Rotation #3
COVID19 replacement lab

controller tuning

What you would have done:

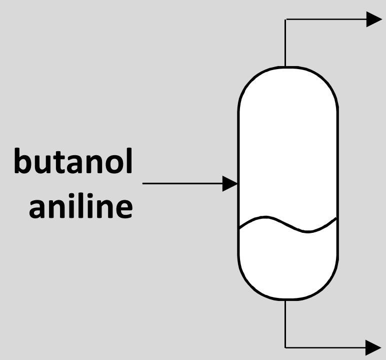
(control on actual equipment)

What you could have done:

(control in HYSYS, but the virtual desktop is too laggy)*

What you will do:

(control in a custom Matlab **≥2019a** simulation)



* the HYSYS simulation is on Canvas for inspection if interested

This exercise answers ***<u>the</u>*** question that ***<u>always</u>*** arises in controls engineering job interviews:

"How do YOU tune a PI controller?"

ANSWER: it depends & I certainly defer to your practices, BUT:

0) know your process*

What keeps the system safe?
Is the controller direct or reverse acting?
From data, calculations, or calibrations what's K_p, τ & θ?

online tuning?

1) **SET** $K_c = 1/K_p$
SET $\tau_I = 2(\tau + \theta)$‡

2) increase K_c until process is responsive to setpoint changes.

3) decrease τ_I until controller removes setpoint offsets aggressively‡

usually what interviewer wants to hear

performance tuning?

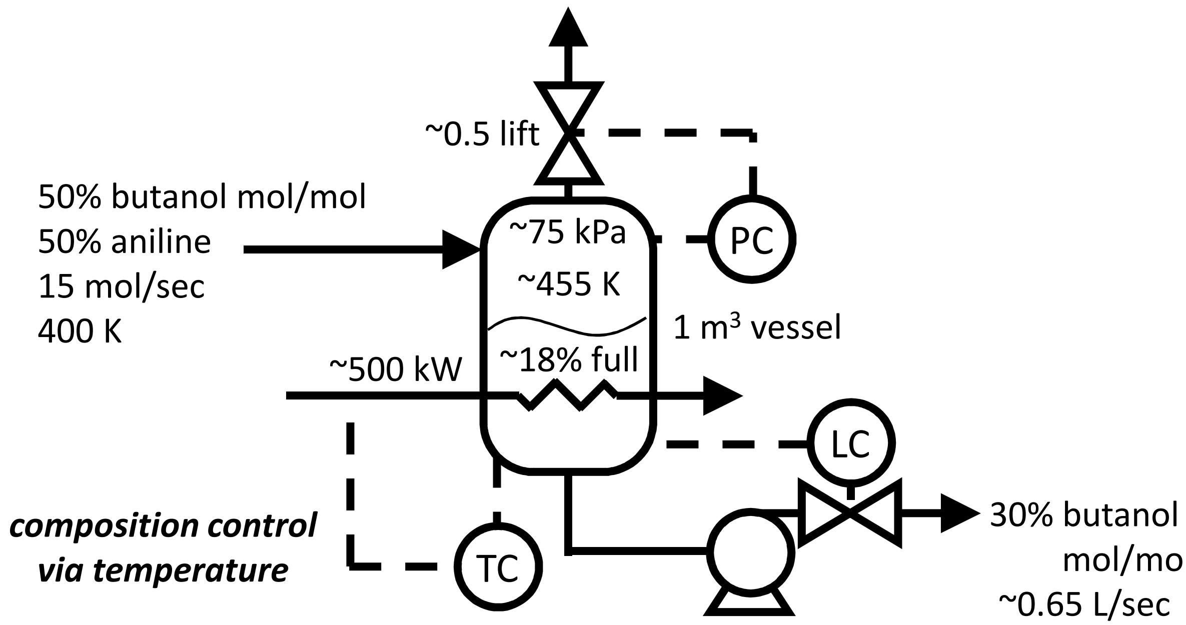
a) input step test for process model

b) Nyquist Plot to select K_c & τ_I via robustness & model tolerance

‡) for integrating/inventory/level $\tau_I = \infty$

*) wholly reconfigure controllers OR do performance tuning on inverse processes.

