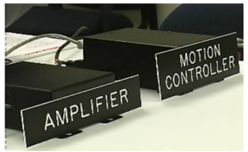
Tuning Servo Systems: Basic Techniques

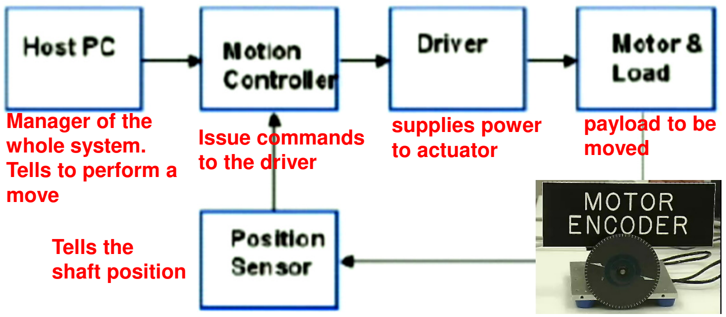
Prof. Rohan Munasinghe
Department of Electronic and Telecommunication Engineering
Faculty of Engineering
University of Moratuwa 10400

Overview

- Description of Elements
- System Compensation
- Autotuning using SDK (Servo Development Kit)
- SDK Commands
- Programming Servo Motion
  - Point-to-point
  - Linear and circular
- Stored Program
- Motion Diagnostics

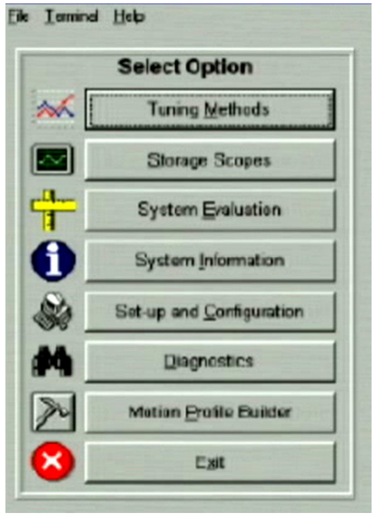
Closed Loop Motion Control System





Closed loop motion control systems ⇒ Servo systems
Some motion control systems are open loop (step motor systems)
- There's no feedback loop, no tuning involved

Servo Development Kit



Software tool to aid in tuning and analysis of servo systems

Communication Formats
PCI, PC/104, VME
RS232, USB, Ethernet

System Compensation with PID

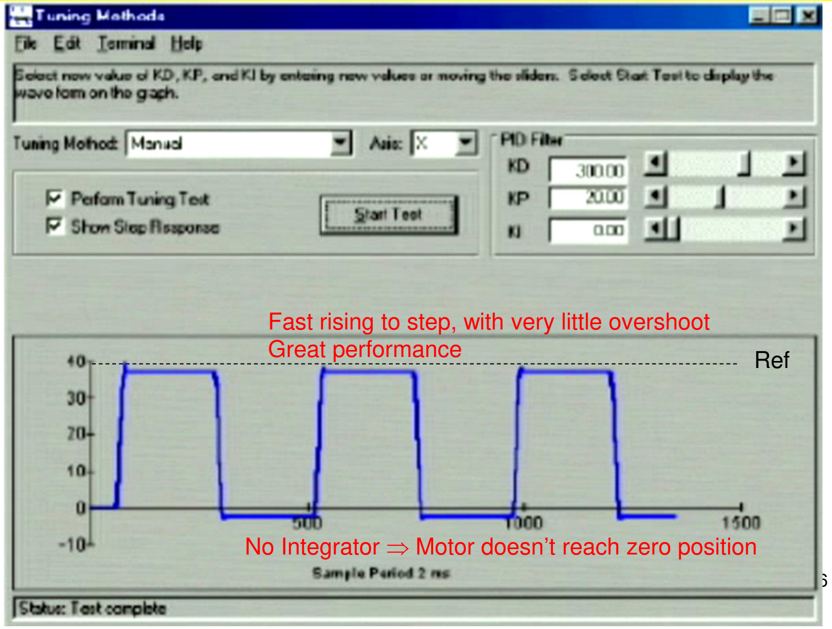
Servo systems must be "tuned" for stable performance

