LESSON PLAN: PID Basics

OBJECTIVES:

- Provide students with a basic understanding of the PID control algorithm and the effect each term has on the control response.

- Familiarize students with the need to tune PID controllers and with procedures they can use to tune controllers.

SAMPLES OF BEHAVIOR:

- Describe the effect of proportional control
- Describe the effect of integral control
- Describe the effect of derivative control
- Explain what the interval is in a computerized control system, and how is it related to the gains
- Explain what is meant by "hunting"
- When should you tune a PID controller? (Two answers)
- How do you pick the initial gains for a PID controller
- What is the difference between open loop and closed loop tuning?

TEACHING METHOD: Lecture, with computer projector used to demonstrate tuning program

REFERENCES:

- Technical Handbook, Chapter 12 "Control Strategies"

- Eikon files "PID_DEMO.EIK", "TUNE_OL.EIK", and "TUNE_CL.EIK"

- Draw files "STATUS.GFX, RESPONSE.GFX, PID_CL.GFX, PID_OL.GFX, PIDO_TR1 thru TR4.BMP, and PIDC_OS1 thru OS5.BMP"

(All programs in directory C:\PID_LESS on instructors computer in training room. Start lecture w/ PID_DEMO.EIK running in foreground and STATUS.GFX in background.)

ORGANIZATION: See attached lecture notes

TEACHING AIDS:

- Load Eikon & Draw files (see references) on computer/projector

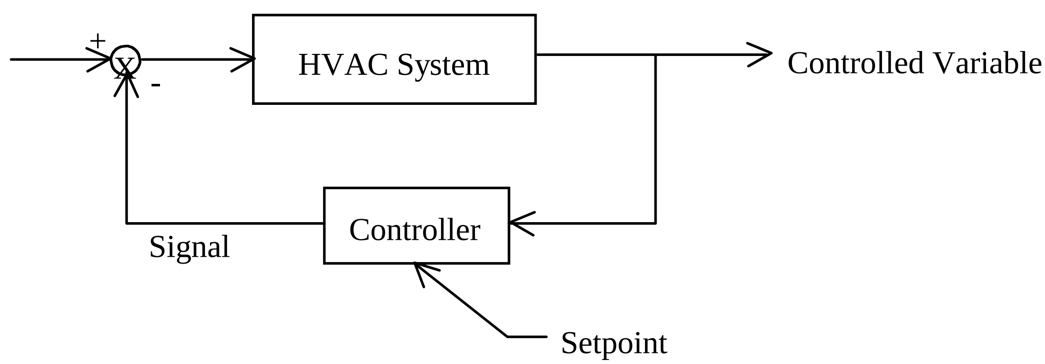
- Red & blue poker chips for "Pass the Chips" demonstration

HANDOUTS: "PID Basics" student outline guide (attached - Ami Pro file PID_LESS.SAM)

PID BASICS
Student Outline Guide
WITH INSTRUCTOR'S LECTURE NOTES

Objective: This lesson will describe the basic function of each part of the Proportional + Integral + Derivative (PID) Control Algorithm and will provide a method for "tuning" this controller. (i.e. setting the gains)

References: Technical Handbook, Chapter 12 "Control Strategies;" and Eikon files "TUNE_OL.EIK" and "TUNE_CL.EIK" (Available on ALC bulletin board or Support+ CD)



ERROR = CV - SP

"Traditional" Control System Diagram
(DRAW ON BOARD)

PID Terms:

Every control loop has a *controlled variable* (temperature, humidity, airflow, etc.) and a *setpoint*. (where you want the controlled variable to be) The difference between the actual measured value of the controlled variable and the setpoint is the *error*. The output of the controller is often called the *signal*.

(Class example: Boat Approaching Dock)
REMEMBER WALT TOLD YOU ABOUT LUTHER? 20" COLLAR AND SIZE 5 HAT?
IMAGINE TRYING TO TELL LUTHER HOW TO DOCK A BOAT IN A RIVER

MIGHT SAY "JUST BACK OFF THE THROTTLE AS YOU GET CLOSER TO THE DOCK"

WALT DIDN'T TELL YOU THIS, BUT LUTHER IS AN ENGINEER FROM (PICK RIVAL COLLEGE) HE'LL PUT THIS INTO AN EQUATION

Proportional Control: The output of the controller (ex. voltage to the actuator, pressure to the control valve, etc.) is proportional to the error:

$$P_{out} = (P_{gain})(Error)$$

where P_{gain} is the proportional gain

Computers measure the inputs and compute the control output at set increments of time called the *interval*. Thus the proportional output at interval "0" would be:

$$P_0 = (P_{gain})(Error_0)$$

LUTHER'S NOT REAL GOOD AT FIGURIN'. HE'S GOT TO SIT DOWN AND SPEND SOME TIME WITH PEN & PENCIL TO DO HIS CIPHERS.

