

## Visual ModelQ Training



## Tuning a PI Controller

This unit discusses

- Zone-based tuning for a PI controller
- Adjusting gains with “Live Constants”
- Retuning the system from the Training unit “Building a PI Controller”

## Install *Visual ModelQ*

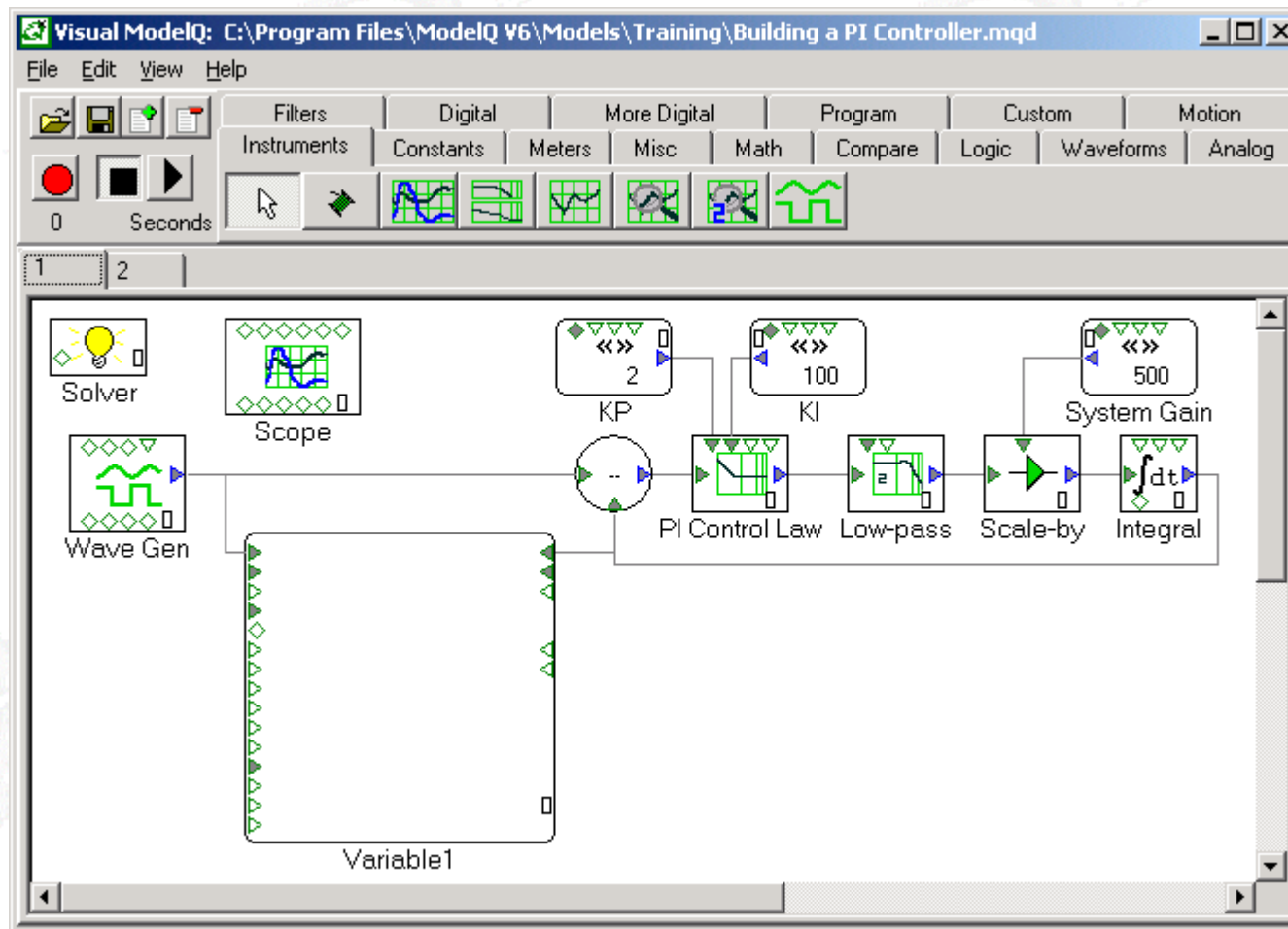
To run *Visual ModelQ* the first time:

- Click here to visit [www.QxDesign.com](http://www.QxDesign.com)
- Download *Visual ModelQ*\*\*
- Run *Visual ModelQ* installation
- Launch *Visual ModelQ* using the Windows start button or clicking on the icon 
- The “default model” should appear



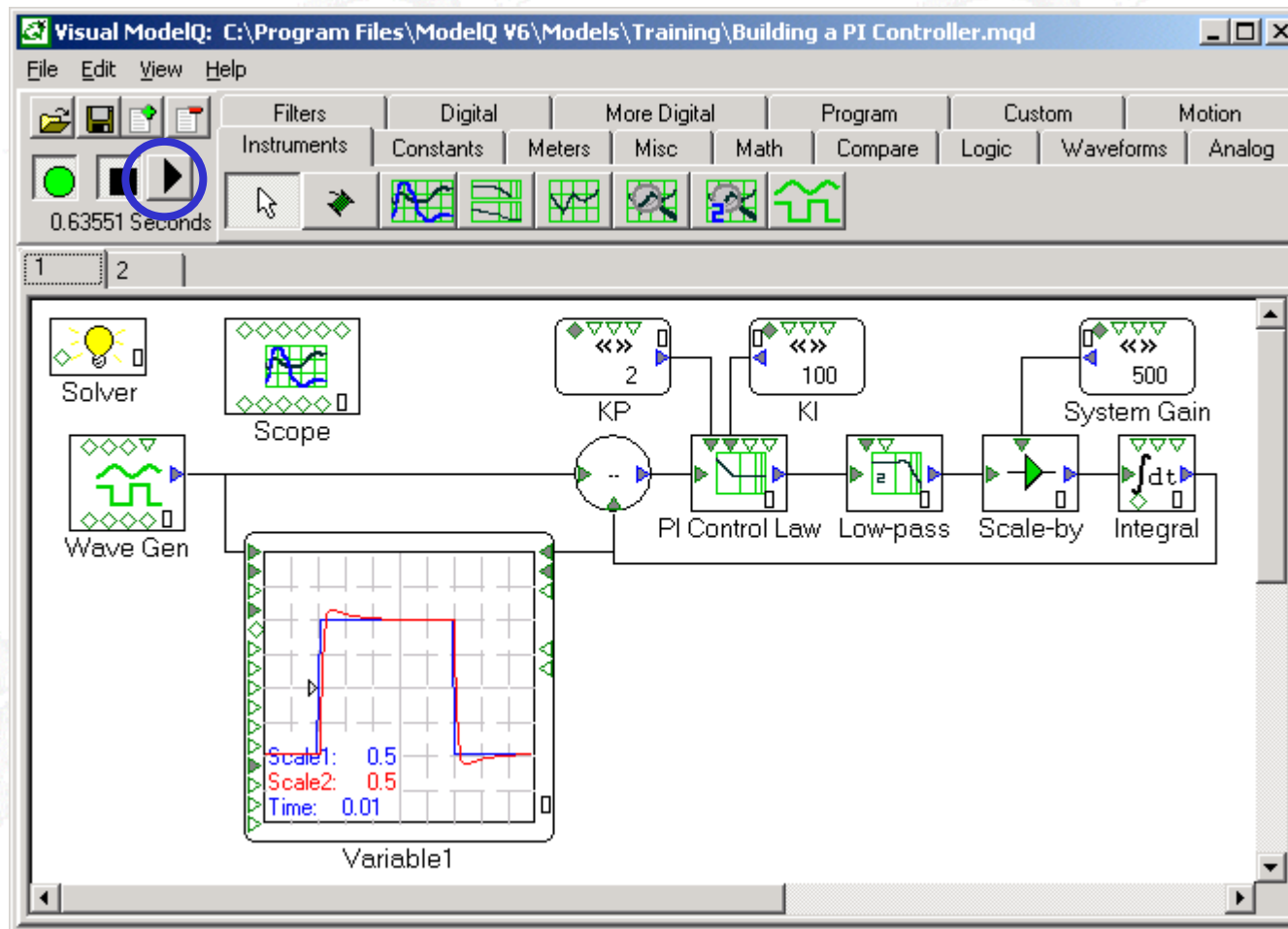
## Start with the model “Building a PI Controller”

- Visit [www.QxDesign.com/VisualModelQ#Training](http://www.QxDesign.com/VisualModelQ#Training)
- Download “Building a PI Controller.mqd” or build the model yourself using the presentation “Building a PI Controller.pps”



## Check performance with the initial values

- Click the run button to see the performance with the initial gains of  $K_P = 100$ ,  $K_I = 2$ , and with a System Gain of 500.
- Looks good; feedback (red) responds fast, with little overshoot.



