

**Lawson Labs, Inc.**

**3217 Phoenixville Pike**

**Malvern, PA 10355**

**www.lawsonlabs.com**

**800 321-5355**

**610 725-8800**

**September 1, 2020**

## **General Purpose Data Acquisition and Control Excel® Template for Lawson Labs Precision USB Data Systems**

### **INTRODUCTION**

Our purpose here is to recover the promise of an earlier phase of personal computing. The open-ended capability for harnessing the power of hardware and software should be accessible to both the scientist and the tinkerer. In addition, we aim to provide protection from predatory pricing and obsolescence. The mechanism is to use the spreadsheet, which is now almost universally familiar, as the programmable element to acquire, massage, and present your data while performing the necessary control and housekeeping tasks.

The combination of Lawson Labs precision USB Data Acquisition and Control hardware with a generalized Excel® template provides a powerful, flexible platform for acquisition and control that is accessible without requiring programming skills. If you are familiar with Excel®, you are halfway up the learning curve before you begin. Particular attention has been paid to the control functions. We provide a flexible control engine that will outperform conventional PID controls while requiring a minimum of loop tuning

The Excel® template is provided at no cost. Up to ten data systems can run concurrently in one spreadsheet. The user can modify the spreadsheet template for their desired application, or, changes can be requested from Lawson Labs. In either event, generous support is provided to get started.

### **A) Software Installation**

On a computer with Excel® 2012 or newer installed, make a new folder for your project. Unzip the software archive and place the files in the new the folder. You can rename the GeneralUse.xlsm file as desired.

The host computer should be internet-connected when the Model 302 Interface is first plugged in to the USB so that the computer can automatically find and install the correct signed device driver. If you are running an older 32-bit operating system, back to Win XP, follow the Model 302 website instructions for installing the 32-bit drivers at:

[http://www.lawsonlabs.com/model30x/30x\\_GetInstallFiles2.php?model=302](http://www.lawsonlabs.com/model30x/30x_GetInstallFiles2.php?model=302)

Make a backup copy of the spreadsheet before continuing. Run the .xlsm file to begin. It should invoke Excel® when starting. The LLI\_USB.dll serves for 32 Excel® and LLI\_USB\_64.dll for 64 bit. If both dlls are present in the application folder, the correct one will be used automatically.

## **B) Connections**

We will here assume the hardware is a Lawson Labs Model 302 24-Bit Data Acquisition System. Model 302 documentation regarding analog inputs and digital and analog outputs applies here also.

The power supply connects at the barrel connector and the USB cable connects the Model 302 to a USB port on the host computer. Any individual USB port should work, but remember that driver installation is for a particular port, and must be repeated if you plug the Model 302 into a different USB port.

Connect up to six analog inputs to the Model 302. Three analog outputs are available, as are four digital inputs and 8 digital outputs.

## **C) Operation**

### **Overview**

There is one main sheet for each connected device. As supplied, there is one main sheet and one graphing sheet for that device. All the main sheets should come first, left to right, then the Chart tabs, or other sheets, which can be added as needed, provided they are to the right of the main sheets. With only one device the only requirement is to begin with the main "Sheet1". For multiple devices, the names are fixed at Sheet1, Sheet2, etc,. The corresponding graphs at Chart 1, Chart2, etc. can be renamed or deleted at will.

Begin by entering the USB Device ID number of the first connected device in cell B2 on Sheet1. Enter 0 or no entry on any additional sheets to begin. That will cause any additional sheets to be ignored. Once you have one device running properly, others device ID numbers can be added on subsequent sheets. All entered and connected devices will be initialized together at the start by the Initialize button in the Device Control window. The Device Control message window will show success or failure. After success, each device, on its own sheet, can be independently started and stopped using the Start/Stop button in the Device Control window. Note that the Device Control window tracks with the selected sheet, showing the current device number in the middle left. Also note that the status of all connected devices shows in the status window below the messaging window. If any device is scanning, a warning not to edit cells will be displayed.

The blue button centered on Cell E1 will show or hide the Device Control window. It is possible for the window to be out of sight behind another. When that happens, click show/hide twice in order to bring the Device Control window to the top.

Important: Excel® does not tolerate any editing while data is flowing into any sheet. If even one device is scanning, no editing is allowed on any sheet. The only changes allowed during scanning are in the

Device Control window. The Device Control window allows you to stop a scan, or manually change an analog or digital output. You can navigate among tabs or move around on a spreadsheet during scanning, but be aware that mouse and keyboard activity can disrupt control timing.