AS A RESULT, HE HAS TO MEASURE THE DISTANCE FROM THE DOCK, CALCULATE WHERE HIS THROTTLE SHOULD BE, MOVE THE THROTTLE, AND THEN MEASURE AGAIN

COMPUTER'S ARE A LOT LIKE THIS: THEY CAN'T THINK & CHEW GUM AT SAME TIME

Proportional control provides a fast response to bring the controlled variable close to setpoint, but does not necessarily eliminate the error. The sustained error which results when a proportional control system reaches equilibrium is called the *offset*.

(Class example continued: Boat Approaching Dock)

WHAT'S HAPPENING TO LUTHER? AS HE GETS CLOSE, GETS THROTTLE TO POINT WHERE JUST HOLDS IT'S OWN AGAINST THE CURRENT.

LUTHER WOULD JUST SIT THERE FOREVER, SO YOU'VE GOT TO GIVE HIM MORE INSTRUCTIONS.

"WE'RE NOT MAKING MUCH PROGRESS. WHY DON'T YOU JUST BUMP THE THROTTLE UP A BIT?"

LUTHER WILL START FIGURING AGAIN. "SOMEHOW I'VE GOT TO INCREASE THE OUTPUT IF THE ERROR LASTS A LONG TIME"

Integral Control: The output of the controller is proportional to the integral of the error over time, i.e. to the error multiplied by the length of time you've had the error:

$$I_{out} = (I_{gain})\int(Error)dt$$

where I_{gain} is the Integral Gain

DON'T LET THE INTEGRAL SCARE YOU. PEOPLE WHO HAVEN'T HAD CALCULUS DON'T KNOW WHAT IT IS, AND THOSE WHO HAVE HAD CALCULUS SCREAM "NO! NO! NOT THAT!"

LUTHER COULDN'T FIGURE OUT AN INTEGRAL IF HIS LIFE DEPENDED ON IT, BUT THAT DOESN'T STOP HIM. LET'S LOOK AT IT THE "LUTHER WAY" LIKE A COMPUTER

When a computer calculates the integral output at time "0"

$$I_0 = I_{Previous} + (I_{gain})(Error_0)(DELTA\ T)$$

ACTUALLY, IF YOU KNOW CALCULUS AND DO THE MATH, YOU'LL SEE THERE SHOULD BE "DELTA T" TERM IN THERE, I.E. TIME PERIOD BETWEEN CALCULATIONS. SINCE IT'S A CONSTANT, WE USUALLY JUST LUMP IT IN WITH THE I_{GAIN}, BUT DON'T FORGET THAT YOUR GAIN INCLUDES THE PERIOD BETWEEN SAMPLES.

Since the output of an integral controller keeps building up as long as the error continues, integral control will eventually bring the controlled variable to setpoint and eliminate the error.

(Class example continued: Boat Approaching Dock)

WHAT HAPPENS IF LUTHER IS JUST A LITTLE TOO THROTTLE HAPPY?
YOU YELL "SLOW DOWN!! YOU'RE COMING IN TOO FAST!!" AND LUTHER THINKS "LET'S SEE NOW. IF THE ERROR IS CHANGING TOO QUICKLY. . . THERE MUST BE AN EQUATION FOR THIS. . ." THERE IS, BUT LUTHER WOULD NEVER THINK OF IT

Derivative Control acts to "slow down" the response once the system starts reacting to the controller output. The output of the derivative controller is proportional to the rate of change of the error, or:

$$D_{out} = (D_{gain}) \frac{d(Error)}{dt}$$

or, in computer terms:

AGAIN, DON'T WORRY ABOUT TRYING TO DO CALCULUS

$$D_0 = (D_{gain})(Error_0 - Error_{Previous})/(DELTA\ T)$$

ONCE AGAIN, THERE'S REALLY A "DELTA T" TERM THAT'S JUST BEEN INCORPORATED INTO THE D_{GAIN}

Derivative Control does not force the controlled variable to move toward setpoint, but instead exerts a "braking action" to keep the system from overshooting the setpoint.