## Zone-based Tuning Procedure

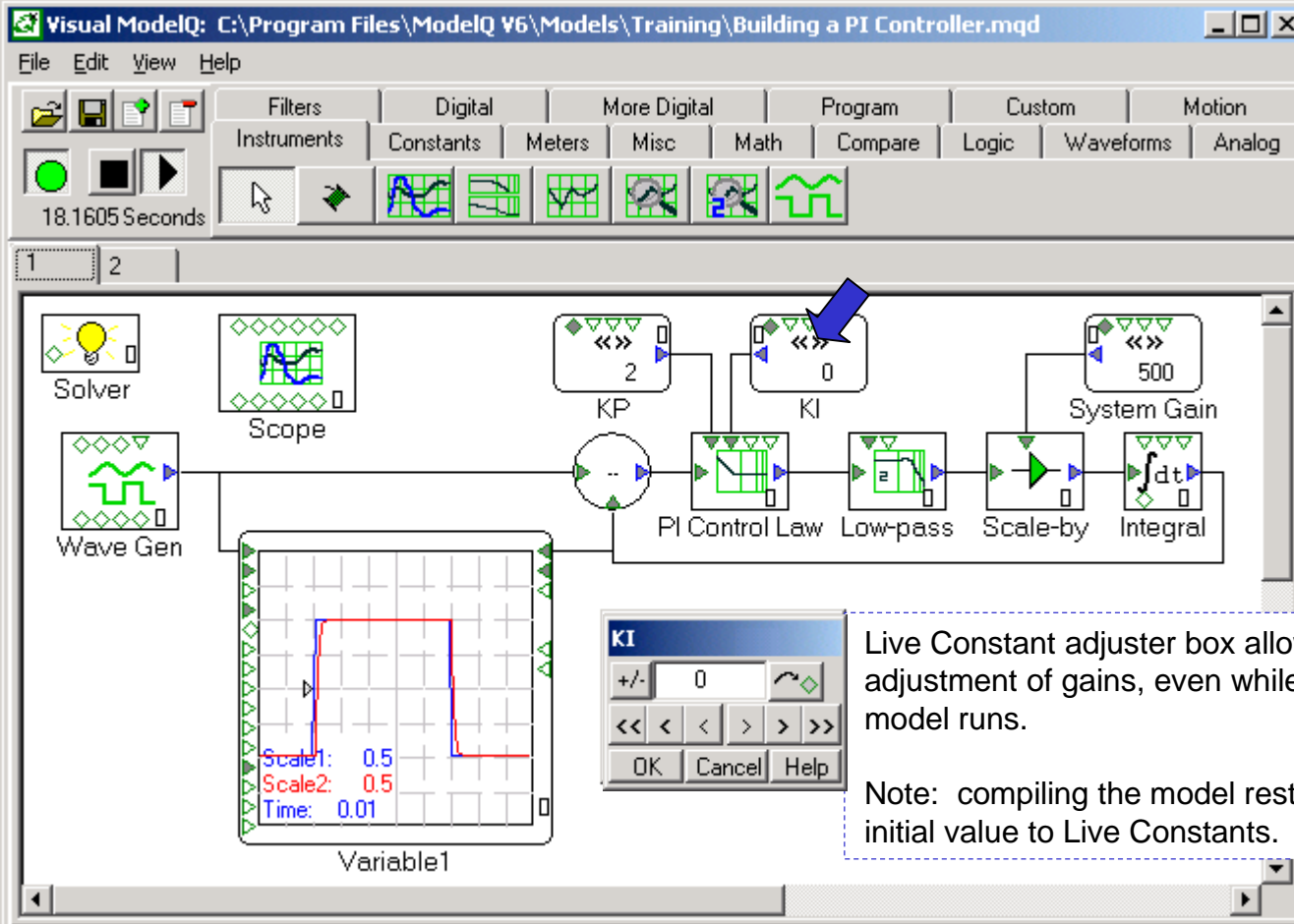
- An overview of the Zone-based Tuning Procedure for a PI (Proportional-Integral) controller:
  1. Set integral gain (KI) to zero\*\*
  2. Set proportional gain (KP) low for system stability
  3. Apply a step command
  4. Increase KP to max value without overshoot to a step++
  5. Increase KI for ~10% overshoot to a step++
- The starting model is tuned with  $KP=2$  and  $KI=100$ . Let's revisit, but ignore the fact that these values are known.
- For more detail on Zone-based Tuning see Control System Design Guide, 2nd Ed., Chapter 3.

