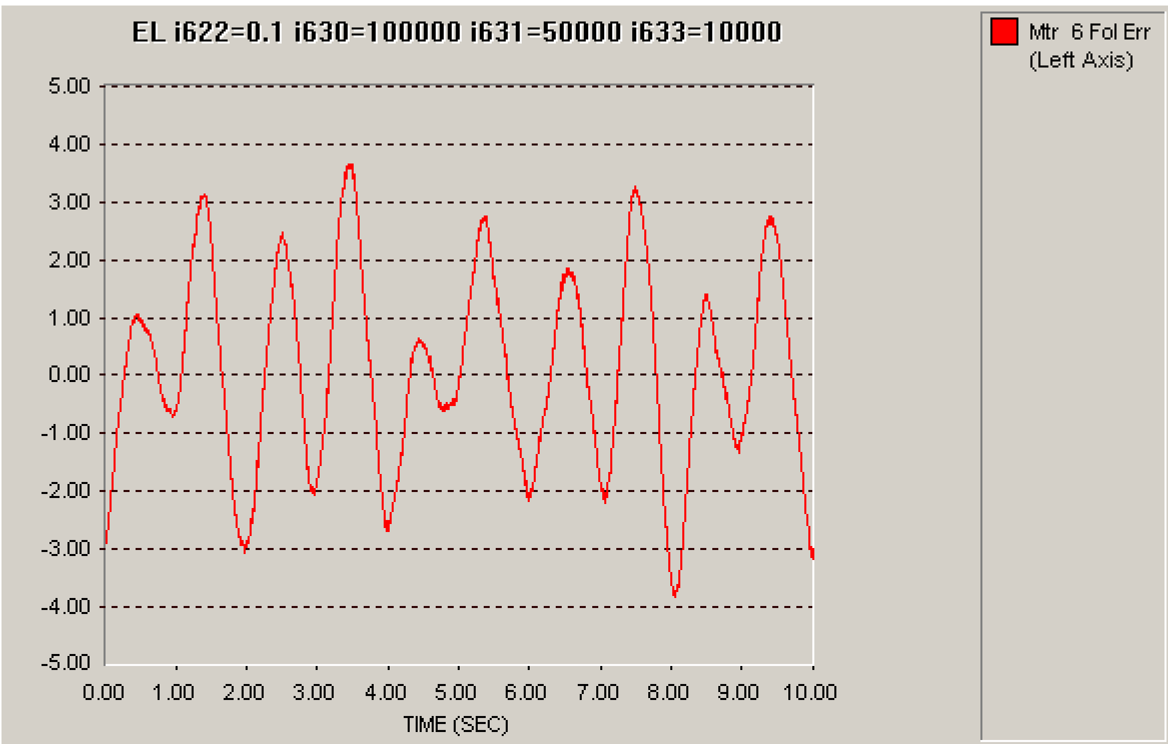
1	Y:$	3501	$078000	1/T extension of location $78000
2	Y:$	3502	$078008	1/T extension of location $78008
3	Y:$	3503	$078010	1/T extension of location $78010
4	Y:$	3504	$078018	1/T extension of location $78018
5	Y:$	3505	$078100	1/T extension of location $78100
6	Y:$	3506	$078108	1/T extension of location $78108
7	Y:$	3507	$078110	1/T extension of location $78110
8	Y:$	3508	$078118	1/T extension of location $78118
9	Y:$	3509	$E00100	Summing Of Conversion Table Entry 1 with Entry 2
10	Y:$	350A	$E00302	Summing Of Conversion Table Entry 3 with Entry 4

22.04.09

Problems with the communication (plc0 too long, shorted → ok)

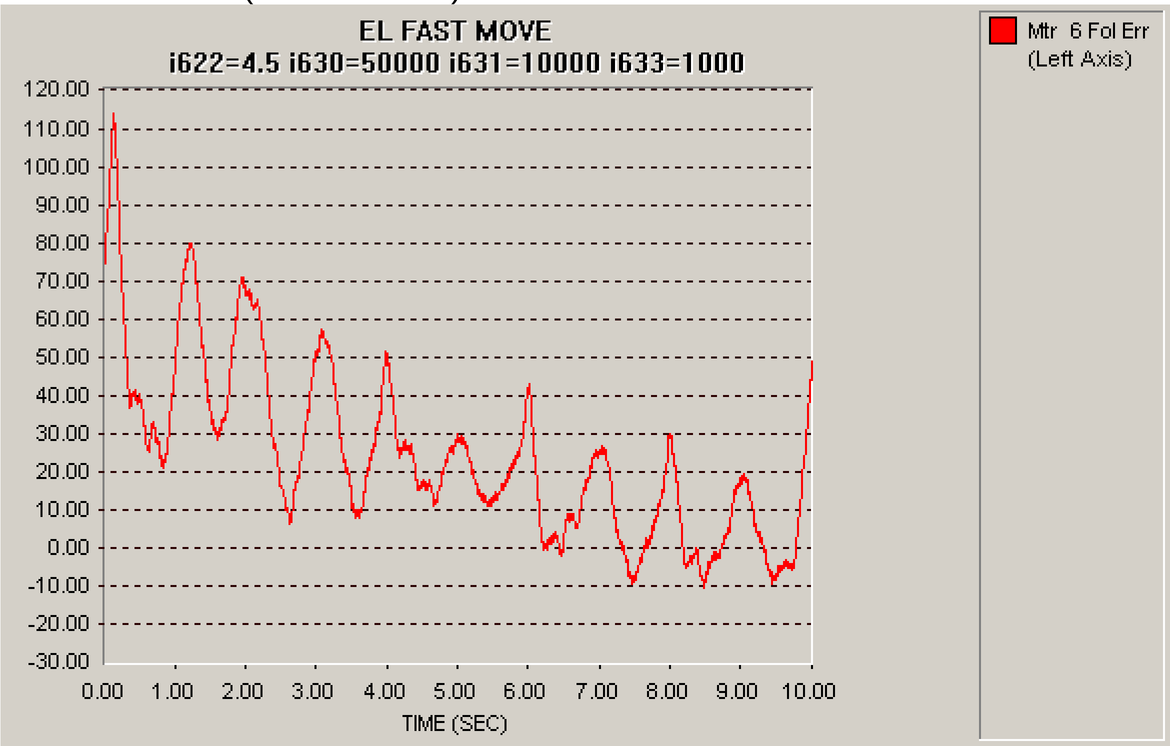
plc0 running / torque offset 1000 / #3o0#4o0 / #3hmz#4hmz / torque offset -1000
#3 = -105000 counts
#4 = 50000 counts → i.e. backlash about 155000 counts (as before measured)

Now doing the position loop (speed i622=0.1 means 40”/s; following error is in counts and scaled by 0.4” per count):

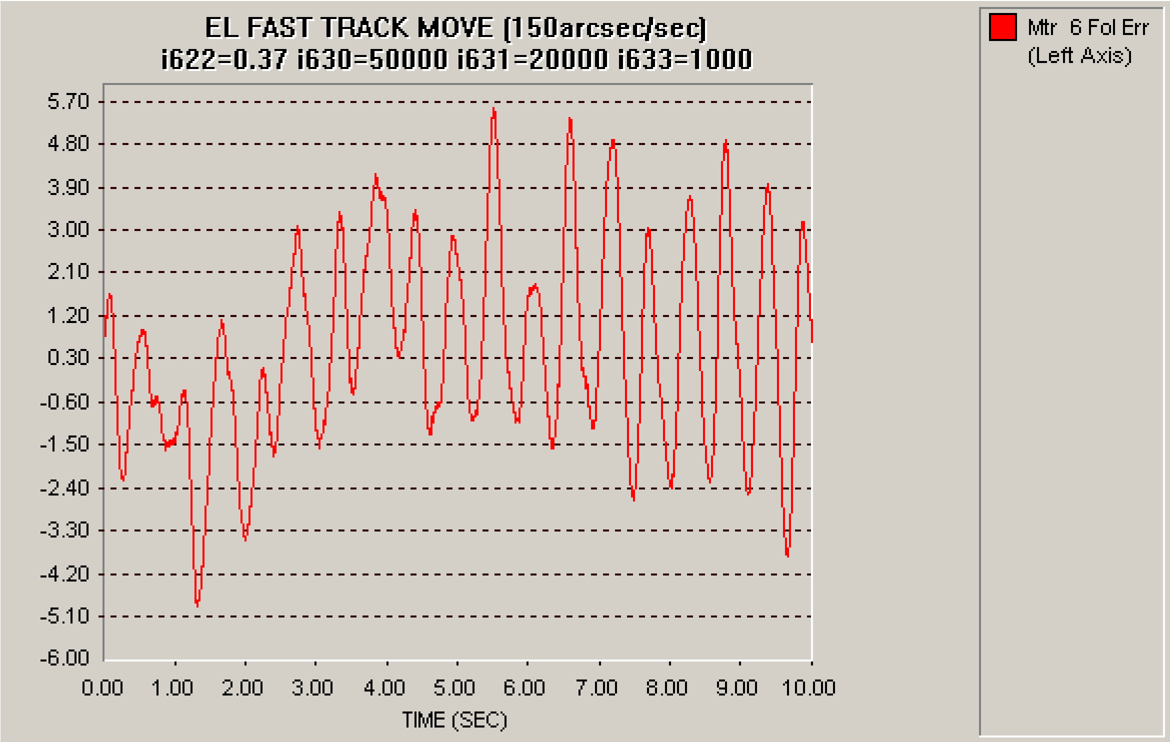


(see file “EL PMAC P Loop i622 0.1.txt”)

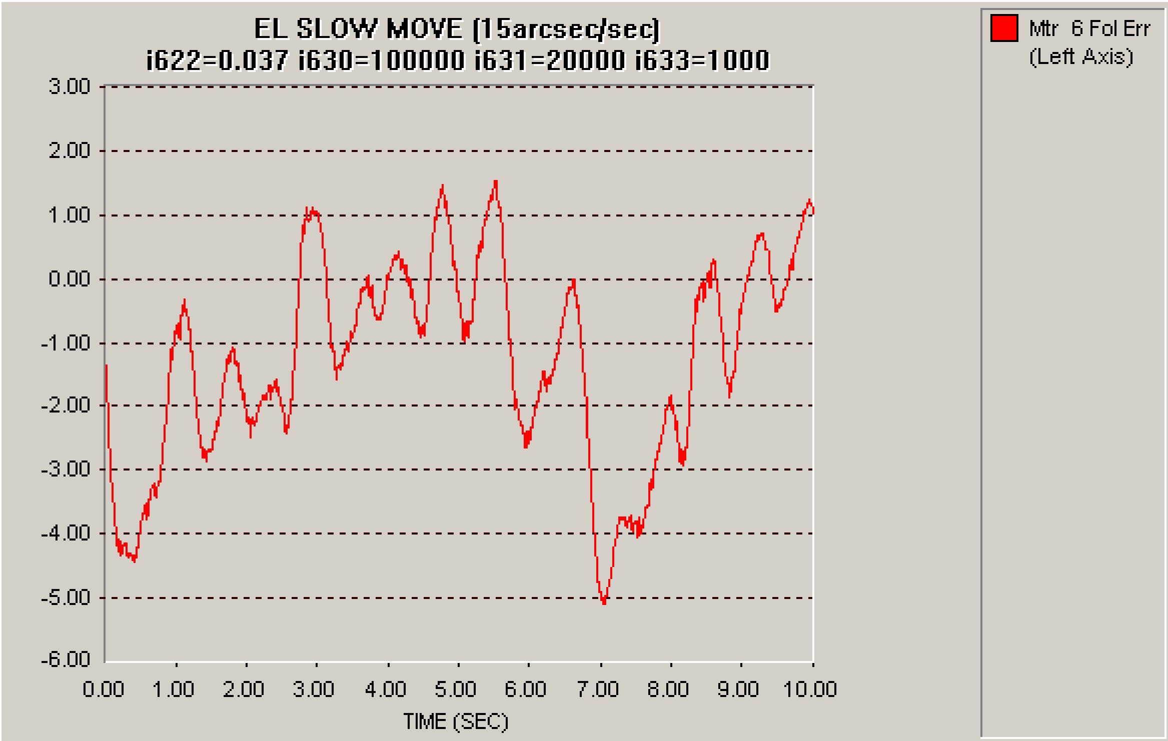
EL FAST MOVE (about 30°/min)