DERIVATIVE CONTROL IS LIKE STIRRING MOLASSES WITH A CANOE PADDLE
WHICHEVER WAY YOU TRY TO MOVE THAT PADDLE, IT'LL FIGHT YOU
CAR SHOCK ABSORBERS ARE A DERIVATIVE CONTROL DEVICE
THEY SLOW DOWN THE BOUNCING BY OPPOSING BOTH UP AND DOWN MOTION

PID Control combines the Proportional, Integral, and Derivative terms into a single output:

$$PID_{out} = P_{out} + D_{out} + I_{out}$$

or, in computer terms:

$$PID_0 = (P_{gain})(Error_0) + I_{Previous} + (I_{gain})(Error_0) + (D_{gain})(Error_0 - Error_{Previous})$$

Note, although the term PID control is commonly used in our industry, very few HVAC control applications actually require derivative control. Most HVAC applications use PI control, with the proportional term providing a quick response to bring the system close to setpoint and the integral term eliminating the offset. If a fast response is not required, some HVAC applications use integral control only.

DERIVATIVE CONTROL IS COMMONLY USED IN FAST REACTING, CRITICAL SYSTEMS WHERE OVERSHOOT IS NOT UNDESIRABLE, LIKE A MOON LANDER
HVAC APPLICATIONS ARE USUALLY SLOW AND CAN TOLERATE A LITTLE OVERSHOOT.

RUN "PID_DEMO.EIK" TO SHOW HOW PID ALGORITHM WORKS

TUNING PID CONTROLLERS:

"Tuning" refers to adjusting the PID gains to give the system acceptable response. It is usually impossible to adjust to the "best" response, because the system changes from day to day.

(Class Example: Pass the Chips)

Designate front row students as "HVAC Ducts" (I got all my ducts in a row), student on far end as "Thermostat." Instructor is "AHU" (full of hot air). Tell Thermostat he/she needs one red chip to be comfortable.

Begin on cold winter day. Room w/ thermostat loses one red chip per turn. No chips in room or duct. Ask thermostat "are you comfortable?" Too cold so put 1 red chip in duct. "Are you comfortable?" etc. Eventually gets 1 red chip per turn, loses one red chip per turn, stays comfortable. Booor - ing.

"Tuning HVAC controls is easy, right? Now lets start over." Gather up chips. "The next morning isn't quite as cold. Today you lose one red chip every other turn. Ready?" When red chips begin to pile up in room, thermostat is too hot. Start putting blue chips (cold air) in duct. Pretty soon room starts cycling between too hot and too cold.

"What did we learn? A system that's tuned just right on one day can go unstable when conditions change. The trick is to adjust the PID gains so you get a good response all the time, but never go unstable. We also saw the effect of a delay time on the system. If I had just been giving my poker chips directly to the thermostat, without using the ductwork, we wouldn't have gone unstable. Unfortunately, that's not how a real HVAC system works. There's always some delay between when a room calls for heating or cooling and when it actually starts to feel the results. The longer this delay time, the easier it is to go unstable.

OK, NOW YOU KNOW WHY TO TUNE A CONTROLLER. WHEN SHOULD YOU DO IT?

When do you tune a controller?

(1) At system start-up

(2) Any time you notice a control loop is "hunting" (cycling above & below setpoint) or won't stay at setpoint

Ideally, the system should be tuned when it is experiencing a typical load. (Occupied, with outdoor temperature hot or cold enough to require significant heating or air conditioning.) Since this is not always practical, tune the system with the best loading you can get and re-check performance when the load changes significantly.

OK, IF IT'S TIME TO TUNE A CONTROLLER, WHERE DO YOU START?

How do you pick initial gains?

(1) Ask experienced engineer in your dealership, or let Dealer Services engineer it

(2) Use values from previous applications with similar equipment.