NOTE: For turnkey applications the spreadsheets can be locked so that a user cannot edit at a disallowed time. Contact Lawson Labs for more information.

Basic set-up information is found in the upper left of the sheet. The cells that define the incoming data have a yellow background. Cells containing input information for the controls are also highlighted in yellow. Cells with brown font are tied into the underlying VBA code, and should be left alone, at least to begin.

See the Model 302 Manual for a full description regarding setting the data acquisition parameters.

## Details

Cell B5 = Internal Data Rate

The data rate should be set for the best noise rejection consistent with adequate scanning speed. Line frequency usually dominates among the noise. The best line frequency rejection will be had at a 60 Hz data rate in the US, or 50 Hz in Europe. If you want the fastest response consistent with good 50 and 60 Hz noise rejection, you will run the Model 302 at 800 Hz, block average 4. That rate provides ten scans per second when sampling 4 channels with active zero drift suppression. Do not tinker with the rate and block average. For good noise rejection during faster scanning, use only 800Hz, Block Average 4 or 8. An internal rate of 400 Hz with Block Average 2 is not even remotely equivalent.

Cell A5 = Block Average (see above)

If your rate is 50 or 60 Hz, you can reduce the quantity of data by increasing the block average to whatever you want. For the 800 Hz internal rate, leave it at 4 or 8 only. Above a block average of 8 you will be better off decreasing the the internal data rate to the line frequency. In any event, the resulting scanning rate will show in cell B6. If your Internal Data Rate is the line frequency, increased block averaging decreases noise as the scanning rate decreases.

Cell B8 = channels to scan

Enter the number of active inputs. Scans always start with channel 1 and proceed incrementally. Adding channels will result in a slower scan rate. The scan rate shows in Cell B6.

Cell B10 = Scan type

Defined types are 1 through 6. Type 6 is often the preferred choice. It provides for multiple channels with active zero drift correction and with digital input information interleaved with the scan data. See the Model 302 Manual for other scan types.

Cell B12 = Channel number for single-channel scanning

If you happen to be looking at only one channel, this cell determines which channel it will be.

Cell B15 = Local average

This variable sets a running average. It does not alter the data rate. Use it sparingly if you are doing real-time control. The data in your spreadsheet will be smoothed by this factor, but latency will increase. If you want to preserve the actual raw data, set Local Avg to 1.

The analog scan data is written into columns starting at J, which contains the scan number. Column K is for channel 1, L is for channel 2, etc. Digital inputs, if enabled, are written into the column after the last analog input channel. If you let the system run indefinitely, it will eventually write to the bottom of the spreadsheet and data preservation will stop, but any controls will continue to run. The most recent scan is also found starting at column A in Line 31, with the most recent Digital input in Cell E33. These fixed locations are for convenience in applying the control rules.

Cell D15 Enter the last scan number here to stop the scan automatically after a certain number of scans. Enter zero here for continuous operation.

Cell D18 If equal to 1, the spreadsheet will be saved at the completion of the number of scans specified in D15. The file name will have a date and time stamp appended to the file name.

The device status is found in Cell A17. The current scan number and current time appear in Cells A19 and 20. If the current scan time is not close to the actual time, the scan is not running.

There is one more important piece of information that appears in Cell A18 during scanning. It is the number of scans collected from the input buffer at any one time. For a control loop to work well, it is essential that the computer is keeping up with the incoming data. That means the number of scans to process should stay small. 2 or 3 is a good maximum. It is OK if the number ticks up occasionally and comes right back down, but that should be a rare occurrence. You can watch the buffer function by holding a mouse button down on the title bar during scanning. That will stop program execution, but not the data acquisition. When you release the mouse button, the Scans to Process number will jump up to 99, and then drop back down once the spreadsheet catches up with the buffered data. A properly functioning system will catch up quickly.

The button at A22 clears the spreadsheet of your data. be sure to save the data first if you want to preserve it. Starting a new scan also clears the old data, so do not start a scan unless you have saved any earlier data that you want to preserve.

Note: The template can be saved at any time, scanning or not. Press Alt F S to save with the same name, or Alt F A and provide a new name. The data will collect in the input buffer while you are keying in the new name. If control loops are running, don't take too much extra time entering the new name.

Do not move cells around without extra caution because some cells are referred to by absolute addressing. Pay particular attention to cells with brown font, since those cells are referenced in the underlying VBA code.

