PID Tuning of Liquid Level in Tank

John O'Hara, Ph.D.
Wavetech, LLC
Created: Feb. 8, 2014

1 Introduction

This document shows an example of semi-analytical tuning of the parameters in a proportional, derivative, integral (PID) control. This particular process consists of a cylindrical tank into and out of which liquid is continuously pumped by two pumps. Some of the incoming liquid also boils off inside the tank and is expelled through a vent. The process demands that the liquid level within the tank be maintained at the $50 \pm 10\%$ mark. The user sets the flow rate of the filling pump, while the PID controls the flow rate of the emptying pump, thereby maintaining the level.

2 Process Constraints

The following constraints limit how the PID may be tuned:

1. The fill pump may operate anywhere in the range of 2-18 gallons per minute (GPM), but is expected to run at nearly full capacity most of the time.
2. The controls must maintain the 50% level regardless of fill rate.
3. The emptying pump has the same capacity as the fill pump.
4. The controls must bring the tank to the 50% level even if it starts empty or full.
5. The volume of liquid boiled off inside the tank is always equal to 10% of the incoming liquid volume.

3 Physical Picture

Figure 1 shows the physical layout of the system. The tank is cylindrical with a horizontal axis. Let's assume the tank capacity is 300 G (gallons). It's radius r is 18" (inches) and its length, l, is 68". For this analysis, it is not strictly necessary to attach numbers to the tank length or radius, or even the flow rates. But numbers add some realism and give a better intuition about what is actually occurring in the controls. Again, both the fill pump, P1, and the emptying pump, P2, can pump at any rate in the range of 2-18 GPM.

Now, 10% of the incoming liquid is boiled off and vented, and this effectively de-rates the filling capacity of pump P1. That is, it can pump 18 GPM, but only 90% of that actually fills the tank with liquid. So the maximum fill rate $\max(R_f) = 0.9 \times 18 = 16.2$ GPM. Boil-off doesn't affect pump P2 so its maximum emptying rate is unchanged, $\max(R_e) = 18$ GPM.

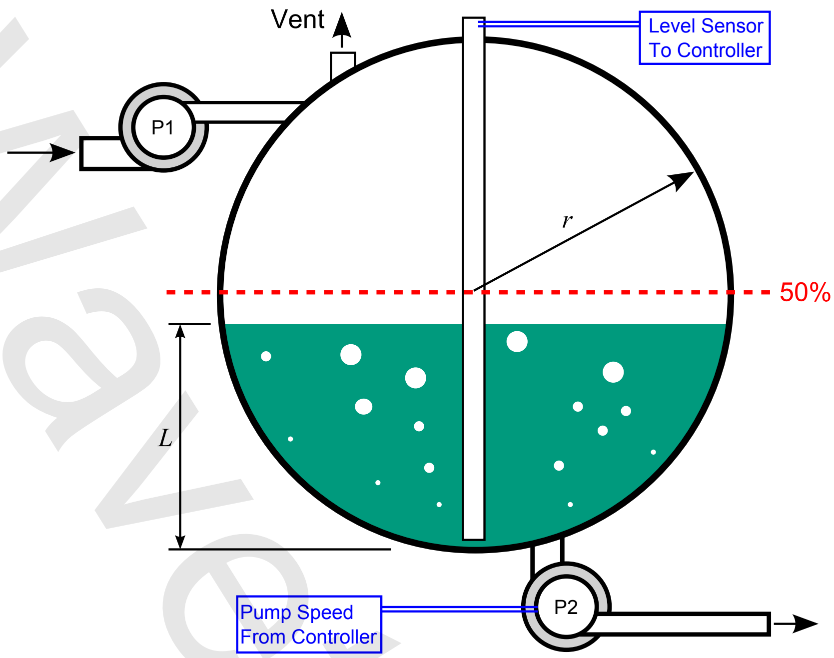

Figure 1: Physical schematic of process.

4 Level/Volume Relation

In this example it is possible, without too much trouble, to fully describe the process with mathematical equations. If the general relations between flow rates and tank level can be determined, then the PID can be tuned without much guesswork. The flow rates are intimately tied to the volume of liquid in the tank. So the next step is to link the volume of liquid in the tank with its level. The details of the derivation are withheld, but it can be shown that the volume of liquid in the tank is given by

$$V = l\left[\frac{\pi r^2}{2} + (L-r)\sqrt{r^2-(L-r)^2} + r^2 \arcsin\left(\frac{L-r}{r}\right)\right] \tag{1}$$

where L is the liquid level in the tank, as shown in Fig. 1, and l is the tank length. As a sanity check we can test three simple cases: $L = 0$, $L = r$, $L = 2r$, which signify the tank is empty, half full, and totally full, respectively. If $L = 0$, then we find $V = 0$. If $L = r$, we find $V = l\pi r^2/2 = 150$ G. If $L = 2r$, we find $V = 300$ G. The three test cases all behave as expected and thus indicate that the equation is correct. Unfortunately, it's also a nonlinear equation. Many may know that trying to tune any nonlinear system is not trivial, and the PID level control will be no exception.

To better describe the problem with nonlinearity, consider the rate at which the level in the tank rises (in derivative notation: $\mathrm{d}L/\mathrm{d}R_f$) when the tank begins at empty and P1 turns on. The level will rise very quickly, or $\mathrm{d}L/\mathrm{d}R_f$ will be large, because there is very little cross-sectional area to fill at the bottom of the tank. Then, as the level approaches 50%, the level will rise much more slowly, or $\mathrm{d}L/\mathrm{d}R_f$ gets small. Why? Because the pump rate has not changed but the cross-sectional area of the tank is much larger when the level is near 50%. Finally, as the level approaches 100%, the rate of change of the level will increase again for a fixed pump rate. The fact that the level does not change *proportionally* to the fill rate means we're going to run into problems trying to use a *proportional*-derivative-integral type control.