(3) Use "generic" defaults:

Application	P_{gain}	I_{gain}	Interval
Valves	12	2	60 sec
Mixed Air Dampers	10	2	30 sec
Zone Temp	20	2	60 sec
Static Pressure	2	0.5	10 sec

(Note: These are one engineers' "seat of pants" recommendations. If your experience shows other values work better - let me know!)

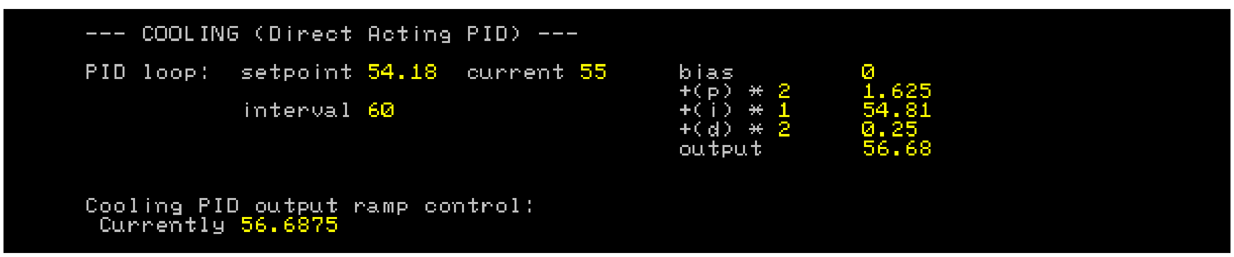
TWO COMMENTS ABOUT THE INTERVAL. FIRST, NOTICE THAT THE FASTER A SYSTEM RESPONDS, THE SHORTER YOU NEED TO MAKE THE INTERVAL BETWEEN CALCULATIONS. SECONDLY, PLEASE REMEMBER THAT THE INTEGRAL AND DERIVATIVE GAINS INCLUDE A "DELTA T" TERM. IF YOU CHANGE THE INTERVAL, YOU MAY NEED TO CHANGE THE GAINS TO KEEP IT STABLE.

How do you tune a controller?

- "Eyeball Method" (Tech Handbook pg. 12-2)

Change the setpoint and watch how system responds, either by trending or by watching controlled variable on status page. The status page is especially useful, as it shows how much each term of the PID equation contributes to the total controller output.

USE ALC DRAW TO SHOW STATUS.GFX ON PROJECTOR



PID Outputs on Status Page

HAVE TO REFRESH SCREEN REGULARLY TO WATCH RESPONSE. CAN ALSO USE LIVE GFB'S TO SEE DYNAMIC RESPONSE, BUT WON'T SHOW INDIVIDUAL PID TERMS

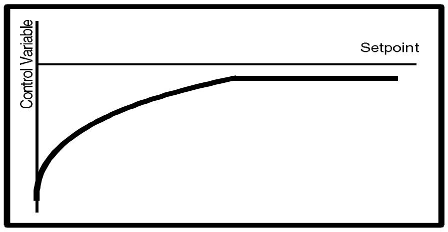
-- If response is sluggish and P contribution is low, increase P gain

-- If system settles on a value other than setpoint and I contribution is low, increase I gain

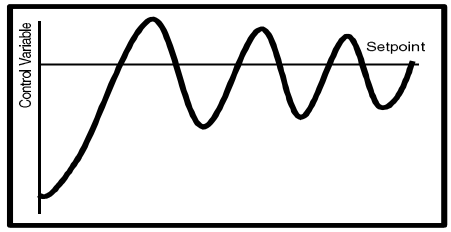
-- If system overshoots & cycles, either P or I gain (or both) is too high. Look at status page to see which term is contributing the most to the controller output and lower that gain.

(See Response Plots on Next Page)