You will execute multiloop control on a flash unit:



<u>Follow Page Buckley's flow for multiloop controller tuning</u>

(Luyben & Luyben <u>Plantwide Process Control</u>, McGraw-Hill 1998)

tune controllers on inventory processes*

↓

tune controllers on fast processes

↓

tune controllers on slow processes

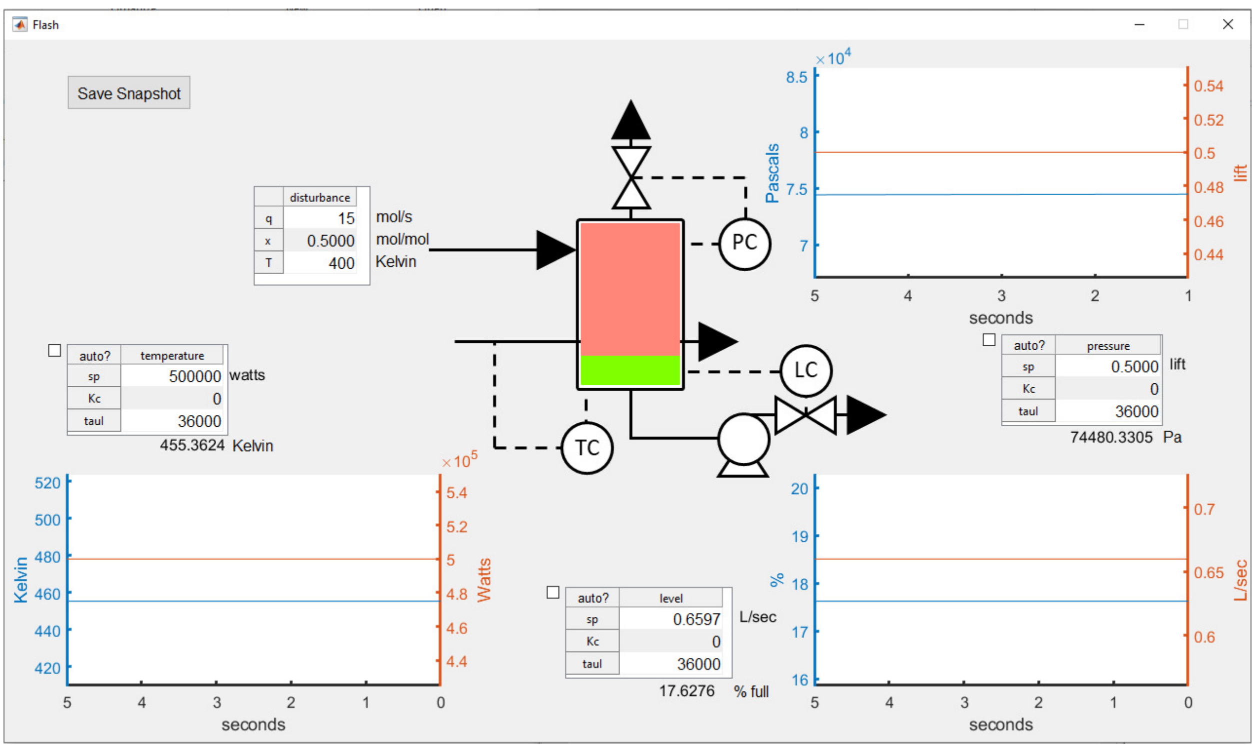
* start with safety & equipment critical controllers first.
(which insures you never run heater(s) or pump(s) dry)

Matlab: complete missing lines of `PI(…)`:

```
% mv is the last mv
% Kc is the PI controller gain
% tauI is the PI controller integral time
% errors is the accumulated integral error
% stpt is the user inputted setpoint (mv in manual mode, pv in auto)
% auto is 0 in manual control & 1 in automatic PI control
function [mv errors] = PI(obj,Kc,tauI,errors,mv,cv,stpt,auto)
      if auto
            err    = 0;         % current controller error
            errors = 0;         % acumúlate control error
            dmv    = 0;         % position mode PI calculation w/Kc & tauI
            mv = mv + dmv;      % position mode PI intervention
      else
            mv     = stpt;       % fully manual mode
            errors = 0 ;         % reset the integral accumulated error
      end
      mv = max(mv,sqrt(eps('double'))); % for numerics disallow mv = 0
end
```

then run `>> Separator(2)` in Matlab's command line:

(insure `Separator.p`, `PI.m` & `flash.png` share the same directory)



"2" is the simulation acceleration.

If your simulation performs poorly restart with <2

EXCEL Sheet 1) safe controller initialization

Calculate from 1st principles the level K_p.
(SEE HELP LAST PAGE)
(show your calculation in the spreadsheet)

Based on pg 2 SET the level controller K_c in simulation.

Start the level controller. ☑ **autos?**

Calculate from 1st principles the temperature K_p.
(SEE HELP LAST PAGE)
Calculate from 1st principles the temperature τ.
(show your calculation in the spreadsheet)

Based on pg 2 SET the temperature K_c in simulation.
Based on pg 2 SET the temperature τ in simulation.

Start the temperature controller. ☑ **autos?**

Estimate pressure K_p & τ from a -0.05 step in lift.
(report your estimates with units in the spreadsheet)

Based on pg 2 SET the pressure K_c in simulation.
Based on pg 2 SET the pressure τ_I in simulation.