Motion controller provides PID compensation where P I and D gains are adjusted for best performance

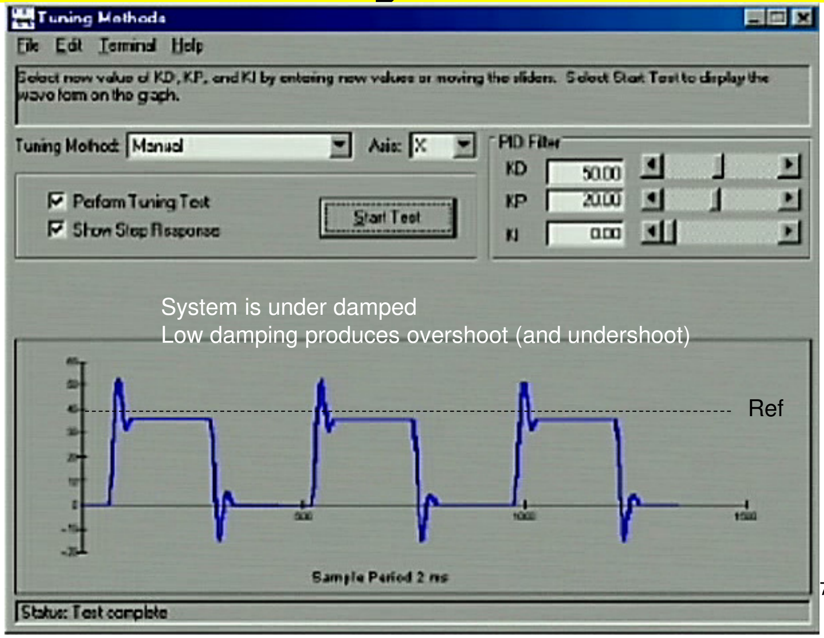
P – Proportional – for stability
I – Integral – for Accuracy
D – Derivative / Damping – for Stability

SDK helps with system tuning, view step response to see the effect of PID gain adjustments

