CHAPTER 13
PROPORTIONAL + INTEGRAL CONTROL

We have used the basis weight response of paper machines as our tie to reality in studying process behavior and process control. The first model we introduced was the influence coefficient model, one form of which appeared as:

\[ BW = 0.7 \times FS \]

Where \( BW \) is the change in basis weight in pounds per ream and \( FS \) is the change in stock flow in gpm. The coefficient 0.7 quantitatively describes the influence that stock flow changes have on basis weight changes: the influence coefficient. In using this model, we ignore the details of the time response of basis weight following a stock flow change and focus only on the change in weight after a "long time".

It was asserted that we could not resolve the key issue of tuning limitations in feedback controllers using this model. This idea was quantified by proposing a new model to describe aspects of the time (transient) response of basis weight following a change in stock flow. Two closely related models were offered: the gain lag model and the discrete gain lag model.

The gain lag model describes the response of basis weight to changes in stock flow (or any process response to an input variable) in terms of two parameters: the influence coefficient (or gain) and the time constant. This model is based on the properties of a first order constant coefficients differential equation. Its potential applicability in to process situations is best established empirically, although we showed in one situation that an appeal to physical principles can suggest this form directly. The gain lag model for basis weight would be:

\[ \tau \times \frac{dBW}{dt} + BW = 0.7 \times FS \]

We also introduced the notion of the "step function" as a test device for showing the essential characteristics of the transient response of processes. The characteristics of the step function are that it remains at zero for a long time and then at a certain time (time zero) takes on a non-zero value and holds that value for a long time. "Long time" means long enough so basis weight "lines out" or stops changing with time. The response of the gain lag process to a step change in stock flow is mathematically given by:

\[ BW(t) = 0.7 \times FS \times (1 - \exp(-t/\tau)) \]

Where \( FS \) is the height of the step change in stock flow. This response is sketched in Figure 1 for a \( \tau \) of 2 minutes. The properties of the response are that the change in BW is 63% complete in one time constant (2 minutes in the Figure), 85% complete in two time constants, and 95% complete in three time constants. We normally say that a "long time" is 3 to 5 time constants.
The discrete gain lag model was introduced to provide a feasible approach to simply computing the response of processes. It is also the logical model to use when you are looking at a process response from the point of view of a digital control system. Such systems sample the input, carry out control
calculations, set the output, and then wait for a finite time interval (the sample time) before repeating the process. The output is constant between sample times. The process response between sample times is not "known" to the digital system. At the sample times, the results from the discrete gain lag model can exactly match those of the gain lag model.

The discrete gain lag model for basis weight response is:

\[ BW(t + T) = A \cdot BW(t) + 0.7 \cdot (1 - A) \cdot FS(t) \]

Where BW and FS are known only at the sample times that are spaced T time intervals. Specifically, the time variable t is allowed to take on the values nT where n is a positive or negative integer.

Generally, the sample time T is under control of the designer and is selected in relationship to the time constant \( \tau \) of the underlying gain lag process. Specifically, we tend to choose T to be between one half and one tenth of the time constant. Thus, for a time constant of 2, we might choose \( T = 0.5 \) and obtain a simplified discrete gain lag model of:

\[ W(t + 0.5) = 0.78 \cdot BW(t) + 0.154 \cdot FS(t) \]

where we have used the relationship:

\[ A = \exp(-T/\tau) \]

which establishes the correspondence between the discrete gain lag model and the underlying gain lag model and guarantees that the points generated from the discrete model match those from the gain lag model at corresponding times.

Using the discrete gain lag model we have been able to introduce the notion that oscillatory and unstable behavior can result when feedback controllers are used on processes. We have shown that the model generates such behavior in numerical computations. We have also shown that the discrete gain lag model admits such behavior when the parameter A, the "lag factor" goes negative and has a magnitude exceeding one. We have shown in practice the old control engineering rule that to set the gain of a proportional controller, one should find the maximum gain for stability and use one half of that value. The possibility of unstable behavior from feedback controllers is ever present and represents a cost that must be born to get the big return: a automatic counter action even to disturbances that are unknown to the control system designer.

While the discrete gain lag model serves quite well to introduce the key issues of feedback control, it fails to describe most actual situations with sufficient precision to set real controller tunings. However, it can be modified to produce a process model that has very wide applicability.
THE DISCRETE GAIN DELAY LAG PROCESS MODEL

For a process with input U and response X, the general form of the discrete gain delay lag model is:
\[ X(t + T) = A X(t) + K(1 - A) U(t - TD) \]

Where the new parameter, TD, is called the delay time of the process. If we lay the discrete gain lag model down near the discrete gain delay lag model, the essential difference shows through. Thus, the discrete gain lag model for such a process is:
\[ X(t + T) = A X(t) + K(1 - A) U(t) \]

The difference is entirely in how the process input is described. In particular, if the delay time TD is taken to be zero, the discrete gain delay lag model reduces to the discrete gain lag model. The discrete gain lag model is a special case of the discrete gain delay lag model where the delay time is zero.

