13.49 Homework #6

1. The parameters governing the surface maneuvering of a high-speed container ship are given below for reference:

m'	0.00792
I'_{zz}	0.000456
x'_G	-0.05
L	$175.0m$
U	$8m/s$
$Y'_{\dot{v}}$	-0.00705
$Y'_{\dot{r}}$	0.0000
Y'_v	-0.0116
Y'_r	0.00242
Y'_δ	-0.00258
$N'_{\dot{v}}$	0.0000
$N'_{\dot{r}}$	-0.000419
N'_v	-0.00385
N'_r	-0.00222
N'_δ	0.00126

Note that the center of vessel mass is located *aft* of the origin; for this model, the origin coincides with the center of added mass, so that $Y_{\dot{r}} = N_{\dot{v}} = 0$. The nondimensional system with states $\vec{x}' = [v', r']$ evolves according to $d\vec{x}'/dt' = A\vec{x}' + B\delta$, where

$$A = \begin{bmatrix} -0.90 & -0.42 \\ -4.8 & -2.3 \end{bmatrix}, \quad B = \begin{Bmatrix} -0.13 \\ 1.4 \end{Bmatrix}.$$

The relevant output is yaw rate: $C = [0 \;\; 1]$ and $D = 0$. For the purposes of autopilot design, however, the transfer function $\phi(s)/\delta(s)$ is needed; this is a result from Quiz 1.

The following steps create two heading autopilots, using the root-locus and loopshaping techniques. In addition to the Matlab commands listed below, you will find very useful the convolution function `conv()` which can be used to combine systems, e.g., for numerators, `numPC = conv(numP,numC);`. Also, be sure that you equalize axis scaling for your plots in the complex-plane, by using `axis('equal');`.

(a) Use the Matlab command `tf()`, or `ss()`, to create a system model of the open-loop transfer function $P(s)C(s)$, using the plant above and a PID-type controller:

$$C(s) = k_p \left(1 + \tau_d s + \frac{1}{\tau_i s}\right). \tag{0.1}$$

The actual numerical values for k_p, τ_d, and τ_i are to be found in the next step.

(b) Using $\tau_d = 2$ and $\tau_i = 6$ as suggested values, use the Matlab command `rlocus()` and then `rlocfind()` to select a controller gain k_p, that puts the three slow poles in the following sector: 1) minimum undamped frequency (nondimensional) of 0.3, 2) maximum frequency of 0.5, and 3) minimum damping ratio 0.7. Give a root locus plot, with your pole locations clearly marked on top of the trajectories taken as k_p varies. You don't need to show the fourth, fast pole, which will be quite far to the left.

(c) Apply the k_p you selected to $P(s)C(s)$, and then use the Matlab functions `feedback()` to create the resulting feedback system, and `step()` to plot the closed-loop system response to a step input in desired heading.

(d) Use the Matlab command `nyquist()` to make a Nyquist plot of $P(s)C(s)$ for your design. Make a visual estimate of the gain and phase margins.

(e) An alternate approach for controller design of this stable plant is loopshaping: For the open-loop function $L(s) = \omega_c/s$, where $\omega_c = 2.0$, invert the plant to come up with a compensator: $C(s) = L(s)/P(s)$. This design has infinite gain margin and 90 degrees phase margin.

(f) As above, create the feedback system, and plot the closed-loop step response.

(g) The loopshaping control is *not quite* a PID-controller; how does it differ, and what would $L(s)$ have to contain to make it a PID?

(h) The controllers you just designed are in nondimensional time coordinates; give the P,I, and D gains for use on a real time scale, for the root-locus design.

2. Following on the LQR example in class, consider the state-space system and LQR design:

$$
\begin{aligned}
A &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \\
B &= [1 \;\; 0]^T \\
C &= [0 \;\; 1] \\
D &= 0 \\
Q &= C^T C \\
R &= \rho.
\end{aligned}
$$

The plant is an undamped oscillator with undamped poles at $\pm j$. Note that the plant output is *position* for this problem.

(a) What is the control gain K in terms of ρ? **Hints:** There are two solutions for p_{12}; choose the positive one. Also, the expression for p_{22} is messy; luckily, you won't need to use it.

(b) Determine the limiting approximations for K with ρ very small and very large – these are the cheap control and expensive control problems.

(c) Derive the limiting closed-loop pole locations for $\rho \longrightarrow 0$, giving the frequency and damping ratio of the Butterworth pattern in terms of ρ. You can get the characteristic equation for the poles as $det(sI - (A - BK)) = 0$, and then make it fit the form $s^2 + 2\zeta\omega_n + \omega_n^2 = 0$.