5 Simplifications

The controls can be designed to avoid the nonlinear problem, in a process known quite generally as linearization. First, we can avoid the use of the PID altogether except when L is near 50%. Let's assume the PID will only control P2 when $L = 50 \pm 10\%$. In this case the quantity $(L - r)$ is going to be small compared to r since L will approximately equal r. Therefore, two approximations can be made in Equation 1(b).

$$r^2 \arcsin\left(\frac{L-r}{r}\right) \approx r(L-r) \tag{2a}$$

$$(L-r)\sqrt{r^2-(L-r)^2} \approx r(L-r) \tag{2b}$$

Now the relation between the liquid level and the volume of liquid becomes

$$V \approx l\frac{\pi r^2}{2} + 2lr(L-r) \tag{3}$$

This approximation is now linear in L, which makes it much more amenable to PID control. It can be tested for accuracy within the range of $50 \pm 10\%$. When the level sensor reads 60%, $L = 21.6$ in, the approximation finds the volume of liquid in the tank to be 187.97 G. The exact equation finds it to be 187.71 G. Similarly, when the level sensor reads 40%, $L = 14.4$ in, the approximation finds the volume of liquid in the tank to be 111.67 G, and the exact equation finds it 111.92 G. Both equations find the exact same volume when the level sensor reads 50%. So the approximation is very accurate in the range of interest and we can trust the linearized equation for the purpose of level control. Quick note, L is expressed in inches. Typically we will rather use it as percent full. To get L in units of "percent full", multiply it by $100/(2r)$.

The Process Constraints demanded that level control be exercised all the way from empty to full, whereas the PID only takes control in the $50 \pm 10\%$ range. To address this, the controls can be designed such that when the level is in a different range (below 40% or above 60%), a simpler control scheme is employed. Recall also that *the user sets the desired fill rate*, R_d. Note that this is not the *actual* fill rate, R_f. It's just what the user wants. For this example, we will thus use a simple scheme where the controls temporarily slow either the filling or emptying rate to bring the liquid level to 40% or 60%. For example, when $L < 40\%$ we set the emptying rate to 75% of the user-desired fill-rate, or $R_e = 0.75R_d$. Then, we set the filling rate equal to the user-desired fill-rate, or $R_f = R_d$. Since the emptying is occurring more slowly than the filling, the tank level will eventually rise to 40%, where the PID can take over control. Similarly, if $L > 60\%$, then we set $R_f = 0.75R_d$ and $R_e = R_d$. Since the emptying is occurring faster than the filling, the tank level will eventually lower to 60% where the PID will take over.

There is another advantage of this control arrangement. It keeps the pumps running continuously and always near the speed the user desires. This is likely to improve efficiency and pump longevity.

6 PID Tuning

Arguably the most difficult part of this control problem is tuning the PID. Below, equation 4 describes the PID control method. Rigorously solving a PID equation obviously involves calculus, which very often becomes troublesome in systems with complex equations governing the physical process. However, a simple computer program can be implemented that effectively models the calculus without performing actual derivatives or integrals.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,\mathrm{d}\tau + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) + B \tag{4}$$

First some definitions are required. In the PID equation, the term $u(t)$ is called the *control variable* or *controller output*. In our example, it is the signal sent to P2 to control its speed. Note, the variable we want to control (in this case the level in the tank, L) is sometimes called the *process variable*. The term $e(t)$ is the error signal. It is defined as the difference in the level set point and the actual liquid level in the tank, or

$$e(t) = L_{set} - L \tag{5}$$

where L_{set} is the desired liquid level in the tank, set by the user in the control program. Sometimes this equivalently denoted

$$e(t) = SP - PV \tag{6}$$

or error is equal to the set point minus the process variable. The remaining terms of the PID equation are K_p, K_i, K_d, and B. These are the *proportional gain*, the *integral gain*, the *derivative gain* and the *bias*. These are numbers we can tune to make the control loop do what it's supposed to. It's best to start with the simplest case and then add complexity. Therefore this tutorial begins with proportional and control alone. It is assumed that the bias can also be added since its effect on the equation is simple.

6.1 Proportional Only

With proportional-only (P-only) control, the parameters K_i and K_d are set to zero. Thus they play no part in the actual process. In this case the control variable is totally controlled by the error signal and the proportional gain. The bias is used to make the output nonzero when $e(t) = 0$. In other words, when there is no error signal we still want a non-zero output to P2. Mathematically,

$$u(t) = K_p e(t) + B. \tag{7}$$

When the set point L_{set} and measured level L are equal (or $L = L_{set}$), then the error signal is zero and the output equals the bias only, or $u(t) = B$. This sounds like what was desired. In the original specifications, P2 should be on pretty much all the time, it's just *balanced* with P1 so that the level stays at 50%. Let's say that $u(t)$ will be some number between 0-100, representing the percentage of maximum possible speed that P2 can run. For example, if $u(t) = 25$ then P2 will run at 25% of its maximum capacity.

It's also important to mention that P2 should not regularly shut off in normal operation, even if the liquid level gets *below* the set point. Recall from the Simplifications section that the PID control will never turn on until the liquid level is somewhere between 40-60%. Also, if the level has just reached 40% then P2 has been running at 75% capacity; this is slower than P1 so that P1 can fill the tank to 50%. This means two things. First, the worst case error signal will only ever reach a numerical value of ±10, or $e_{\max,\min} = L_{set} - L = \pm 10$. Second, in order for the PID to *smoothly* take control of the pumps when the liquid level reaches 40% it should also make P2 run at 75% when the liquid level is at or near 40%.