Figure 2 depicts the situation that this new model is describing. The figure shows the process behavior rather than computed results from the model. We still use the step input to show the behavior. A comparison to Figure 1 is appropriate. There, the process begins to respond as soon as the step occurs. In Figure 2, there is a period of time between the step change and the first sign of movement on the part of the process response. Thereafter, the response is just like that in Figure 1. In other words, the response in Figure 2 is the same as that in Figure 1 except all response points are shifted to a later time and a period of zero response fills in the gap.

For basis weight, a very appropriate version of the discrete gain delay lag model would be:
\[ BW(t + .5) = 0.78 \times BW(t) + 0.154 \times FS(t - 1) \]

Where we have suggested a delay time of 1 minute for this process. We can still establish the response of this model by direct numerical calculations as follows for a step of 5 gpm in stock flow.

<table>
<thead>
<tr>
<th>Time</th>
<th>BW</th>
<th>FS</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>process is lined out to start with</td>
</tr>
<tr>
<td>-0.5</td>
<td>0</td>
<td>0</td>
<td>step hits</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>5</td>
<td>BW = 0.78<em>0 + 0.154</em>0 (FS(0 - 1) = FS(-1) = 0)</td>
</tr>
<tr>
<td>.5</td>
<td>0</td>
<td>5</td>
<td>BW = 0.78<em>0 + 0.154</em>FS(-.5) = 0</td>
</tr>
<tr>
<td>1.0</td>
<td>0</td>
<td>5</td>
<td>BW = 0.78<em>0 + 0.154</em>FS(0)</td>
</tr>
<tr>
<td>1.5</td>
<td>0.77</td>
<td>5</td>
<td>BW = 0.78<em>0.77 + 0.154</em>5</td>
</tr>
<tr>
<td>2.0</td>
<td>1.37</td>
<td>5</td>
<td>BW = 0.78*.75 + 0.154*5</td>
</tr>
<tr>
<td>2.5</td>
<td>1.84</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>2.20</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>2.49</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>2.71</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>2.88</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Figure 2. Gain lag delay basis weight response to a stock flow change (TD = 1)
We do not need a plot of these values. The table itself is the best way to confirm our understanding of the discrete gain delay lag model. The final response of 3.5 is obtained from the influence coefficient 0.7 times the step size 5. We can investigate the time at which the process reaches 63% of its final value or $0.63 \times 3.5 = 2.2$. The time is 3.0 from the table. However, we based this model on the assumption that the time constant was 2 minutes. There is an apparent discrepancy. The reason is that the 2 minutes lag of the time constant must be measured starting after the delay time of one minute. All points on this response are delayed one minute from the lag response we would compute without delay. Otherwise they are identical. Thus, the process reaches 85% of 3.5 or 2.97 at time 5, which consists of a delay of one minute and two time constants.

JUSTIFICATION OF THE DISCRETE GAIN DELAY LAG MODEL

It is a matter of fact that the discrete gain delay lag model plays a key role in the formulation of process response models for advanced process control systems. This importance far and away exceeds the role this model played in the development of classical control theory or over the developments of the 1960s, which came to be known as "modern control theory". There should be some discussion of the reasons for the central role it plays in computer control of industrial processes.

Traditionally, the concept of delay in a process response was associated with the notion of "transport delay", the time it takes for material to move from a process to a point of measurement. Thus, if there were a conveyor moving product away from a process and a convenient point to locate a sensor turned out to be at a point down the conveyor, one would ascribe a transportation delay to the time it took for material to get to the sensor. This delay would play a role in the tuning of a feedback controller based on the sensor output.

The "fit" between this concept and the issue of measuring and controlling basis weight on a paper machine is very good. Since the control is over bone dry weight, our understanding of the paper machine tells us that the processing which determines basis weight is quite complete on the wire or former of the machine. However, the web must travel all the way to the reel before we get a chance to scan it for actual basis weight. Therefore, there is a delay time in the measured response that is the time it takes for the web to travel from the former to the reel. In most paper machines, the web traverse time is comparable in magnitude to the time constant of the response of the forming process. This is the situation that was explored in the table above where we handled a case in which the time constant was 2 minutes and the time delay was one minute.

Other paper industry processes also suggest the presence of "transport delay". The lime kiln, which was discussed earlier, clearly has material gradually moving through a process so that changes in the feed might not be reflected fully in product qualities for periods of one to four hours due to transit time. Bleaching is usually carried out in plug flow reactors with the chemical treatments applied at the inlet. The effects of variations in the treatments are not measurable until the product emerges minutes to hours later. The discrete gain delay lag model has clear justification for such applications.
However, there are many situations where a delay model is used but the physical situation does not match the "transport delay" idiom. For example, the causticizing process has been modeled using a discrete gain delay lag model but there is no transport delay in the process. The processing consists of mixing the reagent lime with the carbonate rich green liquor in a short residence time slaker and then passing the mixture through a series of three stirred tanks where the calcium carbonate precipitates and assumes a form that permits separation of the slurry by gravity in a clarifier. In traditional process control this would be recognized as a situation where the appropriate model would consist of three or four gain lag models connected in a "cascade".