Keep regular back-ups of your work. It is easy to make an unintentional change which might have non-intuitive consequences. To aid troubleshooting, it is better to save regularly and to be prepared to back up a step if an unexpected error crops up.

## Operation example

Start with a single device to begin. Enter the device ID number on Sheet 1 in the upper left. If the control box is hidden, click the button labeled show/hide device control. You may need to click twice to view a hidden Control box. Click the Initialize button in the control box. Wait a few seconds for the "Ready" message. Once initialized, you can start a scan. Do not attempt to change cell values while scanning. A "Do not Edit" message in the control box reminds you that editing is not allowed during scanning. If you need to edit the spreadsheet, stop all scans first, then edit, then restart the scans.

Verify that your data is correctly appearing in the spreadsheet columns. If you intend to do real-time control, confirm that the Scans to Process number stays mostly at 2 or less .

A sample chart sheet is included for the first device. Use the Excel® charting capabilities to obtain the presentation you want. The device sheets must be named Sheet 1, Sheet 2, etc. The chart sheets can be named at will, but must follow the device sheets in the tabs. Huge charts are heavy users of processor cycles. If you have large charts, be sure you are not falling behind, causing data to accumulate in the input buffer.

Editing when not allowed will stop all scans immediately. The status change will be noted after a wait of 30 seconds. It should be possible to restart by clicking start/stop scan once on each stopped device sheet. (Remember to save any data first, since restarting clears the previous data.) The status should then show as scanning. If you cannot restart all the scans, stop any running scans and click Initialize. You may need to repeat that step. Worst case, you may need to quit and restart the software before continuing.

## D) Additional Devices

A sheet can be copied to add a device. First, stop any scanning devices. Right click on the tab, Select move or copy, check the create a copy box, and chose the location. The Device tabs must be in the first 1 to 10 locations, and must be labeled Sheet1, Sheet2, etc.). This limitation is an expedient that could be lifted with more programming. When prompted, yes, you wish to copy everything over. Then, enter the new device ID and rename the second sheet to Sheet2 or third sheet to Sheet3 or fourth sheet to Sheet4 etc. (case sensitive). It is good practice to save, exit and relaunch after adding main sheets.

Initialize the new device(s). All active devices will be initialized together. Then, complete the set up of your added device(s). Thereafter, scans are started or stopped independently on the various sheets. Still, any spreadsheet editing, anywhere, will stop all running scans, so be sure to set up everything in advance before beginning any long-running tests. If a scan is running on another sheet, you will see the editing-not-allowed notice in the Device Control box. That notice is a warning not to edit or you will stop all other scans in progress. If you do edit and stop all scans, you will see that the status of stopped devices is still shown as "Scanning" for half a minute. To recover, click start/stop in the Device Control box to first stop, and then again to restart a scan.

To temporarily disable a device, enter 0 for the device ID number. To re-enable, enter the device ID number and click Initialize.

We have not fully explored the limit of how many devices can be run at once. That number will be determined by the computer operating system, memory and speed, as well as as by scan rates and control requirements. For now, there is an arbitrary limit of ten devices.

## **Control**

Analog outputs may be changed manually from the Device Control window whenever a device is initialized. You need to click on the button to the right in order to actually send the analog output update. Remember that changing a DAC value manually while a control loop is running in the background may temporarily disrupt the control. The most recent DAC output values show in Cells A32-34, whether those values were set manually or automatically. Digital outputs are active, so they will respond immediately from the Device Control window or to any programmatic changes. If Digital inputs are included in the scan, you will see the input word at the end of each scan, and the last digital input status will show in Cell E33. To set a DAC output according to your own rules, put the formula for the output voltage in Cell D27 for DAC1, D28 for DAC2, and D29 for DAC3. The DACs will be updated at the intervals entered in B27-29. For manual DAC setting only, enter 0 for the interval. To use the control rules defined below, cells D27-29 should be set equal to cells C27-29, which contain the automatically calculated output voltages as defined here.

## **Background and Overview**

There is a tendency to take too much data too fast, and then to average it to remove noise. The result is much the same as taking quieter data at a lower data rate, except for all the extra computing overhead involved. Start by determining the highest frequency of meaningful data coming from your sensors. There is rarely a benefit to be had from a sampling interval less than a fourth of the time constant of the inputs. Often, one scan per second is more than enough. For more than 10 scans per second, hardware control is usually indicated. Contact Lawson Labs for hardware control options.