From this information it's now possible to solve for K_p and B. Since there are two unknowns, two equations are required. These equations represent the worst case error conditions, when the set point is at 50% and the liquid level is at 40% or 60%, respectively,

$$u(t) = K_p(-10) + B = 75 \tag{8a}$$
$$u(t) = K_p(+10) + B = 100 \tag{8b}$$

The solution of these two equations works out to

$$K_p = 1.25 \tag{9a}$$
$$B = 87.5 \tag{9b}$$

The entire P-only control loop is now defined.

Matlab (or any programming language) can now be used to show how this tuning performs. Figure 2 shows these simulation results when the liquid level is set to begin at 60% full. The program actually calculates the level in this condition at 57.8% because the program is using the approximate relation between volume and liquid level, Eq. 3. By the way, the Matlab file used to calculate all the results in this white paper is given at the back. Matlab also has built-in PID simulators if you purchase the Control System Toolbox.

With the K_p and B settings just defined the result is unexpected. The tank level settles at 52% instead of 50%. Why? This is a phenomonenon called *droop*. It is a result of the fact that the system was not originally tuned for zero error. Consider the zero error situation where $u(t) = B = 87.5$. If the feed pump P1 is running at 18 GPM, and 10% of that is lost to evaporation, the effective feed rate is 16.2 GPM. But $u(t) = 87.5$, which translates to an actual pump rate (P2) of $0.875 \times 18 = 15.75$ GPM. So when the *error* signal is zero, the pumping rates are *NOT* matched and the tank will begin to fill past 50%. The tank level *has* to change at zero error signal! The pump rates match each other only when there is a small error signal. In this case the rates equalize at 16.2 GPM obviously, since that's the feed pump rate. This translates to a tank level of 52%. Notice in this case the output is $u(t) = 1.25 \times (52 - 50) + 87.5 = 90$. And thus P2 pumps at $0.9 \times 18 = 16.2$ GPM.

You may ask, why not just set the bias to $B = 90$ instead of $B = 87.5$? This would effectively tune the controls at zero error so that the pumps equalize when the tank level is at 50% as was the original intention. In fact, sometimes this procedure *is used* to remove droop effects. It doesn't work in this situation because it requires pump P2 to run beyond its maximum capacity. Consider the case where the tank level is at 60%, then the error is 10. If the bias was increased, then $u(t) = 12.5 + 90 = 102.5$. This means the controls are calling for the pump to operate at 102.5% of its capacity, which it obviously can't do!

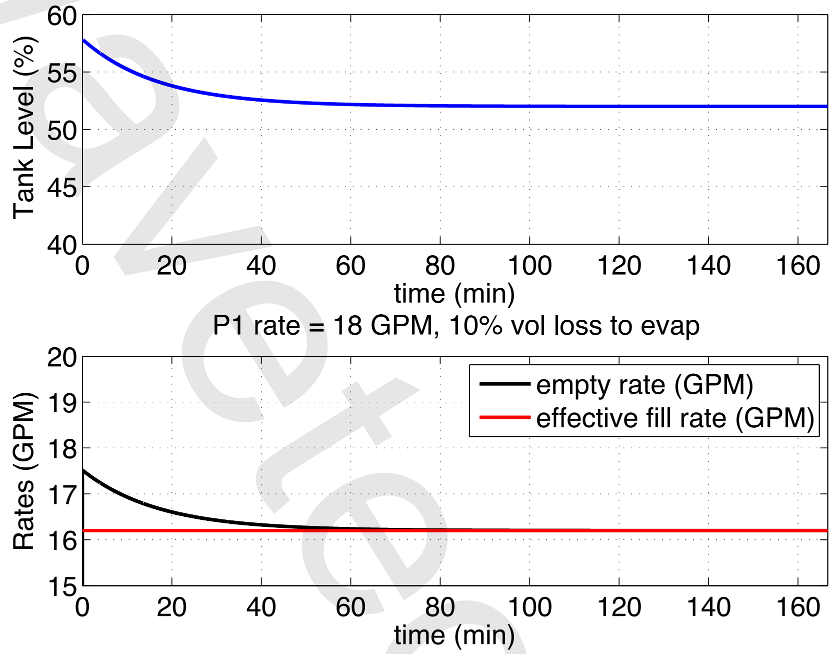

Figure 2: Tank level and pumping rates over time using proportional-only control and starting at 60% full tank.

6.2 Proportional-Integral Control

Another way to fix droop is to add an integral control term. The way it works is as follows. Integral gain adjusts the output signal $u(t)$ according to *accumulated* error. How can we understand this? Pretend that a counter increments every second when there is a positive error between the tank level and its set point. And it decrements every second when there is a negative error. You operate a control that adjusts the output based on the counter value. This is the same function as the integral term.

A simple example helps illustrate. Assume the tank level has settled out at 52% as it does with proportional-only control. Then we turn on a counter that starts incrementing or decrementing every second. Since the error is positive the counter begins to increment. As the counter value gets larger and larger, you add more and more output $u(t)$ thus speeding up pump P2. Eventually this brings the tank level to 50%. But that doesn't mean your counter goes to zero too. It still has some positive value that has accumulated over all the time there was error. Thus pump P2 continues to run, and a bit too fast. The tank level thus dips below 50%, which finally causes the counter to decrement. After some time the counter reaches zero but the tank level is now below 50%. The negative error over time causes the counter to reach larger and larger negative values, which slows pump P2 and thus the tank level begins to rise again. The whole process repeats causing the tank level to oscillate around the set point. Eventually it settles at the proper set point of 50%.