Control engineers have found it easier to base their process modeling on an approximate fit of the discrete gain delay lag model to the process rather than try to estimate the full details of a model like a series of three or four gain lag models cascaded even if a theoretical basis exists for believing the latter is the correct form. Classical control theory was developed before the availability of digital computers. By classical tools, delay is very difficult to handle and there was a preference for series of gain lag models. Now, the reverse is true. With digital computers, delay is easy to handle both in simulation studies of controllers and in their implementation. It is used to simplify the computations and the approximation has proven to give very reliable results in practice.

FITTING THE DISCRETE GAIN DELAY LAG MODEL TO DATA

We can illustrate the process of fitting a discrete gain delay lag model to data in terms of the causticizing situation since this is representative of the general situation in which such a model is found. Figure 3 shows some ideal "data" that fits the concept we have discussed of the causticizing process. The Figure represents a situation in which the lime flow to the slaker was increased by 10 tons per day and the activity was measured out of the third causticizer tank. The long run behavior of the data suggest a gain of 0.4 percent activity per ton per day of lime.

The time response has a very common characteristic of process behavior. The step in lime flow was made at time zero. The sampling time of the system collecting the activity data was 10 minutes. Instead of showing response on the first sample interval after the step, there appear to be several samples during which there is little or no change in activity. After all, this is measured three large tanks away from the point of reagent addition. Then, the activity begins to change slowly at first, gaining "speed" as it goes, until rather late in the response, past time 100, a maximum slope of the activity curve is reached and then the response rate gradually decreases and we approach an asymptote.

The point to notice is that for all time after about time 150, it is easy enough to "see" in this response the characteristics of the discrete gain lag model. However, in the early time, up to 150 the response does not look like that of a discrete gain lag process. Control engineers have learned to see through this apparent discrepancy.
Figure 3. White liquor activity response to a lime flow change of 10
Figure 4 shows the data from Figure 3 with a line generated by a discrete gain delay lag model with a sample time of 10 minutes. The model is presented in the figure and is repeated here:

\[ \text{Act}(t + 10) = 0.915 \times \text{Act}(t) + 0.4 \times (1 - 0.915) \times \text{Lime}(t - 90) \]

The gain of the model is 0.4 and exactly matches that estimated in the data. Inspection of the right hand term shows that the delay time of the model was selected to be 90 minutes. The line drawn in the Figure confirms that the model response remained at zero through the sample time at 90 and then began its lag response.

The lag factor of 0.915 can be plugged into our usual formula to determine the time constant implied by the discrete gain delay lag model as:

\[ \tau = -\frac{T}{\ln A} = -\frac{10}{\ln 0.915} = 113 \text{ minutes} \]

A control engineer, who might not be well versed in the theory of causticizing, would be content to describe this response as: gain, delay, lag, with gain 0.4, delay 90 minutes, lag 113 minutes. If she were old enough to have worked through examples in which this model and a "three one hour lags with a gain of 0.4" model were compared in actual control simulations, she would feel comfortable that very good control results could be obtained using the discrete gain delay lag representation.

Figure 5 presents the discrete gain delay lag model for a causticizer alone so the properties of the model will be clear. The model shows a zero response for each sample up to the delay time of 90 minutes. It then responds with its maximum rate of change in the next sample time and the rate of change declines thereafter. The asymptotic change is measured by the gain times the step size. The model response is the discrete gain lag response delayed by 90 minutes. The model behavior is much simpler than the actual behavior of the process, which in addition to a true delay period, had a period in which the rate of change of activity increased before the decreasing final period was reached. However, if the gain delay lag model is correctly fitted, good control results will be obtained by using it.

This presentation has not provided a "cook book" for fitting the gain delay lag model to data. However, it may be possible to piece a recipe together. First, the gain to be used is found in the usual way: look far out on the response and find the change in the response from its initial value; divide that by the height of the step change in the input variable. Second, locate the point at which the response reaches 50% (or so) of the final value. Draw a tangent to the curve at that point and extrapolate it back to the abscissa. The time elapsed between the point where the step occurred and the time where the tangent intersects the origin is a good estimate of the delay time. Third, estimate the time at which the response reaches 63% of its final value. The difference between the tangent intersection and the "63% time" is an estimate of the time constant. Fourth, compute the response of a discrete gain delay lag process for the same size step input and plot it together with the data to see if the fit is satisfactory. You may want to fine tune your estimates of the gain, delay, or time constants to feel you have truly captured the essence of the data within the limits of the model used.
Figure 4. Fit of discrete gain lag delay model to causticizer response ($A = 0.915$, $K = 0.4$, $TD = 90$)
Figure 5. Discrete gain lag model alone.
PROPORTIONAL CONTROL OF A DISCRETE GAIN DELAY LAG PROCESS