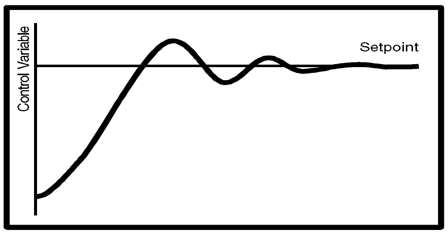
USE ALC DRAW TO SHOW RESPONSE.GFX



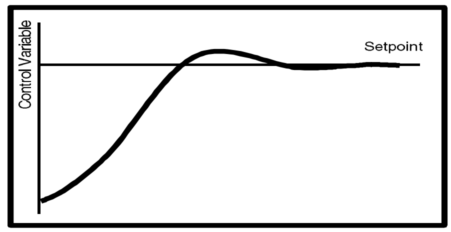
I gain too Low
P gain may be too low



D gain too low



Good Tuning using
P and I gain alone



Good Tuning using
P, I, and D

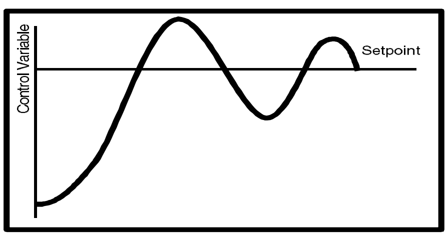


Figure 12-3: P or I gain too High

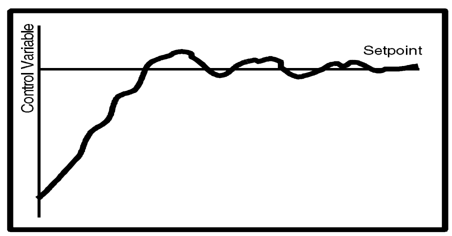


Figure 12-6: D gain too High

Typical Control System Responses

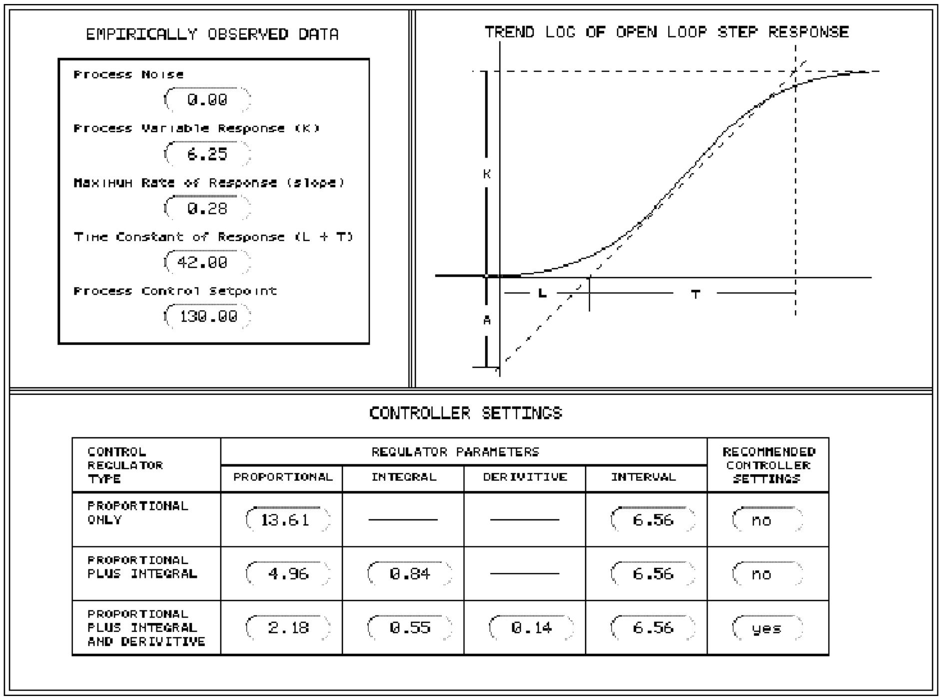
- Ziegler-Nichols Open Loop Tuning:

"Open Loop" tuning means the control system output is locked at a fixed value. (i.e. you no longer have a closed loop where the sensor feeds information back to the controller that adjusts the controlled variable going back to the sensor. . . etc.)

REMEMBER "TRADITIONAL" CONTROL SYSTEM DIAGRAM? WE'RE GOING TO BREAK THE LINE FROM CONTROLLED VARIABLE BACK TO CONTROLLER

With the controller taken out of the loop, you manually change the system input from one extreme to the other (ex. damper fully closed to fully open) and measure the response of the controlled variable. This tells you how the system reacts to a change in its input, and the Ziegler-Nichols open loop equation will calculate the proper PID gains from this response.

Procedures for open loop tuning are given on pages 12-7 to 12-10 in the Technical Handbook, and in the Eikon file TUNE_OL.EIK. These references tell you how to measure the open loop response, and you can input these measurements into the Eikon file and it will calculate the proper PID gains for you.