That's the idea behind integral control. By tracking *accumulated* error we can remove effects like droop. The price is that this causes the output to oscillate. It turns out that you can cause the output to reach the set point quite fast, but it will oscillate widely. If it's made to act more slowly, the oscillations will be smaller. So, now the question is, how does one tune K_i to make the output settle quickly but without massive oscillations? The quick answer is usually trial and error! But a more intelligent approach is possible.

To make things simpler, start at the steady-state condition where the output has stabilized at 52%. Next realize that the pumps are not terribly fast with respect to the tank volume, particularly when they're working in a near-balanced condition. For example, when the effective feed rate is 16.2 GPM and the emptying pump (P2) rate is 18 GPM (its maximum rate), then the tank can only empty at 1.8 GPM. It can't go any faster! Working through the numbers you can find it takes at least 4 minutes to move the tank level from 52% to 50%. This provides a characteristic time over which accumulated error can be estimated.

Now, going back to the equation, the output has to be $u(t) = 90$ when $e(t) = 0$ so that the pumps equalize at 16.2 GPM. It's reasonable to assume that the error decreases linearly from 2 to 0 over about 4 minutes as the pump P2 is increased in speed by the integration control. This is illustrated in Fig. 3. Assuming the integral control was turned on at time zero, the accumulated error in 4 minutes is the area underneath the error signal. This is shown by the hatched region in Fig. 3. In this case, the accumlated error after 4 minutes is 4 %-min. Accumulated error always has units of "something" multiplied by time. This is why the integral gain has units of inverse time.

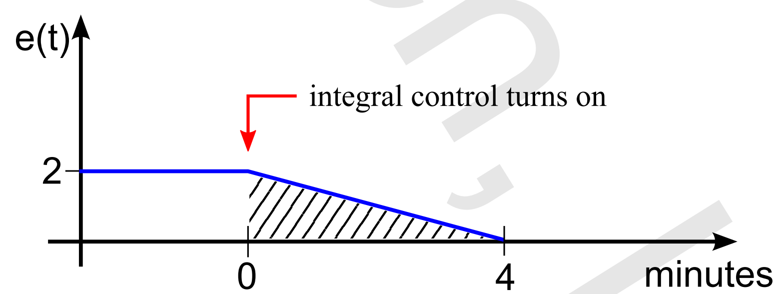

Figure 3: Error signal over time when the integral control is turned on at time zero to eliminate steady-state droop. Hatched area defines accumulated error.

The simulation software actually steps through time by the second instead of by the minute, so it's necessary to express the accumulated error as %-sec. After this conversion the accumulated error is 240 %-sec. These results are plugged back into the PI-control equation and evaluted at time equals 4 min.

$$u(t) = K_p e(t) + K_i(240) + B = 90 \tag{10}$$

Since the error signal $e(t)$ is zero at 4 minutes, the proportional gain does nothing. The bias B is already established at 87.5. So $240K_i$ must equal 2.5 in order for the equation to be true. Using this, we finally arrive at a value for the integral gain by solving

$$240K_i = 2.5, \tag{11}$$

or $K_i = 0.01 \text{ sec}^{-1}$. It's quite a lot smaller than the proportional gain! Still, it's actually a bit too large. We have assumed that the correction to the tank level would run as fast as it could. This may not be such a good idea since at larger errors of ± 10 the accumulated error could get quite large, thus trying to force the pump P2 to run very fast or very slow. Recall the integral gain obtained here assumed a pretty small error of only 2%. In order to allow for the fact that accumulated error could get much larger, it's necessary to downsize the integral gain. We know the error can get up to 10%, which is 5 times larger than we assumed to calculate integral gain. So let's also assume accumulated error can get 5 times larger. Then the integral gain has to get 5 times smaller to avoid large oscillations. As a good estimate in tuning then, we set

$$K_i = 0.002 \text{ sec}^{-1} \tag{12}$$

The plots in Fig. 4 show the result of adding this integral term.

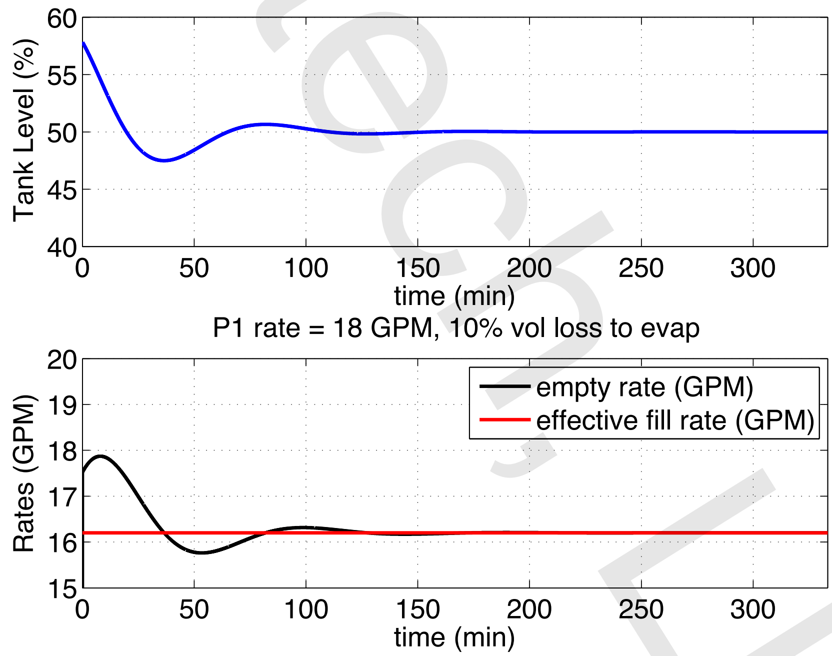

Figure 4: Tank level and pumping rates over time using proportional-integral control ($K_i = 0.002$) and starting at 60% full tank.

