Substituting this into the original equation for PV_{max}

$$PV_{\max} = m_{\min} \times k_{\mathrm{p\ max}} \times SP_{\max}$$

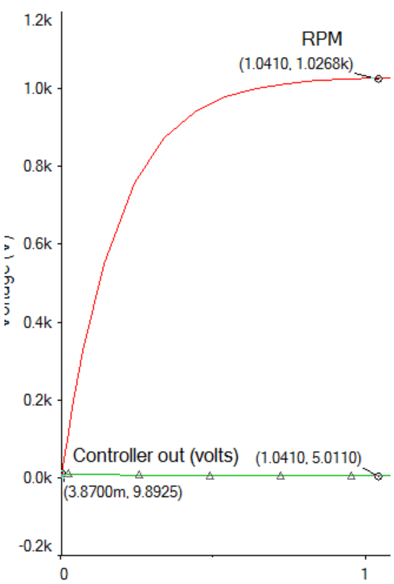
$$\frac{PV_{\max}}{SP_{\max}} = m_{\min} \times k_{\mathrm{p\ max}}$$

The whole goal of negative feedback is to cause *PV* = *SP*.

$$1 = m_{\min} \times k_{\mathrm{p\ max}}$$

Worst case
k_p

$$k_{\mathrm{p\ max}} = \frac{1}{m_{\min}}$$



Proportional only control
SP = 2060 RPM

k_p = 1/m produces fast response with 50% residual error

This is a pretty small value and results in 50% residual error. When the system does not experience full-scale transients or errors, then these calculations should be completed with the practical limits for that system.

Integral Control

The proportional controller can never produce a perfect result. Increasing k_p reduces the residual error, until the output of the controller is driven to its maximum, and control is lost. There's a better way.

The *integral* controller's output depends on the integral of the error.

$$co = k_{\mathrm{i}} \int error\ dt$$

where *co* and *error* are functions of *time*.

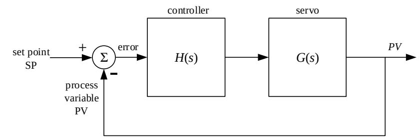
To understand why this relationship can drive the *error* to zero, differentiate both sides of the equation

Integral controllers drive the residual error to zero.

How rapidly the controller output, *co*, changes depends on the *error* and its constant, k_i. Right after a step in set point, the *error* is large and the controller output changes quickly. It continues to change, though more slowly, until the *error* is zero. If there is *error* the output changes.

Once the *error* is driven to zero, the controller output stops <u>changing</u>. It holds whatever value is necessary to make *error* = 0.

$$co = k_{\mathrm{i}} \int error\, dt$$



$$\boldsymbol{\frac{PV}{SP} = \frac{GH}{(1+GH)}}$$

In the Laplace domain this becomes

$$CO = \frac{k_{\mathrm{i}}}{s} \times ERROR$$

In terms of the servo tracking block diagram

$$H = \frac{CO}{ERROR} = \frac{k_{\mathrm{i}}}{s}$$

The overall, closed loop transfer function is

$$\frac{PV}{SP} = \frac{GH}{1+GH} = \frac{\dfrac{m}{\tau s+1} \times \dfrac{k_{\mathrm{i}}}{s}}{1 + \dfrac{m}{\tau s+1} \times \dfrac{k_{\mathrm{i}}}{s}}$$

$$\frac{PV}{SP} = \frac{\dfrac{mk_{\mathrm{i}}}{s(\tau s+1)}}{1 + \dfrac{mk_{\mathrm{i}}}{s(\tau s+1)}}$$

$$\frac{PV}{SP} = \frac{\dfrac{mk_{\mathrm{i}}}{s(\tau s+1)}}{\dfrac{s(\tau s+1)}{s(\tau s+1)} + \dfrac{mk_{\mathrm{i}}}{s(\tau s+1)}}$$

$$\frac{PV}{SP} = \frac{\dfrac{mk_{\mathrm{i}}}{s(\tau s+1)}}{\dfrac{s(\tau s+1) + mk_{\mathrm{i}}}{s(\tau s+1)}}$$

$$\frac{PV}{SP} = \frac{mk_{\mathrm{i}}}{s(\tau s+1) + mk_{\mathrm{i}}}$$

$$\frac{PV}{SP} = \frac{mk_{\mathrm{i}}}{\tau s^2 + s + mk_{\mathrm{i}}}$$

Using an *integral* controller turns a first order lag element (e.g. a motor) into a <u>second</u> order system. The standard second order transfer function has the coefficient of s^2 set to 1. So, divide numerator and all the terms in the denominator by τ.