(see file "EL PMAC P Loop i622 4.5.txt")

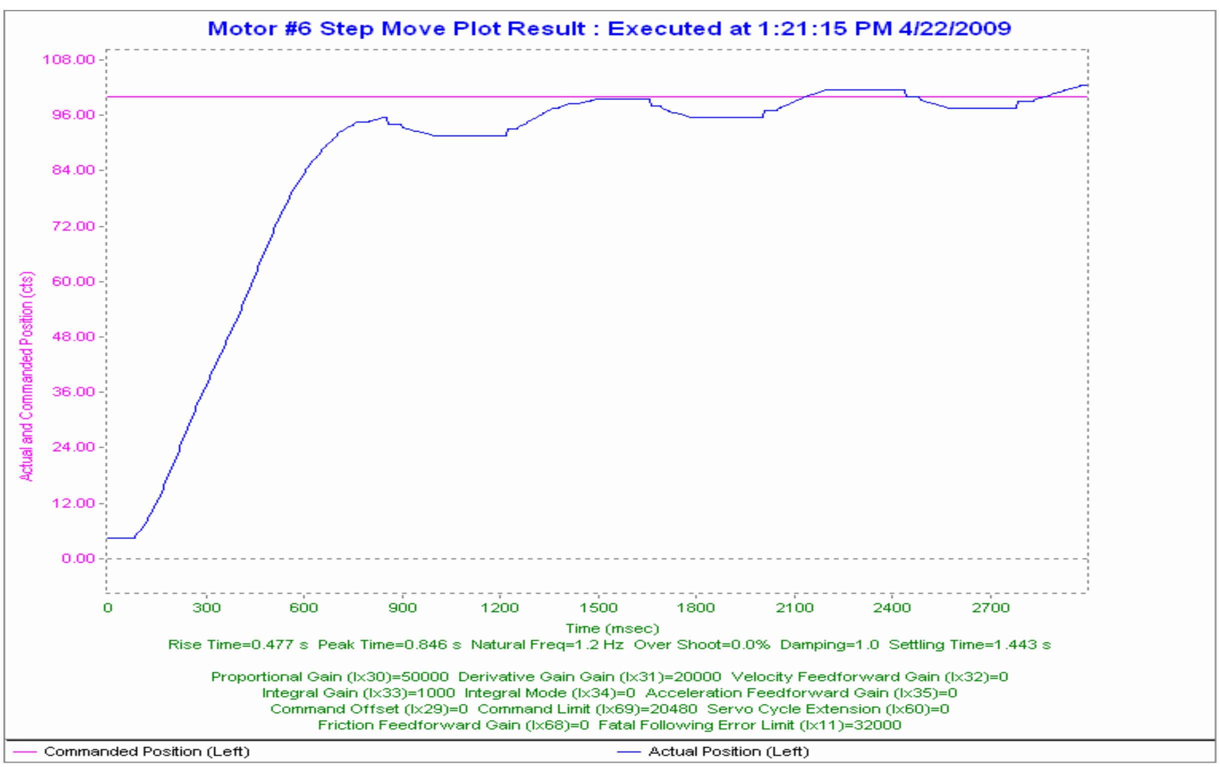


(See file "EL PMAC P Loop i622 0.37.txt")

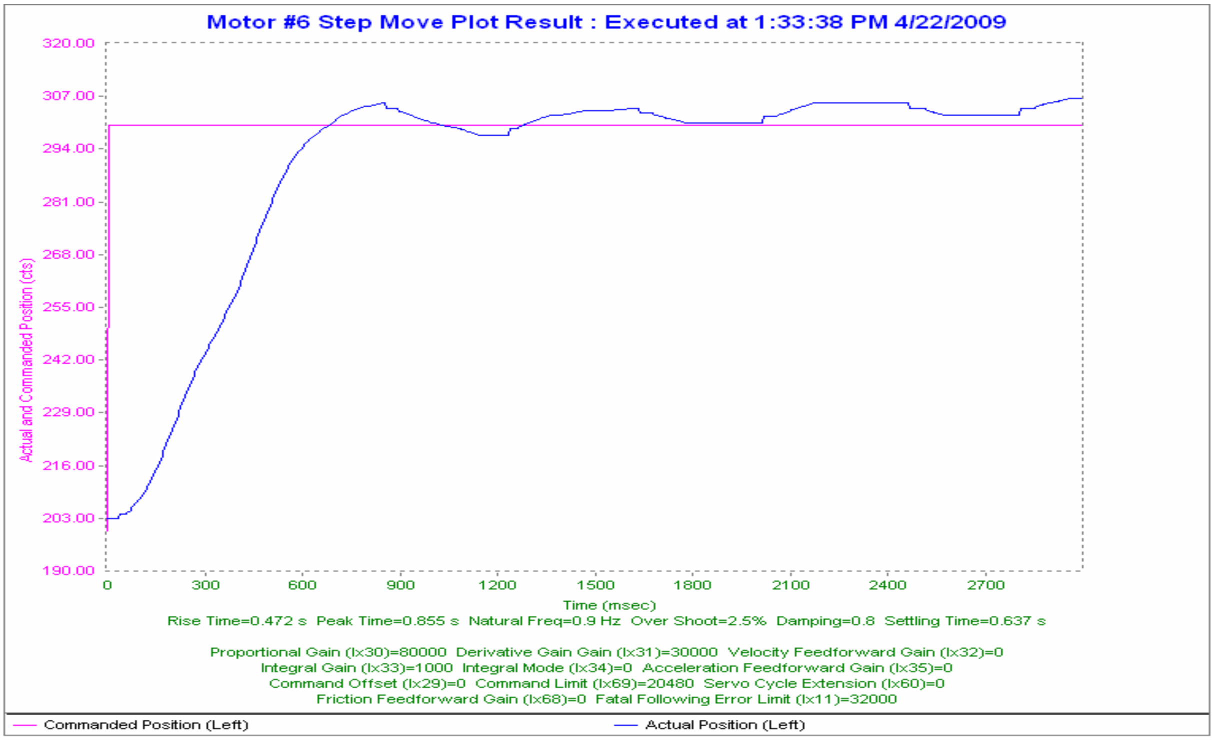


(see file "EL PMAC P Loop i622 0.037.txt")

Step Response EL:



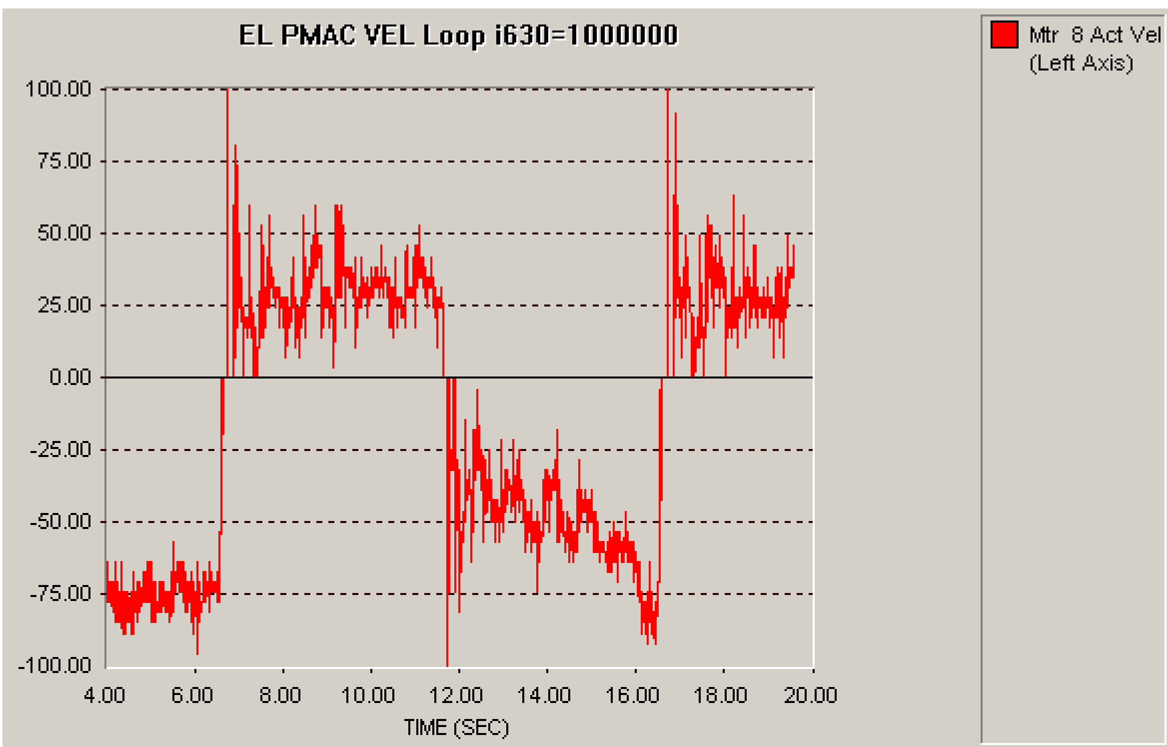
(see file "EL Step Response i630 50000 i631 20000.TXT")



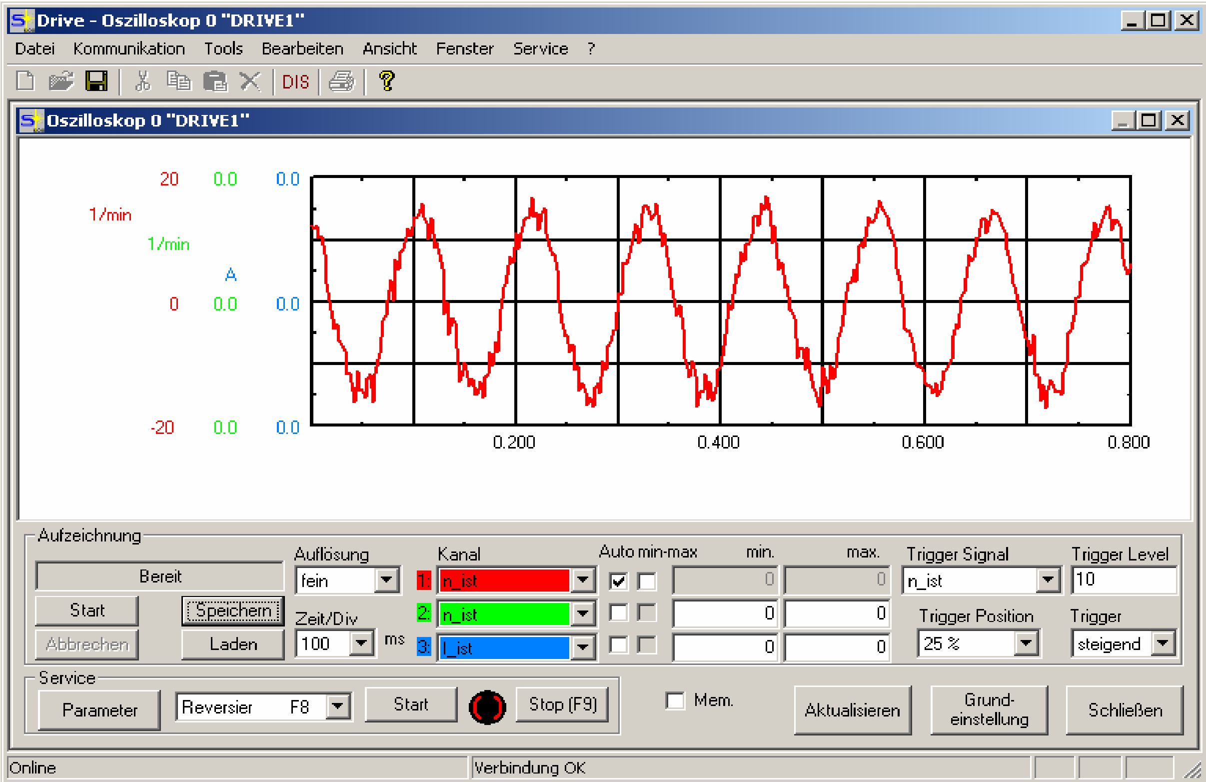
(see file “EL Step Response i630 80000 i631 30000.TXT”)