The physical system being monitored has a maximum rate of change. For example, when controlling the temperature of an oven or bath, it can only heat or cool so fast. Adjusting a system aggressively at a rate faster than the system can respond is an invitation to oscillation. Cycling faster than the response time can make the control worse, not better. That said, you *will* need to cycle fast enough to do the job. A computer-in-the-loop control system's performance will be bounded by the worst-case latency, not by the average latency, so leave a timing margin.

Remember: The object is to match the control timing to the response of the physical system. Faster is not necessarily better.

There are three built-in control loops, one per analog output. These can be modified to suit, but have configuration parameters that should allow a non-programmer to get superior control results in a minimum of time. We use predictive rules that are different from normal PID control, so tuning these control loops necessarily diverges from the usual practice. Please apply the techniques described below instead of ones you may be in the habit of using. You will be rewarded for your efforts.

There are several gain and timing factors that determine loop response. That complexity is needed to optimize loops with long intrinsic response times. Faster, or non-critical loops will work fine with default values and a minimum of tuning.

## Steps to set up a control loop:

Warning: A safety override mechanism should be part of any control loop that can create a hazardous condition. The override should be independent of the control software, so that regardless of what happens to the computer, conditions are always safe.

1) Arrange a feedback signal that is representative of what you are controlling; temperature, pressure, position, etc.

2 Analog output 1, (or DAC channel 1,) is matched to analog input channel 1 for feedback. DAC2 is matched to analog channel 2, and 3 to 3. Connect an analog output to the means of proportional control for the loop. For example, a PDr4 may run a heater for a temperature control. Three control loops are pre-defined in columns A-F, beginning at line 30.

3) Determine the polarity of the system. If increasing the output voltage increases the temperature, then the gain will be positive.

4) Put the desired feedback reading in the setpoint cell (B41 for DAC1). The cell can contain a formula. You may wish to scale the feedback and setpoint in engineering units, or you can match the setpoint to the +/-5 volt feedback voltage.

5) Start with typical settings for the rest

Data rate	60 Hz in US (or 800 Hz with block average 4)
Smoothing count for running average	10
Update interval	1 second
Predictive gain factor	0.5
Proportional gain factor	50
Self-learning gain factor (integral)	0.5
Differential term for oscillation suppression	0.01

The control loop accepts the input, compares it to a setpoint, and produces a proportional output based on the pre-defined settings.

## E) Tuning the Control Loops

### Easy Tuning summary:

Start from default values.

If oscillating, reduce the Proportional gain until oscillation stops.

If not oscillating, increase the proportional gain until oscillation starts, then reduce it marginally to stop oscillation

Disturb the system. One way is to make a small change in the setpoint. If the system over-corrects, increase the Predictive gain factor. If it under-corrects, reduce the Predictive gain factor. Repeat.

The screen shots below show the results of too much or too little gain or prediction.

Why does it work?

The underlying problem with loops with long delays is a tendency toward oscillation. Feedback easily gets out of phase, since time constants can change with conditions. You can go to great lengths to try to match the response to the delay, and some of that effort is essential, but the match is never perfect, so the amount of gain the system will tolerate without oscillation will always be less than the ideal. Our approach looks at changes in averages to obtain the prediction, thereby avoiding excessive sensitivity to noise. It then bases the corrective response on extrapolated future conditions.

When you inevitably settle for reduced gain, the DC error gets bigger, so you often need an integral term to remove any offset should the control settle at not quite the exact value. If the integral term has a short time constant it will act much the same as additional gain, causing oscillation. With a longer time constant for the integral term, the fine tuning will be sluggish after a change in conditions.

So, instead of an integral, we use a self-learned offset to correct the DC response. With computer-in-the-loop control, the self-learning is easier and more controllable than attempting to fine-tune the gain to the ragged edge of stability. The self-learning is only applied when the system is stable, to prevent an upset from allowing the self-learning to detune the system. We call the learned term the bias. The bias factor determines how aggressively the self-learning is applied. To relate the above to conventional PID, the proportional term is analogous, the predictive gain is a form of derivative term, and the bias factor is a sort of integral term.