The plots show that the integral term in the PI control scheme did the trick. It has reduced the droop to zero and the tank level has settled in about 60 minutes with only a small amount of oscillation. It's worthwhile to look at some other values of K_i to see the effect. The first example is when $K_i = 0.01$ which was the initial guess based on only 2% error. This is shown in Fig. 5.

In this case the output has been clamped so that the rate of P2 never exceeds 18 GPM. But it's obvious that the high K_i value is *trying* to force the output much higher. In this case the droop is almost immediately removed, in that the tank level oscillates around the proper value of 50%. And, it may not be terribly apparent, but the oscillations don't last quite as long as in the case of $K_i = 0.002$, but they are much larger. This is a less-optimal solution and even runs the risk of causing the system to become unstable.

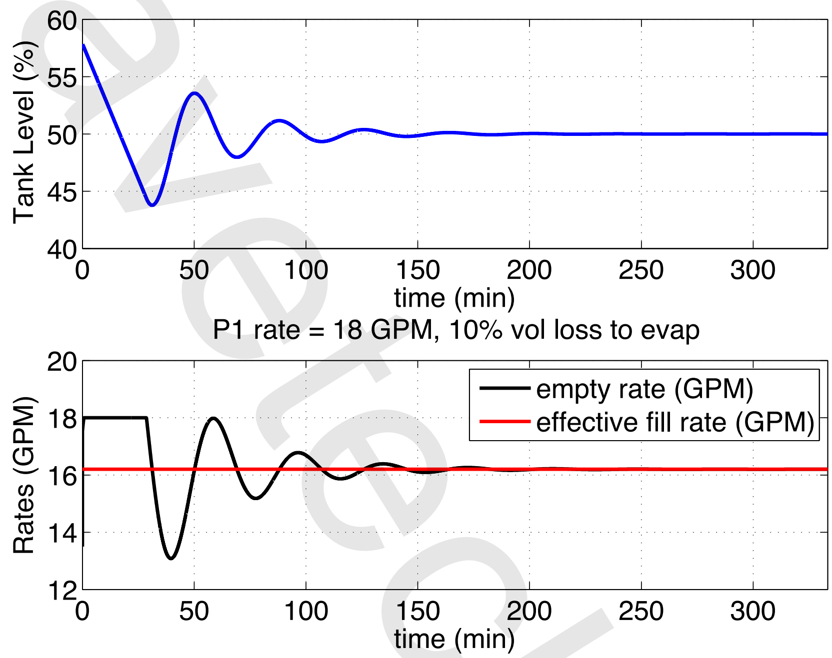

Figure 5: Tank level and pumping rates over time using proporitional-integral control ($K_i = 0.01$) and starting at 60% full tank.

In the last example integral gain is reduced to $K_i = 0.0004$, which is 5 times smaller than 0.002. These results are shown in Fig. 6. In this case the oscillations are very small which amounts to the system settling in only about 40-50 minutes. Further experimentation reveals that this integral gain is about as good as it gets.

There is still another thing to check. How do these settings work when the liquid level starts at 40% instead of 60%? Figure 7 shows the system behavior when it starts in this condition with $K_p = 1.25$, $B = 87.5$, and $K_i = 0.0004$. Because the system droop tends to push the tank level above 50% the system responds by overshooting then going back down to 50%. So this setting isn't quite as optimal in this condition. In Fig. 8, the integral gain is increased back to $K_i = 0.002$. In the end, we find that the original best-guess of $K_i = 0.002$ is probably best overall setting. It has a little overshoot, but settles out quickly whether starting at 40% or 60% tank levels.

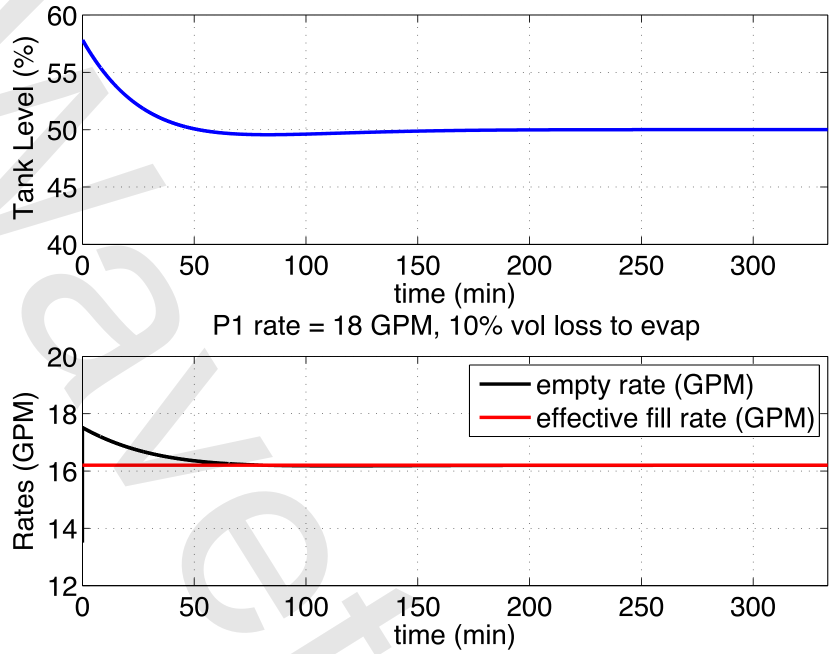

Figure 6: Tank level and pumping rates over time using proportional-integral control ($K_i = 0.0004$) and starting at 60% full tank.

