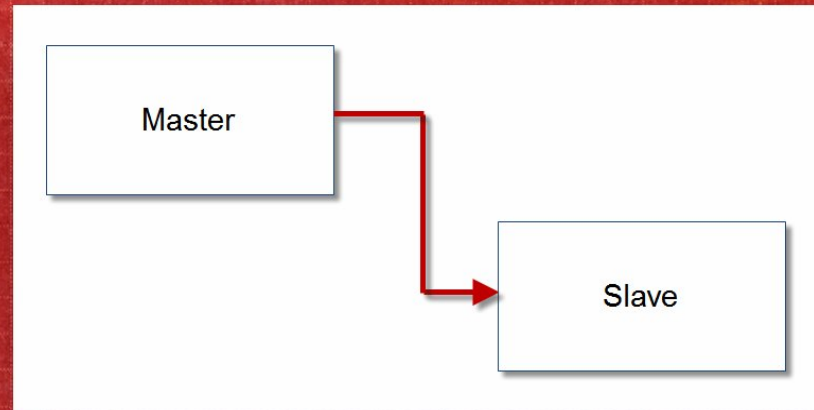


Cascade Control

The Definitive Guide



What it is.
When to use it.
How to use it.
How to tune it.

CONTENTS

CONTENTS.....	1
An Introduction to Cascade Control : The Theory	2
Cascade Control in Practice: A Start Up Feed Heater.....	4
Here's how it works.	7
OK, but why use a cascade architecture in the first place?.....	7
How a single loop strategy would work:	8
Without Cascade: Handling a steam flow (level) disturbance.	10
Other practical reasons for cascade.....	12
Note: Valve positioners.....	12
Deciding when to use cascade	13
Some other examples of cascade control:	13
Level control:.....	13
Temperature control via heat exchanger:	13
Crystal Growth Control	13
Operating a Cascade Control Loop.....	14
How to operate a Cascade Loop: Going from Manual to Fully Automatic Operation.....	15
Step 0	16
Step 1	17
Step 2	18
Step 3	19
Step 4	20
Step 5	21
Tuning a Cascade Control System	22
1. Tune the Slave.....	22
2. Tune the Master Loop.....	23
Stop Winding Me Up!	26
Another windup situation	27

An Introduction to Cascade Control: The Theory

A Cascade control system uses two PID controllers. Depending on who you listen to these two controllers are either called the:

- Primary Controller and Secondary Controller
- Outer Controller and Inner Controller
- Master Controller and Slave Controller

Too keep things simple, I will use Master/Slave from now on.

The overall aim of a cascade control loop is to tightly control ONE variable to its setpoint. This is the Master Loop's Process Variable.

The Slave loop has a PV too. But the only reason we care about the Slave PV is because having individual control of this Slave PV gives us better (faster) control of the Master PV.

Sorry slave PV. You are just a pawn in the Master's bid for ever better control of the Master PV. It's an unfair world out there.

Cascade Control: The Definitive Guide

Here's a block diagram of a theoretical Cascade Control System showing all the component parts and signals:

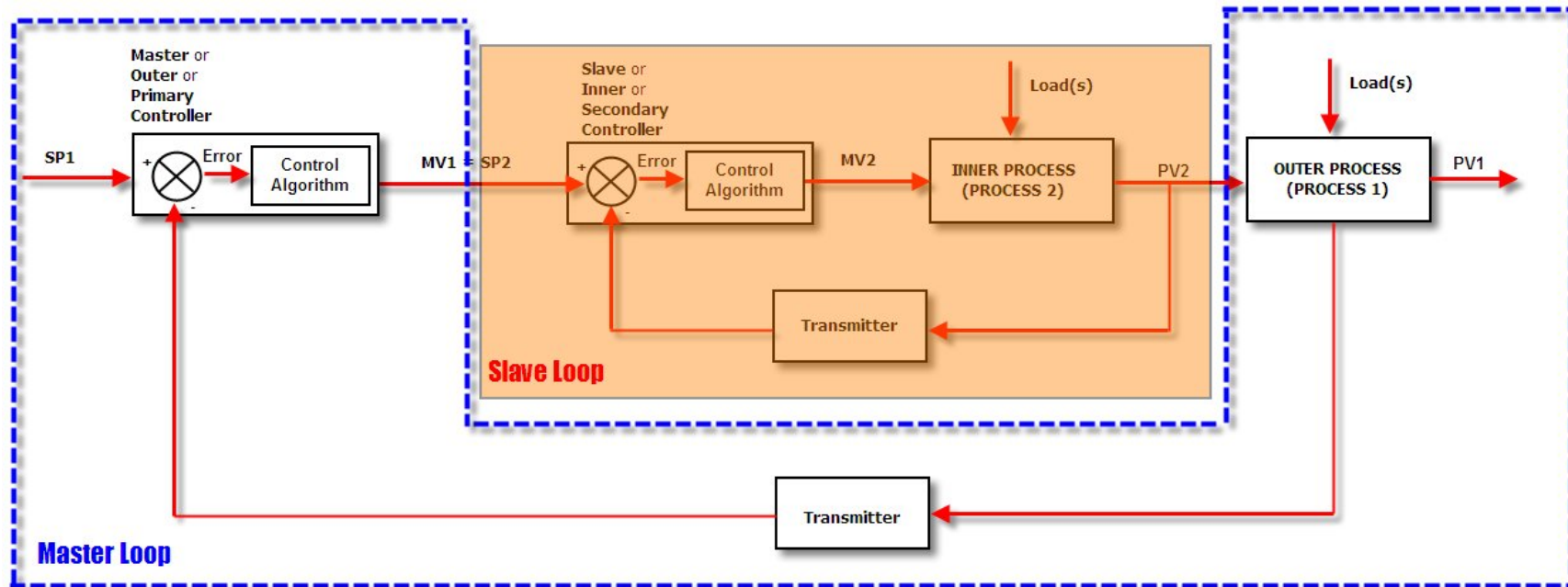


Figure 1 - A Theoretical Cascade Control System

Now, I'm not going to talk you through this diagram because I'm reckoning that any explanation of how a system works is going to be a lot easier to understand if it is based on a real life example. So to make the whole concept of cascade control clear, let's think about a temperature cascade system. In fact let's use the first control loop I ever tuned, back when I was a spotty 19 year old.

Cascade Control in Practice: A Start Up Feed Heater

In the Nuclear plant I used to work in, we needed to heat the feedwater going into the boilers so that it matched the temperature of the metal boiler tubes (to avoid thermal shock, ruptured boiler tubes and nuclear catastrophe).

The feedwater is heated up by putting the pipes, carrying the feedwater, through a pressure vessel. The pressure vessel is full of steam at the top and condensate at the bottom.

Just like in this picture:

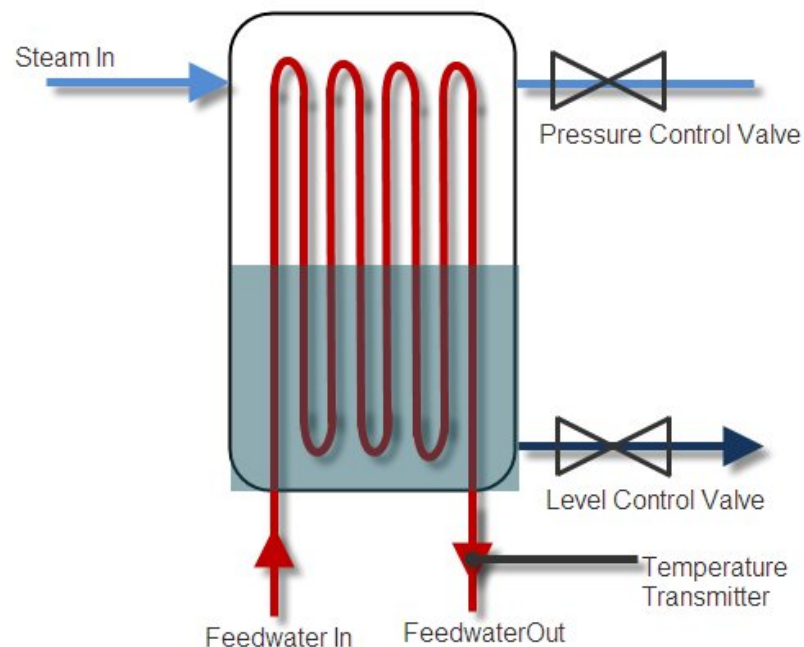


Figure 2 - A Start Up Feed Heater (SU FH)

So steam goes into the pressure vessel. When it hits the feedwater tubes it condenses. This creates a column of water in the bottom of the vessel.

The steam pressure is controlled by a standalone pressure control loop that opens a valve to reduce pressure and closes it to increase pressure. Forget about this pressure control loop. It is not connected into the cascade loop in any way; I just mention it for completeness.

The main aim in life of this “Start Up Feed Heater (SUFH)” is to get the feedwater to a desired temperature setpoint. The cunning way that this is achieved is by varying the level of the condensate with a level control valve that drains the condensate to reduce the level.

To increase the feedwater temperature:

- Open the level control valve
- The condensate level drops
- More feedwater piping is exposed to the steam.
- Feedwater temperature increases.