Consider our improved estimate of the behavior of basis weight on a paper machine:

\[ BW(t + .5) = .78 \times BW(t) + .0044 \times FS(t - 1) \]

This is a discrete gain delay lag process with gain lag factor 0.78, gain 0.02, and delay of 1 with a sample time of 0.5. If we apply a proportional controller to this process its representation would be as before:

\[ FS(t) = K_c \times (BWSP(t) - BW(t)) \]

It is our intention to work a specific example to learn how this new process model reacts in a feedback control situation. Since it is an extension of the lag model, we should be prepared to see such things as oscillation and instability arise depending on the tuning we choose for the controller. For a given \( K_c \), we can calculate the response \( BW \) provided we use the method in which we step along in both the process model and the controller equation. In fact, this is the only feasible approach to computing closed loop response: no closed form solution is available.

In our work with proportional control of basis weight using the discrete gain lag model, we started with a loop gain of 2, then moved up to 4 and found a modest overshoot. The maximum loop gain for stability was found to be 8.48. If we accept our new model here has a more correct representation of the process dynamics, we are about to find that the earlier results were wildly optimistic in terms of the loop gains that will work. In anticipation of that result, we begin here with a loop gain of one. If the magnitude of the loop gain is 1, the controller gain is found from:

\[ .02 \times K_c = 1; \ K_c = 50. \]

The control response is found by stepping together on the process equation:

\[ BW(t + .5) = .78 \times BW(t) + .0044 \times FS(t - 1) \]

and the controller equation:

\[ FS(t) = 50 \times (BWSP - BW(t)) \]

The procedure is much like our work with the discrete gain lag model but now we must find the flow term in the process model by looking further back in time as shown in the table below.

<table>
<thead>
<tr>
<th>TIME</th>
<th>BWSP</th>
<th>BW</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>5.00</td>
<td>0.00</td>
<td>250.00</td>
</tr>
<tr>
<td>0.50</td>
<td>5.00</td>
<td>0.00</td>
<td>250.00</td>
</tr>
</tbody>
</table>
Notice that although the setpoint change hits at time zero and the stock flow makes a jump at that time, the process model does not begin to move until time 1.5 when the effect of the flow change at zero is first felt. In computing basis weight at times prior to t=1.5, we must rely on the fact that the flow was incrementally zero for times prior to zero, equivalent to the process being "lined out" before our setpoint change was made.

The notes also show that it is only on and after time 3 that the stock flow values used in the calculation are tracking a changing stock flow. These are the real and natural consequences of delay on the proportional controller. The controller held its output constant until the process response first showed signs of changing.

Now before looking at a graphical representation of the output, we should use what we know about proportional control to decide what the final value will be. The numerical work suggests something in the vicinity of a change of 2.5 for basis weight even though we are "looking for" a change of 5 pounds. Our old results enable us to compute the closed loop gain of the process as:

\[ K_{cl} = \frac{K\times K_c}{1 + K\times K_c} = \frac{.02\times50}{1 + .02\times50} = \frac{1}{1 + 1} = .5 \]

Thus we are indeed headed for a change in basis weight of 2.5 pounds, just half of the desired value. We learned early that to approach perfect control, the loop gain needs to be high: one is not a high loop gain.

Figure 6 is a graph of the basis weight data from the table. Using our standard approach to characterizing closed loop responses, we must call this stable and oscillatory. We can also estimate the overshoot: from the table, the peak value is 2.94; from our incremental calculation the final value is 2.5. Therefore,

\[ \text{overshoot} = 100\times(2.94 - 2.5)/2.5 = 17.6\% \]

Given our understanding of process operators and overshoot, we do not anticipate being able to improve the performance by raising the controller gain.
At this point, we should take a closer look at Figure 6 and see that there is an essential difference from our earlier results with stable oscillatory control responses. In our work based only on the discrete gain lag model, when the process oscillated, it had a period of oscillation which was 2*T, twice the sample time of the controller. Now, the degree of oscillation we see in Figure 6 is small so it is a little tricky to estimate the period of the oscillation but it is quite clear that it is longer than the one minute we would have seen earlier. In fact, we can make a fairly reliable estimate by noting the time of the first peak and the time of the first minimum after that. We will call the difference between these times one half the period. The table suggests the peak occurs at time 3.5 and the minimum at 6, so the full period is estimated as

\[
\text{period} = 2 \times (6 - 3.5) = 5 \text{ minutes}
\]

While our earlier results served well to orient us toward the oscillations that can occur in feedback control situations, the results here are much more typical of actual behavior. The period of oscillation is rarely closely tied to the sample time. It is a complex function of the sample time, the time constant, and the delay in the process. It is usually estimated only via simulations, which amount to the kind of calculations we are doing here, or by actual determination on the process during tuning exercises.