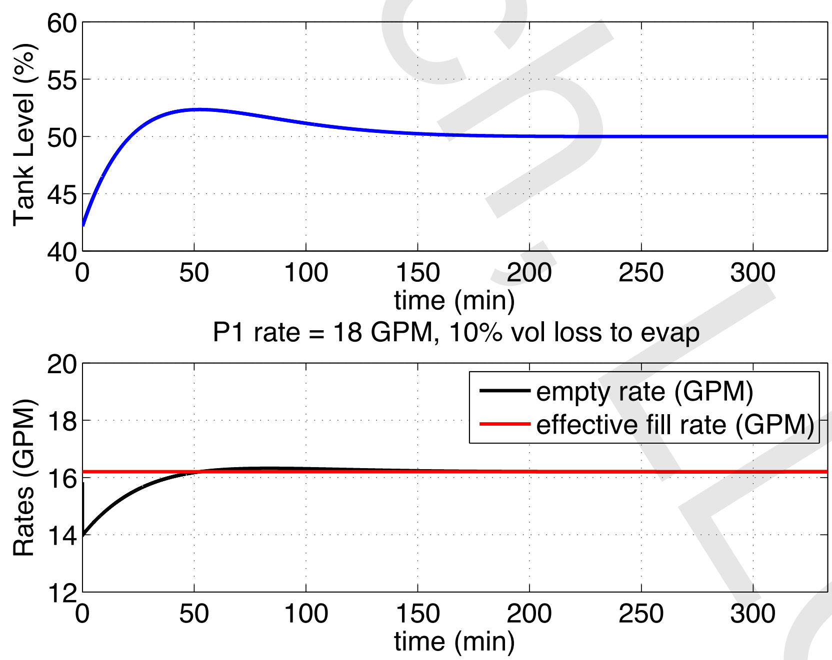

Figure 7: Tank level and pumping rates over time using proportional-integral control ($K_i = 0.0004$) and starting at 40% full tank.

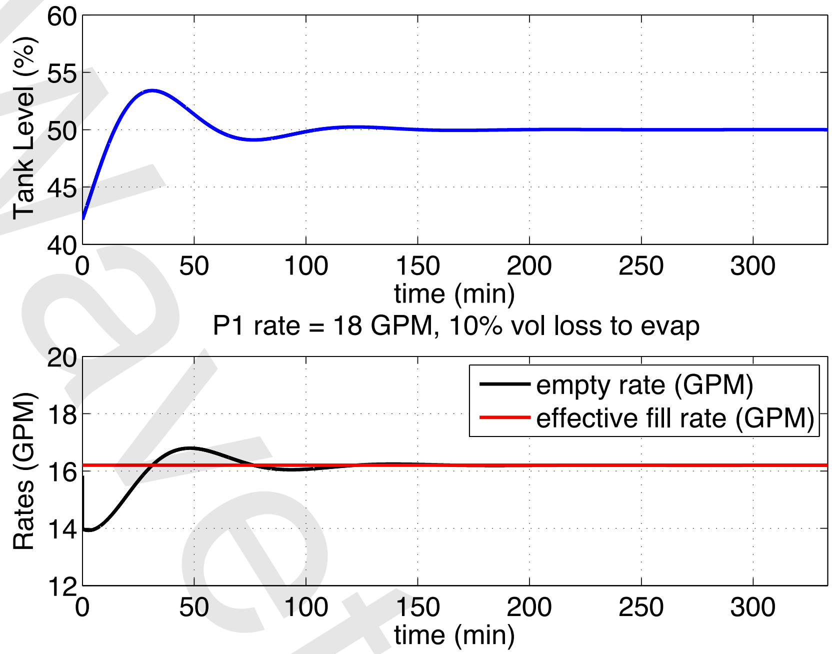

Figure 8: Tank level and pumping rates over time using proportional-integral control ($K_i = 0.002$) and starting at 40% full tank.

6.3 Proportional-Integral-Derivative Control

Finally, it's possible to add derivative control. Often times the addition of derivative control is not necessary. Our example application is one such case. And in some cases the derivative term can actually cause instability because it's pretty sensitive to system noise. The derivative term can basically amplify a small error and turn it into a big one, that gets further amplified by the derivative term! This might occur, for example, when the set point is suddenly changed by the user. However, the derivative term can also buy some advantage causing the output to settle faster. The derivative term can even aid stability in some cases.

In the case of our tank application, the derivative term does almost no good. But the reason is somewhat subtle. Before trying to explain this in text, let's first look at the results of adding some derivative gain. Note that the derivative gain has to be negative in this application otherwise it actually makes the settling time worse. In the first example, the derivative gain is set to $K_d = -1100$ sec. The tank level starts at 40% and $B = 87.5$, $K_i = 0.002$, $K_p = 1.25$ as before. The response is shown in Fig. 9. Note the time scale of the plots has been significantly reduced to show how dramatically the settling time has improved. The derivative term seems to have helped a lot by reducing the emptying rate of pump P2 to nearly zero, which obviously fills the tank quickly. The system settles at a tank level of 50% in about 30 min with almost no oscillation.