#### **Details for the controllable factors in the control template:**

Data rate

Update interval (The loop is sampled, i.e. it recalculates once per sample interval)

Smoothing count for running average (sets effective timebase when combined with data rate)

Proportional gain factor

Predictive gain factor (derivative)

Self-learning bias factor (integral)

Differential term for oscillation suppression

Max out voltage

Min out voltage

Data rate:

The choice may be made for you. The best line frequency rejection will be had at a 60 Hz data rate in the US, or 50 Hz in Europe. If you want the fastest response consistent with good 50 and 60 Hz noise rejection, you will run the Model 302 at 800 Hz, block average 4. That rate provides ten scans per second when sampling 4 channels.

Note: We generally recommend leaving the running average at 1 unless your loop is very slow or your data is too noisy.

Update interval:

The range is 0.2 seconds to several hundred seconds. More frequent updates can give faster loop response, but frequent updates increase the effect of any timing imperfections and increase the loop sensitivity to noise. If you want to measure the slope of a line accurately, it helps to pick two points that are not right next to each other. So, updating the loop too fast can sometimes make the loop response worse instead of better. A one-second update interval is usually entirely satisfactory. The update interval should not be shorter than the scan interval to insure that there is always new data available.

Smoothing count (an internal running average):

Generally, a count minimum of 3 is required to suppress oscillation. The maximum can be over a hundred. Larger counts may be better for systems with longer time constants. Smaller counts can provide faster response, but may require a reduction of the proportional gain to maintain stability. You can usually get good loop performance with almost any count setting, if other factors are adjusted to match. A count of 10 is a good place to start for a loop with more than a little built-in delay.

Proportional gain factor:

This factor is the underlying loop gain. The more gain, the faster the response, and the more tendency to overshoot and oscillate. As a rule of thumb, reduce the gain until the system is stable. Then, if you want to push the limits, increase the gain until you see a little oscillation. After that, tune the other factors to stop the tendency toward oscillation. If you succeed at that, boost the gain again, and repeat. Usually, you will want to keep the system from hitting the stops unless there is a big disturbance. If you see the output voltage alternating between the maximum and the minimum values, you almost certainly need to reduce the gain.

Note: Some systems run fine with excess gain. You might see the output alternating between too high a value and too low a value. As you reduce the gain, the two values tend to converge. If you stay out of the stops, settings that cause this type of alternating behavior can be entirely satisfactory, if a little noisier than settings that result in a slowly changing output voltage.

Predictive gain factor:

The sample interval multiplied times one half the average count define an effective base period. With a one second interval and a count of 10, the time base for calculating the slope is 5 seconds. You would then be effectively looking 5 seconds ahead for your predictive control with a Predictive gain factor of 1. If your system response has a long built-in delay, 5 seconds may not be long enough. A Predictive gain factor of 2 would cause the loop to look 10 seconds ahead. If your system response is quick, 5 seconds may be too long. A Predictive gain factor of 0.5 would cause the loop to look 2.5 seconds ahead, etc. In practice, it is simpler. If the system undershoots after a step change or other disturbance, decrease the Predictive gain factor. If it overshoots increase it. Successfully increasing the Predictive gain factor can allow for an increase in Proportional gain. To the extent that these factors interact, they tend to go up or down together for a loop that is reasonably well-tuned.

Note: The prediction is symmetrical. That may not be the case in your system. You can make the prediction smarter if you want to take the time. For example, your system may heat faster and cool more slowly. But remember that relative rates of heating and cooling will not match at a high temperature and at a low temperature. Unless you need the fastest response possible, you do not need to concern yourself with tuning out asymmetry imperfections in the prediction, but there may be response improvements to be had, if you want to investigate adding asymmetry.

Self-learning bias factor (integral):

This term performs a function equivalent to a conventional integral term. The loop will, over time, gradually eliminate any difference between the process set point and the feedback value through the action of this term. If the Self-learning gain factor is large, say 1 or more, it will act very much like increased Proportional gain, and may induce oscillation. Smaller values mean it will take longer to remove any error, but stability is not degraded. The self-learning result is seen in the cell labeled "bias". It sets a default output voltage which should find its way close to the ideal output voltage under stable

conditions. If the system is disturbed, and falls temporarily out of regulation, the self-learning is turned off until the system re-stabilizes.

Note: If you know the desired bias for a particular setpoint, you can write it into the bias cell. The Self-learning gain factor will then proceed from that value. If the Self-learning gain factor is zero, you can programmatically set the bias term wherever you want, whenever you want.

Differential term for oscillation suppression:

You can leave this term at 0.02 or less in most applications. This term is the most anticipatory indicator of a change at the feedback point. If the output is going up, you can expect the feedback will go up in the future. By subtracting a small percentage of this factor from the output, oscillation can be suppressed. For most applications, this factor can be set to 0.01 and ignored.

