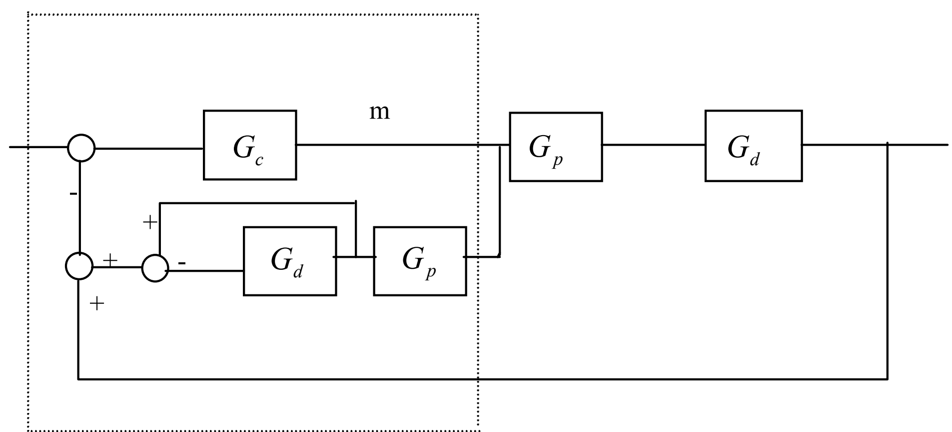
Lesson 15 **Dead Time Compensation Using the Smith Predictor**

Students will be able to:

1. Understand why dead time is a killer.
2. Understand how the "dead time" can be compensated by Smith predictor.

Procedure:

1. Show how the Bode plot is for dead time. It does not affect the amplitude ratio plot, but does add more phase angle lag as frequency increases.
2. Show the diagram of Smith predictor as follows and explain how it counter-acts the dead time. Process G_p is the portion of process without dead time. The dead time portion of the process is G_d. The technique used below is used to cancel the dead time portion of the process. Consider the path mG_pG_d in the outer loop. The same in the inner loop negates it. What is left is the G_p only. Therefore the controller can be tuned based on the process minus the time delay; thus allowing a higher gain. The dashed line box will have to be implemented by computer as the dead time compensator.
3. The method is based on the assumption that the model is identified correctly. If the disturbance process also has dead time, it is not negated based on this configuration. However a feed-forward controller can compensate that.



Discussion:

Summary of this unit:

1. Construct a third order process without dead time using "custom process".
2. Fit it to a FOPDT. Find the corresponding PI tuning.
3. Use this tuning to control the third order "real" process. (Base case)
4. Add dead time to the "real" process and control it with the same tuning.

5. Refit the process with a FOPDT (also SOPDT used later in step 9) and find the corresponding tuning.
6. Control the "real" process (third order with dead time) with the new tuning. You should notice that the performance has deteriorated because of the deadtime.
7. Select the Smith Predictor Control (SPC). This controller has a model imbedded in it. Enter the FOPDT model parameters obtained from step 5. The SPC also has a PI tuning in it. Hand-calculate (use the correlation) the PI value based on the FOPDT with a small, modified dead time (0.1 of the first order process time constant). Theoretically this dead time should be zero because the SPC should cancel the process dead time. We enter a small value to take into account the sampling time. Control the "real" process with the SPC. Theoretically the performance should equal that of the base case. But it is a little worse because of the dead time, albeit small. (How about adding the sampling time to the process dead time in the SPC and use the PI tuning obtained from step two? This is an exercise. I think the use of sampling time is merely to satisfy the correlation requirement.)
8. Study the effect of model mismatch of the SPC.
9. Use SOPDT for the SPC and the PI from step 2 (and therefore ignore the sampling time because we are not using the correlation for PI calculation). Control it (close the loop) with set point change. It should be better than FOPDT based SPC.
10. So far we have been monitoring set point tracking. How about disturbance rejection? I think it will not benefit from the cancellation of process dead time from the SPC. Summary of this step is:
    a. Test the disturbance rejection using process from step 1 (3^{rd} order w/o dead time), tunings from step 2 (not the SPC), and disturbance process same as process.
    b. Use the same tuning but add dead time of 25 to both the process and the disturbance process. Test the disturbance rejection. The performance should deteriorate.
    c. Use the FOPDT as the SPC model as the PI tuning from STEP 7 (correlation calculation). Repeat the disturbance rejection test. The result is uncertain. SPC is never meant for improving disturbance rejection.