Implementation of Conventional and Neural Controllers Using Position and Velocity Feedback

Progress Report 1

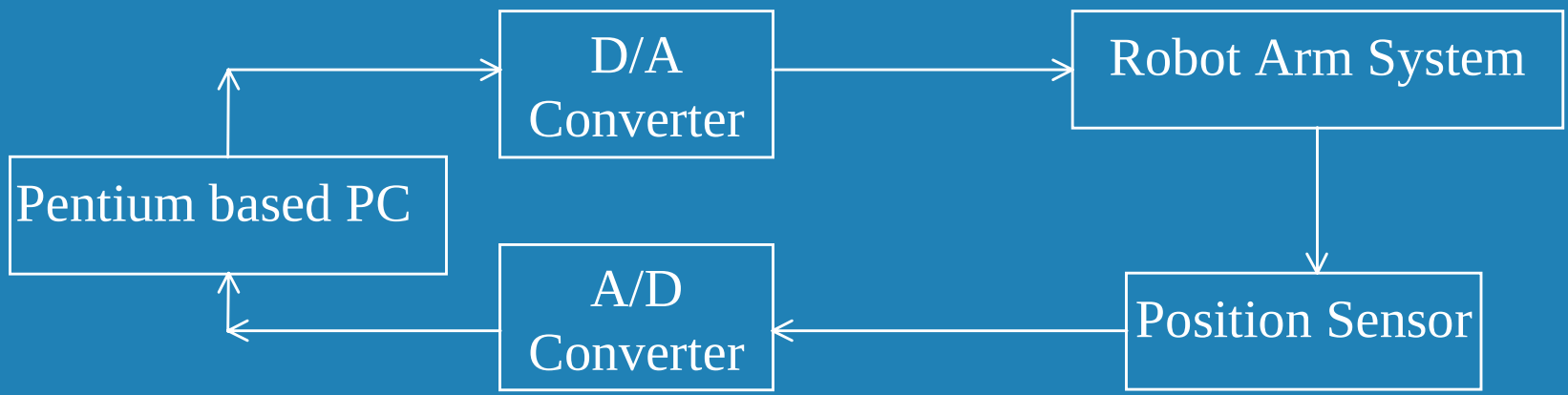
By:
Chris Spevacek and Manfred Meissner

Advisor:
Dr. Gary Dempsey

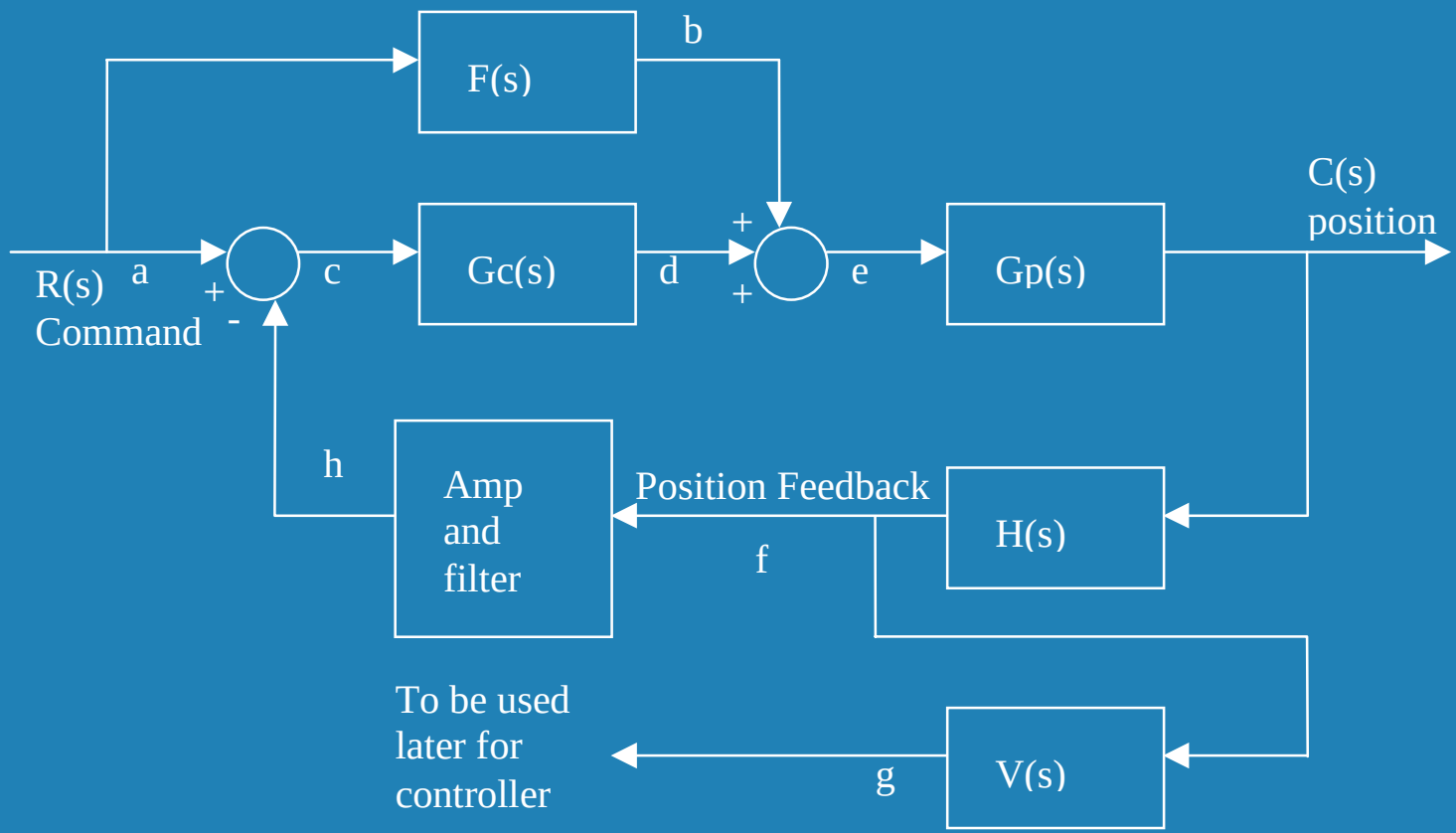
Summary

- Design and Compare Conventional and Neural Controllers for a Small Robot Arm
