Solution to Problem 7.13

(a) is the same as (a) for Problem 7.12.

(b) See the root locus in part (b) of 7.12. The branches of the root locus run right along the imaginary axis for gains higher than the gain where the breakaway point occurs, resulting in a marginally stable closed loop system. To stabilize the system, we will need to “bend” the root locus branches to the left and into the left half-plane. This requires **lead compensation** of some sort, which could be a PD controller (as in 7.12), a phase lead compensator, or even a PID controller, though a PID will require a compensator zero to lie near the compensator pole at the origin to avoid stability problems.

(c) We can use the design equations (7-53) for the compensator form of (7-46). The dc gain is given as 0.1, so \( a_0 \) is 0.1. We find \( a_1 \) and \( b_1 \) from the formulas. As in Problem 7.12 where the same values of \(-0.25 \pm j0.25\) were given for the desired closed loop poles, we have \( s_i = .3535 \) angle(135 deg) and \( \beta = 135 \) deg. Also, just like Problem 7.12, we have that \( G(s_i) = 7.321 \) angle(103.4 deg), which means \( \Psi = 103.4 \) deg. So:

\[
\begin{align*}
  a_1 &= \frac{\sin(\beta) + a_0 |G(s_i)| \sin(\beta - \Psi)}{|s_i| |G(s_i)| \sin(\psi)} = \frac{\sin(135^\circ) + (0.1)(7.321) \sin(31.6^\circ)}{(0.3535)(7.321) \sin(103.4^\circ)} = 0.4333 \\
  b_1 &= \frac{\sin(\beta + \Psi) + a_0 |G(s_i)| \sin(\beta)}{-|s_i| \sin(\psi)} = \frac{\sin(135^\circ + 103.4^\circ) + (0.1)(7.321) \sin(135^\circ)}{-0.3535 \sin(103.4^\circ)} = 0.9714
\end{align*}
\]

So, the compensator transfer function is:

\[
G_C(s) = \frac{0.4333 s + 0.1}{0.9714 s + 1} = \frac{0.1 (s/0.2308 + 1)}{s/1.0294 + 1} = \frac{0.4333 (s + 0.2308)}{s + 1.0294}
\]

where the transfer function is written above in both Bode form (illustrating directly the dc gain of 0.1) and in the more standard transfer function form.

(d) We can verify this just by checking the roots of the denominator of the closed loop transfer function (called ‘dencl’ below):

```matlab
» numol=[0.9407];denol=[1 0 -0.0297];
» numc=[0.4333 0.1];denc=[0.9714 1];
» dencl=conv(denc,denol)+[0 0 conv(numc,numol)]
   dencl =
         9.7140e-001  1.0000e+000  3.7875e-001  6.4370e-002
» roots(dencl)
ans =
    -5.2935e-001
   -2.5005e-001 +2.5032e-001i
   -2.5005e-001 -2.5032e-001i
```
Notice that there is now a third closed loop pole at -.5294. Its real part is negative and more than twice as large as the real part of the desired closed loop poles, so the time response terms that are due to the extra pole converge twice as fast as the ones for the desired closed loop poles, and hence these terms contribute little to the total response after a brief transient period.

Another way to check the closed loop poles is to draw the root locus for the compensated system, see if the locus passes through the desired closed loop poles, and then to make sure the gain is the right one to get the desired closed loop poles. The root locus drawn by Matlab for the compensated system is at the right (you will have to figure out the arrows on the branches), and it does indeed pass through the desired closed loop poles. We get the required compensator gain from the magnitude criterion. Using the polynomials defined above to find the closed loop poles, we solve for $1/|GCG|$ evaluated at the desired closed loop pole $s_1 = -0.25 + j0.25$ to get the compensator gain required. Since the compensator we got already had the gain specified, we should get unity for the “additional” gain required:

```matlab
» s1=-.25+i*.25;
» k_c=abs(polyval(denc,s1)*polyval(denol,s1)/polyval(numc,s1)/polyval(numol,s1))
k_c =
9.9959e-001
```