\*\*Note that in practice, some systems should not be operated with zero integral gain because external loading may cause undesirable consequences (example: in a motion-control system where an unbalanced load could fall with integral gain of zero).

++Note that steps 4 and 5 are “rules of thumb”. The settings for KP and KI may be somewhat higher or lower for a given application. The point of Zone-based tuning is that, usually, having a fixed procedure with specific performance criteria speeds the tuning process and produces more consistent results.

## Step 1: Zero Integral Gain

- While model runs, double-click in KI block; adjuster box appears.
- Click in edit box (initially “100”) and type “0” to disable integral.
- Press OK button and adjuster box will hide.



The screenshot shows the Visual ModelQ interface with a PI controller model. The model includes a Wave Gen block, a Scope block, a Solver block, and a PI Control Law block. The PI Control Law block is composed of a summing junction, a KI block, a Low-pass block, a Scale-by block, and an Integral block. The KI block is highlighted with a blue arrow, and its adjuster box is open, showing a value of 0. The Scope block displays a plot of Variable1, which is a square wave. The adjuster box for the KI block has a text field containing '0', and buttons for '+/-', '<<', '<', '>', '>>', 'OK', 'Cancel', and 'Help'.

Visual ModelQ: C:\Program Files\ModelQ V6\Models\Training\Building a PI Controller.mqd