- Position and Velocity Feedback Design
- User Friendly Interface Design

High Level Block Diagram



Control Block Diagram



Progress

- System Identification
- Help Menu

System Identification

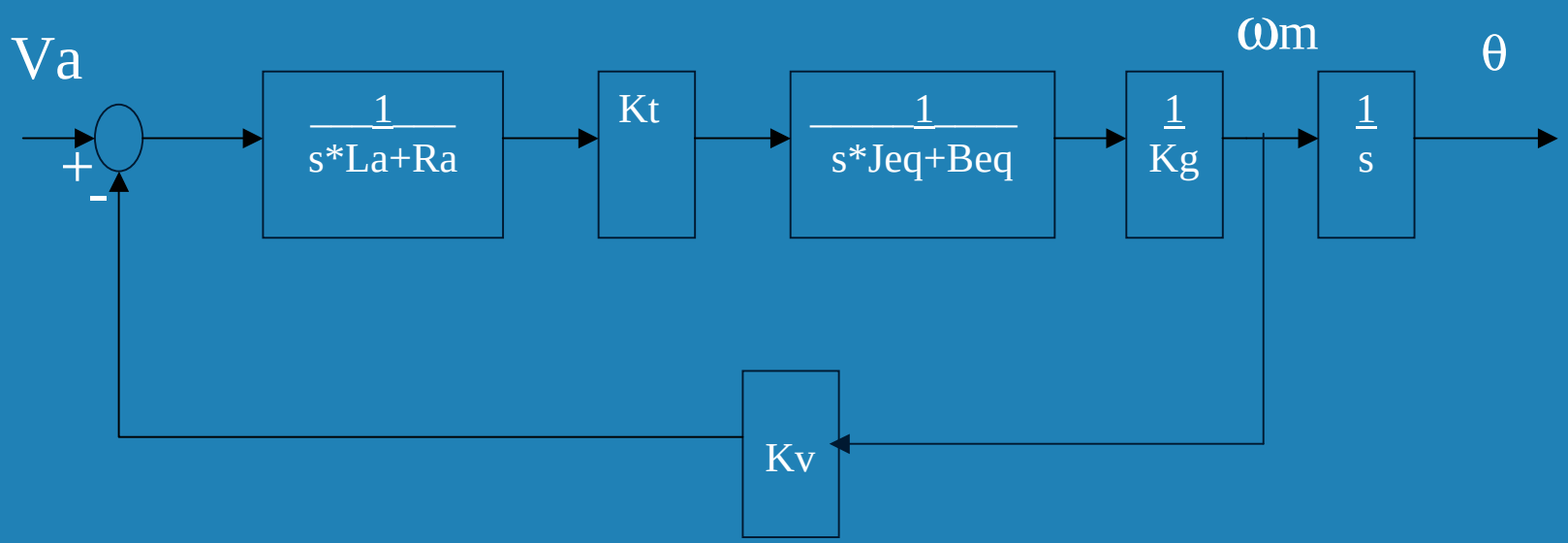
System

- Level arm with light load

Objective:

- Find System Model from Theory
- Derive S-Plane
- Find Plant Model

Robot Arm System Block Diagram

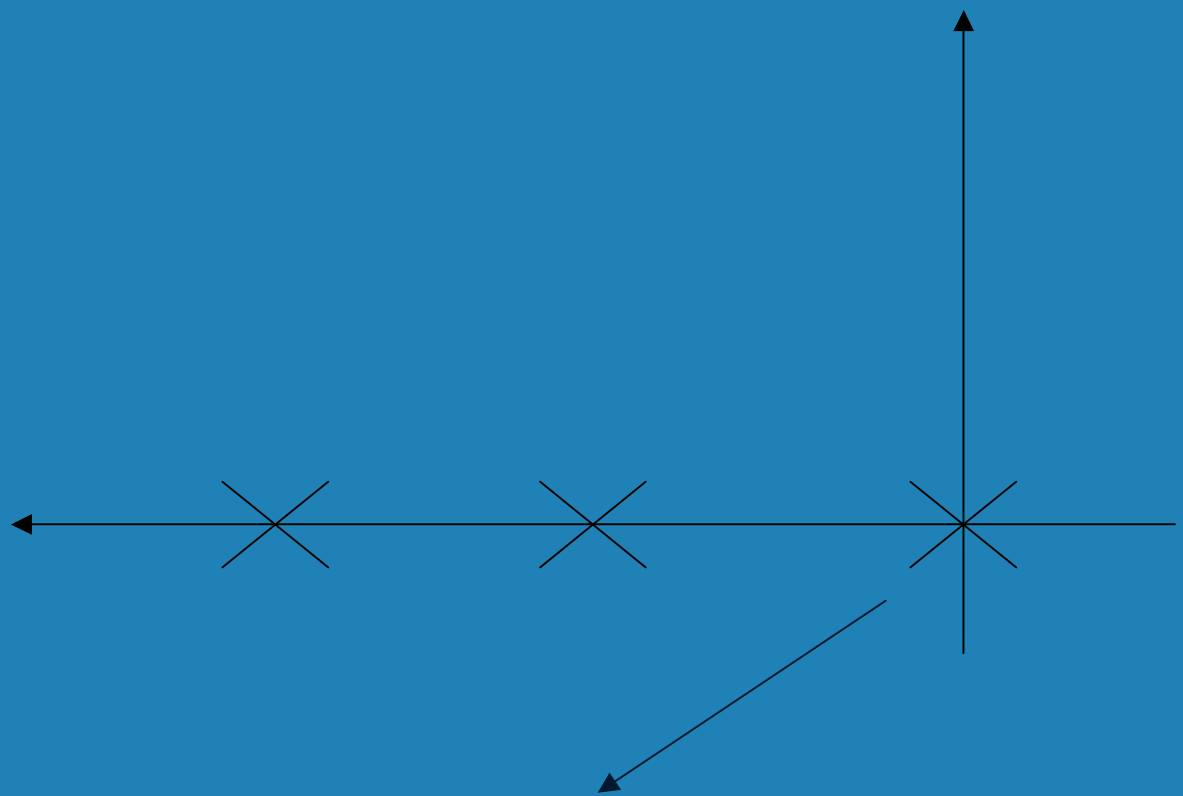


Ra=2.6Ohms La = 0.18mH Gives a pole at 340Hz

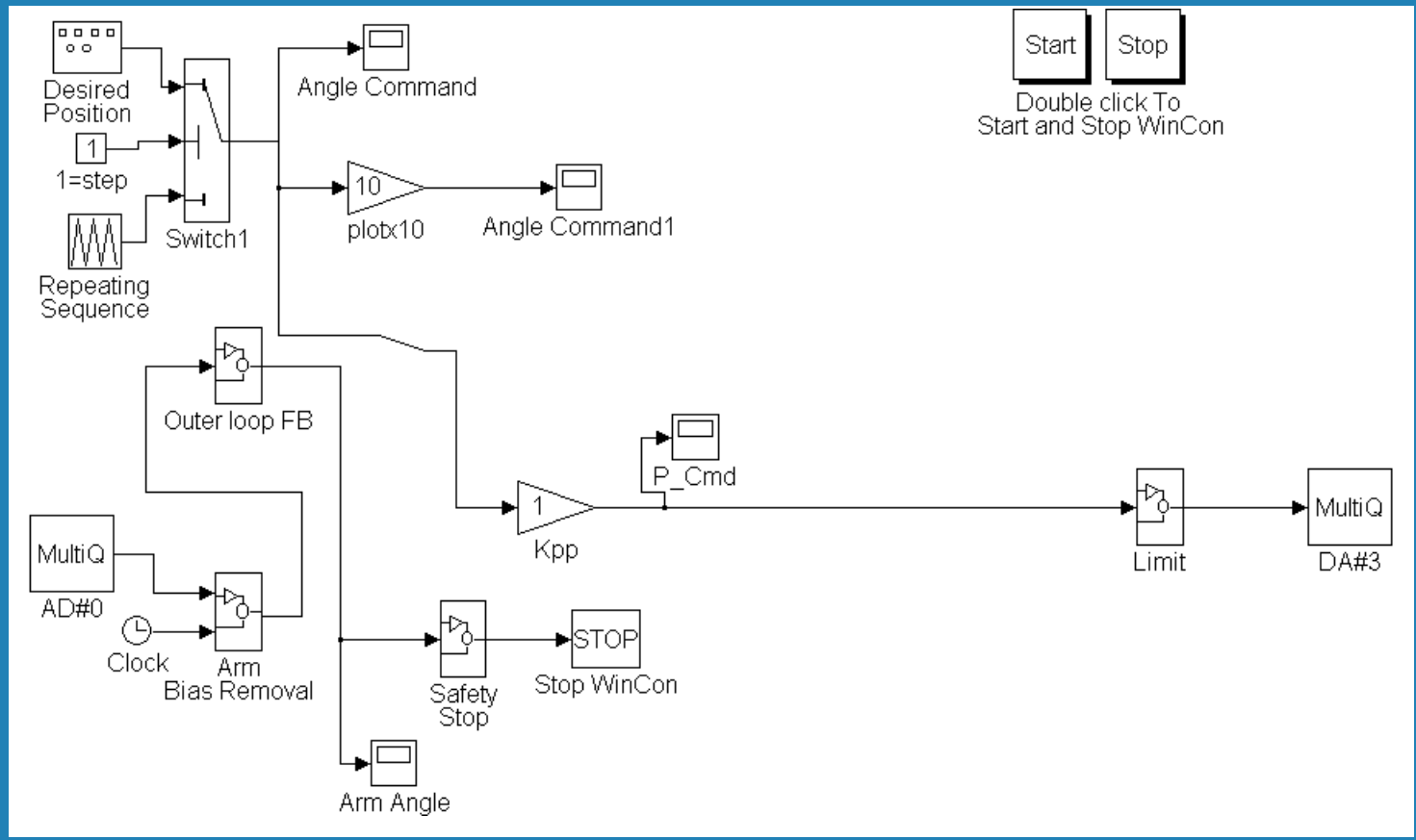
Kt=0.00767Nm/amp Kv=0.00767V/rad/sec Kg = 1/70