Figure 7 presents a graph of the flow settings used in this control situation. The incremental flow jumps from zero to 250 gpm when the setpoint change is made. It stays there for the next two sample intervals because the basis weight has still not responded. Then it declines and settles down with modest oscillation at the final value of 125 gpm. Of course, the 250 gpm was just what we really needed to achieve our desired weight change but we are by now used to the behavior of proportional controllers and aren't even especially disappointed that it was not smart enough to stay with the correct value. Unfortunately, we don't have any prospects to improve the situation by raising the loop gain since we are already beyond the maximum overshoot we allow ourselves.

Even so, we should consider additional cases to fill out understanding of proportional control with realistic process models. Since the calculation procedure is identical with that above, the additional results will only be presented graphically. We will investigate loop gains both smaller than one and larger. The maximum controller gain for stability will be of interest but we can only find it by trial and error. The simulation results presented here were generated on a spreadsheet simulation of this situation.

Figure 8 presents a case with a very low loop gain. The character of the response suggests that this could be modeled as a discrete gain delay lag process with a lag factor corresponding to a time constant of about 1 minute. Since the basis weight process response is 2 minutes we see that at low loop gains we retain the stable monotonic character of response yet provide faster response as indicated by the smaller time constant. Of course this response falls very short of perfect control since the response is only a little over one pound when the setpoint change was 5 pounds. With the discrete gain lag process we were in a position to prove that adding proportional control gave a closed loop response which was itself a discrete gain lag process but with altered gain and lag factors (K’ and A’) compared to that of the original process.
We cannot prove the same thing for the discrete gain delay lag process but the numerical responses here suggest that such an interpretation is possible.

Of much more interest is how the behavior changes as the loop gain is increased. Figure 9 shows the results with a loop gain of 2. Here the controller will reach 2/3 of the change requested but there is a large overshoot and continued oscillation. Figure 10 shows that the process is unstable with a loop gain of 4. Figure 11 represents an attempt to find the maximum gain for stability by trial and error on the spreadsheet simulation: the estimate 3.1 was obtained. In Figure 12 the loop gain was set at 1.55 to test the rule of thumb that a gain of one half the maximum should be suitable. The degree of overshoot is not acceptable in process situations although it might be acceptable in electromechanical systems where the rule of thumb was developed. We already knew from Figure 6 that a loop gain of 1 exceeded our overshoot criterion. However, in our simplified work based on the discrete gain lag model we had found that the rule of thumb worked well. The results here are more typical of process situations: a loop gain of one half the maximum for stability usually produces unacceptable overshoot. Perhaps the rule should be "use 33% of the maximum gain for stability".

Given that the discrete gain delay lag model we are using here is an accurate representation of the basis weight process we must conclude that proportional control will not produce a satisfactory basis weight control system. This is in fact the case for most process situations. However, we are in a position to remedy the situation through the introduction of a more advanced control algorithm than we have considered so far. However, before doing so, we should examine the results that can be obtained with integral control.

INTEGRAL CONTROL OF BASIS WEIGHT

In the last section, integral control was introduced in connection with the one step process. However, it is not limited to use in such situations. As long as we are willing to use numerical calculations to evaluate alternative tunings of the integral controller, it can be applied to the discrete gain delay lag process. Without going so far as to recommend it, we evaluate such a controller here as an intermediate step on our way to a more advanced control algorithm.

We will step through the process and controller equations as was done above. The process equation will be taken to be the same one as above:

\[ BW(t + .5) = .78 \times BW(t) + .0044 \times FS(t - 1) \]

and the controller equation for integral control is:

\[ FS(t) = FS(t - .5) + (\frac{1}{2}) \times (BWSP - BW(t)) \]

where we have kept the sample time at 0.5 minutes and we will continue with the setpoint change of 5 pounds per ream.
Integral control is not nearly as well studied as proportional control so there is not much to go on regarding the tuning of the controller. In our earlier discussion, we pointed out that TI in the formula above is a dimensional quantity. The dimensionless quantity in this situation is $K^*T/TI$ where $K$ is the process influence coefficient. In our case $K^*T = .02*.5 = .01$, so the dimensionless quantity "equivalent" to the loop gain is $.01/TI$. Trial and error was used to obtain the value $TI=.1$ for use in the presentation below: for reference, $T/TI = 5$ and $K^*T/TI = .1$ for this case. The relevant calculations are as follows.