St

$$\frac{PV}{SP} = \frac{\frac{mk_\mathrm{i}}{\tau}}{s^2 + \frac{1}{\tau}s + \frac{mk_\mathrm{i}}{\tau}}$$

$\boldsymbol{s^2 +}$

Equating coefficients with the standard form gives

$$\omega_\mathrm{n}^2 = \frac{mk_\mathrm{i}}{\tau}$$

$$\omega_\mathrm{n} = \sqrt{\frac{mk_\mathrm{i}}{\tau}}$$

$$A\omega_\mathrm{n}^2 = \frac{mk_\mathrm{i}}{\tau} = \omega_\mathrm{n}^2$$

In the steady state

PV = SP

There is no residual error!

$$A = 1$$

$$2\xi\omega_\mathrm{n} = \frac{1}{\tau}$$

$$\xi = \frac{1}{2\omega_\mathrm{n}\tau}$$

$$\xi = \frac{1}{2\tau\sqrt{\frac{mk_\mathrm{i}}{\tau}}}$$

Closed loop system damping can be controlled by adjusting

$\boldsymbol{k_\mathrm{i}}$

$$\xi = \frac{1}{2\sqrt{mk_\mathrm{i}\tau}}$$

Example 5-3

a. Calculate k_i for the motor from previous examples to produce a *critically* damped closed loop and verify its performance.

b. Alter k_i by 2 or ½ to make the system underdamped and verify its performance.

From Example 4-2

m= 206 RPM/V
τ= 0.362 sec
ω_o = 277 RPM

$$G = \frac{mV}{(\tau s + 1)}$$

Solution

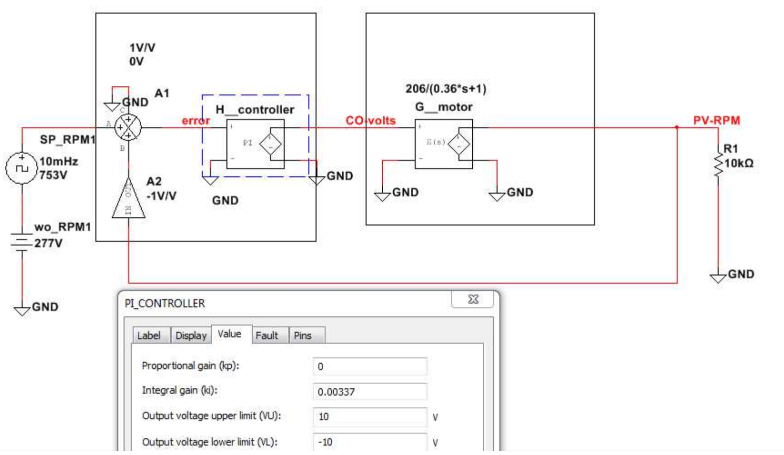
$$\xi = 1 = \frac{1}{2\sqrt{mk_i\tau}}$$

$$1 = \frac{1}{4mk_i\tau}$$

$$k_i = \frac{1}{4m\tau}$$

$$k_i = \frac{1}{4 \times 206 \times 0.36 \text{ sec}} = 0.00337/_{\text{sec}}$$

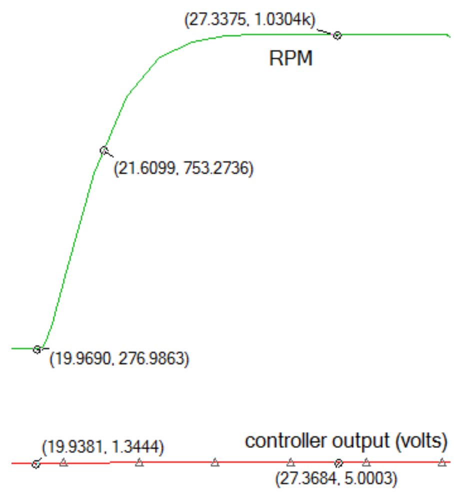
The simulation schematic is shown in Figure 5-15 (a), and the transient analysis of *PV* in Figure 5-15 (b). The integral controller block cannot have its initial conditions set. So, there is nothing to drive the motor to its initial speed at t = 0 sec. Therefore, the **wo_RPM** source has been removed, and the *SP* step size set to 50% = 617 RPM.