Power Amp pole at 60kHz

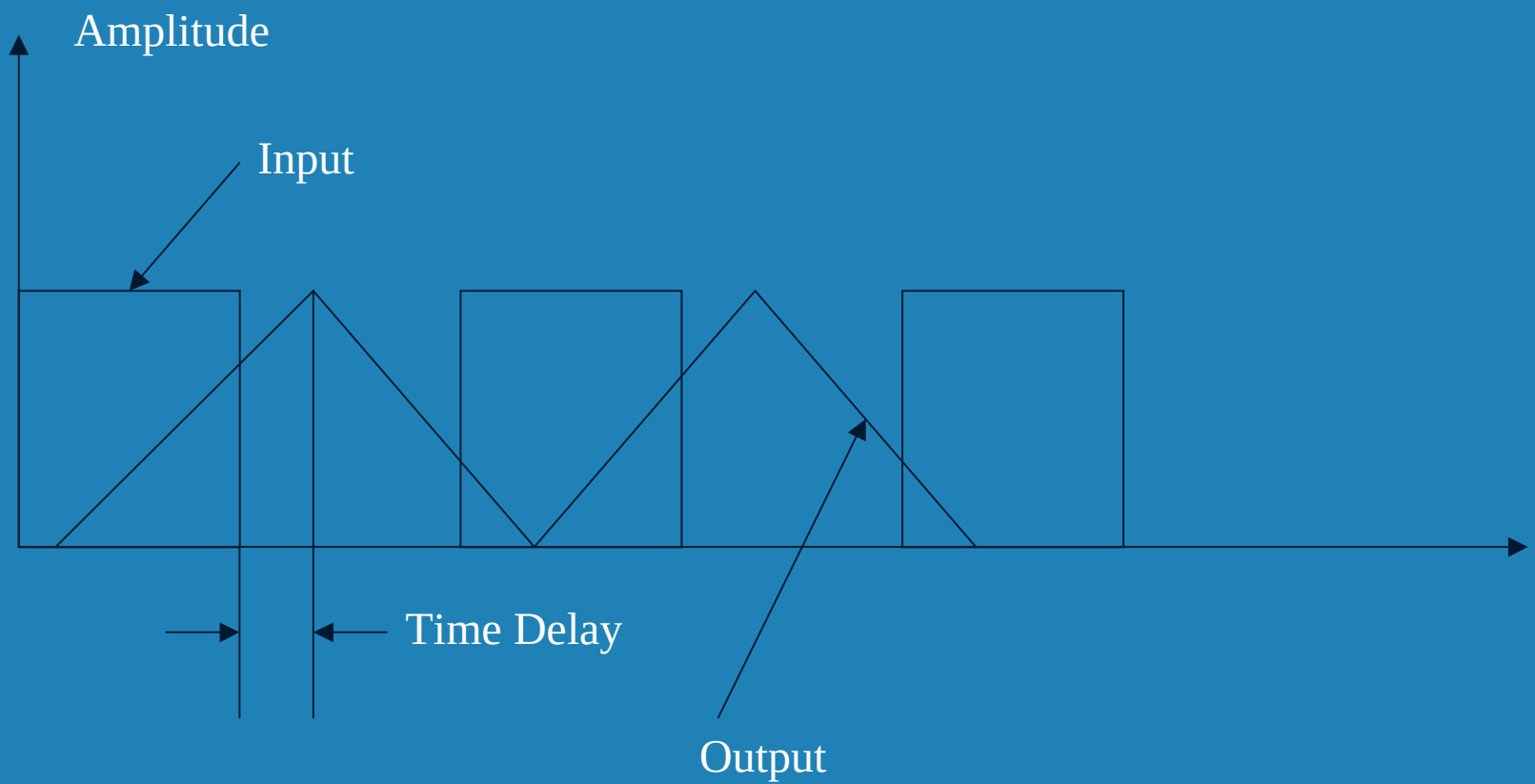
S-Plane of System



Open Loop WinCon



Time Delay Measurement

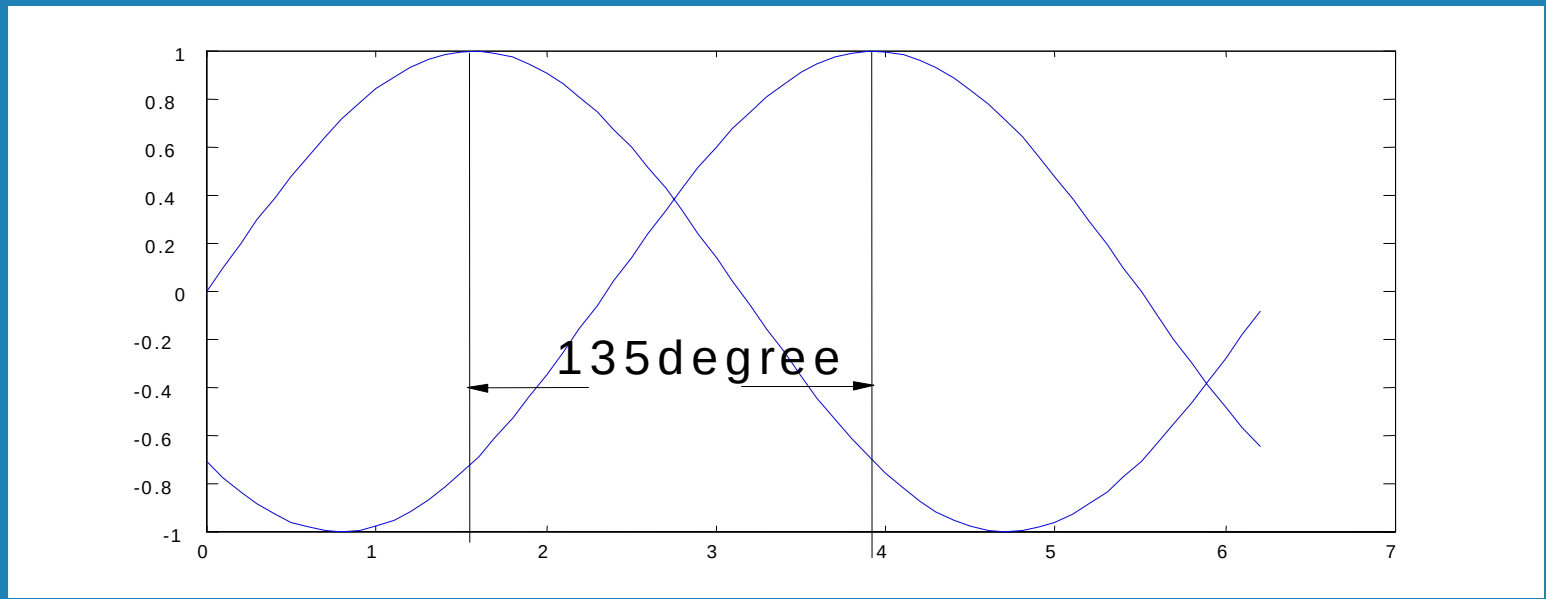


Time Delay

- 25ms counterclockwise
- 15ms clockwise
- Design worst case

Second Pole Search