Input/Output Page from TUNE_OL.EIK

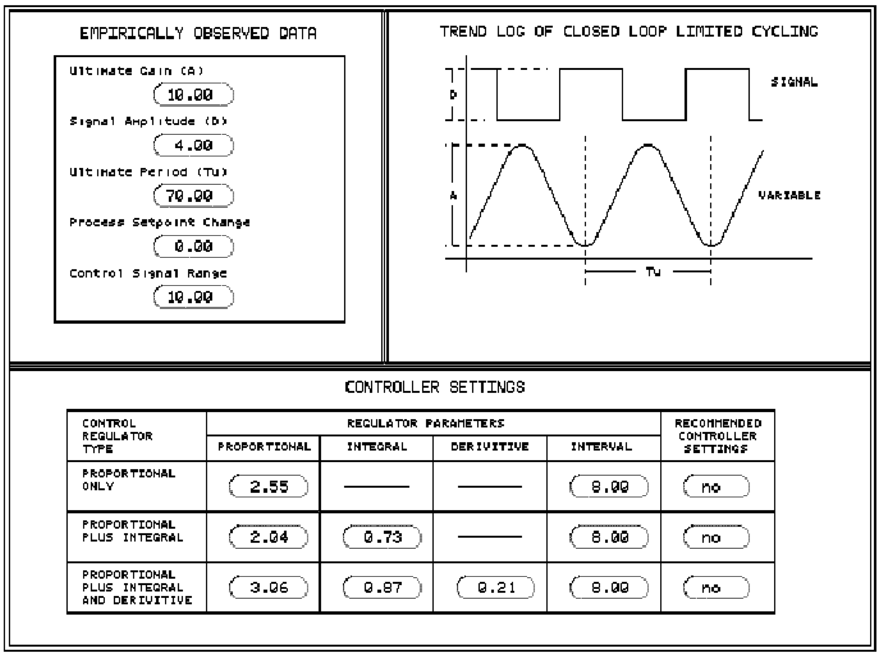
(Class Example - Open Loop Tuning)

Used Eikon tuning program to adjust gains in VAV fan speed control supplying this building
Show Eikon File TUNE_OL.EIK in Simulate mode. Want to measure controlled variable change, slope, & time period
Show ALC Draw File PID_OL.GFX. First screen: This shows the VAV fan static pressure output vs signal to fan. Since we're trying to adjust fan, had to lock rest of system in fixed mode so it wouldn't react. In this case, we locked all VAV boxes so the boxes wouldn't open & close as we were changing the fan speed, as that would affect static pressure readings. This screen shows safe, stable operation when control signal between 2V and 8.5V, so we stabilized system at 2V and then gave step input of 8.5V.
Page down one screen in Draw file. This is the same response, with the X-axis (time) expanded. We have drawn a straight line approximation of the slope in red, and can measure K=12.73, slope = 0.76, and L+T = 27 sec
Switch back to Eikon. Plug in values, using Noise = 0, K=12.73, Slope = 0.76, and L+T = 27. SP=?? Look @ Draw plot
Switch back to Draw. Looks like SP = 15. Reasonable? Do you set your VAV fans at 15" static pressure? See note at top of scale. Pressure readings are 10x actual pressure. We typically do this with analog inputs (sensors) that have a small range (like 0 - 2"wg) to get better resolution. This will be covered in more detail in Diversified SVW class. For now - trust me. If enter Y-axis values x10, calculations will be more accurate but will have to multiply gains x10.
Switch back to Eikon. Enter SP=15 and step. Recommended PID gains are P = 1.54, I = 0.75, D = 0.19, Interval = 3 sec
Switch back to Draw & page down. This is result when set gains (x 10). Looks pretty good, maybe some overshoot
Page down one screen. Experimented with dropping P gain to 12 to reduce overshoot. Looks good!

- Ziegler-Nichols Closed Loop Tuning:

If you have tried open loop tuning and the system is still not performing properly, the next step is to try closed loop tuning. "Closed loop" means the control system is still in the loop and is controlling the output. To perform closed loop tuning, you set the Integral and Derivative gains to zero and increase the Proportional gain until the system becomes unstable and starts "hunting." Measurements taken from this unstable system can be used by the closed loop Ziegler-Nichols equation to determine proper gains.