MULTISIM

(a) Example 5-3 Integral controller of the motor simulation schematic

MULTISIM



(a) Motor response to a set point step from 277 RPM to 1030 RPM

Figure 5-15 Integral control of the example motor

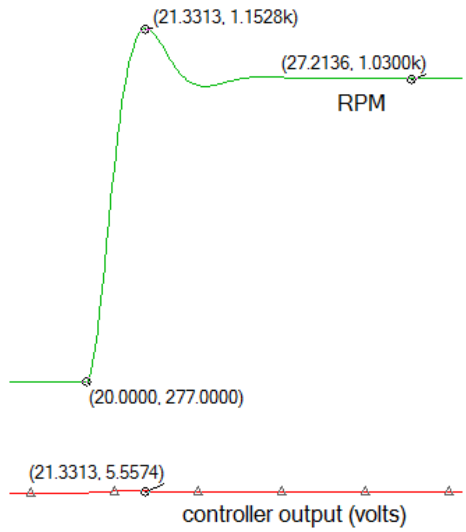
The *PV* speed climbs steadily to its final value of 1030 RPM, which equals the set point. Indeed, there is no residual error. The controller output never approaches its maximum, also steadily climbing from 1.3 V to 5.0 V, half of what it takes for the motor to run full speed.

The time for the system to rise 63% of its steady state value is 1.6 seconds, considerably longer than the motor's time constant.

To lower the damping by half means that k_i must be increased by a factor of 4, because k_i is in the square root in the denominator.

$$\xi = \frac{1}{2\sqrt{mk_i\tau}}$$

Figure 5-16 shows the transient response for the same system with the same set point step, only k_i has changed. With a larger integral constant, the damping drops, producing a substantial overshoot before reducing the residual error to zero.



MULTISIM

Figure 5-16 Effect of increasing k_i by 4 lowers the damping by ½ leading to ringing

Proportional-Integral Control

Proportional control provides a quick response to a step in set point, but there is residual error. When the system settles, no longer changing in response to the set point change, the output , *PV*, may be as much as 50% away from the *SP*. Integral control drives residual error to zero, since the controller output, *CO*, continues to change until *error* = 0. But, to keep from overshooting and ringing back and forth, that process may be considerably longer than 5τ, the system's open loop full response time.

Integral control gives *good* results, proportional control gives *fast* results. Does combining the two types of controllers produces *good*, *fast* results?

controller servo

set point SP + Σ error – $H(s)$ $G(s)$ *PV*

process variable PV

$$\frac{\boldsymbol{PV}}{\boldsymbol{SP}} = \frac{\boldsymbol{GH}}{(\mathbf{1} + \boldsymbol{GH})}$$

Transfer function

$$H_{\text{prop}} = k_{\text{p}} \qquad H_{\text{integral}} = \frac{k_{\text{i}}}{s}$$

$$H_{\text{pi}} = k_{\text{p}} + \frac{k_{\text{i}}}{s} = \frac{k_{\text{p}}s + k_{\text{i}}}{s}$$

Proportional-integral controller

The example DC motor is a first order lag system element.

$$G = \frac{m}{\tau s + 1}$$

The closed loop, servo tacking, system transfer function is

$$\frac{PV}{SP} = \frac{GH}{1 + GH}$$

$$\frac{PV}{SP} = \frac{\dfrac{m}{\tau s + 1} \times \dfrac{k_\mathrm{p} s + k_\mathrm{i}}{s}}{1 + \dfrac{m}{\tau s + 1} \times \dfrac{k_\mathrm{p} s + k_\mathrm{i}}{s}}$$

$$\frac{PV}{SP} = \frac{\dfrac{m\left(k_\mathrm{p} s + k_\mathrm{i}\right)}{s(\tau s + 1)}}{\dfrac{s(\tau s + 1) + m\left(k_\mathrm{p} s + k_\mathrm{i}\right)}{s(\tau s + 1)}}$$

$$\frac{PV}{SP} = \frac{m\left(k_\mathrm{p} s + k_\mathrm{i}\right)}{s(\tau s + 1) + m\left(k_\mathrm{p} s + k_\mathrm{i}\right)}$$