- At 2nd pole phase lag should be 135º
- 135º=-90º-arctan(ω/pole)- (ω*Td*180/π)

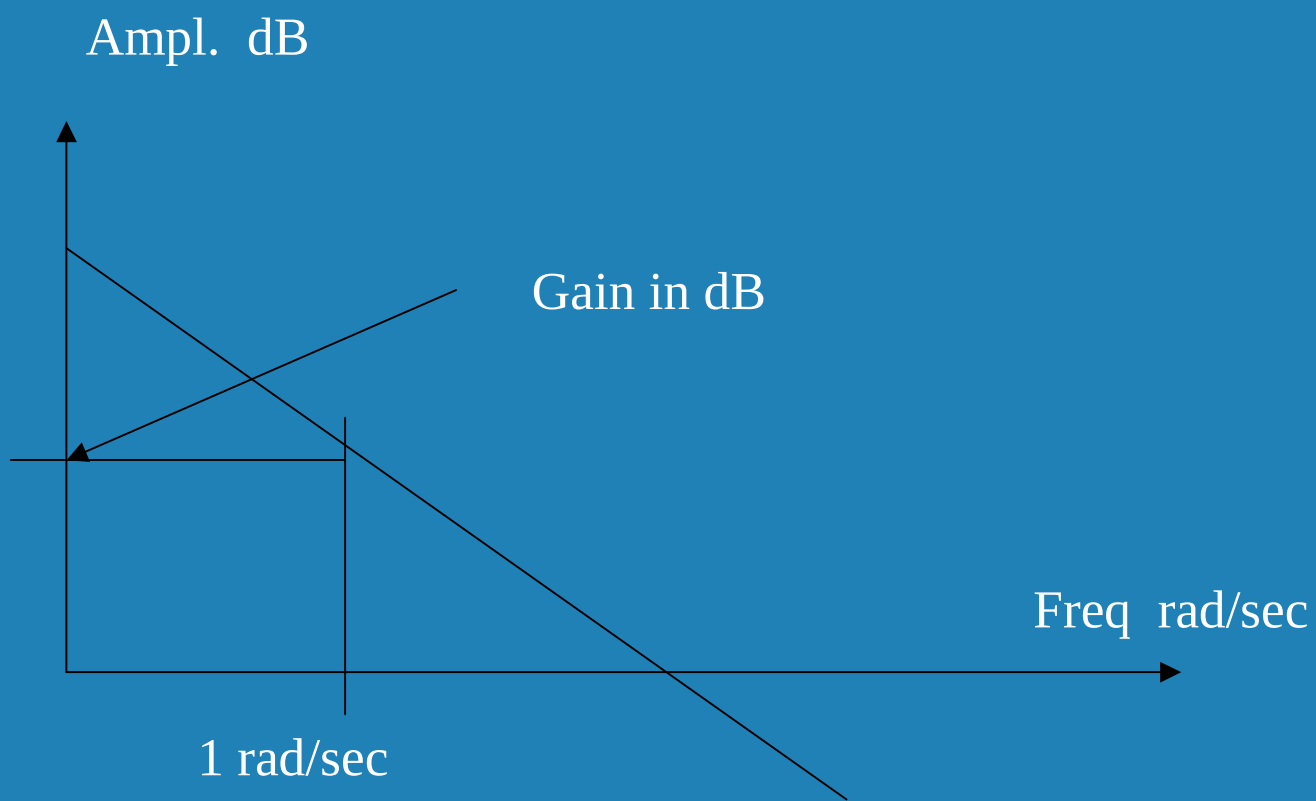


Problem

- Td too big
- ω*Ts*180/π term =45 degree

  $135^{o}=-90^{o}-\arctan(\omega/\text{pole})-(\omega*Td*180/\pi)$
- Pole not verified