<table>
<thead>
<tr>
<th>TIME</th>
<th>BWSP</th>
<th>BW</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.000</td>
<td>5.000</td>
<td>0.000</td>
<td>25.000</td>
</tr>
<tr>
<td>0.500</td>
<td>5.000</td>
<td>0.000</td>
<td>50.000</td>
</tr>
<tr>
<td>1.000</td>
<td>5.000</td>
<td>0.000</td>
<td>75.000</td>
</tr>
<tr>
<td>1.500</td>
<td>5.000</td>
<td>0.110</td>
<td>99.450</td>
</tr>
<tr>
<td>2.000</td>
<td>5.000</td>
<td>0.306</td>
<td>122.921</td>
</tr>
<tr>
<td>2.500</td>
<td>5.000</td>
<td>0.569</td>
<td>145.078</td>
</tr>
<tr>
<td>3.000</td>
<td>5.000</td>
<td>0.881</td>
<td>165.673</td>
</tr>
<tr>
<td>3.500</td>
<td>5.000</td>
<td>1.228</td>
<td>184.533</td>
</tr>
<tr>
<td>4.000</td>
<td>5.000</td>
<td>1.596</td>
<td>201.552</td>
</tr>
<tr>
<td>4.500</td>
<td>5.000</td>
<td>1.974</td>
<td>216.682</td>
</tr>
<tr>
<td>5.000</td>
<td>5.000</td>
<td>2.352</td>
<td>229.923</td>
</tr>
<tr>
<td>5.500</td>
<td>5.000</td>
<td>2.721</td>
<td>241.318</td>
</tr>
<tr>
<td>6.000</td>
<td>5.000</td>
<td>3.076</td>
<td>250.938</td>
</tr>
<tr>
<td>6.500</td>
<td>5.000</td>
<td>3.411</td>
<td>258.884</td>
</tr>
<tr>
<td>7.000</td>
<td>5.000</td>
<td>3.722</td>
<td>265.273</td>
</tr>
</tbody>
</table>

The response is far from complete in the time period covered in the table above. This is so even though the time period is the same as was presented for proportional control in our discussions of basis weight control by that means. Most control engineers associate proportional control with "fast" control and integral control (if they recognize it at all) with slow response. That is supported by our example here.

Figure 13 presents a much longer time period of the response for the case above. The most important thing to notice about the response is that it succeeds in reaching the desired basis weight change of 5 pounds per ream, something the proportional controller was quite incapable of. The response also shows some overshoot, but there may be room for slightly more aggressive tuning, i.e., a TI lower than 0.1. All in all, though, this figure probably represents the capability of integral control on this process. It might not be hard to "sell" process operators on this response. One could point out, for example, that for all time after 10 minutes, the basis weight is very close to the desired value. Most operators would have a hard time achieving a 5 pound weight change by hand in this short a time since they would probably use at least two moves with a period in between to "see where it is going".
Figure 14 shows that with much more aggressive tuning, the integral controller will produce unstable oscillatory response on this process. The TI value is one tenth that used above. The actual minimum TI for stability on this process is about .02, or one fifth of the value used in Figure 13. Note also that the period of oscillation is much longer than the sample time of 0.5 minutes. The period here is about 8 minutes. This is twice the period of the unstable oscillations found with proportional control, again meriting the "slow" label associated with integral control.

The table below shows some of the points calculated by the spreadsheet for this response.

<table>
<thead>
<tr>
<th>TIME</th>
<th>BWSP</th>
<th>BW</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>5.00</td>
<td>0.00</td>
<td>250.00</td>
</tr>
<tr>
<td>0.50</td>
<td>5.00</td>
<td>0.00</td>
<td>500.00</td>
</tr>
<tr>
<td>1.00</td>
<td>5.00</td>
<td>0.00</td>
<td>750.00</td>
</tr>
<tr>
<td>1.50</td>
<td>5.00</td>
<td>1.100</td>
<td>945.00</td>
</tr>
<tr>
<td>2.00</td>
<td>5.00</td>
<td>3.058</td>
<td>1042.100</td>
</tr>
<tr>
<td>2.50</td>
<td>5.00</td>
<td>5.685</td>
<td>1007.838</td>
</tr>
<tr>
<td>3.00</td>
<td>5.00</td>
<td>8.592</td>
<td>513.845</td>
</tr>
<tr>
<td>3.50</td>
<td>5.00</td>
<td>11.287</td>
<td>101.912</td>
</tr>
</tbody>
</table>

Notice that the first controller output after the step called for a flow change of 250 gpm. This is exactly the amount needed to reach a setpoint change of 5 pounds per ream in this situation. The difficulty presented by the delay in the process response shows up clearly in this example. Since the process cannot respond in 0.5 minutes, the integral controller adds an additional 250 gpm to its output at time 1. This and subsequent actions represent severe over control but it is not until time 2.5 that the controller gets feedback from the process which makes this clear to it. The basis for the instability is fully in place at that point.

For future reference, our recommended tuning for this controller occurs at a K*T/TI of 0.1 and instability at K*T/TI of 1. We would need to study the problem further, but there might be a rule that says KT/TI must be in the range zero to one or two. Such a rule would be comparable to our rule for loop gain that it is in the range zero to 10. In both cases the dimensionless character of the quantity lends credence to the notion that such a rule exists.