To reduce the feedwater temperature:

- Close the level control valve
- The condensate level rises.
- Less feedwater piping is exposed to the steam.
- Feedwater temperature decreases.

Cascade Control: The Definitive Guide

The blokes that designed the system implemented this as the cascade control system - shown as a block diagram here:

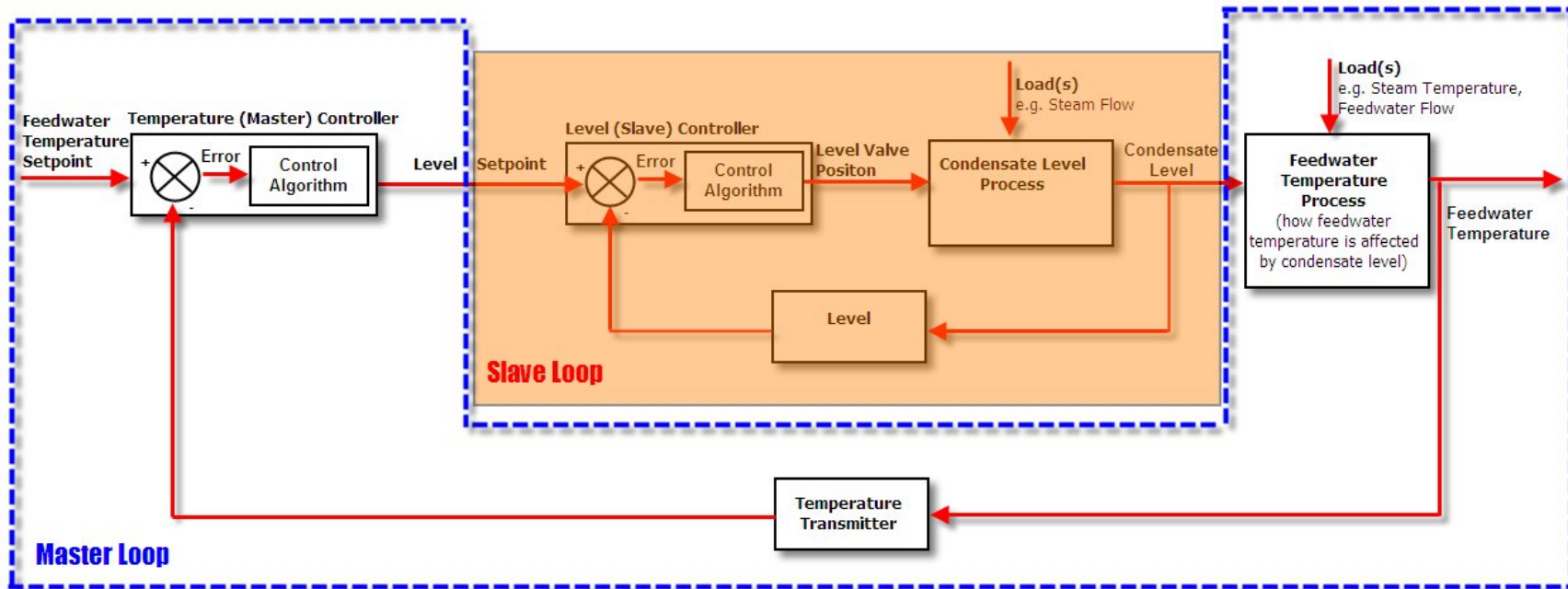


Figure 3 - A real-life example of a Cascade Control System - Feedwater Temperature Control through a Start Up Feed Heater

It looks bloody complicated, but is actually fairly simple when you break it down.

Here's how it works.

The temperature (Master) controller compares the desired feedwater temperature with the actual temperature. If the water needs to heat up, there will be a positive error into the master controller and the reverse-acting master controller will reduce its output.

This will signal to the slave controller that the level needs to drop. The slave controller will open the valve to drop the level; the feedwater temperature will rise.

If the water still isn't hot enough the master controller will ask for an even lower level. This whole cycle repeats until we hit the temperature setpoint. There will eventually be zero temperature error and zero level error and everything will be nice and stable.

OK, but why use a cascade architecture in the first place?

Why use 2 controllers when you could get away with 1?

The diagram over the page shows how we could control the feedwater temperature with just one controller. I've deliberately laid it out so that you can see what has been removed from the cascade diagram. Note that the 2 processes in this system remain unchanged. All that we have done is delete the inner (slave) controller and feedback loop.