K value



Conclusion

$$Gp = \frac{K e^{-std}}{s}$$

With K=Vout/Vin=0.977
and td=25ms

Verifying Result

- Using Simulink to simulate plant
- %Overshoot of experiment did not agree with simulation
- Switch to 2nd order system

$$Gp = \frac{K e^{-std}}{s(s/p + 1)}$$

Measurement

Gain k	1	0.8	0.33	0.2
% Overshoot	44%	41%	10%	0%

Approaches

- MatLab match
  - 2 unknowns pole and K
  - try to match
- Experimental finding gain K
  - use MatLab to match up pole

Velocity Approach

- Time revolutions at different voltages
- Divide velocity by input voltages
- Adjust to average K
- MatLab to match pole

Voltage	Time/Revolution	Gain
4 Volts	12.5 sec	7.2
3 Volts	12.8 sec	9.375
2 Volts	13.0 sec	13.84

Results

Matlab Match $$Gp = \frac{6.9 * e^{-s*0.025}}{s(s/3.2+1)}$$

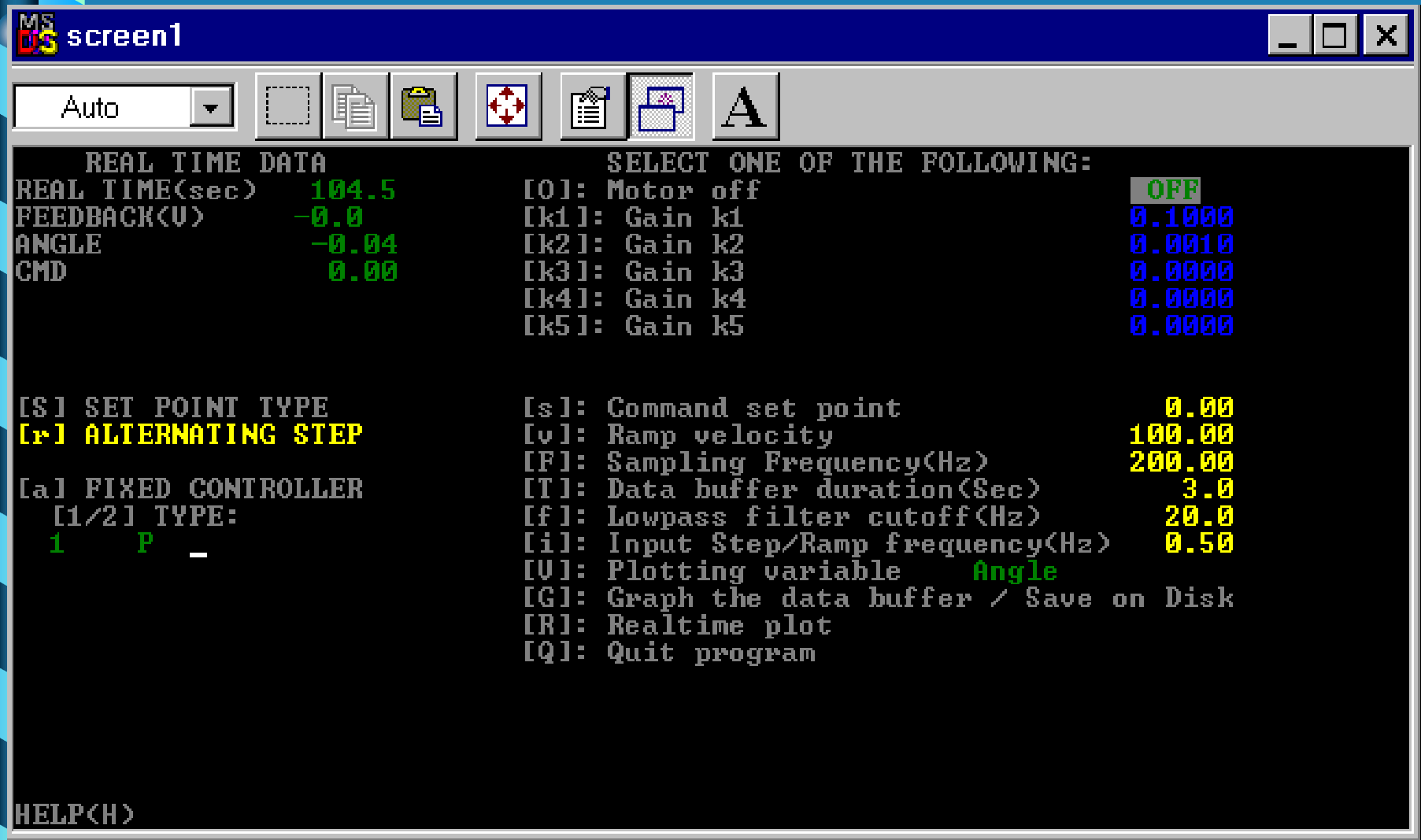
K Measurement $$Gp = \frac{10.1 * e^{-s*0.025}}{s(s/2+1)}$$