Procedures for closed loop tuning are given on pages 12-10 to 12-13 in the Technical Handbook, and in the Eikon file TUNE_CL.EIK. These references tell you how to measure the closed loop response, and you can input these measurements into the Eikon file and it will calculate the proper PID gains for you.



CONTROL REGULATOR TYPE	REGULATOR PARAMETERS				RECOMMENDED CONTROLLER SETTINGS
	PROPORTIONAL	INTEGRAL	DERIVITIVE	INTERVAL	
PROPORTIONAL ONLY	2.55	——	——	8.00	no
PROPORTIONAL PLUS INTEGRAL	2.04	0.73	——	8.00	no
PROPORTIONAL PLUS INTEGRAL AND DERIVITIVE	3.06	0.87	0.21	8.00	no

Input/Output Page from TUNE_CL.EIK

(Class Example - Closed Loop Tuning)

Look @ Eikon TUNE_CL.EIK. Need to make system start hunting. Measure period & amplitude of cycle
Look @ ALC Draw PID_CL.GFX. Same system as before. To make it start hunting, set I & D gains to 0, increase P gain until goes unstable. Ask students "In control terms, what is Static Pressure?" (Ans: Controlled Variable) "What is Supply Fan Variable Frequency Drive Voltage?" (Ans: Signal)
Page down. Expanded scale on X-axis (time) Want smooth sine curve, no "plateaus" where system is maxing out
Page Down: Good curve. Measured A - 14.63, D = 6, Tu = 39.
Switch back to Eikon. Enter Gains. "Process Setpoint" is only used if setpoint is being controlled by another controller during the test. In our case there is no other controller, so = 0. Control signal range is max range of output signal. We're using a 0 - 10V VFD, so range = 10.
Execute "step" and look at calculated values. Interesting. PID is no longer recommended. Use PI control only. P = 2.09, I = 1.34, Interval = 4 sec.
Switch back to Draw and page down. Gains were set at 20, 13 & 0 (remember: 10x) Interval = 4. Bias = 25? What's bias? Bias is a constant value that is added to PID algorithm, sort of a "starting value" or "normal value." PID = P + I + D + Bias Doesn't affect stability or accuracy of control, but can help controller zero in on correct value quicker. Different companies handle different ways. W/ ALC it's a percentage. In this case, we're saying "use 25% as initial starting value, and let PID loop adjust output from there."
With these values, control looks pretty good.
Page down. Expanded Y axis. Holding Static Pressure steady +/- 0.1 inch. Pretty good!

A few notes on PID tuning:

(1) Most HVAC systems are slow to respond. Be patient! Make certain the system has stabilized at one setting before you try another. Trying to speed up the response by putting in high gains leads to instability.

(2) Think about the effects of your actions before you override a controller. Opening the outside air dampers on a cold winter day to tune the mixed air PID loop could lead to cold occupants and frozen coils. Similarly, speeding up a variable speed fan when the VAV boxes are closed could induce stress concentrations beyond the system's elastic deformation limit. (i.e. the ducts will blow up!)

(3) System responsiveness changes from day to day. It is better to err on the side of slow response than to risk having the system go unstable. Very few HVAC applications require fast response, and "hunting" will wear out actuators (and the customer's patience!) in no time at all.

THAT'S QUICK AND DIRTY ON PID CONTROL. HAVE DISCUSSED PROPORTIONAL CONTROL, WHICH IS WHAT? (PROPORTIONAL TO ERROR) DRAWBACK? (OFFSET)

HAVE DISCUSSED INTEGRAL CONTROL WHICH IS WHAT? (ERROR X TIME) WHAT DOES THIS DUE FOR US? (ELIMINATES OFFSET)

WHAT DOES DERIVATIVE CONTROL DO FOR YOU? (SLOWS RESPONSE, PREVENTS OVERSHOOT)

WE TALKED ABOUT TUNING THE CONTROLLER, WHICH IS? (ADJUSTING GAINS)

TWO FINAL QUESTIONS: WHAT'S THE INTERVAL? (TIME BETWEEN CALCULATIONS) AND WHAT DOES IT HAVE TO DO WITH TUNING? (INTERVAL IS "BUILT INTO" I AND D GAINS - IF CHANGE INTERVAL MUST CHANGE GAINS)

THAT'S ALL QUESTIONS I HAVE FOR YOU. DO YOU HAVE ANY FOR ME?