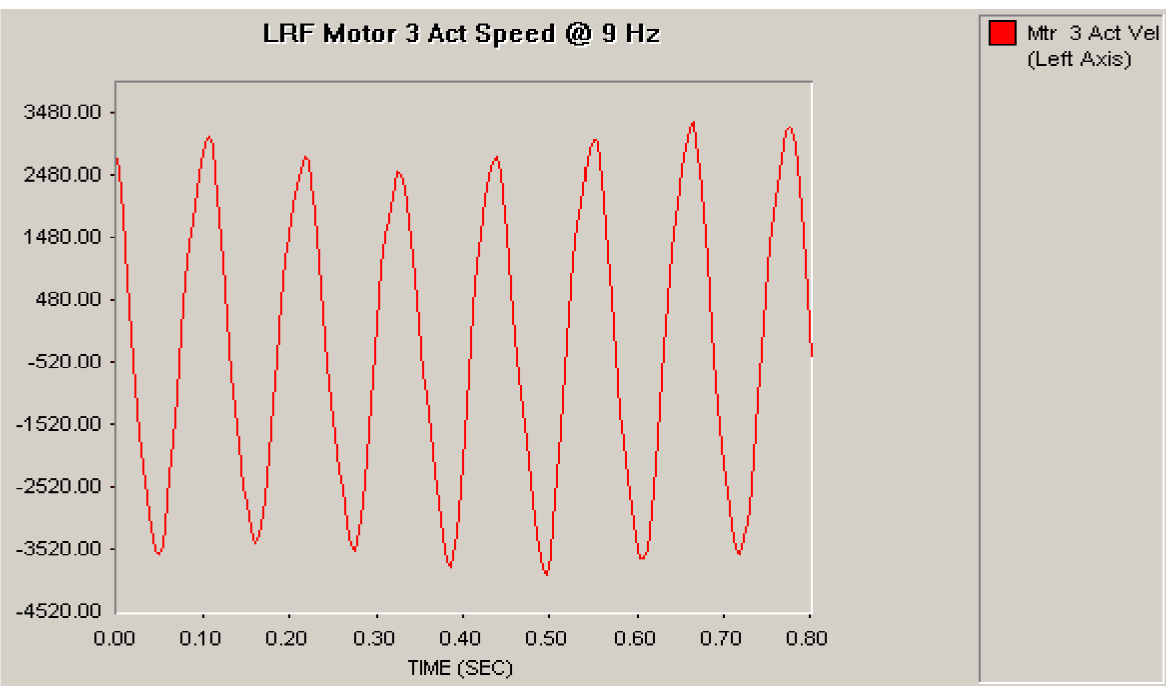
Now doing EL velocity loop with PMAC PID

→ see file: “AZ EL preload velocity 1.pmc”



Task 4 at EL “Locked Rotor Frequency” Test (LRF) (22.04.2009)
Settings as above





Regeneration

Critical situation for regenerations (according GMRT): When the wind speed is increasing from 40km/h to 80km/h the EL axis needs to be moved with high speed (20°/min) from 15° to 90° (i.e. parking position). Reflected speed to the motor is:
20°/min * 821.976*30.55 = 20° / min / 360° * 25111 = about 1400 rpm

Time for that move will be about 4min, during that time max. regeneration can occur.

When using the maximal torque of 20Nm, we get a mechanical power of maximal 2800W per motor.

Since each of the two AZ and EL motors are working in anti-backlash the DC-bus of the two drives should be connected (via X7). This lowers the regeneration load.

Additionally the regeneration load can be checked in the “DRIVE” software of Servostar in the “Monitoring View”.

Annexure – A (Algorithm for pre-loading the two motors with PMAC configured as Position loop)

Algorithm for pre-loading the two motors along with configuration of I variables for channels #1, #2 and #5 where motors 1, and 2 and load encoder are connected**. (File name: - upload 07012009_bsr.CFG) See attached CD.**

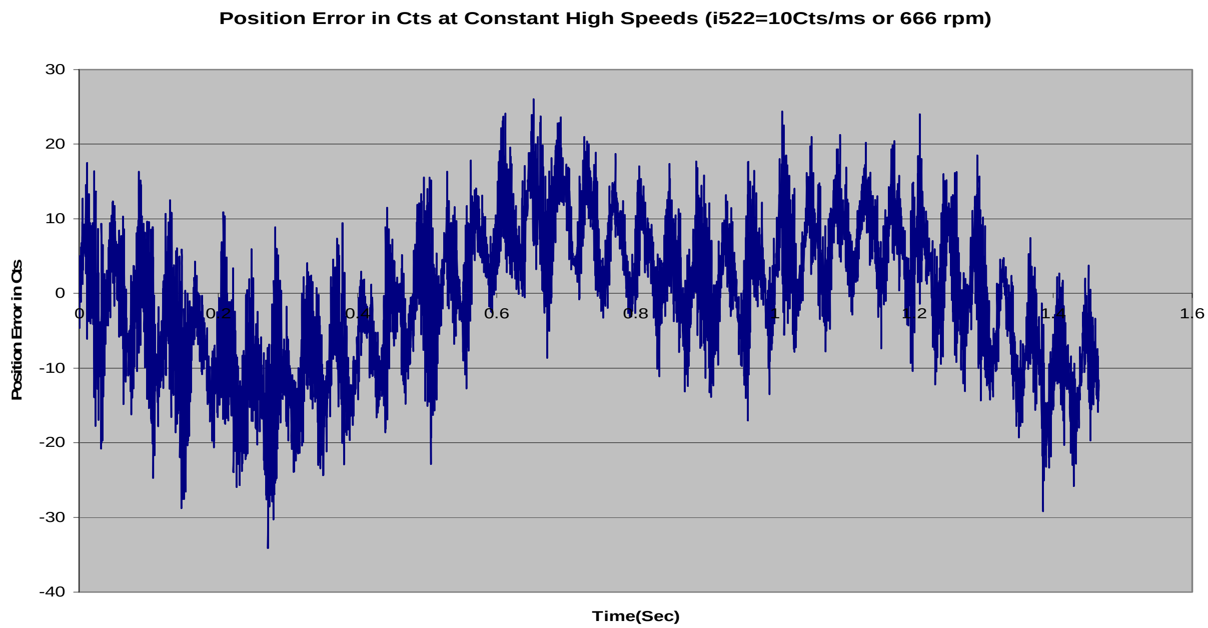
```
; PLC0 algorithm
 Close
 Endg
 Del gat
 i7016, 2, 10=1                ; true DAC Output
 i100, 2,100=1                 ; activate axis
 i122, 2,100=15                        ; slow default speed
 i119, 2,100=1                 ; higher acc + dec
 i123, 2,100=5                 ; homing speed
 i124, 2,100=$20001            ; no limit switches
 i130, 2,100=500000            ; PID Settings
 i131, 2,100=5000
 i132, 2,100=5000
 i133, 2,100=10000
 i134, 2,100=0
 i169, 2,100=16384             ; 10V diffential DAC Output

 ; load encoder setting
 i7110=3                       ; changing counting direction
 ; ---------------------------------------------------------
 ; PLCC0 real-time task for torque offset and active damping
 ; Standard position/speed control loop at axis 5
 ; Control output of axis 5 distributed to axes 1 and 2


 ; adding a torque offset
 ; Axes 1 and 2 must be activated via command O0
 ; When killing axes 1 and 2, Torque Offset must be reset to 0

 #define velocity Load      M574   ; filtered (unfiltered is M166)
 #define velocityMotor1     M174   ; filtered (unfiltered is M266)
 #define velocityMotor2     M274   ; filtered (unfiltered is M366)
 #define torque1            M179
 #define torque2            M279
 #define desTorque          M568
 #define CmdPos             M561
 #define ActPos             M562
```

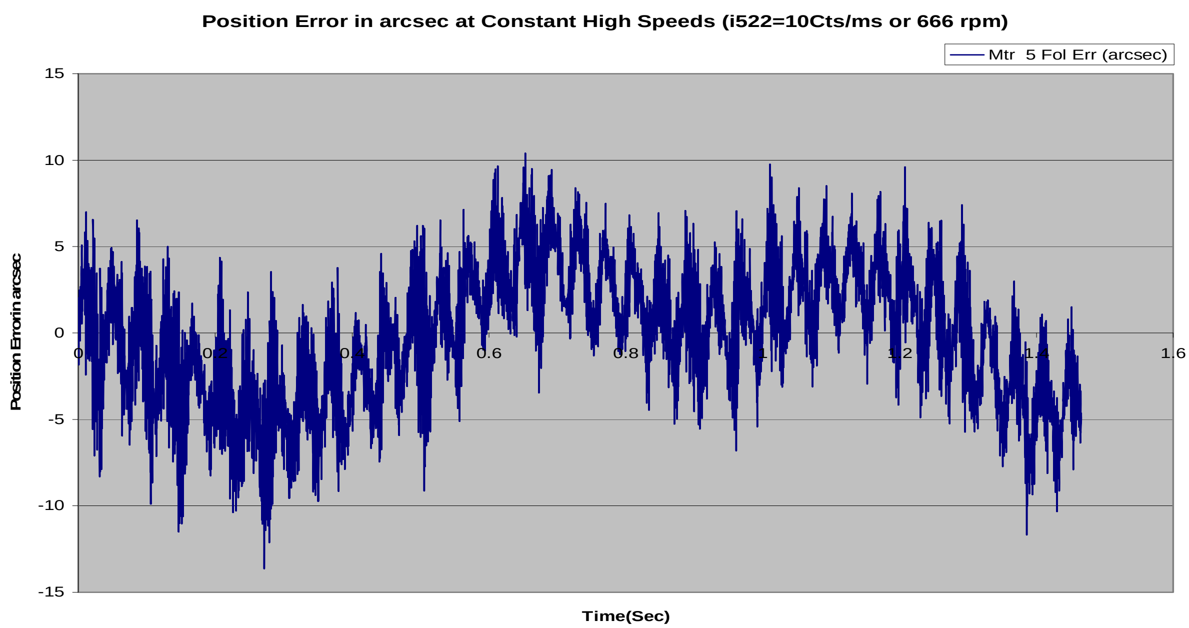
```
#define PosError                    P1     ; position control deviation
#define FRICTION Offset        P2
#define D1                            P3     ; damping coefficient 1
#define D2                            P5     ; damping coefficient 2
#defines GR                         P6     ; gear ratio
#define MAX_TORQUE          P7   ; Nm scaled to 16 bit integer
#define TORQUE_OFFSET      P8   ; Nm scaled to 16 bit integer
D1          = 0
D2          = 0
GR          = 8.64257
MAX_TORQUE    = 32768
TORQUE_OFFSET = 1150


I5 = 3     ; PLC program control enabled
I8 = 0     ; PLCC 0 called every sample
; Motor encoders used for velocity feedback
I8008 = $E00100     ; sum of motor 1 and 2 encoders written into
; Motor 5 velocity feedback register
i500 = 1
i503 = $3505
i504 = $3509
i508 = 96
I509 = 4          ; motor 5 velocity scaling factor
; Half of default value 96 to get average
; Of motor 1 and 2
; considering the different resolution of motor and load
i524 = $20001

OPEN PLC 0 CLEAR
; friction compensation
PosError = cmdPos - actPos;
If (posError > 0)
  desTorque = desTorque + FRICTION_Offset;
EndIf
If (posError < 0)
  desTorque = desTorque - FRICTION_Offset;
EndIf



; Torque offset
If (desTorque < 0)
  torque2 = desTorque/2 - TORQUE_OFFSET;
  If (torque2 < -MAX_TORQUE/2)
    torque2 = -MAX_TORQUE/2
```

```
  EndIf
  torque1 = desTorque - torque2
Else
  torque1 = desTorque/2 + TORQUE_OFFSET;
  If (torque1 > MAX_TORQUE/2)
    torque1 = MAX_TORQUE/2
  EndIf
  torque2 = desTorque - torque1
; active damping
torque1 = torque1 - D1 * (velocityMotor1 - velocityMotor2) - D2 *
(velocityMotor1 + velocityMotor2 - 2*velocityLoad/GR)
torque2 = torque2 + D1 * (velocityMotor1 - velocityMotor2) - D2 *
(velocityMotor1 + velocityMotor2 - 2*velocityLoad/GR)
; Saturation
If (torque1 > MAX_TORQUE/2)
  torque1 = MAX_TORQUE/2
EndIf
If (torque1 < -MAX_TORQUE/2)
  torque1 = -MAX_TORQUE/2
EndIf
If (torque2 > MAX_TORQUE/2)
  torque2 = MAX_TORQUE/2
EndIf
If (torque2 < -MAX_TORQUE/2)
  torque2 = -MAX_TORQUE/2
EndIf
CLOSE   ; PLC 0
```