$$\frac{PV}{SP} = \frac{m k_\mathrm{p} s + m k_\mathrm{i}}{\tau s^2 + s + m k_\mathrm{p} s + m k_\mathrm{i}}$$

$$\frac{PV}{SP} = \frac{\dfrac{m k_\mathrm{p}}{\tau} s + \dfrac{m k_\mathrm{i}}{\tau}}{s^2 + \dfrac{1}{\tau} s + \dfrac{m k_\mathrm{p}}{\tau} s + \dfrac{m k_\mathrm{i}}{\tau}}$$

$$\frac{PV}{SP} = \frac{\dfrac{m k_\mathrm{p}}{\tau} s + \dfrac{m k_\mathrm{i}}{\tau}}{s^2 + \dfrac{1 + m k_\mathrm{p}}{\tau} s + \dfrac{m k_\mathrm{i}}{\tau}}$$

The denominator now looks like a standard second order system. But, the numerator has *two* terms. They can be separated and examined separately, then those results added. Remember, Laplace transforms are additive and superposition applies.

$$\frac{PV}{SP} = \frac{\frac{mk_\mathrm{p}}{\tau}s}{s^2 + \frac{1+mk_\mathrm{p}}{\tau}s + \frac{mk_\mathrm{i}}{\tau}} + \frac{\frac{mk_\mathrm{i}}{\tau}}{s^2 + \frac{1+mk_\mathrm{p}}{\tau}s + \frac{mk_\mathrm{i}}{\tau}}$$

Standard second order form

$$\boldsymbol{\frac{A\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}}$$

The second term looks familiar. Equating coefficients gives

$$\omega_\mathrm{n}^2 = \frac{mk_\mathrm{i}}{\tau} \qquad \omega_\mathrm{n} = \sqrt{\frac{mk_\mathrm{i}}{\tau}}$$

$$A\omega_\mathrm{n}^2 = \frac{mk_\mathrm{i}}{\tau} \qquad A = 1$$

$$2\xi\omega_\mathrm{n} = \frac{1+mk_\mathrm{p}}{\tau}$$

$$\xi = \frac{1+mk_\mathrm{p}}{2\omega_\mathrm{n}\tau}$$

$$\xi = \frac{1+mk_\mathrm{p}}{2\tau\sqrt{\frac{mk_\mathrm{i}}{\tau}}}$$

$$\xi = \frac{1+mk_\mathrm{p}}{2\sqrt{mk_\mathrm{i}\tau}}$$

For full-scale performance without driving the controller to its limits, set

$$\boldsymbol{k_\mathrm{p} = \frac{1}{m}}$$

$$\xi = \frac{1+m\frac{1}{m}}{2\sqrt{mk_\mathrm{i}\tau}} = \frac{1+1}{2\sqrt{mk_\mathrm{i}\tau}}$$

$$\xi = \frac{1}{\sqrt{mk_\mathrm{i}\tau}}$$

$$\boldsymbol{k_\mathrm{i} = \frac{1}{m\tau\xi^2}}$$

$$\frac{PV}{SP} = \frac{\frac{mk_{\mathrm{p}}}{\tau}s}{s^2 + \frac{1+mk_{\mathrm{p}}}{\tau}s + \frac{mk_{\mathrm{i}}}{\tau}} + \frac{\frac{mk_{\mathrm{i}}}{\tau}}{s^2 + \frac{1+mk_{\mathrm{p}}}{\tau}s + \frac{mk_{\mathrm{i}}}{\tau}}$$

The second term produces a typical over damped, or critically damped, or under damped response to a step, eventually settling to *PV* = SP. How the system's components parameters affect that response can be calculated with A, ξ, and ω_{n}.

The first term gives a similar response, but is multiplied by *s*. In the Laplace domain, when a function is multiplied by *s*, then its time domain function is differentiated, i.e. slope or rate-of-change. When the output from the standard second order term is rising, the output from the first term is going to be a big positive, increasing the speed of the response. But, eventually, when the second term is constant, the first term falls to zero (the derivative of a constant is zero). So, the combined response is going to be faster than the second term alone, but will eventually settle to *PV* = *SP* and residual error = 0.

Look at Figure 5-17.