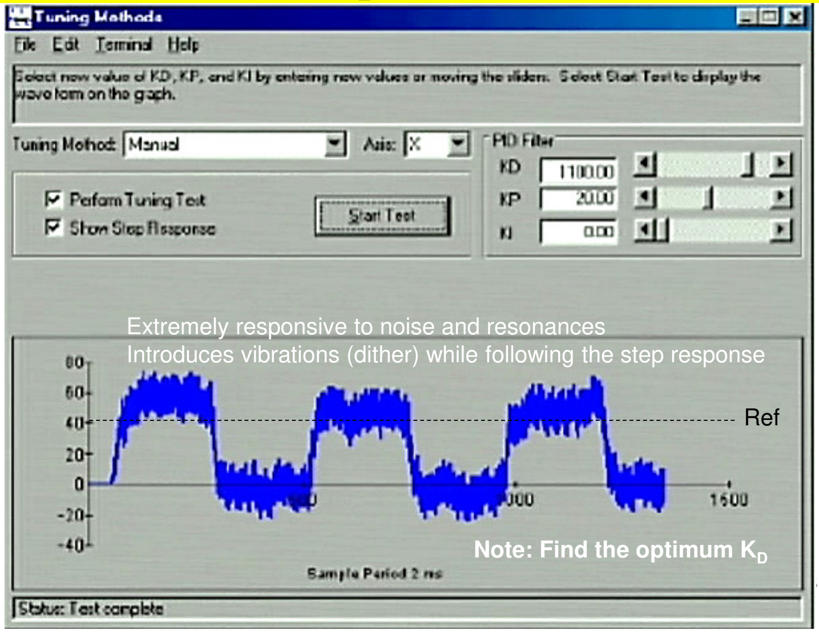
Step Response



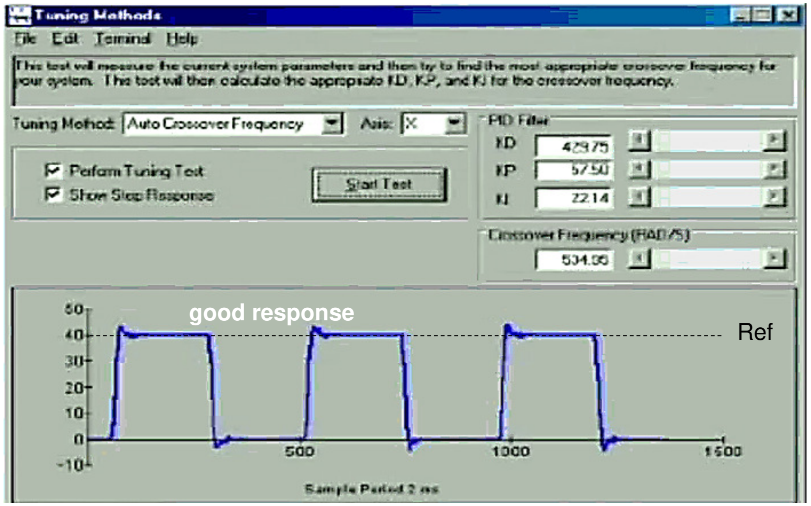
When K_D is too Low



When K_D is too High

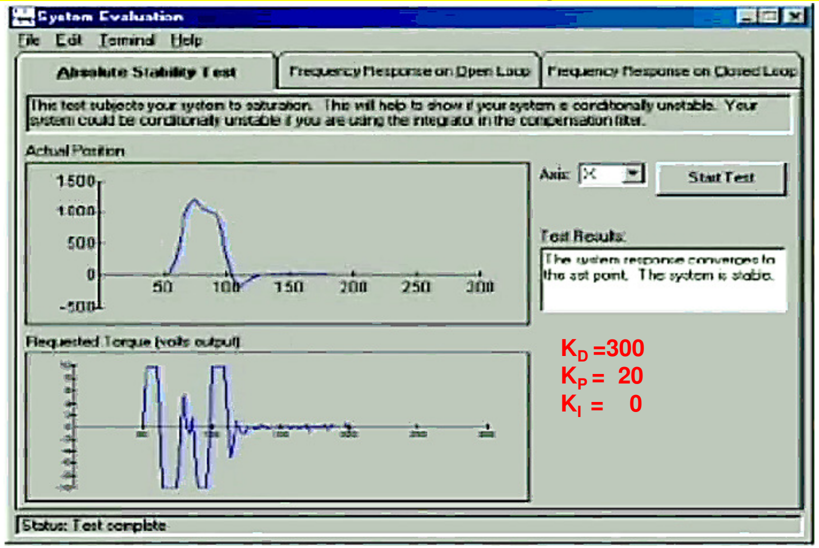


Auto-tuning Crossover Frequency



SDK applies driving signal to the motor and watching over the response of the motor the system parameters and best controller parameters are determined.

Absolute Stability Test



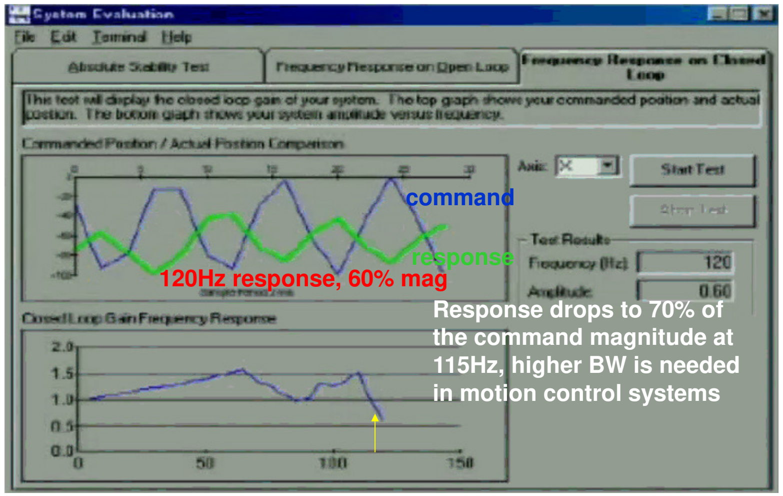
$K_D = 300$
$K_P = 20$
$K_I = 0$

Deliberately introduce A MAJOR DISTURBANCE, which saturates the amplifier, and check and make sure that the position is still stable.