Gain k	1	0.75	0.33	0.2
% Overshoot Experiment	44%	41%	10%	0%
% Overshoot Matlab	44%	36%	13%	3%
% Overshoot K-Measurement	43%	35%	15%	5%

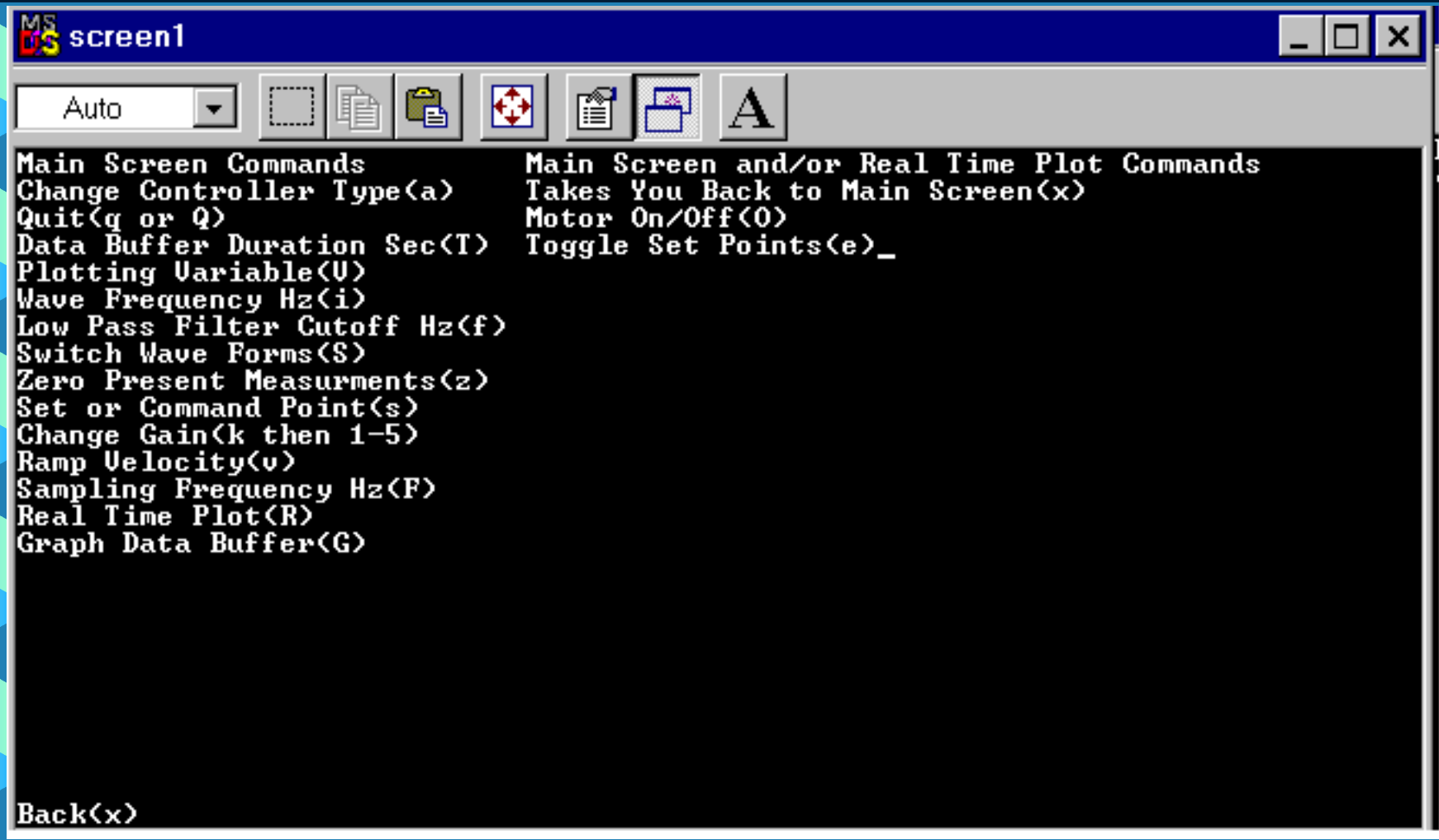
Help Menu

- Some commands are not verified on main screen
- Needed extra space
- Command line added to real time plots