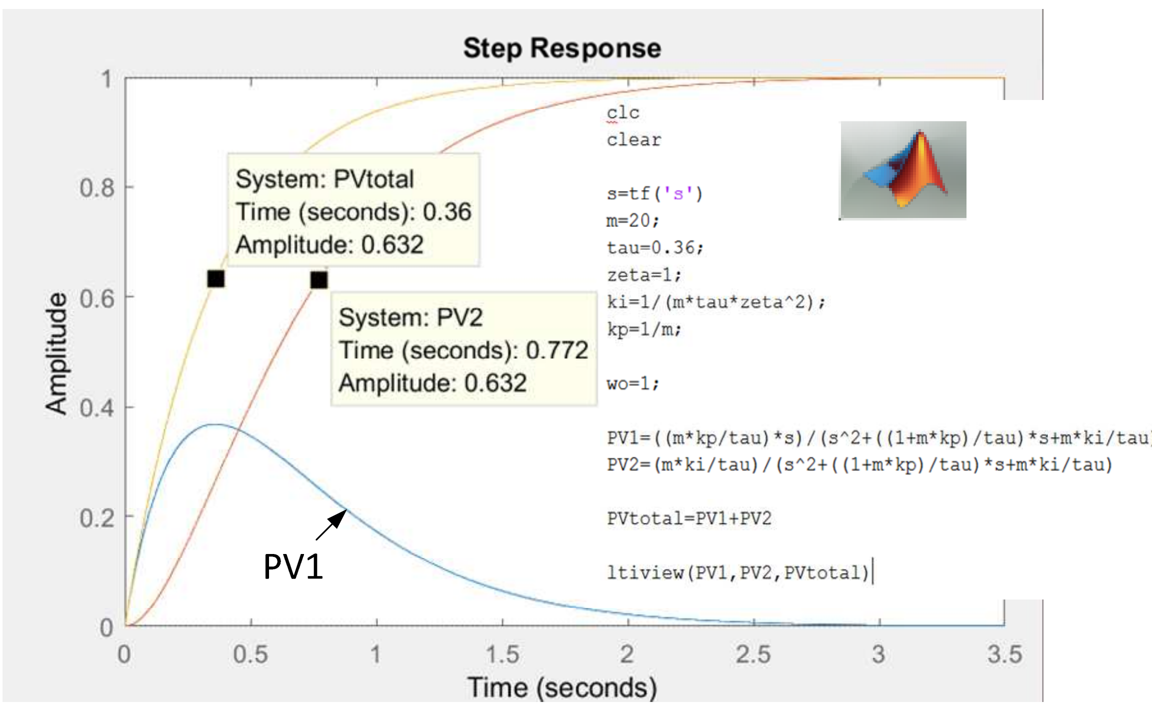


Figure 5-17 Matlab ltiview of two term step response

The second term has its damping set to one, critically damped, and it does not overshoot. The first term gives a bump at first, driving the combined terms up much more quickly than the critical second order response alone.

From Example 4-2

m= 206 RPM/V
τ= 0.362 sec

$$G = \frac{m}{(\tau s + 1)}$$

Example 5-4

Calculate k_i for the motor from previous examples to provide a response to a step from 227 RPM to 1030 RPM with

a. ξ = 0.5 under damped
b. ξ = 1.0 critically damped
c. ξ = 1.5 over damped

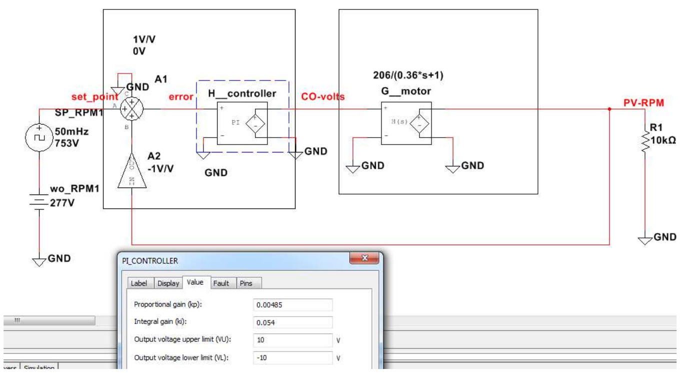
For each damping value, use k_p = 1/m, then run a simulation to determine the residual error and the time to 63% of PV_{final}

Solutions

a.

$$k_{\mathrm{i}} = \frac{1}{m\tau\xi^2} = \frac{1}{206 \times 0.36\ \mathrm{sec} \times 0.5^2} = 0.054/_{\mathrm{sec}}$$