File Edit View Help

Filters Digital More Digital Program Custom Motion

Instruments Constants Meters Misc Math Compare Logic Waveforms Analog

18.1605 Seconds

1 2

Solver Scope Wave Gen

KP KI System Gain

PI Control Law Low-pass Scale-by Integral

Variable1

Scale1: 0.5  
Scale2: 0.5  
Time: 0.01

KI

+/- 0

<< < > >>

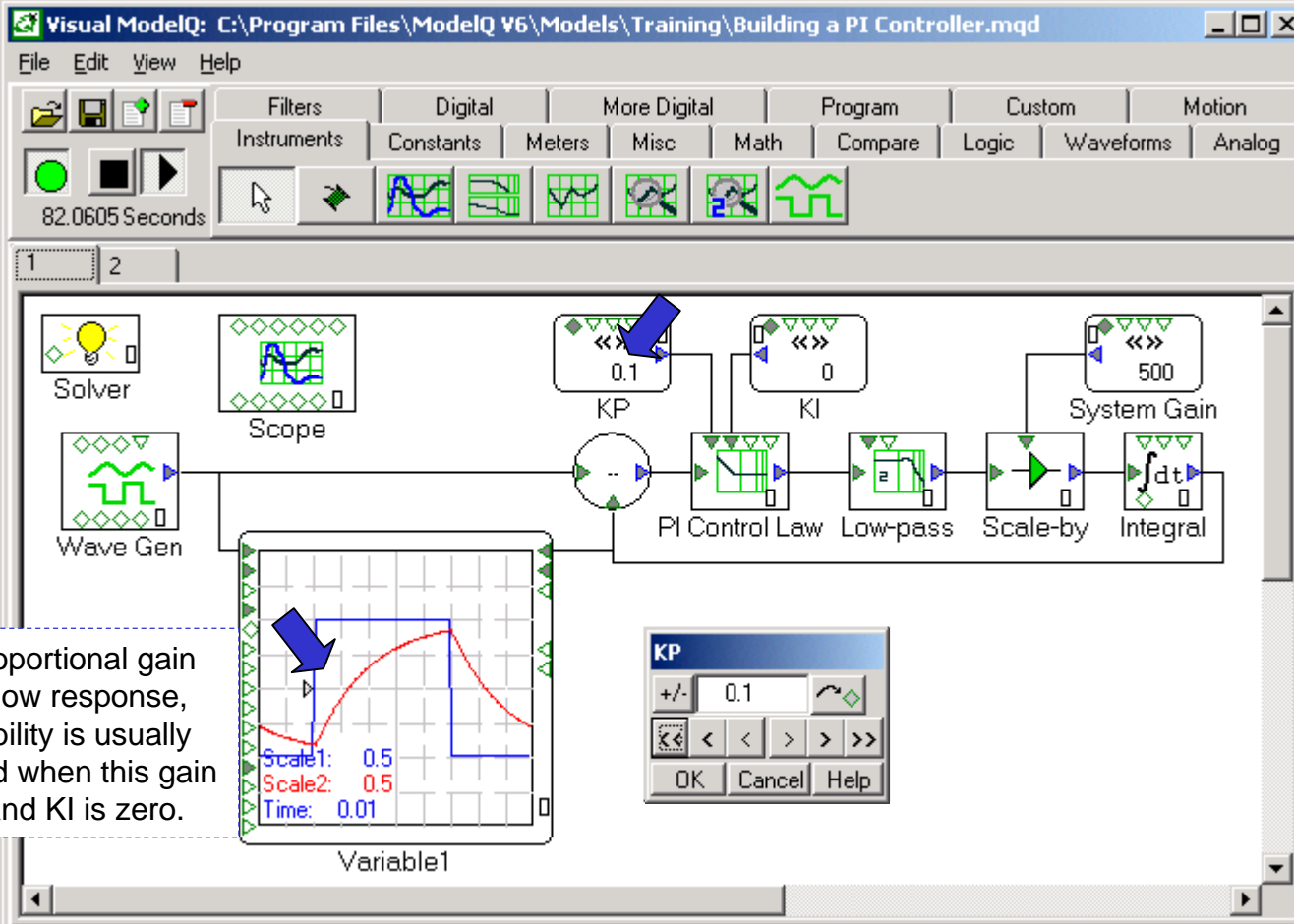
OK Cancel Help

Live Constant adjuster box allows easy adjustment of gains, even while the model runs.

Note: compiling the model restores the initial value to Live Constants.

## Step 2: Set Proportional Gain Low

- Double-click in KP block; adjuster box appears.
- Click in edit box (initially “2”), type “0.1” and press Enter key. Note slow response; that’s okay...here, we just need stability.

