**PSET 1 – Due at 12:45 PM on Aug 28**

1. Sign into SLACK and setup a profile with a picture so Dr. Reuel can remember your name. We will use SLACK to collaborate in class and out of class. See email from Dr. Reuel for the sign up link.
	1. **[See SLACK]**
2. Consider the following PID diagram for a distillation column. How many control loops are there? For each, identify the controlled variable, the manipulated variable, the type of sensor, and the final control element. What are some possible disturbance variables?



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | CV | MV | Type of Sensor | Final Control Element |
| 1 | Pressure head of column | Flow rate to of cold water to condenser | Pressure | Control valve (liquid) |
| 2 | Flow rate | Flow rate of reflux to top of column | Flow | Control valve (liquid) |
| 3 | Liquid level in reflux drum | Outlet valve for distillate | Level sensor | Control valve (liquid) |
| 4 | Temperature in column | Flow of hot water to reboiler | Temperature | Control valve (liquid) |
| 5 | Liquid level at bottom of column | Flow rate of bottoms | Level sensor | Control valve (liquid) |

Disturbance Variables = changes in feed quality, environmental conditions, temperature of hot and cold water lines, corrosion/age of equipment, etc.

1. What is the purpose of a controller? How does it operate?

The controller is the ‘brain’ of the control system. It takes in the transmitted signal from the sensor (a.k.a. analyzer) and compares it to a set point. It then computes the ‘error’ or difference between the analyzed signal level and desired set point and determines what level of actuation is needed to improve the system. It then sends out the actuation signal which is delivered to the final control element. As we saw in class, controllers can be directed by mechanical, pneumatic, electronic, and software components. Most modern controllers are computer chips (microcontrollers) with the tuned control method stored in memory (software). This allows for rapid changes based on changes in process operation.

1. Refresh your memory on how MATLAB can solve ODE problems. What functions are available and what does the syntax look like?

There are many built in MATLAB ODE solvers. A table of these and the rationale in choosing which one is copied below (Matlab help file). The ones that are most common are **ode45** for most problems and **ode15s** for ‘stiff’ problems. For the following single ODE:

where C\* = 0.517 (example we did in class)

I will demonstrate the syntax using in-line anonymous functions:

[t,x] = ode45(@(t,x) (0.517-x)/3, [0 25], 0.5);

plot(t,x)

Also remember that ode45 can be efficiently used with sub-functions (especially for systems of ODEs and higher order ODEs – we will review this on Tuesday): 

function Problem23p10

% Coded by NFR on 11.28.2017

%

yo = [pi/2; 0];

tspan = [0 100];

[tp yp] = ode45(@sys,tspan,yo);

plot(tp,yp(:,1))

end

function dy = sys(t,y)

% y(1) = theta

% y(2) = h

dy = [y(2);

 -9.8/.6\*sin(y(1))];

end



1. Form a team of 3-4 people. This will be for the class LAB + team homework problems. Record your team here for Dillon (our TA) to manage. If you cannot find a team, please email Dillon (dghurd@iastate.edu) and he’ll help you out.

<https://docs.google.com/spreadsheets/d/13eIqJsnzHsw46DnWJ_IzEvRoqqoOSj2JM4CATiEvajs/edit?usp=sharing>