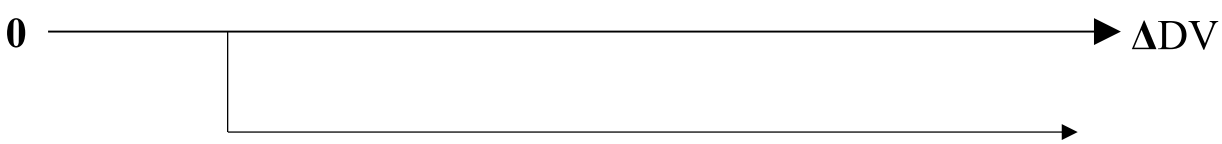
Start the pressure controller. ☑ **autos?**

EXCEL Sheet 2) untuned controller performance

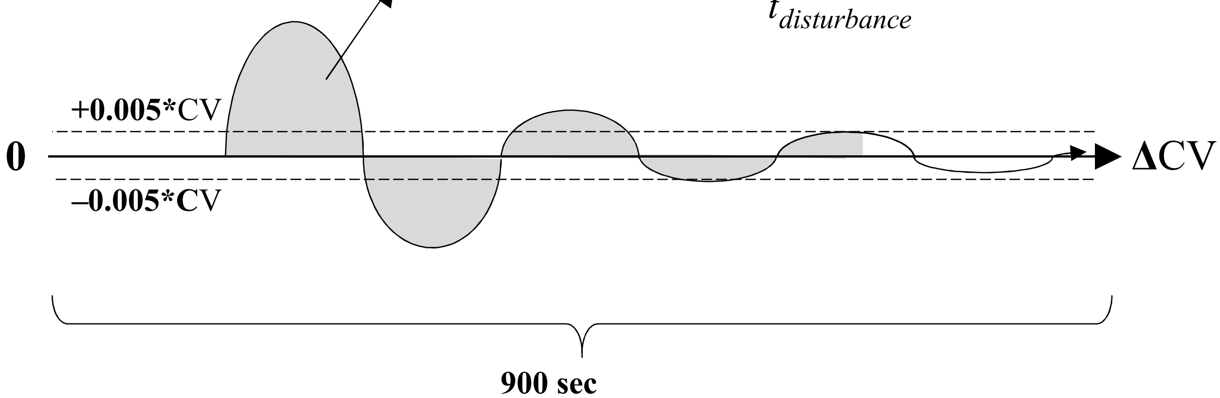
Save a 900 sec snapshot of system performance
& paste a screenshot of the simulation in the Excel sheet
for a simultaneous 33% turndown in mole feed & 6.25% feed temperature increase.

Save Snapshot → **Writes all current data in the shown plots to the current Matlab working directory as "snapshot.xls."**

Find the integral squared error of all three controlled variables for a settling time of ±0.5% the process value.



$$\text{integral squared error} = \int_{t_{disturbance}}^{t_{\pm 0.5\%}} \Delta e(t)^2 \, dt$$



EXCEL Sheet 3) tuned controller performance

Reset the simulation either:

by resetting the feed to 15 mol/sec
and waiting for steady state.

by reopening and setting all controllers (page 5).

Tune the three controllers in Page Buckley's order (RIP).
(page 3)

Use setpoint changes ala 2) & 3)
from the workflow on page 2.

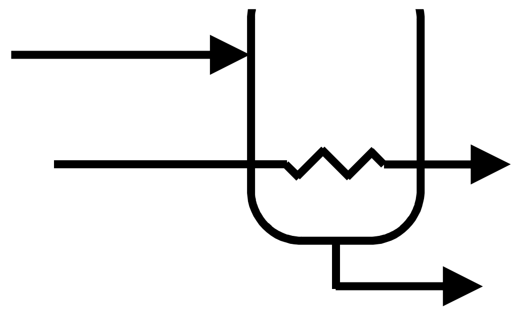
Page S. Buckley

Save a 900 sec snapshot of system performance
& paste a screenshot of the simulation in the Excel sheet
for a simultaneous 33% turndown in mole feed & 6.25% feed temperature increase.

Find the <u>integral squared error</u> of all three controlled variables for a settling time of ±0.5% the process value.

the performance of your tuning is 30% this lab grade.

Model: ponder the system
as a simple heated tank



(ignore VLE and the vapor stream)

For level K_p, start from a material balance:

$$\frac{dV}{dt} = ? \xrightarrow{\text{unit conversions}} \frac{d\%}{dt} = ?$$

and remember the desired input-output standard model form:

$$\frac{dx}{dt} = K_p \Delta u$$

For thermal K_p & τ start from a thermal balance:

$$\frac{d[MC_p T]}{dt} = ?$$

and remember the desired input-output standard model form:

$$\tau \frac{dx}{dt} + \Delta x = K_p \Delta u$$