Annexure – B(Measurement of Position accuracy of large Test setup in position loop)

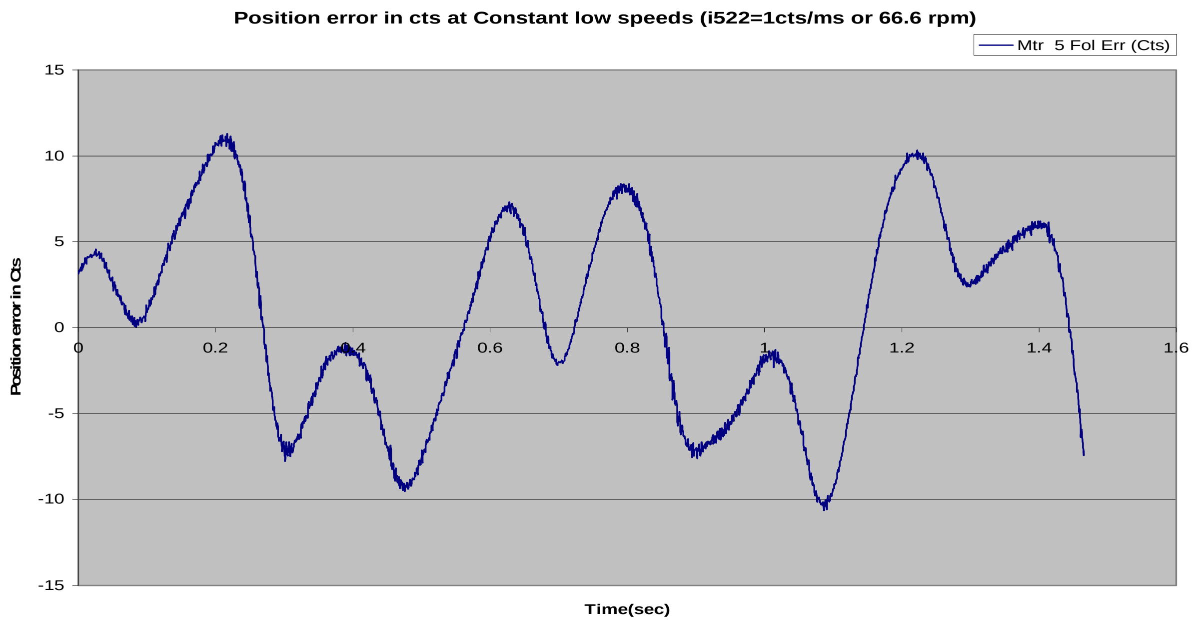
The following plot shows following error with set speed i522=10 counts/m-sec, i.e. 666 rpm of motor and Kp=500,000, Kd=10,000, Kvff=10,000, Ki=10,000

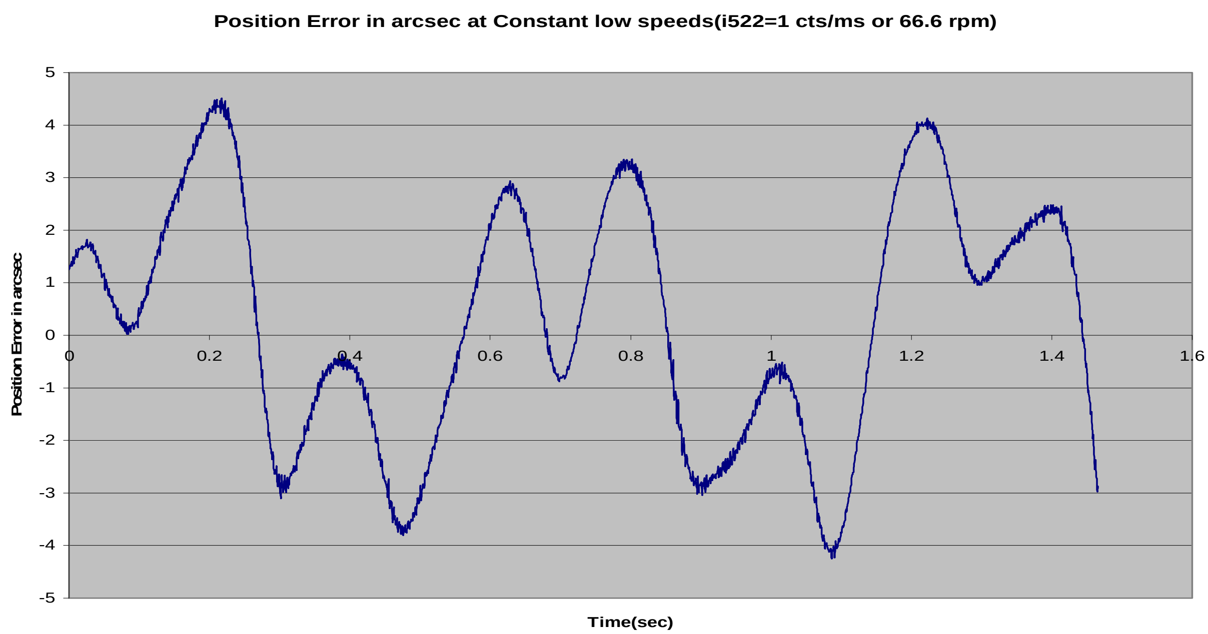




The following error for the plot is between ***+/_ 25 cts*** *or* ***+/-10arcsecs****.*

The following plot shows following error with set speed i522=1cts/ms, i.e. 66.6 rpm of motor and Kp=500,000, Kd=10,000, Kvff=10,000, Ki=10,000

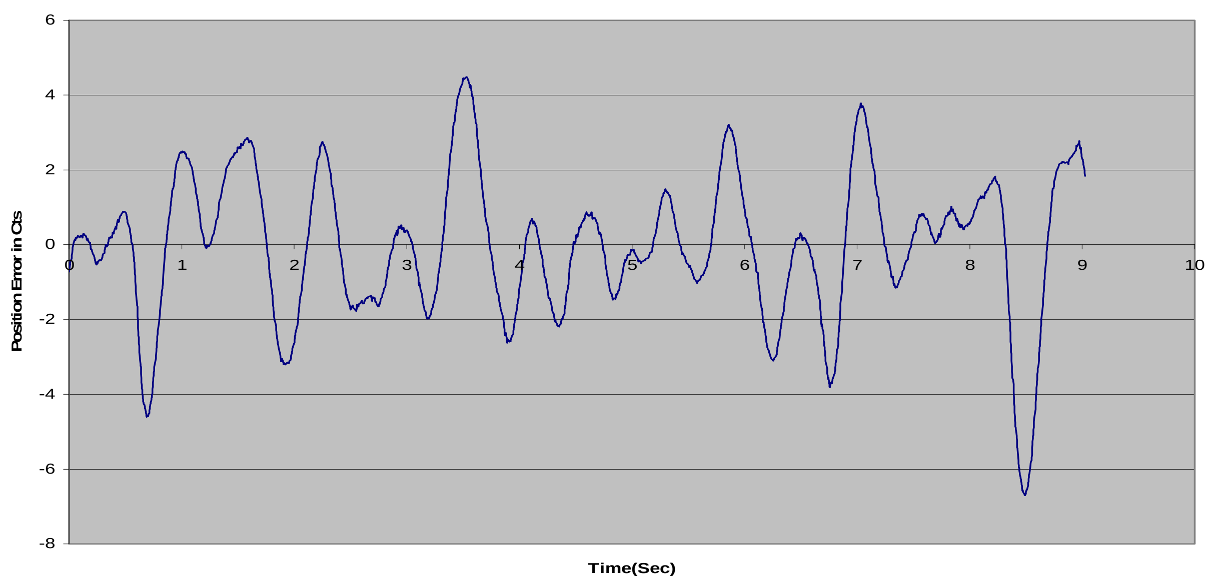




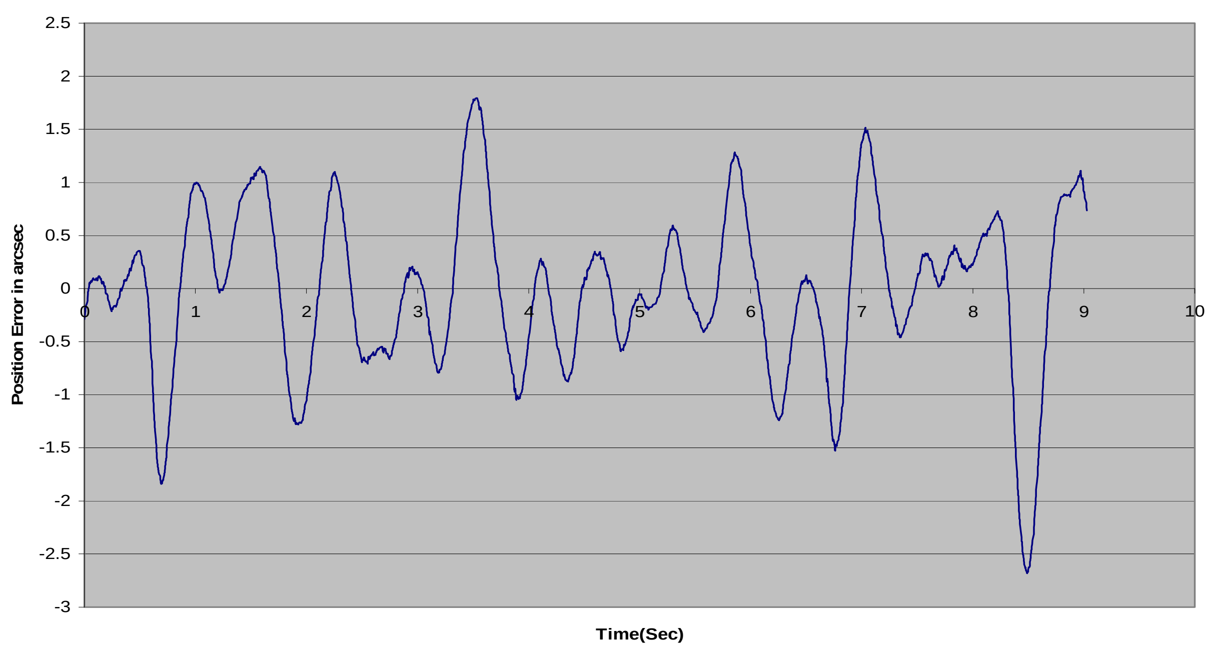
The following error for the plot is between ***+/_ 10 cts*** *or* ***+/-4 arcsec.***

