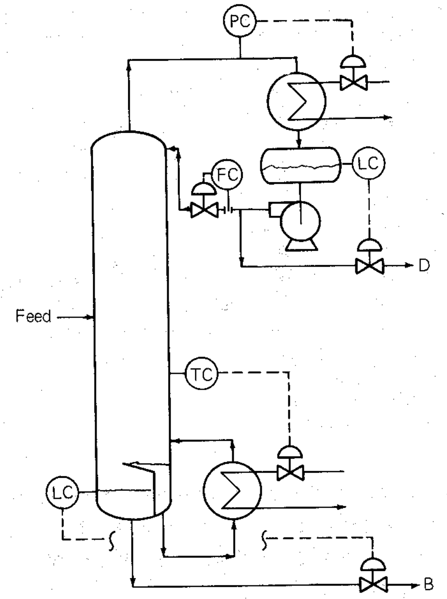
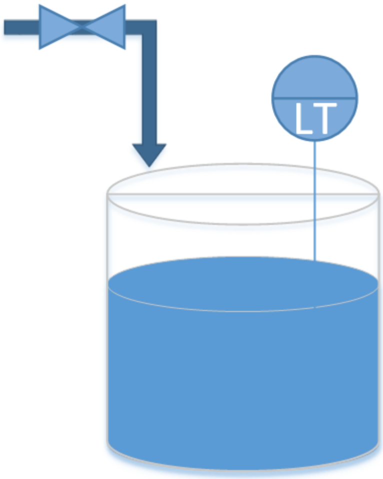
Chemical Engineering 436
Special Problem #1

A) The distillation column below includes five control loops. For each of these loops, identify the controlled variable, the manipulated variable, the type of sensor, and the final control element. Also, draw the block diagram for each of the control loops.



(figure from Distillation Operation, by H. Z. Kister, McGraw-Hill, New York, p. 498 ,1990)

B) Design a P-only controller for the following tank to maintain a level set point of 10.0 *m* for the following tank. The cylindrical tank has no outlet flow and an adjustable inlet flow. The inlet flow rate is not measured but there is a level measurement that shows how much fluid has been added to the tank. The objective of this exercise is to develop a model and controller that can maintain a certain water level by automatically adjusting the inlet flow rate. Show the P-only controller action with different values of K_c, the controller gain.



Solve for the height of the tank by integrating the mass balance equation for a period of 10 seconds. Use a value of *1000* kg/m^3 for density and *1.0* m^2 for the cross-sectional area of the tank. For the valve, assume a valve coefficient of *c=50.0 (kg/s / %open)*. Make sure that the valve does not exceed the limits of 0% to 100% by clipping the requested valve opening to an acceptable range. For example, if the P-only controller calculates a valve opening of 150%, use 100% instead.

For a discussion of the modeling of this tank and a starting Python script, see http://apmonitor.com/pdc/index.php/Main/DynamicModeling or the script below.

For more information on P-only controllers visit http://apmonitor.com/che436/index.php/Main/LectureNotes5

```
# Simulation Script

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# define tank model
def tank(Level,time,c,valve):
    rho = 1000.0 # water density (kg/m^3)
    A = 1.0      # tank area (m^2)
    # calculate derivative of the Level
    dLevel_dt = (c/(rho*A)) * valve
    return dLevel_dt

# time span for the simulation for 10 sec, every 0.1 sec
ts = np.linspace(0,10,101)

# valve operation
c = 50.0          # valve coefficient (kg/s / %open)
u = np.zeros(101) # u = valve % open
u[21:70] = 100.0  # open valve between 2 and 7 seconds

# level initial condition
Level0 = 0

# for storing the results
z = np.zeros(101)

# simulate with ODEINT
for i in range(100):
    valve = u[i+1]  # change this line to implement P-only controller
    y = odeint(tank,Level0,[0,0.1],args=(c,valve))
    Level0 = y[-1] # take the last point
    z[i+1] = Level0 # store the level for plotting

# plot results
plt.figure()
plt.subplot(2,1,1)
plt.plot(ts,z,'b-',linewidth=3)
plt.ylabel('Tank Level')
plt.subplot(2,1,2)
plt.plot(ts,u,'r--',linewidth=3)
plt.ylabel('Valve')
plt.xlabel('Time (sec)')
plt.show()
```