At this point, faced with a choice between proportional control and integral control everyone would choose integral control. It achieves the desired change and if it is slow to respond it may well be that the operators will accept this as appropriately cautious in dealing with a process like papermaking. However, we should look for a controller that can achieve the desired result and still speed things up some. Such a controller is the proportional plus integral controller.
THE P+I CONTROLLER

The intent of the proportional plus integral controller is to achieve the benefits of both of the separate control actions. In principle, this is achieved by adding the two actions together. In practice the control algorithm is modified from that in a way that has been found to facilitate tuning of the two control actions. The control action for a general situation in which \( U \) is the manipulated variable and \( X \) is the process response is given in terms of the error or difference between setpoint and measurement. Thus,

\[
E(t) = X_{SP}(t) - X(t)
\]

\[
U(t) = U(t - T) + K_c \times (E(t) - E(t - T) + \frac{T}{T_I} \times E(t))
\]

The control equation expresses the control output at a given time in terms of the output at the previous time, the error at the present time, and the error at the previous time. The use of the previous control output follows directly from our prior integral control equation. The use of a prior error in addition to the present error is new in this controller. The P+I controller is actually a special case of the most general control formula, which expresses the control output as a weighted sum of prior control outputs and present and prior errors. Here we use one past control output and the present and one past error.

Whereas it was said that the behavior of the integral controller mimicked the behavior many operators use of factoring into their decision prior control outputs, the rule here is probably more complicated than most mere mortals could handle in their head. Even so, it has been around in the automatic control world for at least 60 years and is definitely the most widespread control algorithm in use in the process industries. Most commercial controllers are advertised to be PID controllers, which exhibit an even more complex behavior than that above. However, in use, the "D" term in the controller is usually turned off. Therefore, we count almost all PID controllers in the "fold" of PI controllers.

We can apply the PI controller to basis weight and set the problem up for numerical simulation on a spreadsheet by repeating the process model:

\[
BW(t + .5) = .78 \times BW(t) + .0044 \times FS(t - 1)
\]

adding the error computation of the controller:

\[
E(t) = BW_{SP}(t) - BW(t)
\]

and the control algorithm itself:

\[
FS(t) = FS(t - .5) + K_c \times (E(t) - E(t - .5) + (.5/T_I) \times E(t))
\]

Simulation of the behavior of the controlled situation will be a question of cycling through these three equations in each time step. Hand calculations would be possible but great care is needed in handling the
different time steps represented throughout the models above. In any case, specific values must be selected for Kc and TI before such calculations can proceed.

TUNING THE PI CONTROLLER

There are two approaches to the tuning of controllers. The oldest and best is called closed loop tuning. With this approach, one puts the controller on the process and finds the tuning parameters by a trial and error process. One is guided by experience in moving from one trial tuning to another. A variant of this procedure is to use a process model and a simulation tool to go through the early part of the trial and error process and then move to the actual process when pretty close estimates of Kc and TI are available. In any case, it will be found that guidelines are needed to avoid a protracted period of trial and error. This is so because the tuning parameters interact with one another and present a confusing picture if one is not methodical. I usually compare this process to the issue of tuning the color knobs on a color television or the compensation adjustments on a complex stereo.

There have been many attempts to systematize the process of controller tuning. The method presented here is somewhat home grown but that was done with a rather complete knowledge of the other approaches.

Step 1. Set the TI to a large value so there is effectively no integral action in the controller.

Step 2. Starting with a low loop or controller gain, make a step response test and move to a higher loop gain if the process is exhibiting stable monotonic behavior.

Step 3. When a Kc is found which produces about 15% overshoot, make a note of the value of Kc, and make the best possible estimate of the period of the oscillation. This usually involves measuring the time from the first peak to the first valley and multiplying by 2.

Step 4. Set TI equal to the period of the oscillation as found above. (In the PI algorithm, TI has the units of time.)

Step 5. Run a step response test and check to see if the response shows about 15% overshoot with the overshoot centered on the desired final value.

Step 6. Lower the Kc if the overshoot has increased to an undesirable level; lower TI if the response is not "centered" on the desired final value. Run step tests after each tuning change and do not stray very far from the values found in steps 3 and 4.

Considering this "cookbook", we are in a good starting position because we already have results from the proportional control case and can jump in between steps 3 and 4. Figure 6, with a loop gain of 1, showed about 15% overshoot. Therefore, our first trial tuning will be with a controller gain of 50. In our
discussion of Figure 6, we estimated the period of oscillation as 5 minutes. Therefore, we can start out with a TI of 5 minutes. We hope that this will meet the requirements of step 4 in our procedure above.

Figure 15 presents the results obtained with the tuning above. The response does not overshoot and oscillate about the desired final value of 5 pounds per ream, although the integral action does continue to move it toward that goal so that it is substantially there by the end of the time shown.

This is a case where we need more integral action. Since we are doing this on a simulation, we can afford to "take a chance", and the next value selected was TI=2.5. The results are shown in Figure 16, which is much more satisfactory. However, the response is still "not centered", so a further move towards more aggressive integral action is made to TI=2.

The results with a Kc of 50 and a TI of 2 minutes are shown in Figure 17. This is an ideal response for a PI controller. For reference, the following table presents the first few computed points from the spreadsheet simulation.