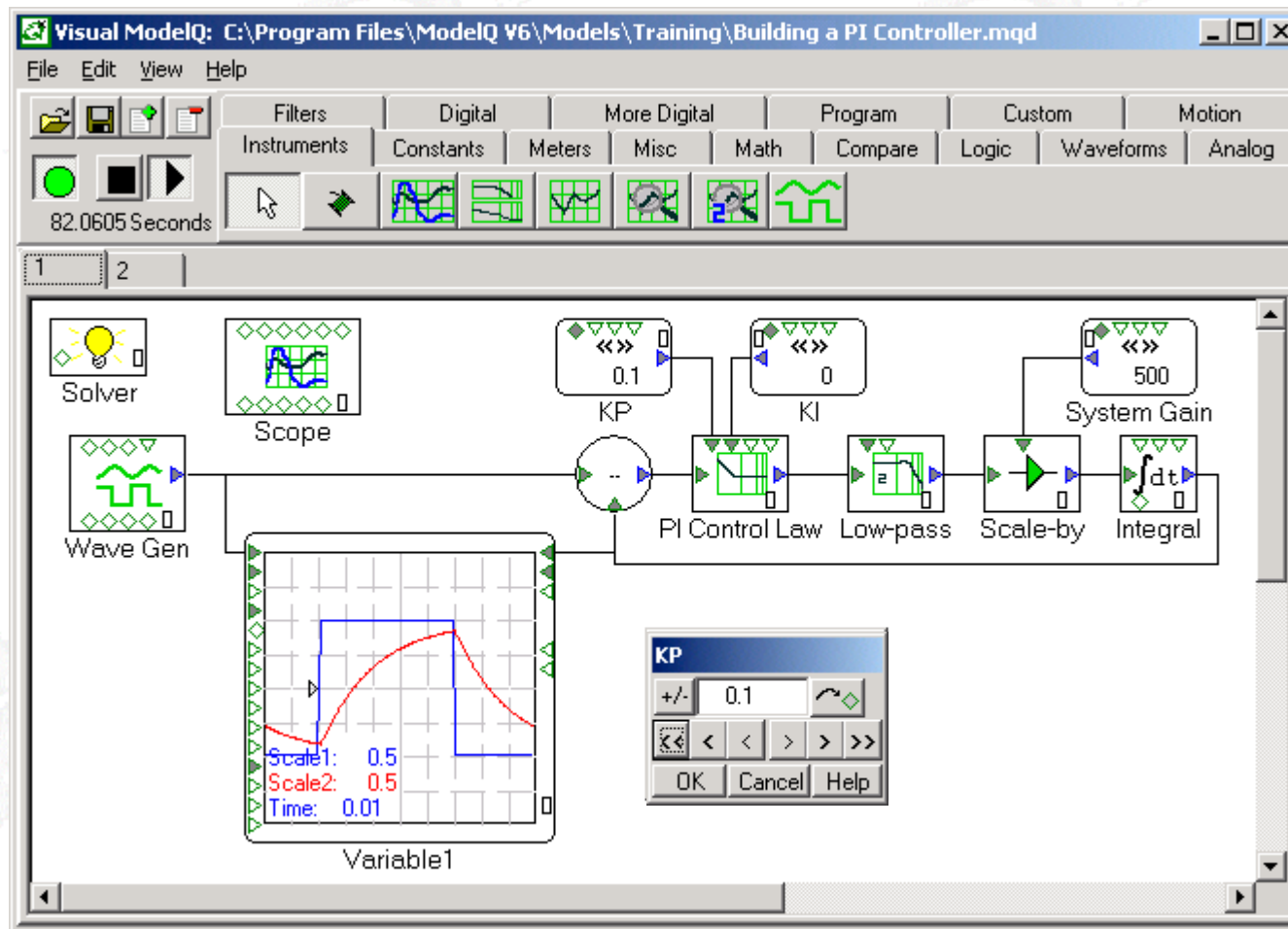


The screenshot shows the Visual ModelQ interface for a PI controller model. The main workspace contains several blocks: Solver, Scope, Wave Gen, KP (Proportional Gain), KI (Integral Gain), System Gain, PI Control Law, Low-pass, Scale-by, and Integral. The KP block is highlighted with a blue arrow, and its value is set to 0.1. A dialog box for the KP block is open, showing the value 0.1 and buttons for OK, Cancel, and Help. The Scope block displays a plot of Variable1, showing a slow response to a step change. A blue arrow points to the plot, and a text box explains that low gain results in a slow response but ensures stability.

Low proportional gain gives slow response, but stability is usually assured when this gain is low and KI is zero.

## Step 3: Apply Step Command

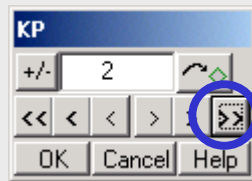
- Already done. This model is built with the step command applied.
- Now that the starting conditions are set, we can start tuning.



## Step 4: Increase KP, avoid overshoot

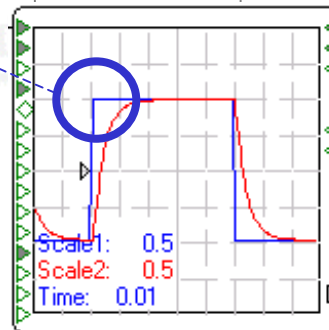
- Increase KP. Response improves up to  $KP \sim 2.0$ ;  $KP > 3.0$  makes overshoot. Several cases are shown below. Choose  $KP = 2.0$ .

