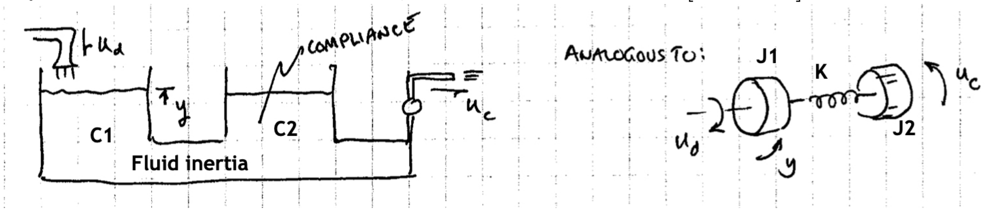
For the 'three-state' system example discussed in lecture, you should have proven that the system is completely controllable and observable, the latter when $\mathbf{C} = [1 \quad 0 \quad 0]$.



Problem 1. Design a full-state feedback (regulator) control (ref. Dorf and Bishop, Section 11.3) assuming all states are measured. Assume that you want to place the three poles for the controlled system at: $-2.12 \pm j2.12$ and -2.12. Find the gain matrix, $\mathbf{K}$.

Problem 2. For the three-state system, use Matlab to solve for the initial condition response assuming $x_1(0) = 1$, $x_2(0) = 0$, and $x_3(0) = 0.5$. Compare the 'uncontrolled' initial condition response to the response when you implement the regulator from Problem 1 above. For the controlled case, also include a plot of the actuator output, $u = -\mathbf{K}\mathbf{x}$.
Note: A Matlab script (three_state_reg_IC.m) is posted on the lecture summary for 12/2/21.

NOTE: we did not cover observer design in our quick overview of state-variable feedback (see Dorf and Bishop, Section 11.4), but this is a topic you may want to be aware of and review at some point.

The next page provides a proposed extra credit related to state-variable feedback control.

Extra credit. Study either Section 11.6 or 11.8 and show me how you would develop a state-variable feedback system with the three-state system (or equivalent of order $n \geq 3$) that has state x_1 tracking a reference input $r(t)$.

For the three-state example, assume the system is initially at an initial state $[1 \quad 0 \quad 0]$, and the initial reference is also at $r(0) = 1$. Let $r(t)$ then take on values as follows:

```
if (t>1 && t<10)
    r = ro + 0.25;
elseif t≥10 && t<30
    r = ro - 0.5;
elseif t≥30 && t<40
    r = ro + 0.25;
else
    r = ro;
end
```

Run a simulation out for about 50 seconds to show how the states behave. Plot results as follows on a single figure:

```
subplot(4,1,1), plot(time,error), legend('e = y-r')
subplot(4,1,2), plot(time,control input), legend('uc')
subplot(4,1,3), plot(time,r,time,x1,time,x3)
subplot(4,1,4), plot(time,x2)
```

You can get 50% just for the formulation and then another 50% for the simulation.