<table>
<thead>
<tr>
<th>TIME</th>
<th>BWSP</th>
<th>BW</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.000</td>
<td>5.000</td>
<td>0.000</td>
<td>312.500</td>
</tr>
<tr>
<td>0.500</td>
<td>5.000</td>
<td>0.000</td>
<td>375.000</td>
</tr>
<tr>
<td>1.000</td>
<td>5.000</td>
<td>0.000</td>
<td>437.500</td>
</tr>
<tr>
<td>1.500</td>
<td>5.000</td>
<td>1.375</td>
<td>414.063</td>
</tr>
<tr>
<td>2.000</td>
<td>5.000</td>
<td>2.723</td>
<td>375.156</td>
</tr>
<tr>
<td>2.500</td>
<td>5.000</td>
<td>4.049</td>
<td>320.747</td>
</tr>
<tr>
<td>3.000</td>
<td>5.000</td>
<td>4.980</td>
<td>274.440</td>
</tr>
<tr>
<td>3.500</td>
<td>5.000</td>
<td>5.535</td>
<td>239.997</td>
</tr>
<tr>
<td>4.000</td>
<td>5.000</td>
<td>5.728</td>
<td>221.210</td>
</tr>
<tr>
<td>4.500</td>
<td>5.000</td>
<td>5.676</td>
<td>215.400</td>
</tr>
<tr>
<td>5.000</td>
<td>5.000</td>
<td>5.483</td>
<td>218.995</td>
</tr>
</tbody>
</table>

Given that the final value is clearly 5, we can see the peak value in the table as 5.728 corresponding to 14.6% overshoot.

It is well to compare these results with those for integral control given in Figure 13. Both responses reach the desired final value. Both show overshoot of acceptable levels. However, there is much difference between them. The PI control response is clearly much faster than the integral control response. One measure of the speed of response is the time from when the setpoint is entered until the response first reaches the desired final value on the way up. For the PI controller the table above gives this as 3 minutes. We can infer from Figure 13 that the integral controller requires almost 10 minutes. The difference is even greater than that because we should subtract the time delay from each number and
compare 2 for the PI control with 12 for the integral control: the PI controller is said to have six times faster "rise time" in this case.

Another measure control engineers use to judge the quality of responses is the "settling time". That is the time from the initial move until the response is always within 95% of its final value. With 15% overshoot, this time must be rather longer than the time to the first peak and often longer than the time to the first valley after the peak. Thus, one might guess that the settling time for the PI controller is about 10 minutes and that for the integral controller is about 15 minutes. On this measure, the difference between the controllers is not very large.

The PI controller has lived up to its advance billing that it would combine the speed of proportional action with tendency of integral control to achieve the desired final value.

OPEN LOOP TUNING

A second approach to controller tuning is to provide formulas that permit the tuning constants, in this case $K_c$ and $T_I$ to be obtained directly from the parameters of the process model itself. We have used a method like this with both proportional control of the discrete gain lag process and integral control of the one step process for the special case of "dead beat" tuning. There, we used our formula for $A'$ with the advance value of zero for that parameter to provide a value for $K$ or $T_I$ in terms of the process $K$ and $A$ values. The development of comparable formulas for PI control of the gain delay lag process has been the subject of considerable research.

Over the years, this author has tinkered with various versions of these rules. The current favorite based on that tinkering is embodied in two formulas:

$$K*K_c = \frac{\tau}{(TD + T)/1.3}$$

$$T_I = \tau$$

where $\tau$ is the time constant of the process, $TD$ is the delay time of the process, $K$ is the gain of the process and $T$ is the sample time. The first formula gives the loop gain in terms of time parameters in the problem. The second simply states that the integral time should be set to the time constant of the process itself.

For the present situation, we have been using a $\tau$ of 2 and a $TD$ of 1 for the process model. Therefore this "open loop" tuning method suggests that the magnitude of the loop gain should be:

$$K*K_c = \frac{2}{(1 + .5)/1.3} = 1.03$$

$$T_I = 2$$
These results are virtually identical to the results obtained above by the closed loop tuning method. Bear in mind, however, that formulas like these may not always produce the type of transient response from the controlled process that you want. They should always be validated in tests. Also, they tend to produce 15% overshoot, so if the situation calls for smoother more conservative response than that, it will probably be necessary to choose a lower gain. Closed loop tuning either on a simulation or the process is always an element of the control design process.
Proportional basis weight control
(loop gain = 1.)

Basis weight vs. Time, min.
Figure 7

Proportional basis weight control
(loop gain = 1)

Proportional basis weight control
(loop gain = 1)
Figure 8

Proportional basis weight control
(loop gain =0.3)
Proportional basis weight control
(loop gain = 2.0)
Figure 10

Proportional basis weight control
(loop gain = 4.0)
Figure 11

Proportional basis weight control
(loop gain = 3.1)
Proportional basis weight control
(loop gain = 1.55, 1/2 max)
Figure 13

Integral basis weight control
(TI = 0.1, K*T/TI = 0.1)

Basis weight

Time, min
Figure 14

Integral basis weight control
\((T_I = 0.01, K^*/T_I/T = 1)\)
Figure 15

PI basis weight control
(Kc = 50, TI = 5)
PI basis weight control

(Kc = 50, TI = 2)