Why not go to a greater gain? When K_d gets to about -1300, the system actually goes unstable. Pump P2 periodically is maxed out and sharply drops and raises again over time, as shown in Fig. 10. At the start, pump P2 also shust off completely for 3 full minutes, then immediately

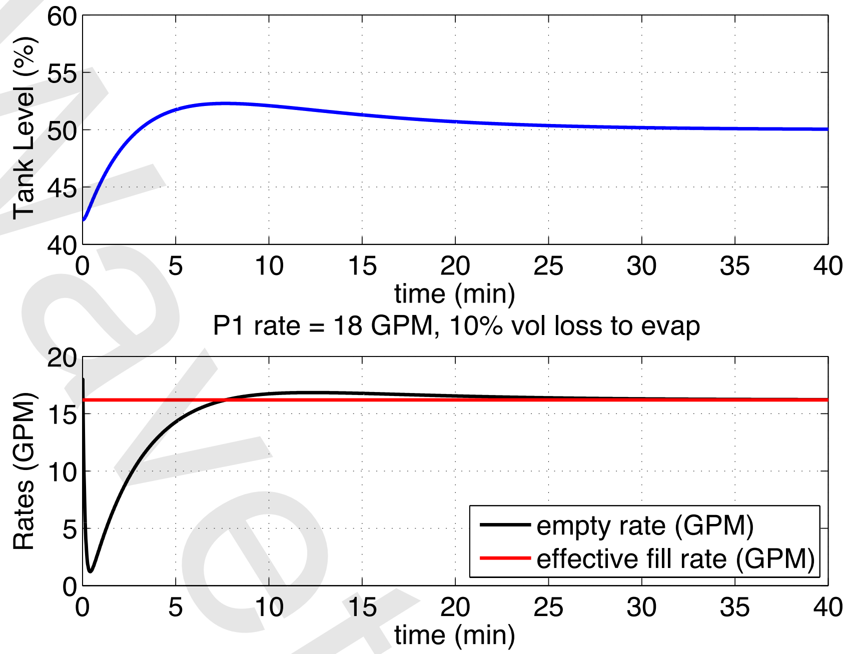

Figure 9: Tank level and pumping rates over time using PID control ($K_d = -1100$) and starting at 40% full tank.

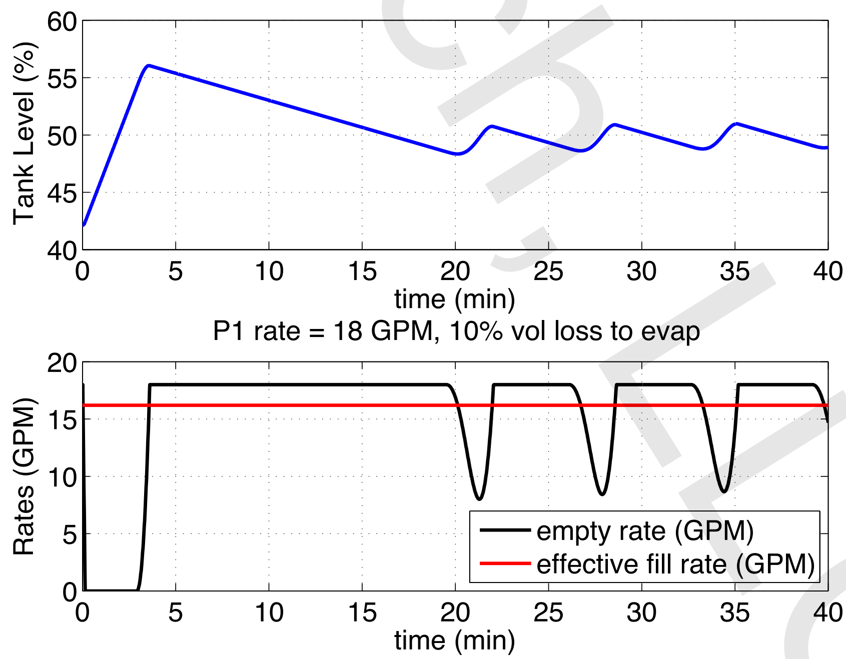

Figure 10: Tank level and pumping rates over time using PID control ($K_d = -1300$) and starting at 40% full tank.

rockets to maximum! This is clearly not a well-tuned system.

If $K_d = -1100$ dramatically reduces settling time, why not use it? The problem arises when the tank level starts at 60%. In this case the large derivative term tries to force pump P2 to run *much* faster than its maximum rate. Figure 11 illustrates the idea showing how pump P2 rate becomes clamped at 18 GPM. Perhaps this is acceptable in some cases, like this one, but in general it's not a good idea. Smooth and well-limited changes are often least difficult on the equipment and least prone to instability problems. The only way to avoid clamping P2 at its max rate is to reduce the derivative gain to $K_d = -100$, which has renders it almost unfunctional, as shown in Fig. 12.

