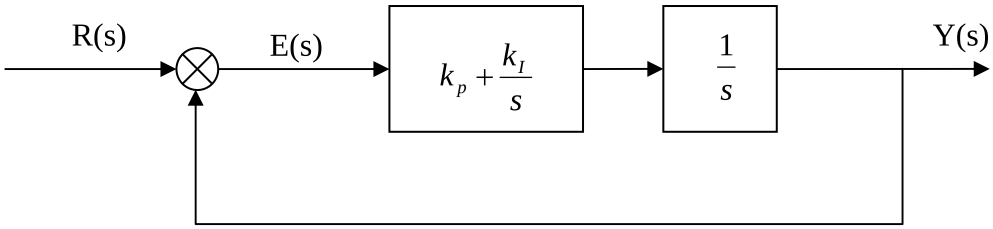
Lesson 27

Problem 27.1

A PI controller is shown below for a plant can be modeled as an integrator. The proportional gain is k_p and the integrator gain is k_I.



a) Determine the closed-loop transfer function as a ratio of simple polynomials.

b) Only using the denominator of your transfer function choose k_p and k_I to satisfy

%OS = 20% and Peak Time, Tp = 0.2 s.

c) Implement in Simulink and plot the unit step response, y(t). Use the subplot to also plot e(t). Does the response actually meet the specifications? Why not? (Hint: there are numerator dynamics!)

d) Also choose k_p and k_I values in order make the system critically damped. Since there is no overshoot for a critically damped system find the parameters so the settling time is the same as what would result from the ζ and ω_n found using the constraints shown in part b). Again simulate and plot y(t) and e(t). Compare and contrast these plots with those found in part (c). Include a printout of your Simulink model.

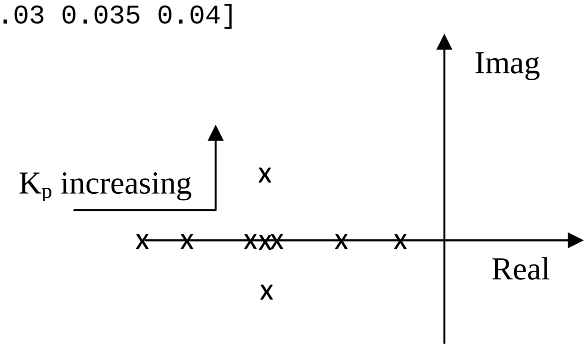
Problem 27.2

Explore pole (roots of the characteristic equation) locations with changing k_p. Start with the p-controller example from class so the characteristic equation is $Js^2 + Bs + k_p$ with B = 0.05 N-s and J = 0.025 kg-m^2. Plot the roots on a Imaginary-Real axes and use values of

k_p = `[0.001 0.005 0.01 0.015 0.02 0.0249 0.025 0.03 0.035 0.04]`

You should get something that looks like the following:

By hand, or using the computer, label the roots for k_p = 0.001, k_p = 0.04 and for the critically damped case.



You may make this plot in Matlab, Excel, Maple or any other program you'd like. If you use Matlab you will find the "roots" command helpful. You just type

> roots([J B kp]) where J, B and kp are numerical values. To find the real part of a number the Matlab command is "real(*number*)" and for the imaginary part you can use "imag(*number*)" where *number* is a complex number.