Max and Min out voltage;

You can set the limits of output voltage swing here. For example, the PDr4 begins to respond at about 0.4 volts. We set the minimum output to 0.4, to match. If your heater could overheat your oven beyond a certain control voltage, set a safe maximum value here. The standard hardware has built-in limits of 0 to 5 volts, or +/- 5 volts.

#### **Detailed Tuning Discussion:**

For reference, typical default settings with cell references for DAC1		Cell
Data rate	60 Hz (or 800 Hz with block average 4)	A5 & B5
Smoothing count for running average	10	B36
Update interval	1 second	B27
Proportional gain factor	50	B43
Predictive gain factor	0.5	B47
Self-learning gain factor (integral)	0.5	D45
Differential term for oscillation suppression	0.01	B44
hiLim, volts	5	B45
LoLim, volts	0.4	B46

Run, and confirm that the loop is acting to cause the feedback to approach the set point.

If no oscillation. increase the Proportional gain until you see oscillation, then decrease it a little. If you do see oscillation, decrease the Proportional gain until it stops.

Watch the settling after a disturbance. If the loop overshoots, double the Predictive gain factor. If it undershoots, halve it. Fine tune once you get close.

If your feedback data is noisy, the prediction may be erratic. Increase the Smoothing count to obtain better stability, or apply a small amount of running average, if necessary. If you double the smoothing count, halve the Predictive gain factor to maintain about the same amount of prediction.

Increase the Self-learning gain factor to improve set point matching under stable conditions. If you see oscillation, back off on the Self-learning gain factor.

Once the loop is running reasonably, you can fine tune as much as you want, or you can move on to the next task. The general strategy for fine tuning is to increase the Proportional gain until a little oscillation appears. Then, increase or decrease the Predictive gain factor by, say, 20% and look for improvement. If you don't need to push for the maximum possible gain, your task is easier. If you want absolutely the fastest settling time, try decreasing the average count, increasing the Predictive gain factor and retuning the Proportional gain. Remember that the time response of the system may be different under different operating conditions, so effort spent fine-tuning a particular transition may not optimize response in another situation. Don't push the proportional gain too close to the limits unless you can tolerate a little oscillation.

### **Displayed values and intermediate calculated control values, and what they mean:**

Cells are given for DAC1.

The last analog input voltage from the feedback channel (Cell B42) is shown in Cell C42

A32 = last DAC output voltage

B27 = update interval (input value)

B41 = setpoint (input value)

B36 = Smoothing count for running average (input value)

C36 = Running average of DAC output

D36 = Difference from previous running average

D42 = the error term, ie the feedback minus the setpoint

D43 = delta error, or the difference between the error and the previous error

B43 = Proportional gain term (input value)

B44 = Dgain leading indicator term to suppress oscillation (input value)

B45 = High limit (input value)

B46 = Low limit (input value)

B47 = Predictive gain term (input value)

D45 = Self-learning adaptive factor (input value)

D44= Pterm, or the error times the proportional gain

D46 = The leading Dterm for oscillation suppression

D47 = Predicted future Error

E47 = Predicted future feedback

F47 = Predicted feedback corrected for the Dterm, proportional gain is applied to this term

F45 = bias term, self-learned

F42 = new output voltage

This block of cells provides some transparency into the process as well as some access for customization. Cells with numbers in brown font are referenced by the underlying VBA code, so those cells cannot be repurposed. If the brown cells do not have a name assigned, they cannot be moved without changing the VBA code. Aside from those limitations, the user is free to change or add function. If you want to add other factors into your control rules, change the formula in Cell G8. There is no theoretic limit to the complexity of the control rules. You could factor in multiple analog inputs as well as digital I/O states.