MULTISIM



(a) Simulation schematic with ξ = 0.5

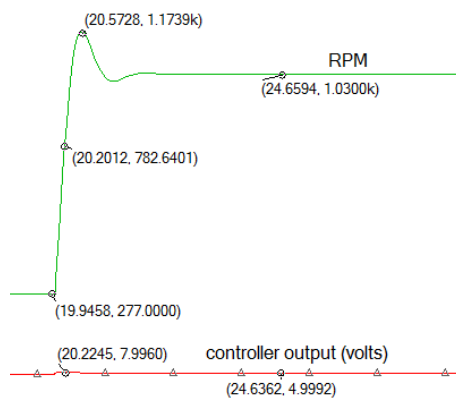


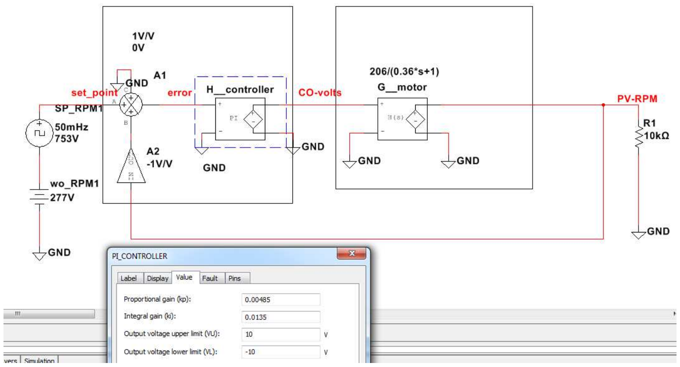
Figure 5-18 Example 5-4 with ξ = 0.5

The set point steps to 1030 RPM and *eventually* the process variable settles to 1030 RPM. Residual error is zero, as expected for a controller with an integral element.

63% of the step from 277 RPM to 1030 RPM arrives at 754 RPM. It takes ~ 200 ms to reach this level. That is considerably smaller than the motor's time constant. However, there is a *large* overshoot, also typical of an under damped system.

b.

$$k_{\mathrm{i}} = \frac{1}{m\tau\xi^2} = \frac{1}{206 \times 0.36\ \mathrm{sec} \times 1^2} = {0.0135}/_{\mathrm{sec}}$$



(a) Simulation schematic with $\xi = 1$

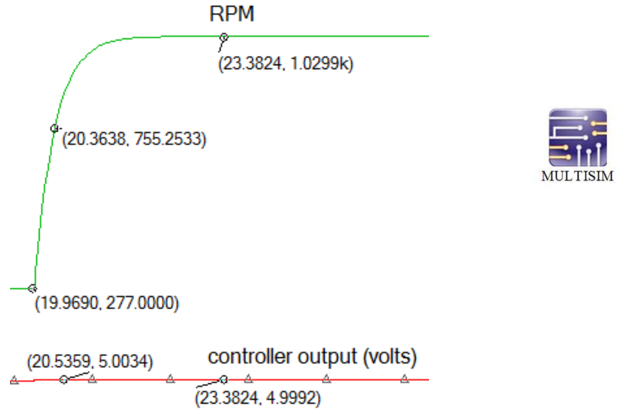


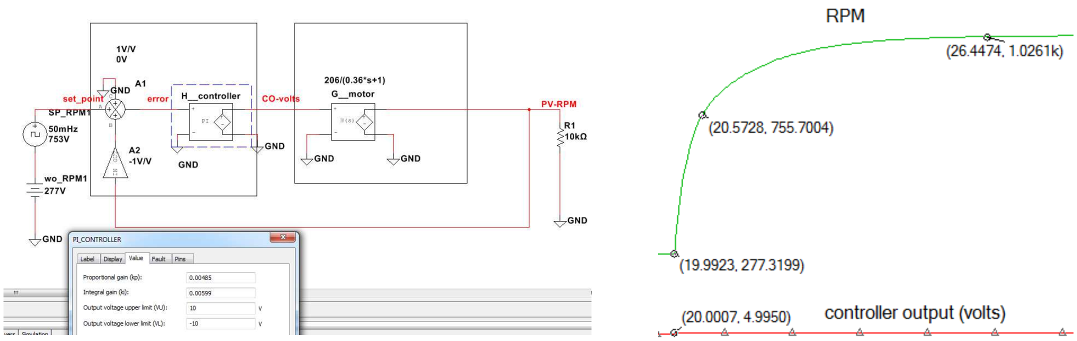
Figure 5-18 Example 5-4 with $\xi = 1$

The set point steps to 1030 RPM and *eventually* the process variable settles to 1030 RPM. Residual error is zero, as expected for a controller with an integral element.

