Cambridge University Engineering Department

**DESIGN PROJECT GF1: CONTROL SYSTEM**

**Week 3: Controllers for $X2$ and $P2$. Robustness.**

# 1 Controllers for $X2$ and $P2$

Design, implement and test a PI controller for the operating pressure $P2$, using the 'linearise + Bode plot' method, as you did for the $L2$ loop last week. Use the cooling water flow rate $F200$ as the control variable. (Section 4.5 and Appendix D of [3], and section 4.5 of [4], discuss how pairings of control inputs and outputs can be decided. But this is a very imprecise black art, which we shall not probe in this project.) Leave the $L2$ controller in place when you do this.

*Note: If your simulation runs start with a long initial transient period during which the system recovers from incorrect initial conditions, set the correct initial conditions on each integrator before starting the simulation. This will save you a lot of time.*

Now leave both the $L2$ and $P2$ controllers in place, and design, implement and test a PI controller for the Product Composition $X2$, using the Steam Pressure $P100$ as the control variable. For this loop use a different, more heuristic, method of design, known as the 'Ziegler-Nichols' method [7]. This method is commonly used in practice. One of the Ziegler-Nichols procedures is to gradually increase the proportional gain until $X2$ just starts to oscillate with approximately constant amplitude. Call this gain $K_u$, and the period of oscillation $T_u$. Then set the proportional gain to $K_p = K_u/2.2$, and the integrator action time to $T_i = T_u/1.2$. (The transfer function of the PI controller is $K_p[1 + 1/(sT_i)]$.)

Leaving the $P2$ and $X2$ controllers in place, check the stability margins of the $L2$ loop, because closing the $P2$ and $X2$ loops might have changed the behaviour of the $L2$ loop. You need to 'break' the $L2$ loop in order to do this, for example at the input of the PI controller, then linearise and obtain the frequency response of the open loop. You should find that the phase margin is now rather low ($10 - 15$ degrees), but that you can increase it to about 25 degrees just by changing the proportional gain — note that the gain has to be *increased* to improve the phase margin, which is rather unusual. Change the gain accordingly. (Ideally, you should now check that the $P2$ and $X2$ loops are still ok, but let's go on.)

# 2 Robustness to process changes

A real process never behaves exactly as modelled, so it is important that controllers are reasonably robust to changes in the process behaviour. Investigate the robustness of your control system when the process parameters vary between the bounds shown in Table 1.

There are several ways of varying the parameters within *Simulink*:

1. Open the appropriate icon in the model, and edit the parameter field.
2. Use `Slider Gain` blocks.
3. Use a parameter in the model instead of a numerical value, and set the value of that parameter in the MATLAB window before running the simulation. For example, in the Separator sub-model, the `Gain` field of the amplifier can be set to `1/rhoA` instead of `1/20`; before running the simulation type `rhoA=20` or `rhoA=40` etc in the MATLAB window. If you find out how to run simulations from the MATLAB window instead of the *Simulink* window (using calls such as `[t,x,y]=sim('process_ss',...)`) this offers the possibility of running a whole set of simulations under the control of a MATLAB program — for example:

```
for rhoA = 10:10:40,
  for M = 15:5:25,
    ...
    [t,x,y]=sim('controlsys');
    ...
  end
end
```

   If you do this you will need to keep the results of each simulation run in the MATLAB workspace or, more safely, save them to file. (Type `help save` to see how to do this.) Also note that if you run `for` loops like these from a MATLAB function, rather than directly from the keyboard, then you will have to declare variables such as `rhoA` and `M` to be `global`, so that they are visible in the workspace.

| Parameter | Min | Max |
|---|---|---|
| $\rho A$ | 10 | 40 |
| $M$ | 15 | 25 |
| $UA2$ | 5.0 | 8.0 |

Table 1: Process parameter bounds

# 3 Integrator wind-up

A serious problem with the use of integral control is that an integrator can continue to integrate, even though the actual control variable which it determines has become saturated, and can no longer increase. This is frequently called 'integrator wind-up'. Usually this is the result of an error signal remaining large, and of the same sign, for a long time. When the error finally reduces to zero there is still a large control signal due to the integrator output, which now causes the error to become large in the opposite direction, since it takes a long time for the integrator output to come back to zero. Large overshoots are therefore induced in the controlled variable, and 'limit-cycle' oscillations may even result. You may have observed integrator wind-up already during initial transients of your simulations runs.

Various solutions are available for this problem. The most straightforward is to switch the integrator on only when the error is relatively small. A more sophisticated one is to monitor the integrator output, and to stop integrating when the output reaches some level, usually slightly smaller than the saturation level of the control variable.

See whether your $P2$ controller suffers from integrator wind-up if there is a pulse disturbance of 20% on $X1$ (*i.e.,* $X1$ increases to 25%), lasting for 100 minutes.

Insert some anti wind-up protection on the $P2$ controller. For more information about anti wind-up schemes see [1, 2].

## 4 Interim report

No report is needed this week. But remember to record material which you will need for your final report.

Specifically, you will need the controller designs you obtained, and how they performed, particularly when the plant parameters varied from their nominal values. You will also need to note how you investigated the robustness of the controllers, and the effects of including 'anti wind-up' in the $P2$ controller.

## References


[1] Åström,K. J., and Wittenmark, B., *Computer Controlled Systems — Theory and Design*, Englewood Cliffs: Prentice-Hall, 1997. (CUED Shelfmark QE 46)

[2] Franklin, G. F., Powell,J. D., and Emami-Naeini, A., *Feedback Control of Dynamic Systems, 4th edition*, Reading MA: Prentice-Hall, 2002. (CUED Shelfmark: QC 253)

[3] Newell, R. B., and Lee, P. L., *Applied Process Control, A Case Study*, New York: Prentice-Hall, 1989. (CUED Shelfmark: QL 30)

[4] Maciejowski, J. M., *Multivariable Feedback Design*, Wokingham: Addison-Wesley, 1989. (CUED Shelfmark QR 25)

[5] *Matlab User's Guide.*

[6] *Simulink User's Guide.*

[7] Ziegler, J. G., and Nichols, N. B., Optimum settings for automatic controllers, *Trans. ASME*, **64**, 1942, p.759.


R. Sepulchre May 2022