TIP: Use adjustment such as ">>" (large increase) to quickly vary gains while the model runs

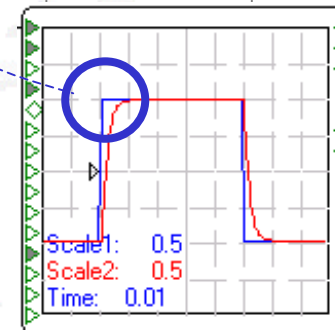


Slow Response

KP = 0.5

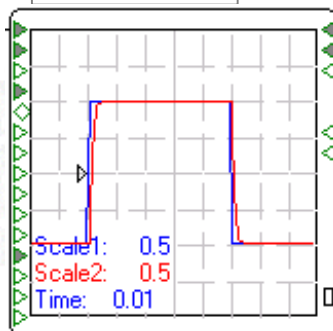


KP = 1.0



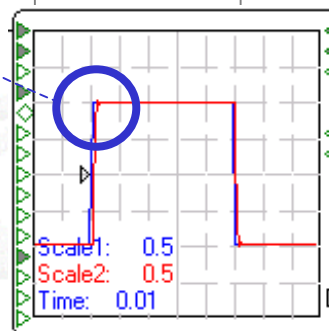
Choose this!

KP = 2.0



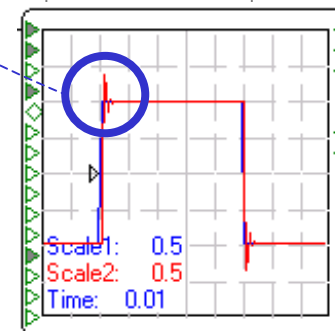
Slight overshoot

KP = 3.0



Overshoot and ringing

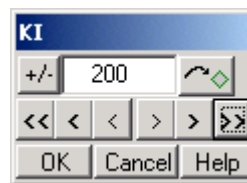
KP = 5.0



## Step 5: Set KI for ~10% overshoot

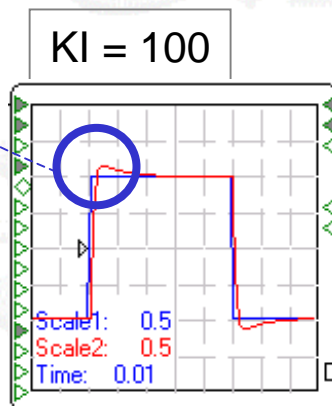
- Start with KI low (say, 1)
- Increase KI for ~10% overshoot. Choose KI = 200.

TIP: To save a value so it isn't reset by each compile, click the "save as initial value" button right of the value.

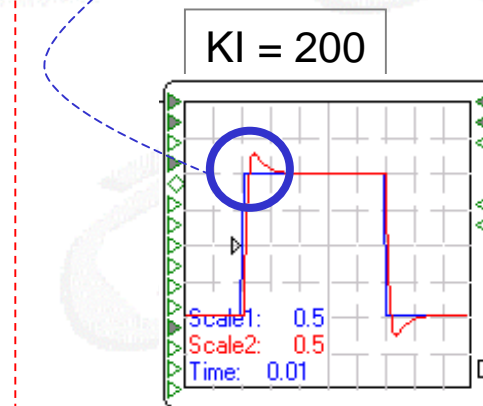


Choose this!

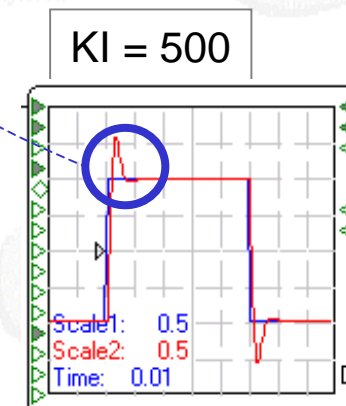
~5% overshoot



~10% overshoot



25% overshoot



Visit [www.QxDesign.com](http://www.QxDesign.com) for information about software and practical books on controls.



Click here for information on [Control System Design Guide \(2nd Ed.\)](#), published by Academic Press in 2000



Click here for information on [Visual ModelQ](#)



Click here for information on [Observers in Control Systems](#), published by Academic Press in 2002