The following plot shows following error with set speed i522=0.1cts/ms, i.e. 6.6 rpm of motor and Kp=500,000, Kd=10,000 , Kvff=10,000 ,   Ki=10,000

Position error in Cts with i522=0.1cts/ms or 6.66 rpm

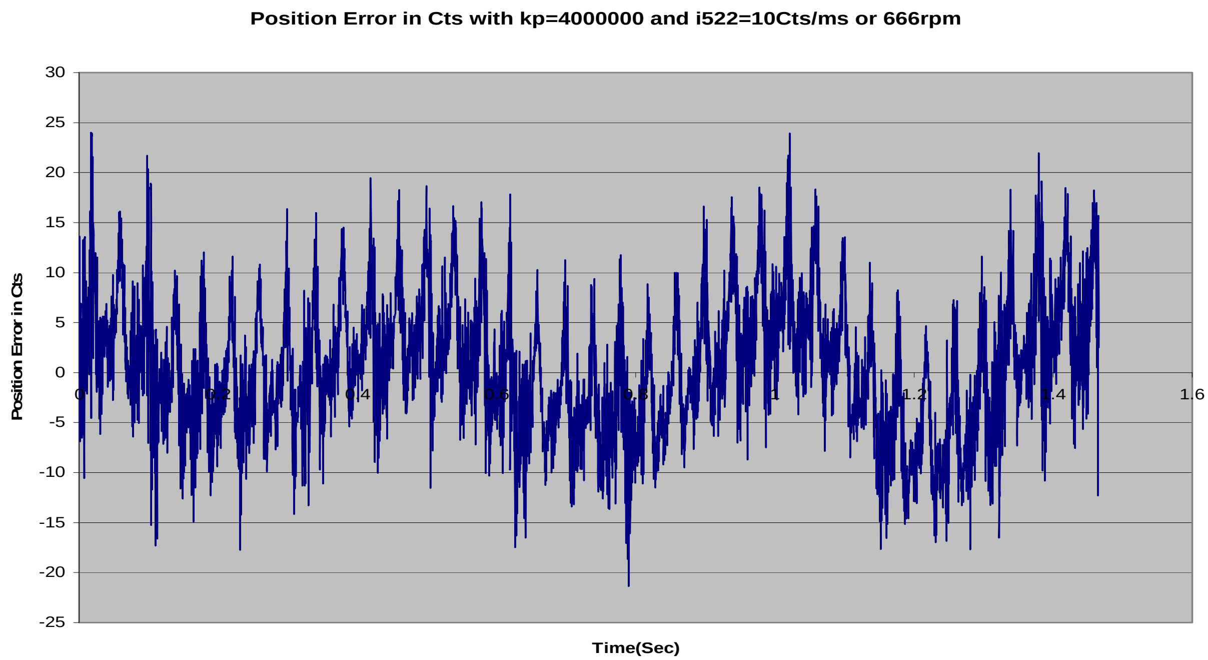


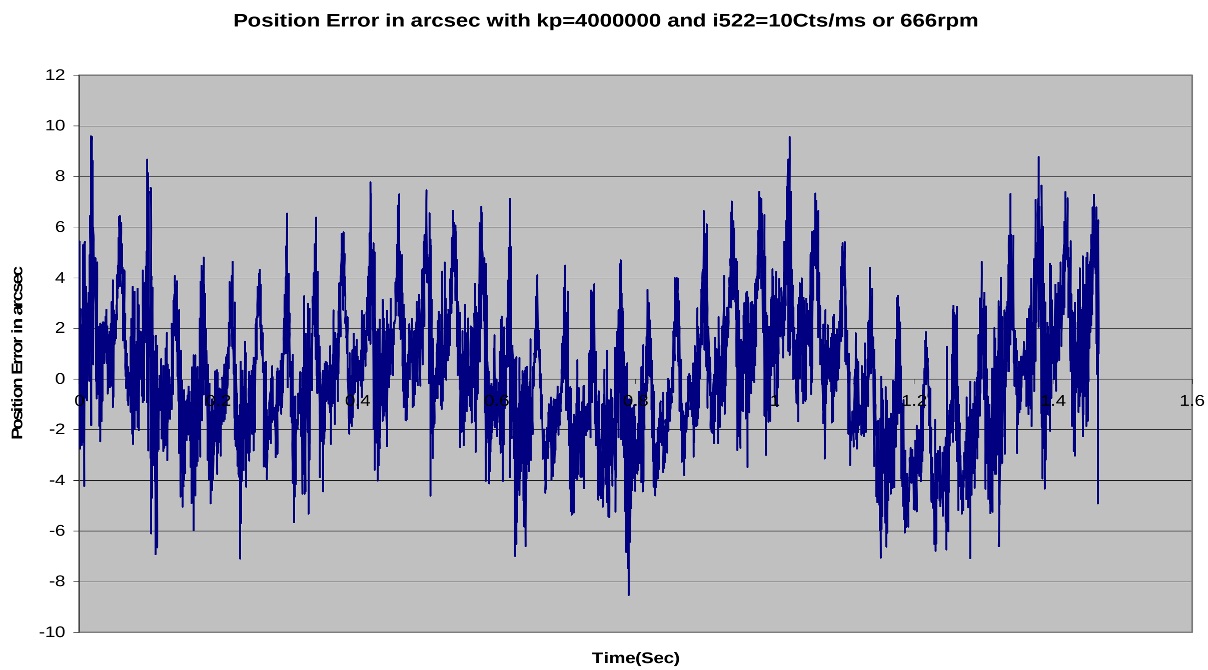
Position Error in arcsec with i522=0.1cts/ms pr 6.66 rpm



The following error for the plot is between +/_ ***4 cts*** *or +/-* ***1.6 arcsecs****.*

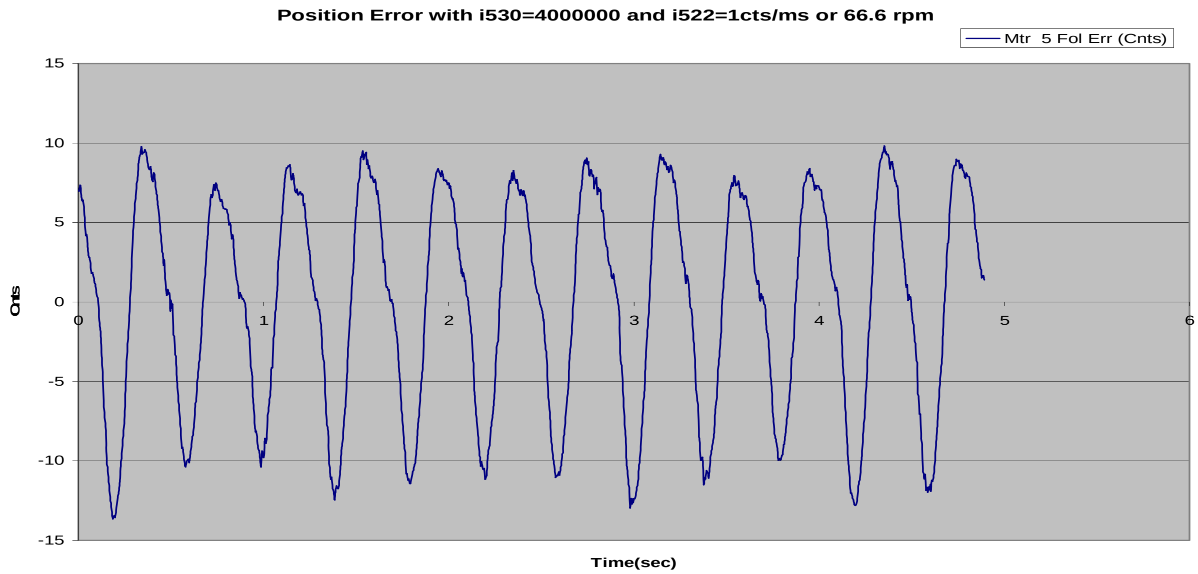
The following plot shows following error with set speed i522=10 cts/ms, i.e. 666 rpm of motor and Kp=40, 00,000, Kd=10,000,   Kvff=10,000, Ki=100,000

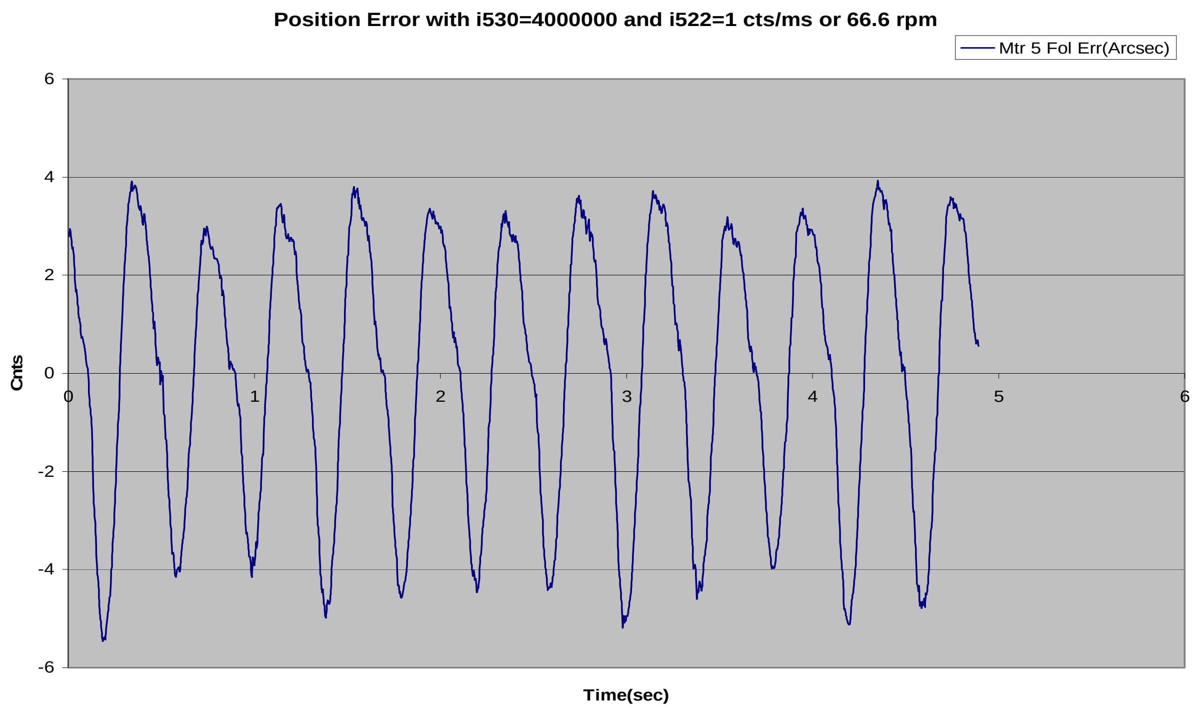




The following error for the plot is between ***+/_ 20 cts*** *or* ***+/-8 arcsecs.***