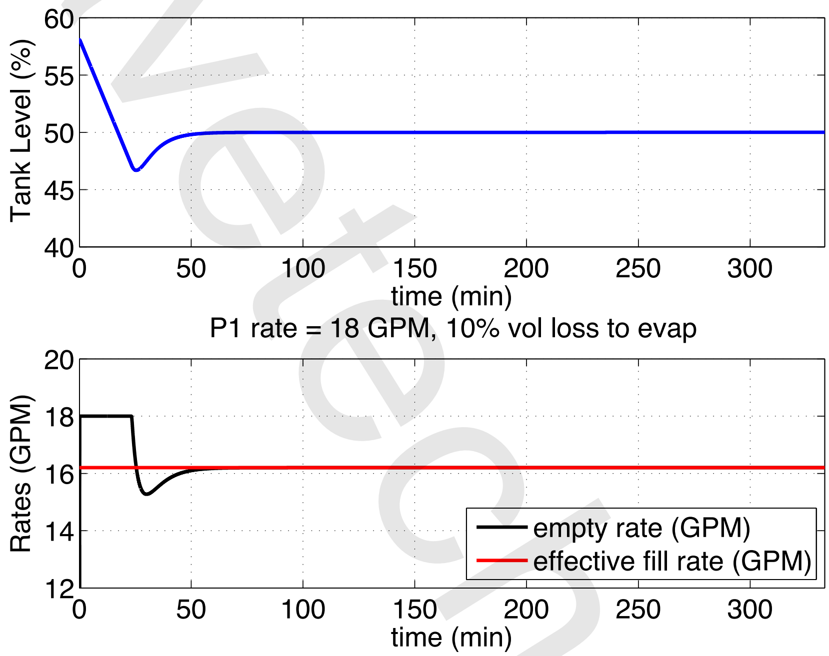

Figure 11: Tank level and pumping rates over time using PID control ($K_d = -1100$) and starting at 60% full tank.

Finally, it's worthwhile to verbally explain what the derivative term is doing. The derivative term observes the change in error from one step to the next. Recall when studying the effect of integral gain, we looked at how error accumulated with each time step. In this case we are looking at how error changes from one step to the next. For example, if the error at step 1 (time = zero) is $e(t) = 10$, and the error at step 150 (time = 2.5 minutes) is $e(t) = 5$, then the derivative term is 5/150=0.033 $\sec^{-1}$. In this example the error is reducing pretty quickly, so if we don't *slow down* the correction the output will overshoot the set point in a couple of minutes. To stop this, we can decrease the pump P2 speed by 25%, for example. Mathematically we want to cause $u(t)$ to be reduced by 25.

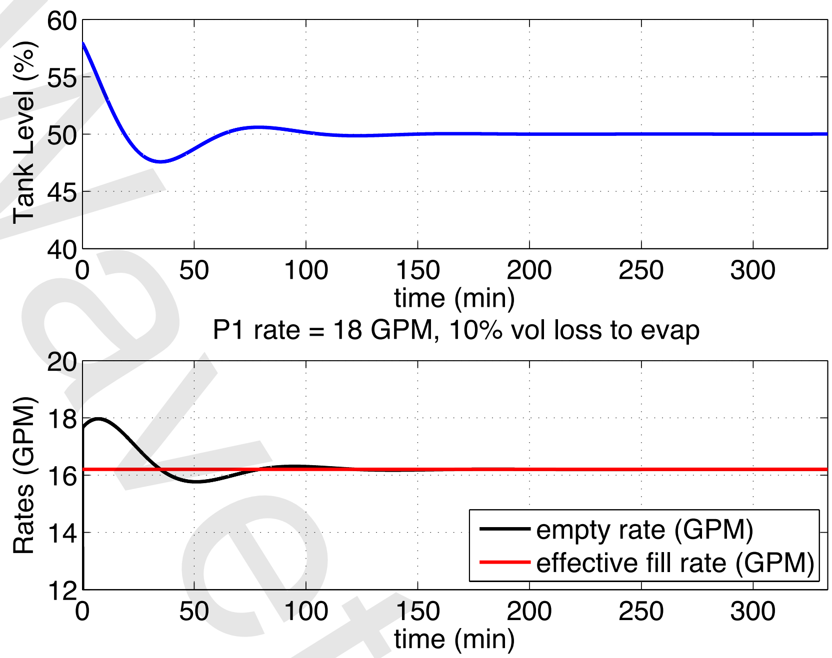

Figure 12: Tank level and pumping rates over time using PID control ($K_d = -100$) and starting at 60% full tank.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,\mathrm{d}\tau + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) + B \tag{13a}$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,\mathrm{d}\tau - 25 + B \tag{13b}$$

In other words, $K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) = -25$. If $\frac{\mathrm{d}}{\mathrm{d}t} e(t) = 0.033$, then

$$K_d = -758 \text{ sec}^{-1}. \tag{14}$$

This value is quite close to what we found by experimentation. Again, this would be a decent value to use for the derivative gain, *if we could use it*. But we can't because it forces the output to saturate when the tank level is above 50%.

7 Low Pump Rates

The original Process Constraints also indicated that the fill pump rate may be as low as 2 GPM, instead of the 18 GPM we've been assuming. To be complete, the PID tuning should also remain operable at low and intermediate fill rates. While this is not detailed, a single adjustment can make this possible. The bias must be adjusted according to the user-defined fill rate with the relation

$$B_{adj} = \frac{F_{act}}{F_{max}} B \tag{15}$$

where the adjusted bias B_{adj} is a version of the original bias B, scaled by the ratio of the actual fill rate versus the maximum fill rate. For example, if the user specifies 2 GPM fill rate, then $B_{adj} = 2/18 \times B = 9.72$. Then B_{adj} is substituted in for B in all the equations. Without this adjustment pump P2 behaves as though P1 is still running at 18 GPM. Since P1 is actually running much more slowly, pump P2 runs dramatically too fast. The adjusted bias corrects this.

8 Conclusions

As can be seen, the tuning of a relatively simple PID control loop has to take into account many parameters. However, when these are intelligently approached, the actual tuning is relatively straightforward. In our example, the tuning only required a few assumptions about the tank, the pumping rates, and the pumping rate limits.