Frequency Response

- Motor is driven with a sinusoidal position reference of increasing frequency.
  - For slow variations motor follows the reference accurately
  - However, motor finds it difficult to follow the reference at high frequencies. (System attenuates high frequencies)
- Reference and response are displayed on the screen together with response vs. frequency plot

Frequency Response



- **System is able to cope up to 115Hz.**
- **At 120Hz, the response has dropped to 60% of the reference amplitude.**
- **System bandwidth is the frequency at which the response drops to 70% of the reference amplitude.**
- **Motion control systems have 20Hz<BW<70Hz BW with the rated load.**
- **Under no load condition the system has more bandwidth.**

Motion Commands

- Motion Commands
  - BG: Begin motion
  - PR: Position relative
  - SP: Speed
- Interrogation Commands
  - TP: Tell position
  - TE: Tell error
  - TT: Tell torque

Point-to-Point Motion

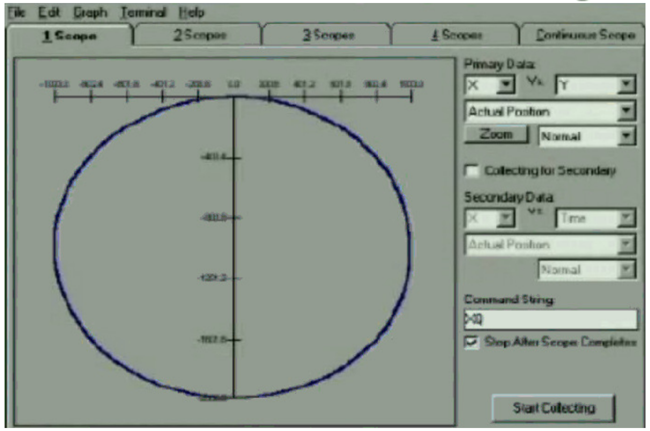
PR 4000	*Sets distance at 4000 counts*
SP 100000	*Sets slew speed at 100000 c/s*
AC 500000	*Sets acceleration*
DC 500000	*Sets deceleration*
BG X	*Begins motion of X-axis*

Speed and acceleration can be changed during motion

Trajectory Tracking using two motors

Traverse a circle in X-Y plane

VM XY	*Vector mode XY*
VS 5000	*Set vector speed*
VA 1000000	*Set vector acceleration*
VD 1000000	*Set vector deceleration*
CR 1000, 0, 360	*Specify circular move* *Radius = 1000* *Starting angle = 0* *Travel angle = 360 degrees*
VE	*End segment*
BGS	*Begin motion*