Note for programmers:

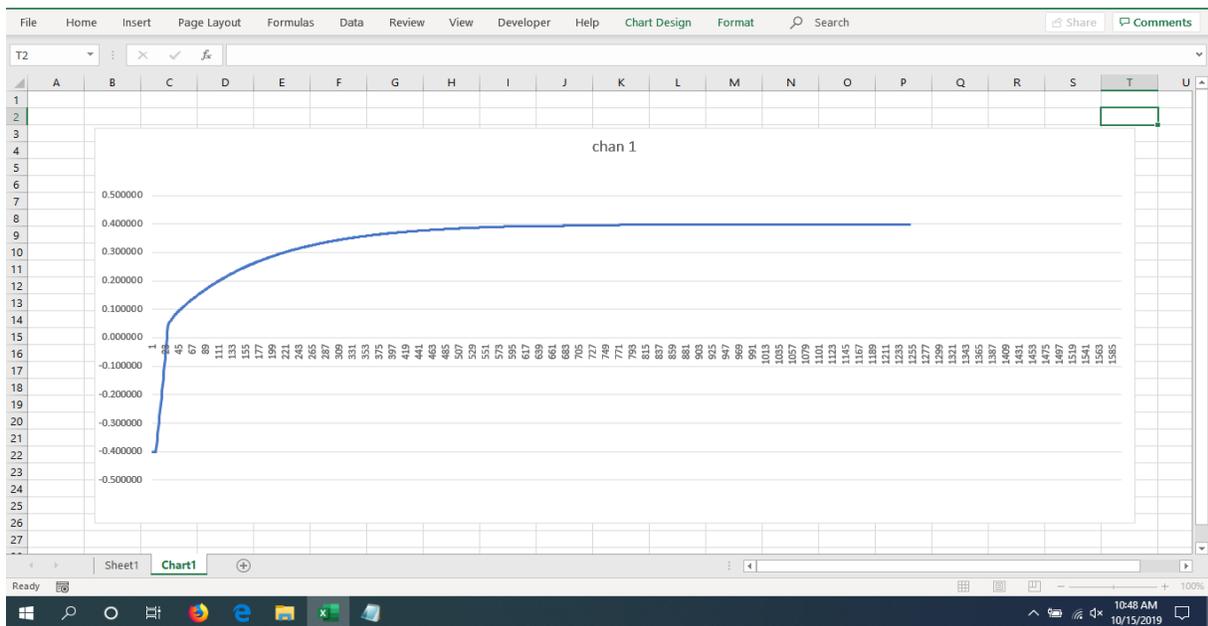
The control rules run once every control interval. Spreadsheet formulas update according to the spreadsheet recalculation rules. Time sensitive variables like deltas need to be calculated inside the

timed code, not in spreadsheet formulas. If we have not provided the timed information you would like to see for your application, please contact us for custom modifications.

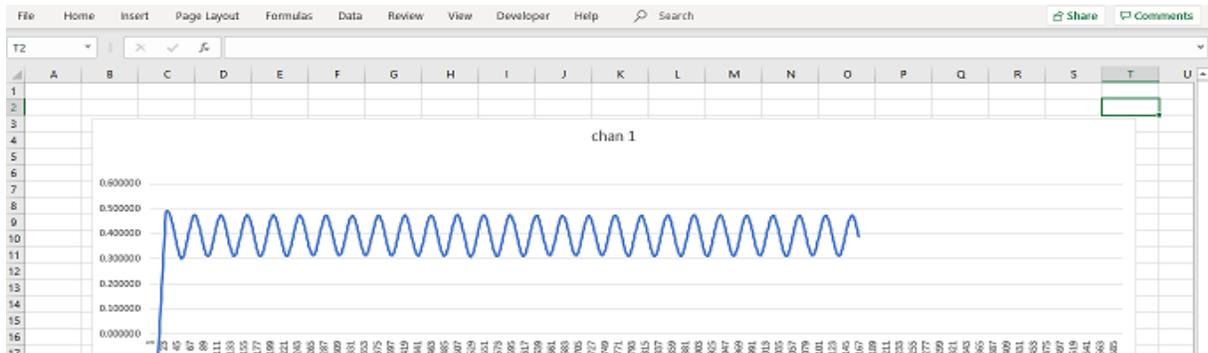
### Quick and easy tuning, better-than-most control

- 1) Set everything at defaults.
- 2) If control oscillates, decrease Proportional gain until it stops.  
If it doesn't oscillate, increase Proportional gain until it starts, then back the gain down a bit
- 3) Apply a step change. If control overshoots, increase Predictive gain. If it undershoots, decrease it. Repeat #3 until close enough
- 4) Done