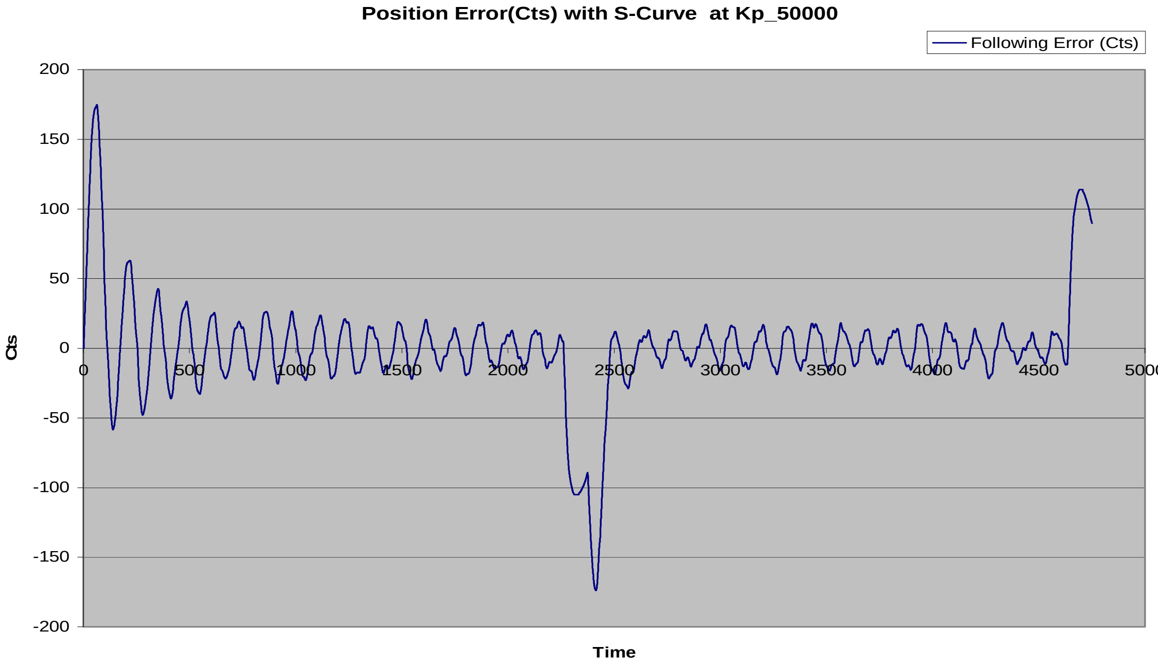
The following plot shows following error with set speed i522=1cts/ms, i.e 66.6 rpm of motor and Kp=40, 00,000, Kd=10,000, Kvff=10,000, Ki=100,000

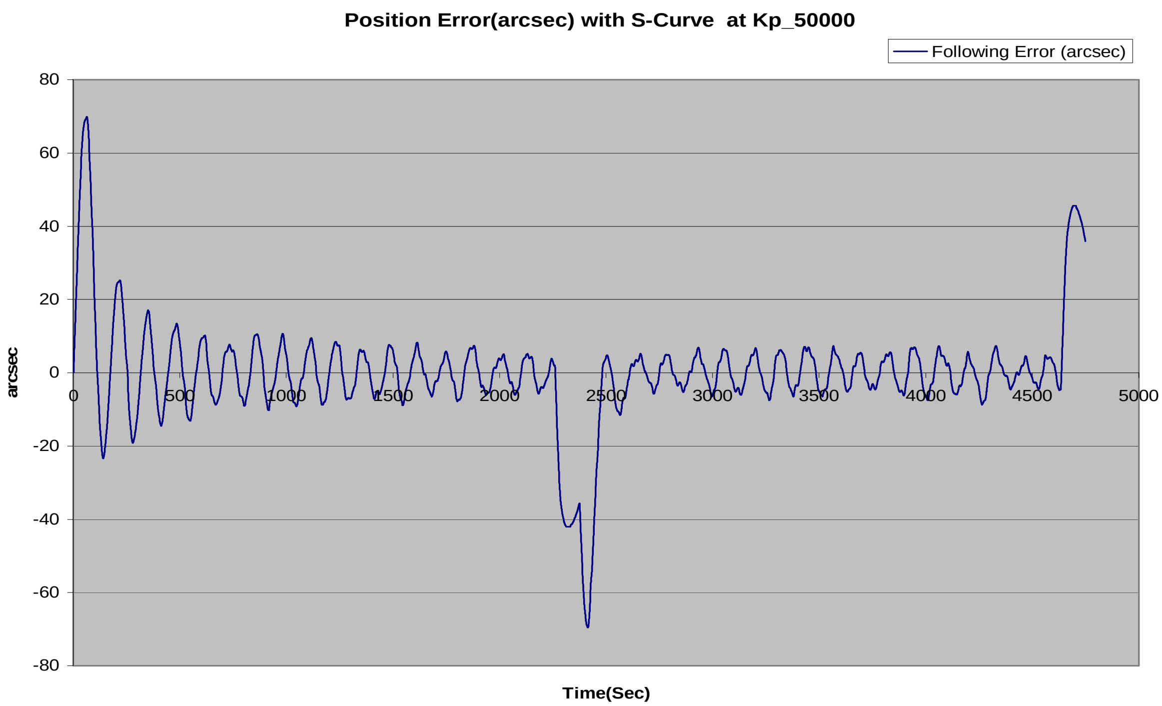




The following error for the plot is between ***+/- 10cts*** *or* ***+/-4arcsec****.*

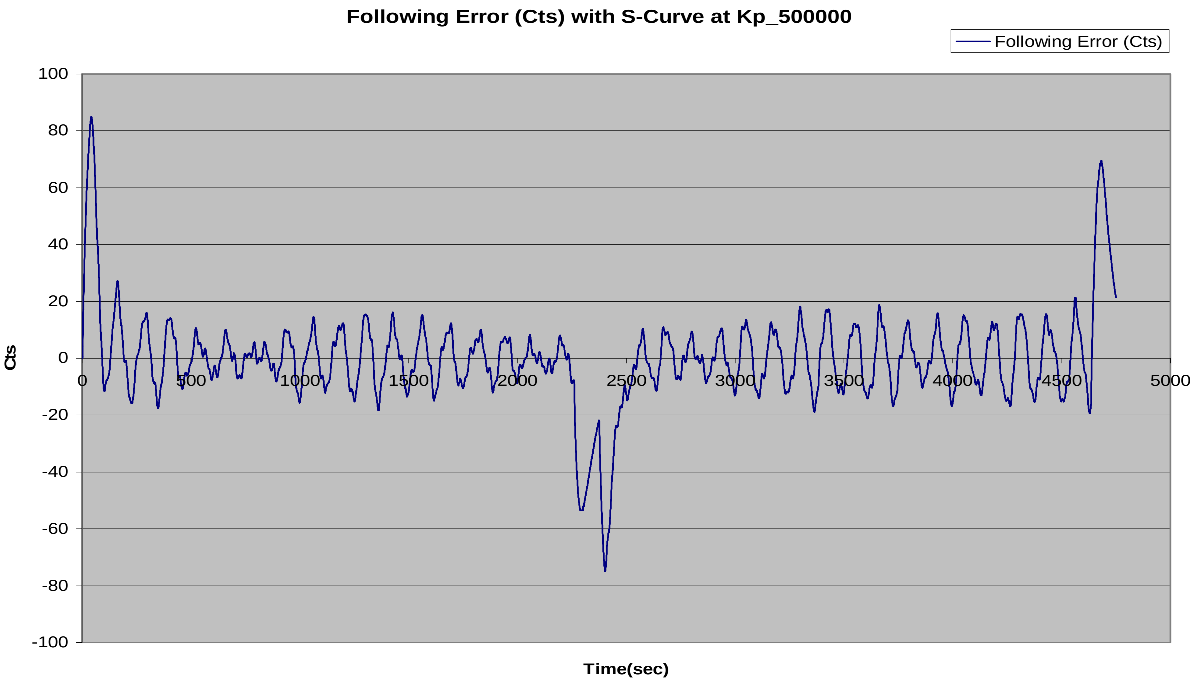
Annexure – C (Brushless motor tuning with S-curve profile)
The following shows Channel #5 (Encoder) S Curve move plot with Kp=50,000, Kd=10,000 Kvff=10,000, Ki=10,000

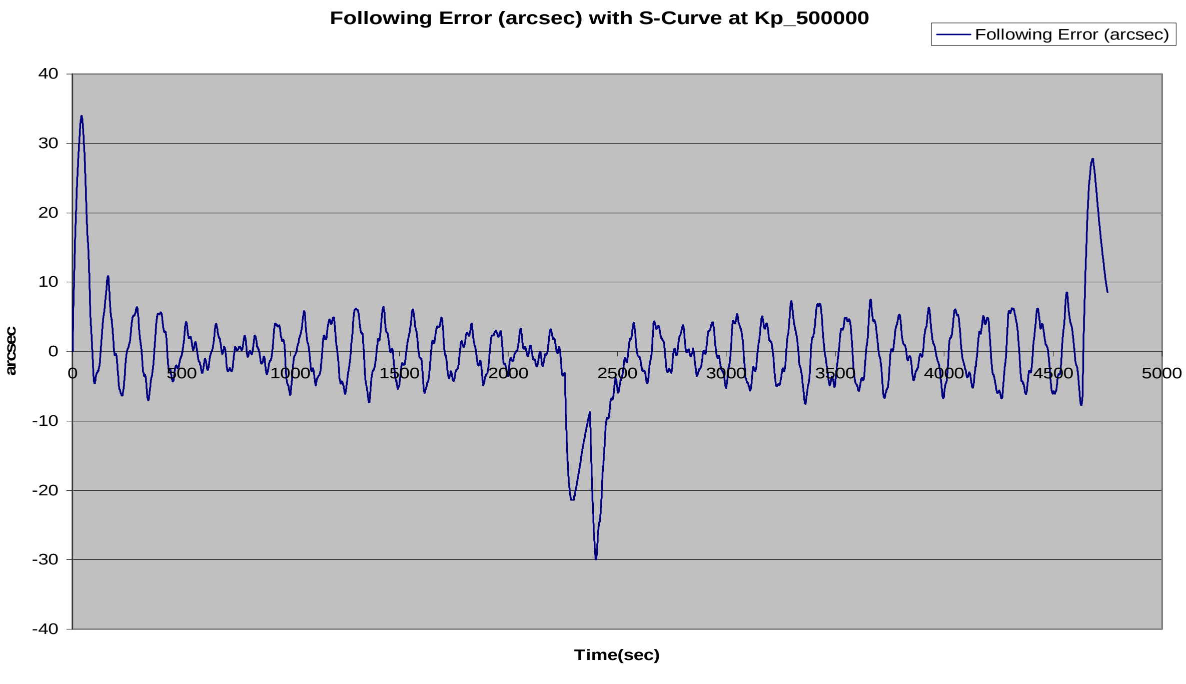




The following error for the plot is between +/- 20 cts or +/- 8 arcsec.