Repetitive Motions

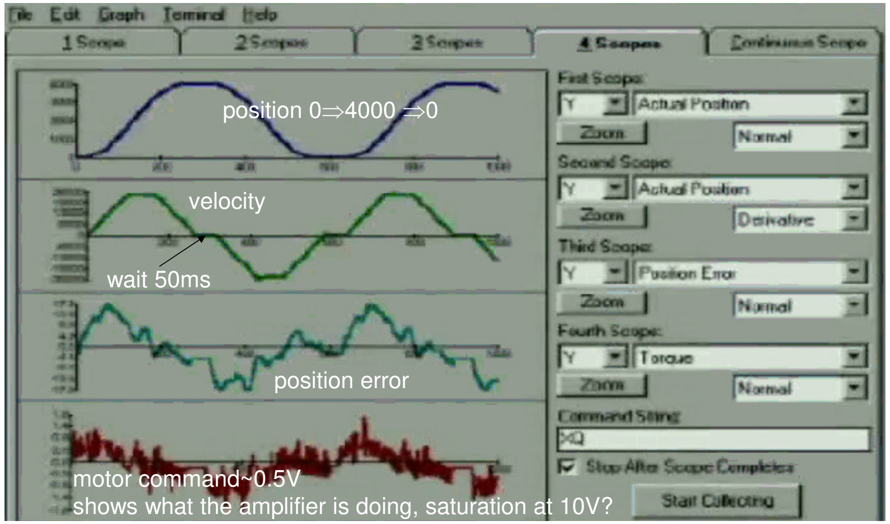
Repeated Step Program

#STEP	*Label for main*
#LOOP	*Label for loop*
PR 4000:BG X	*Position relative 4000 and begin*
AMX:WT 50	*Wait for motion done & 50msec*
PR -4000:BG X	*Repeat in reverse*
AMX:WT 50	*Wait for motion done & 50msec*
JP # LOOP	*Repeat cycle*
EN	*End program*
XQ #STEP	*Execute program*

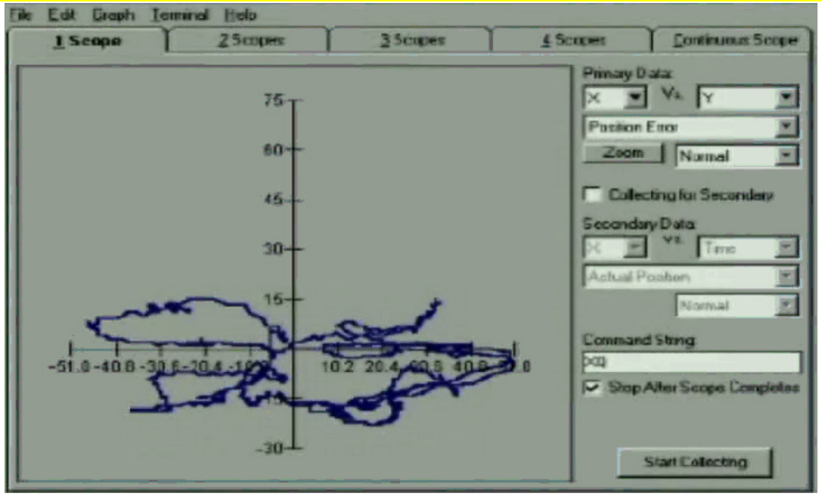
Program is downloaded from the host PC to the controller and is stored, and executed from there

Diagnostics

- Use WSDK software to capture actual motion data while motor is moving
- View step response and velocity profile
- View position error and torque to adjust PID parameters and maximize speed and acceleration

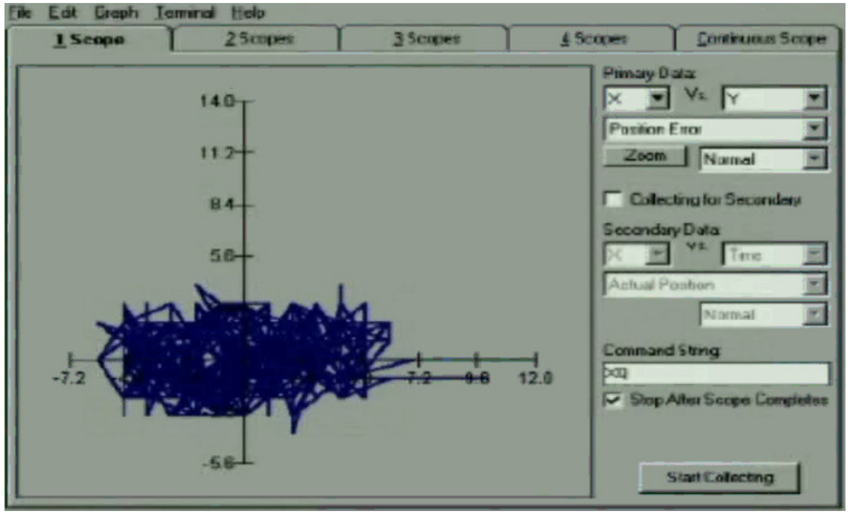


PID Tuning Using X-Y Motion



PID filter is tuned to shrink the error envelop as small as possible.
Tuning PID filter: Start with initial PID gains, watch the error, change the gains, repeat motion, obtain error envelop, calculate the change of error, update gains …

Error while Tuning



The tuning method is directly on hardware, results are 100% trustworthy