63% of the step from 277 RPM to 1030 RPM arrives at 754 RPM. It takes ~ 390 ms to reach this level. That iscomparable to the motor's time constant. However, there is no overshoot, typical of a critically damped system.

c.

$$k_{\mathrm{i}} = \frac{1}{m\tau\xi^2} = \frac{1}{206 \times 0.36\ \mathrm{sec} \times 1.5^2} = 0.0617/_{\mathrm{sec}}$$



(a) Simulation schematic with $\xi = 1.5$ **(b)** Simulation schematic with $\xi = 1.5$

Figure 5-19 Example 5-4 with $\xi = 1.5$

The set point steps to 1030 RPM and the process variable rises quickly, to begin with, but then increments *slowly* until it eventually reaches 1030 RPM. Residual error is zero, as expected for a controller with an integral element.

63% of 617 RPM arrives at 754 RPM. It takes 580 ms to reach this level. That is slower than the critically damped system. The approach to the final value is even slower since the integral adds only a small amount each step. This system is over damped.

The purpose of closed loop, negative feedback, servo tracking is to cause the system's output. *PV*, to match and follow the set point. Combining proportional and integral control produces a response that perfectly tracks the steady state value of the set point, and moves as quickly as the controlled element (τ) can move. Proper adjustment of the k_{p} constant assures that the system is always under control. The integral constant, k_{i} then is adjusted to produce a critically damped system.

If the motor's characteristics change (m and τ), the performance of the *open loop* system varies proportionally. As the system ages, is damaged, and is repaired or up graded, its open loop performance changes.

However, the closed loop system compares the system performance, *PV* with the desired output, *SP*, and drives the motor differently to correct any error. The equation that determines *how* to drive the motor differently is the controller transfer function, *H*. For the PI controller

$$H = k_{\text{p}} + \frac{k_{\text{i}}}{s} = \frac{k_{\text{p}}s + k_{\text{i}}}{s}$$

Figure 5-20 compares the performance of three sets of different motor parameters running with and without a PI controller (k_p = 0.05, k_i = 0.139/sec). The *open* loop performance varies widely. But, with a Proportional Integral controller, the *closed* loop performance varies only slightly even for wide variations in the motor's parameters.

MULTISIM

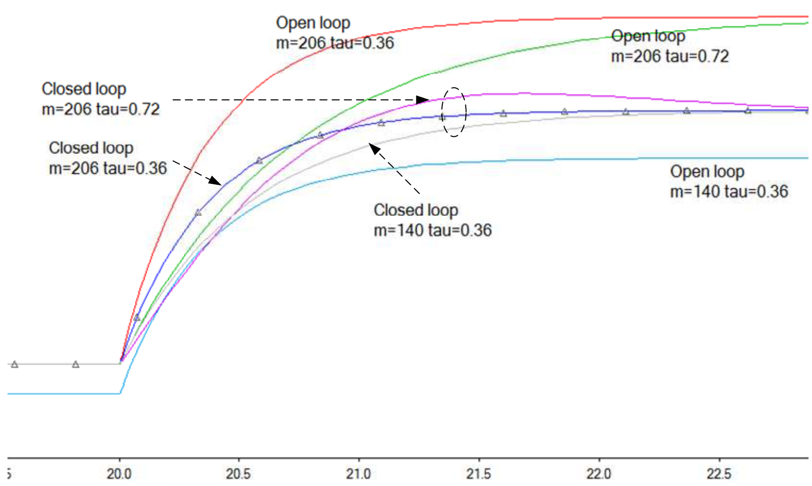


Figure 5-20 Open and PI closed loop performance comparisons

Derivative and PID Speed Control

Proportional control provides a quick response to a step in set point, but there is residual error. Adding the integral of the error to the controller's output *eventually* eliminates that residual error, driving the *PV* to match the *SP*, producing no steady state error.

The transient response, i.e. how quickly the system settles, can be improved by adding a term to the controller's output that is dependent on how quickly the error *changes*. How quickly the error changes is determined by its time derivative.