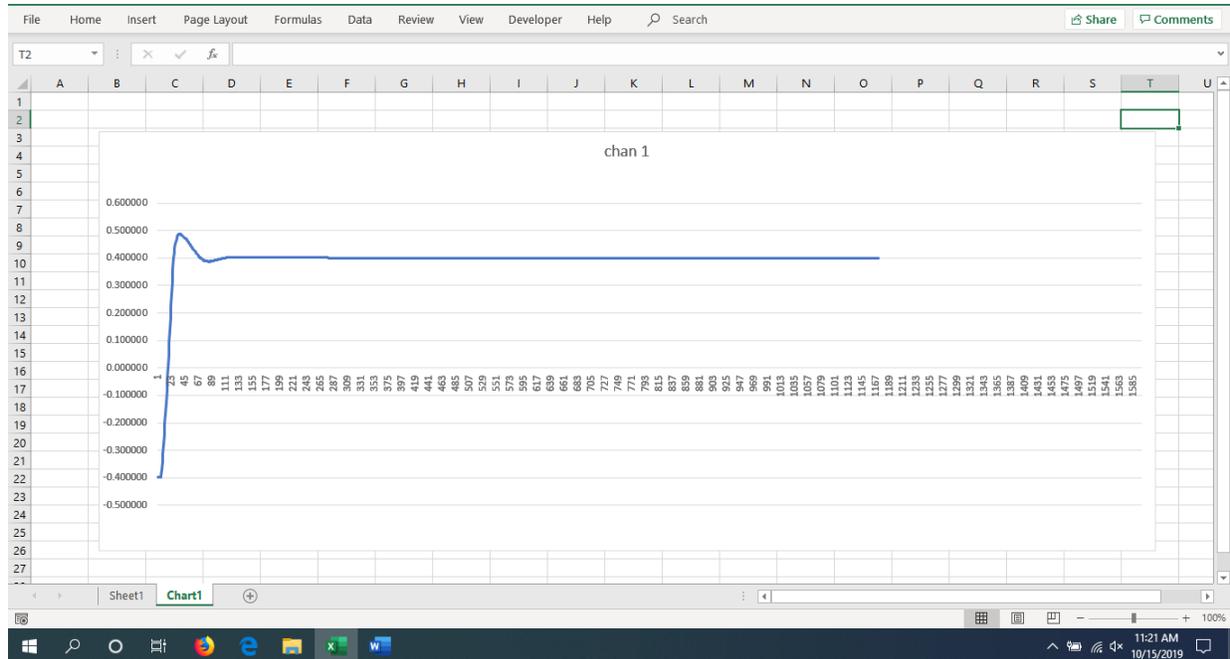
The screenshots to follow show characteristic plots for aid in setting the gain.



### Too little Proportional gain or too much Prediction



## Too much Proportional gain



## Gain and Prediction about right

If you have difficulty finding a good gain setting, try increasing the control interval to 2 seconds, or maybe more. Longer control intervals slow loop response, but with reduced gain, that may result in higher stability.

## F) Troubleshooting

This software is being continually improved. There may be occasional issues. For example, cell locking during scanning is not implemented. It is possible to crash Excel® if worksheets are edited during scanning. If you have difficulty restarting after a crash, unplug all USB cables for a second. That should allow Excel® to fully unload prior to a new start. If not, look in the task manager for Excel®, and End Task for any Excel® remnant still loaded.

If you should have two copies of the spreadsheet open at once, nonsensical behavior will result. In that case, close all Excel®, and check the task manager, as above, for any Excel® remnants to unload.

Keep a backup copy of your work, and do not rely solely on the Excel® document recovery features.

In order to run long-term tests, you will need to turn off any power saving features that would stop the scanning. In Win10, you need to specifically turn off power saving on the USB ports. You should also turn off automatic updates.

## **G) Resources and Contact Information**

Specifications and Documentation

[www.lawsonlabs.com/model30x/Model302.php](http://www.lawsonlabs.com/model30x/Model302.php)

Installation issues are covered extensively at

[www.lawsonlabs.com/model30x/30x\\_GetInstallFiles.php?arg2=table&model=302](http://www.lawsonlabs.com/model30x/30x_GetInstallFiles.php?arg2=table&model=302)

General discussion of noise issues

[www.lawsonlabs.com/HowtoMinimizeNoise.pdf](http://www.lawsonlabs.com/HowtoMinimizeNoise.pdf)

Call or email Lawson Labs for help with customization, or for noise or stability issues.

Phone 610 725-8800

email:

[lawsonlab@lawsonlabs.com](mailto:lawsonlab@lawsonlabs.com)

website:

[www.lawsonlabs.com](http://www.lawsonlabs.com)

***Lawson Labs, Inc.***

3217 Phoenixville Pike

Malvern, PA 19355