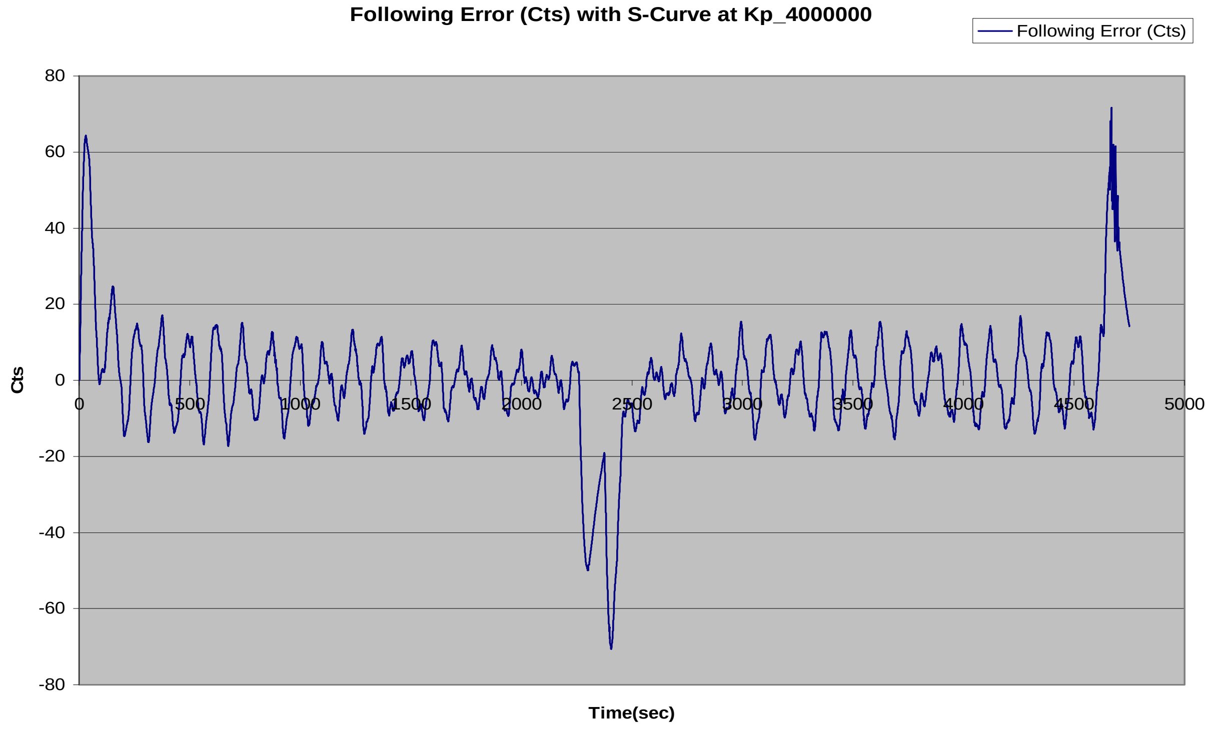
The following shows Channel #5 (Encoder) S Curve move plot with Kp=5, 00,000, Kd=10,000 Kvff=10,000, Ki=10,000

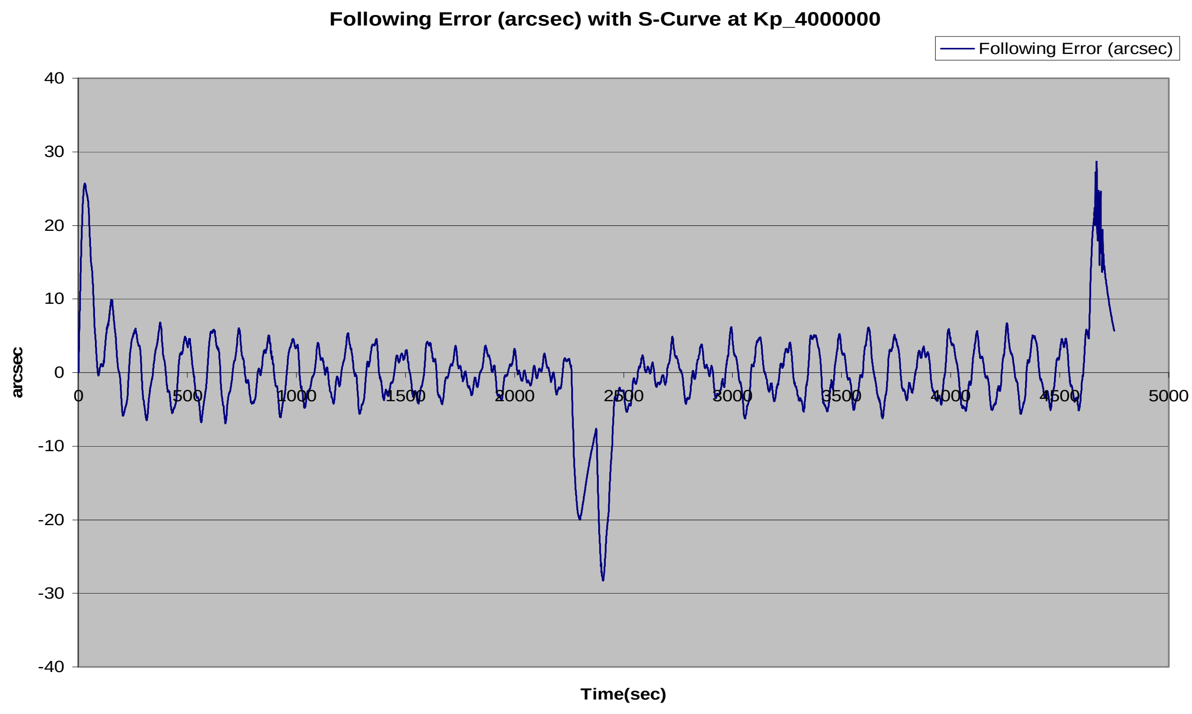




The following error for the plot is between +/- 18 cts or +/- 7 arcsec.

The following shows Channel #5 (Encoder) S Curve move plot with Kp=40, 00,000, Kd=10,000 Kvff=10,000, Ki=100,000





The following error for the plot is between +/- 18 cts or +/-7 arcsec

<u>Annexure D(</u>Algorithm for pre-loading the two motors with PMAC configured as software Velocity loop)

Algorithm for preloading the two motors along with configuration of I variables for Channels #1, #2 and #5 where motor 1, motor 2 and load encoders are connected. (**File name:-upload 09012009_bsr.CFG**) see attached CD.

```
Close
Endg
Del gat


i7016, 2, 10=1                    ; true DAC Output
i100, 2,100=1                     ; activate axis
i122, 2,100=15                          ; slow default speed
i119, 2,100=1                     ; higher acc + dec
i123, 2,100=5                     ; homing speed
i124, 2,100=$20001                ; no limit switches
i130, 2,100=500000                ; PID Settings- Proportional Gain
i131, 2,100=5000                   ; Derivative Gain
i132, 2,100=5000                   ; Velocity Feed forward Gain
i133, 2,100=10000                  ; Integral Gain
i134, 2,100=0                      ; Integral Mode
i169, 2,100=16384                  ; 10V diffential DAC Output

; load encoder setting
i7110=3                            ; changing counting direction

; Definitions for the analog input reading

M5063->Y:$78115,8,16,s ;ch7 A-D channel
i7106=$1FFFFF                      ; ADC strobe word

; ---------------------------------------------------------
; PLCC0 real-time task for torque offset and active damping
; Standard position/speed control loop at axis 5
; Control output of axis 5 distributed to axes 1 and 2
; adding a torque offset
; Axes 1 and 2 must be activated via command O0
; When killing axes 1 and 2, torque offset must be reset to 0

#define velocity Load        M574      ; filtered (unfiltered is M166)
#define velocityMotor1      M174      ; filtered (unfiltered is M266)
#define velocityMotor2      M274      ; filtered (unfiltered is M366)
#define torque1             M179
#define torque2             M279
#define desTorque           M568
```

```
#define cmdVelSignal        M5063     ; analog input

#define FRICTION Offset          P2
#define D1                       P3      ; damping coefficient 1
#define D2                       P4      ; damping coefficient 2
#defines GR                      P5      ; gear ratio
#define MAX_TORQUE               P6      ; Nm scaled to 16 bit integer
#define TORQUE_OFFSET            P7      ; Nm scaled to 16 bit integer
#define cmdVelScaling            P8      ; analog input -> real velocity
; (same units as velocity Load)
#define cmdVelOffset             P9      ; only if required
; (same units as velocity Load)
#define cmdVel                   P10     ; scaled commanded velocity
#define proportional Gain        P11
#define integral Gain            P12
#define velError                 P13
#define velErrorIntegral         P14
#define integral Lock            P15

FRICTION Offset    = 0
D1                 = 0
D2                 = 0
GR                 = 8.64257
MAX_TORQUE         = 32768
TORQUE_OFFSET      = 1150

CmdVelScaling   = 1.0
CmdVelOffset    = 0.0

Proportional Gain = 1.0
Integral Gain     = 0.0

I5 = 3     ; PLC program control enabled
I8 = 0     ; PLCC 0 called every sample

; Motor encoders used for velocity feedback

I8008 = $E00100     ; sum of motor 1 and 2 encoders written into
; Motor 5 velocity feedback register
i500 = 1
i503 = $3505
i504 = $3509
i508 = 96
```

```
I509 = 4          ; motor 5 velocity scaling factor
; Half of default value 96 to get average
; Of motor 1 and 2
; considering the different resolution of motor and load
i524 = $20001



OPEN PLC 0 CLEAR

; Velocity PI controller for axis 5

CmdVel = cmdVelSignal * cmdVelScaling + cmdVelOffset
VelError = cmdVel - velocity Load

If (integral Lock < 0)
  VelErrorIntegral = velErrorIntegral + velError
EndIf
DesTorque = proportional Gain * velError + integral Gain * velErrorIntegral

; Friction compensation

If (cmdVel > 0)
  DesTorque = desTorque + FRICTION Offset;
EndIf

If (cmdVel < 0)
  DesTorque = desTorque - FRICTION Offset;
EndIf

; Torque offset

If (desTorque < 0)
  torque2 = desTorque/2 - TORQUE_OFFSET;
  If (torque2 < -MAX_TORQUE/2)
    torque2 = -MAX_TORQUE/2
  EndIf
  torque1 = desTorque - torque2
Else
  torque1 = desTorque/2 + TORQUE_OFFSET;
  If (torque1 > MAX_TORQUE/2)
    torque1 = MAX_TORQUE/2
  EndIf
  torque2 = desTorque - torque1
EndIf
```

```
; active damping

torque1 = torque1 - D1 * (velocityMotor1 - velocityMotor2) - D2 * (velocityMotor1 +
velocityMotor2 - 2*velocityLoad/GR)
torque2 = torque2 + D1 * (velocityMotor1 - velocityMotor2) - D2 * (velocityMotor1 +
velocityMotor2 - 2*velocityLoad/GR)

; Saturation

Integral Lock = 0

If (torque1 > MAX_TORQUE/2)
  torque1 = MAX_TORQUE/2
  Integral Lock = 1
EndIf

If (torque1 < -MAX_TORQUE/2)
  torque1 = -MAX_TORQUE/2
  integralLock = 1
EndIf

If (torque2 > MAX_TORQUE/2)
  torque2 = MAX_TORQUE/2
  integralLock = 1
EndIf

If (torque2 < -MAX_TORQUE/2)
  torque2 = -MAX_TORQUE/2
  Integral Lock = 1
EndIf

CLOSE   ; PLC 0
```

<u>Annexure E (</u>Algorithm for pre-loading the two motors with PMAC configured as a Velocity loop with PMAC PID Filter)