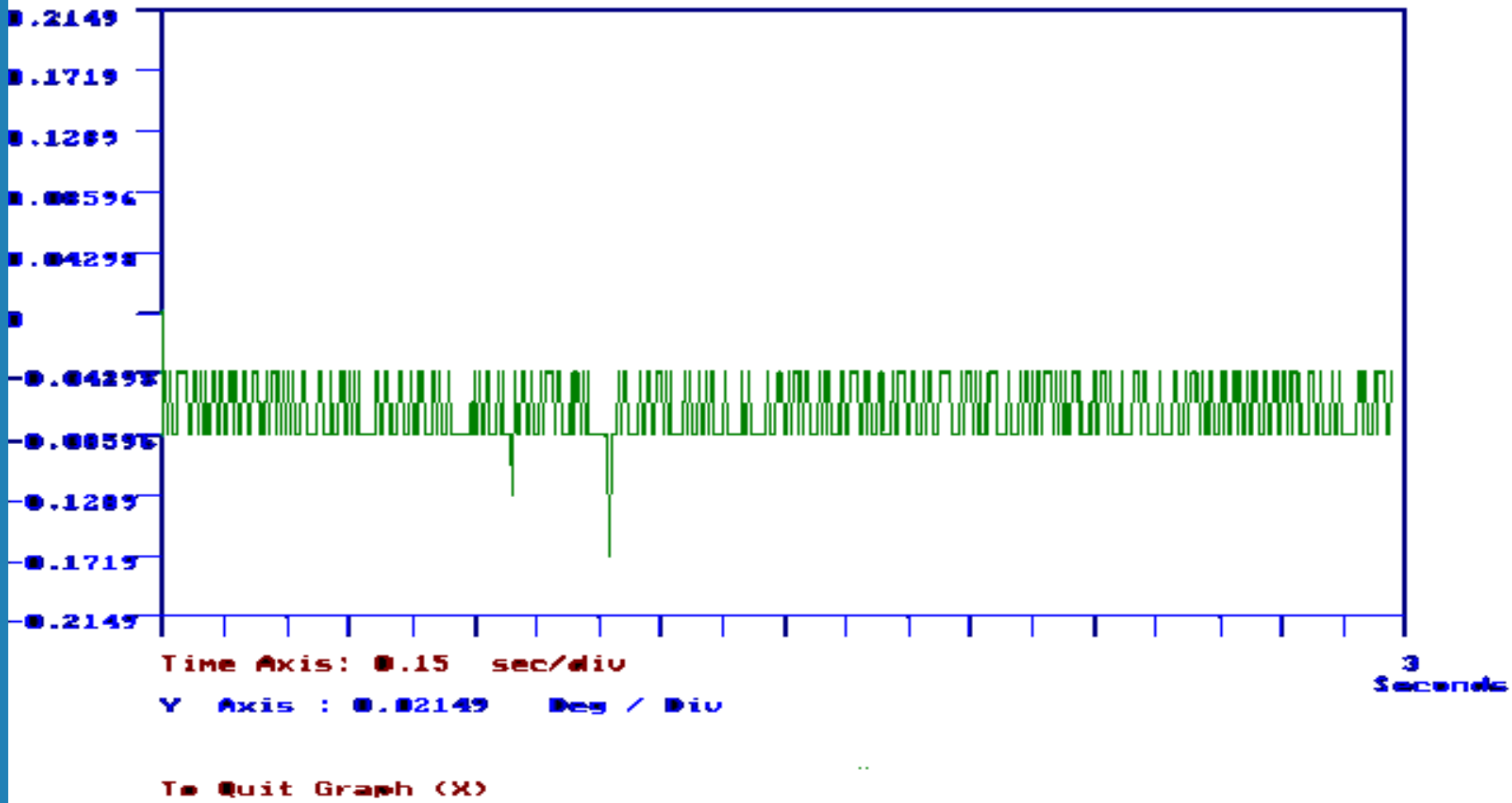
Main Screen


screen1
Auto
REAL TIME DATA
REAL TIME(sec) 104.5
FEEDBACK(V) -0.0
ANGLE -0.04
CMD 0.00
SELECT ONE OF THE FOLLOWING:
[O]: Motor off OFF
[k1]: Gain k1 0.1000
[k2]: Gain k2 0.0010
[k3]: Gain k3 0.0000
[k4]: Gain k4 0.0000
[k5]: Gain k5 0.0000
[S] SET POINT TYPE
[r] ALTERNATING STEP
[a] FIXED CONTROLLER
[1/2] TYPE:
1 P
[s]: Command set point 0.00
[v]: Ramp velocity 100.00
[F]: Sampling Frequency(Hz) 200.00
[T]: Data buffer duration(Sec) 3.0
[f]: Lowpass filter cutoff(Hz) 20.0
[i]: Input Step/Ramp frequency(Hz) 0.50
[V]: Plotting variable Angle
[G]: Graph the data buffer / Save on Disk
[R]: Realtime plot
[Q]: Quit program
HELP(H)

Help Screen


screen1
Auto
Main Screen Commands
Change Controller Type(a)
Quit(q or Q)
Data Buffer Duration Sec(T)
Plotting Variable(V)
Wave Frequency Hz(i)
Low Pass Filter Cutoff Hz(f)
Switch Wave Forms(S)
Zero Present Measurments(z)
Set or Command Point(s)
Change Gain(k then 1-5)
Ramp Velocity(v)
Sampling Frequency Hz(F)
Real Time Plot(R)
Graph Data Buffer(G)
Main Screen and/or Real Time Plot Commands
Takes You Back to Main Screen(x)
Motor On/Off(O)
Toggle Set Points(e)_
Back(x)

Real Time Plot



Revised Schedule

Subproject	Time in Weeks	Progress
System Identificatio	3	0.5 weeks left
Menu	1	Done
P-Controller Design and Testing	1	0.5 weeks left
Investigate & Implement Neural Networks with P-controller	2	Not Started
Velocity Algorithm	2	Not Started
Two Loop Design With Neural Networks	1	Not Started
Feed-Forward Control & Impementation in Neural Networks	1	Not Started
Digital Control Analysis	1	Not Started

The
End