Algorithm for preloading the two motors along with configuration of I variables for Channels #1, #2 #5 and #7 where motor 1, motor 2 and load encoders are connected at Ch #1,#2 and #5. Velocity loop implemented in PMAC PID. Here in velocity loop Position loop needs to be Zero, So after closing #5, the position feedback to ch #5 is zero. For gathering data from load encoder we are taking ch #7**.**
(**File Name: - AZ EL preload velocity 1_bsr.pmc**) see attached CD.

```
; Leo's antenna vel.loop upload file of C04 is modified
; By bsr/ss on 29th march for testing setup large
; 4 changed to 2 in consign of I100 to 17000 as only
; Two motors are used for large test setup
; All EL.parameters are commented
; M5063 is used for ADC i/p in ch#7
; Vel.cmd.scale factor-P9 is varied from 1 to 200
; P9 = 1 (ADC=10) test setup did not rotate, P9 = 50
; (ADC=10) rotation
; started and P9 = 200 (ADC=1) speed is high
; ADC i/p changed from 1V to 10V
; With vel.loop in software closing ch#5 loop was
; done in program but when PID firmware loop is
; used for ch#5 the loop has to be closed externally
; by giving cmd #5j/
; P9 value to be arrived after trial and error so that
; We get max.speed of 30deg/min and min speed of
; 15"/sec as done by Leo in C04
; I508 is not used for vel.loop so i509 = 96 is made
; 48 as two motor velocities are added
; ADC i/p is to be calibrated for max speed of 30deg per minute
; And min.speed goes 15"/sec. In order to do this
; We measure the load encoder vel. by making i703=$3505
; In cmd window and enabling the 7th axis and data gather
; The plot gives the velocity actual and note ADC i/p from
; Power supply.
; 21.04.2009

#include "Tp2mvar.pmc"
Close
Endg
Del gat

i7016, 2, 10=1                    ; true DAC Output
```

```
i100, 2,100=1                         ; activate axis
i122, 2,100=15                              ; slow default speed
i119, 2,100=0.2                             ; higher acc + dec


i123, 2,100=5                         ; homing speed
i124, 2,100=$20001                     ; no limit switches

i130, 2,100=120000                    ; PID Settings- Proportional Gain
i131, 2,100=1050                      ; Derivative Gain
i132, 2,100=1050                      ; Velocity feed Forward Gain
i133, 2,100=10000                     ; Integral Gain
i134, 2,100=1                         ; Integral Mode
i169, 2,100=16384                     ; 10V diffential DAC Output
i700=1                                   ; #7 Channel enable
i703=$3505                               ; giving position feedback to channel #7
i711=1, 00,000                            ; fatal following error
i724= $020001                            ; no limit switches.

; AZ load encoder setting (connected to ENC5 input)
i7110=7                                     ; changing counting direction
; load encoder at antenna needs
; That orientation (17.04.2009)

; #3 encoder is EL1 encoder and needs an opposite counting
; Direction since the motor is mounted in the opposite orientation
; Output of the BLC is hardwired in the opposite way (Servo star input)
; i7030=3
; EL load encoder setting (connected to ENC6 input)
; i7120=3

; Definitions for the analog input reading
I7106 = $1FFFFF                             ; ADC strobe word
M5063->Y:$78115,8,16,s                      ; ch7 A-D channel
; M5064->Y: $7811D, 8,16,s                  ; ch8 A-D channel

; ---------------------------------------------------------
; PLCC0 real-time task for torque offset and active damping

; Standard AZ position/speed control loop at axis 5
; Control output AZ of axis 5 distributed to axes 1 and 2
; adding a torque offset
; Axes 1 and 2 must be activated via command o0
; When killing axes 1 and 2, torque Offset must be reset to 0

; Standard EL position/speed control loop at axis 6
; Control output EL of axis 6 distributed to axes 3 and 4
```

```
; adding a torque offset
; Axes 3 and 4 must be activated via command o0
; When killing axes 3 and 4, torque offset must be reset to 0

#define AZvelocityLoad          M574    ; filtered (unfiltered is M566)
#define AZvelocityMotor1        M174    ; filtered (unfiltered is M166)
#define AZvelocityMotor2        M274    ; filtered (unfiltered is M266)


#define AZtorque1               M179
#define AZtorque2               M279
#define AZdesTorque             M568
#define AZcmdPos                M561
#define AZactPos                M562


/*
#define ELvelocityLoad          M674    ; filtered (unfiltered is M666)
#define ELvelocityMotor1        M374    ; filtered (unfiltered is M366)
#define ELvelocityMotor2        M474    ; filtered (unfiltered is M466)
#define ELtorque1               M379
#define ELtorque2               M479
#define ELdesTorque             M668
#define ELcmdPos                M661
#define ELactPos                M662
*/
#define AZposError              P1      ; position control deviation
#define AZFRICTION_Offset       P2
#define AZD1                    P3      ; damping coefficient 1
#define AZD2                    P5      ; damping coefficient 2
#define AZGR                    P6      ; gear ratio
#define AZMAX_TORQUE            P7      ; Nm scaled to 16 bit integer
#define AZTORQUE_OFFSET         P8      ; Nm scaled to 16 bit integer
#define AZcmdVelScaling         P9
/*
#define ELposError              P11     ; position control deviation
#define ELFRICTION_Offset       P12     ;
#define ELD1                    P13     ; damping coefficient 1
#define ELD2                    P15     ; damping coefficient 2
#define ELGR                    P16     ; gear ratio
#define ELMAX_TORQUE            P17     ; Nm scaled to 16 bit integer
#define ELTORQUE_OFFSET         P18     ; Nm scaled to 16 bit integer
#define ELcmdVelScaling         P19
*/
AZD1                    = 0
AZD2                    = 0
AZGR                    = 8.64257
```

```
AZMAX_TORQUE              = 32768
AZTORQUE_OFFSET                  = 1000
AZcmdVelScaling            = 50
/*
ELD1                       = 0
ELD2                       = 0
ELGR                       = 8.64257
ELMAX_TORQUE                     = 32768
ELTORQUE_OFFSET                  = 1000
ELcmdVelScaling           = 200



*/
I5 = 3                    ; PLC program control enabled
I8 = 0                    ; PLCC 0 called every sample

; Motor encoders used for velocity feedback
I8008   = $E00100     ; sum of motor 1 and 2 encoders written into
                      ; Motor 5 velocity feedback register
; I8009 = $E00302     ; sum of motor 3 and 4 encoders written into
                     ; Motor 6 velocity feedback register

i500 = 1
i503 = $350B
i504 = $3509
i506 = 1                         ; enable master encoder (hand wheel) in order
                                   ; For setting the desired vel input signal via
; m567 (scaled by 1/(32*i507)
i507 = 96
i508 = 96
I509 = 48                                 ; motor 5 velocity scaling factor
                                   ; set to half of the scaling of i508 since
                                   ; Summation of to input = resolver are considered
i530 = 5000000                          ; PID Settings
i531 = 128
i532 = 0
i533 = 0
i534 = 1
i538 = 0
i539 = 0
i519 = 0.0002
i522 = 1
i523 = 1
i524 = $20001
i568 = 0                         ; Friction FF term
i511 = 0
```

```
i512 = 0                            ; setting the error limits to zero in order to
                                    ; avoid any influence to the velocity loop
/*
i600 = 1
i603 = $350B
i604 = $350A
i606 = 1                            ; enable master encoder (hand wheel) in order
                                    ; For setting the desired vel input signal via
                                    ; m667 (scaled by 1/(32*i607)
i607 = 96
i608 = 96
I609 = 48                           ; motor 6 velocity scaling factor
                                    ; see scaling calculation of 21.04.09
                                    ; Of motor 3 and 4
                                    ; considering the different resolution of motor and
load


i630 = 1000000                      ; PID Settings
i631 = 128
i632 = 0
i633 = 0
i634 = 1
i638 = 0
i639 = 0
i619 = 0.0002
i622 = 1
i623 = 1
i624 = $20001
i668 = 0                            ; Friction FF term (not checked yet)
i611 = 0                            ; disable following Error
i612 = 0
*/
OPEN PLC 0 CLEAR
;
============================================================
; AZ PART
;
============================================================
; Analog input reading and scaling
m567 = 96 * 32 * m5063 * AZcmdVelScaling / 32767

; Friction compensation
/*
AZposError = AZcmdPos - AZactPos;

If (AZposError > 0)
```

```
  AZdesTorque = AZdesTorque + AZFRICTION_Offset;
EndIf

If (AZposError < 0)
  AZdesTorque = AZdesTorque - AZFRICTION_Offset;
EndIf
*/
; torque offset

If (AZdesTorque < 0)
  AZtorque2 = AZdesTorque/2 - AZTORQUE_OFFSET;
  If (AZtorque2 < -AZMAX_TORQUE/2)
    AZtorque2 = -AZMAX_TORQUE/2
  EndIf
  AZtorque1 = AZdesTorque - AZtorque2
Else
  AZtorque1 = AZdesTorque/2 + AZTORQUE_OFFSET;
  If (AZtorque1 > AZMAX_TORQUE/2)
    AZtorque1 = AZMAX_TORQUE/2
  EndIf
  AZtorque2 = AZdesTorque - AZtorque1
EndIf


; active damping
/*
; Remark: consider the filtered velocity needs to be checked, because of the steps
in the
signal!!!
AZtorque1 = AZtorque1 - AZD1 * (AZvelocityMotor1 - AZvelocityMotor2) -
AZD2 *
(AZvelocityMotor1 + AZvelocityMotor2 - 2*AZvelocityLoad/AZGR)
AZtorque2 = AZtorque2 + AZD1 * (AZvelocityMotor1 - AZvelocityMotor2) -
AZD2 *
(AZvelocityMotor1 + AZvelocityMotor2 - 2*AZvelocityLoad/AZGR)
*/
; saturation

If (AZtorque1 > AZMAX_TORQUE/2)
  AZtorque1 = AZMAX_TORQUE/2
EndIf

If (AZtorque1 < -AZMAX_TORQUE/2)
  AZtorque1 = -AZMAX_TORQUE/2
EndIf

If (AZtorque2 > AZMAX_TORQUE/2)
```

```
  AZtorque2 = AZMAX_TORQUE/2
EndIf

If (AZtorque2 < -AZMAX_TORQUE/2)
  AZtorque2 = -AZMAX_TORQUE/2
EndIf

/*
;
============================================================
; EL PART
;
============================================================
; analog input reading and scaling
m667 = 96 * 32 * m5064 * ELcmdVelScaling / 32767

; friction compensation

/*
ELposError = ELcmdPos - ELactPos;

If (ELposError > 0)
  ELdesTorque = ELdesTorque + ELFRICTION_Offset;
EndIf

If (ELposError < 0)
  ELdesTorque = ELdesTorque - ELFRICTION_Offset;
EndIf
*/

; torque offset
/*
If (ELdesTorque < 0)
  ELtorque2 = ELdesTorque/2 - ELTORQUE_OFFSET;
  If (ELtorque2 < -ELMAX_TORQUE/2)
    ELtorque2 = -ELMAX_TORQUE/2
  EndIf
  ELtorque1 = ELdesTorque - ELtorque2
Else
  ELtorque1 = ELdesTorque/2 + ELTORQUE_OFFSET;
 If (ELtorque1 > ELMAX_TORQUE/2)
    ELtorque1 = ELMAX_TORQUE/2
  EndIf
  ELtorque2 = ELdesTorque - ELtorque1
EndIf
*/
```

```
; active damping
/*
 ; Remark: consider the filtered velocity needs to be checked, because of the steps
in the
    signal!!!
ELtorque1 = ELtorque1 - ELD1 * (ELvelocityMotor1 - ELvelocityMotor2) -
ELD2 *
(ELvelocityMotor1 + ELvelocityMotor2 - 2*ELvelocityLoad/ELGR)
ELtorque2 = ELtorque2 + ELD1 * (ELvelocityMotor1 - ELvelocityMotor2) -
ELD2 *
(ELvelocityMotor1 + ELvelocityMotor2 - 2*ELvelocityLoad/ELGR)
*/

; saturation
/*
If (ELtorque1 > ELMAX_TORQUE/2)
  ELtorque1 = ELMAX_TORQUE/2
EndIf

If (ELtorque1 < -ELMAX_TORQUE/2)
  ELtorque1 = -ELMAX_TORQUE/2
EndIf

If (ELtorque2 > ELMAX_TORQUE/2)
  ELtorque2 = ELMAX_TORQUE/2
EndIf

If (ELtorque2 < -ELMAX_TORQUE/2)
  ELtorque2 = -ELMAX_TORQUE/2
EndIf
*/
CLOSE   ; PLC 0
```

Annexure-F

Converting Counts in arcsec in motor end and Encoder end:-

Motor end:-

Resolution of the motor is 8192 counts	
1 rotation of motor	= 8192 counts
360 degree	= 8192 counts
1 degree	= 8192/360 counts
	= 22.7 counts
1 arcmin	= 0.37 counts (22.7/60)
1 arcsec	**= 0.0063 counts (0.0063/60)**

Encoder End:-

Resolution of the encoder is 8192*400 counts (400 when using interpolator)	= 3276800 counts
1 rotation of encoder	= 3276800 counts
360 degree	= 3276800 counts
1 degree	= 3276800/360 counts
	= 9102.2 counts
1 arcmin	= 151.70 counts (9102.2/60)
1 arcsec	**= 2.5 counts (151.70/60)**

Note:- 1 arcsec = 2.5 counts
2.5 counts = 1 arcsec
1 count = 0.4 arcsec.

Pending Issues:-

1. Resonant frequency of the Brushless Motor (Large Test setup) with PMAC configured in Velocity Loop by using Sine Sweep profile.

2. Interfacing Absolute Encoder with PMAC without interpolator.