So now the output of the temperature controller directly sets the position of the level control valve instead of going through a second controller:

Cascade Control: The Definitive Guide

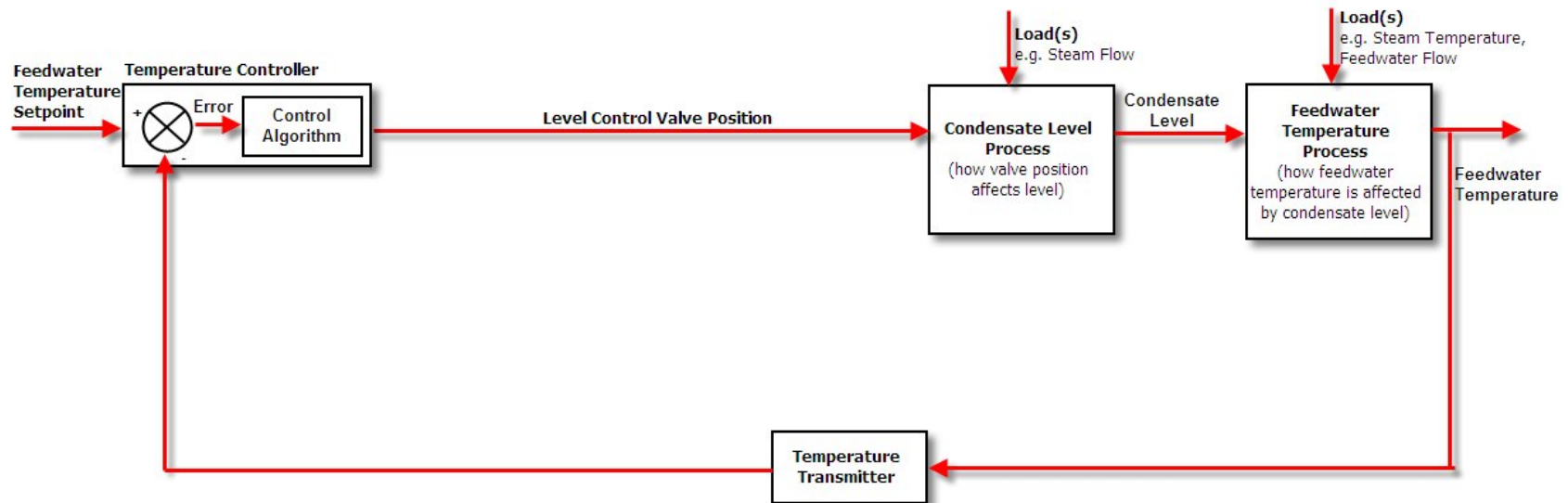


Figure 4 - Controlling feedwater temperature with just one loop

How a single loop strategy would work:

The temperature controller compares the temperature setpoint with the actual feedwater temperature.

If the feedwater is too cold, the temperature controller opens the level control valve, the level drops, and the feedwater gets hotter.

That is exactly what we want to happen. So why bother with the second loop?

Well, just as with a single PID loop we learned that if it wasn't for the disturbances we wouldn't need a feedback loop (Blueprint page 7), with cascade control systems, if it wasn't for disturbances we wouldn't need a second feedback loop.

We implement cascade control so that our system will better handle some disturbances.

Note the emphasis on "some".

Look again at Figure 4. There are 2 distinct sets of disturbances. Those that directly affect the temperature control process. And those that affect the level control process. With cascade control, the disturbances that affect just the temperature process aren't handled any better. This kinda makes sense when everything is drawn as a block diagram.

(That's why I always say "Block Diagrams Rock!" and "P&ID's suck")

You can see from the block diagram (Figure 4) that the length of the feedback loop that senses and corrects for disturbances that directly affect the temperature is no shorter in the single loop version. If anything it is a teeny bit shorter because it doesn't go through a second controller (which saves 1 sample-time worth of dead time).

The block diagram makes it clear however, that the existence of an inner loop provides a much shorter and direct feedback loop to counter any pesky disturbances that will mess with our level control.

To make it crystal clear, let's walk through two scenarios. One with cascade. One without.

Without Cascade: Handling a steam flow (level) disturbance.

The steam flow increases, which means that our condensate level starts filling more quickly. This means that the level in the pressure vessel rises. The effect of the level rise has to filter through the thermal characteristics of the system as a whole before the rise in level manifests itself as a decrease in feedwater temperature. Only when the effect on the temperature is picked up by the temperature transmitter does the controller realise that it has to do something.

The temperature of the feedwater rises. The temperature error increases and the controller opens the valve. The condensate level falls, the temperature increases. The system achieves steady state once again, but not without a noticeable dip in the feedwater temperature.

This diagram shows the path of the disturbance with one only controller:

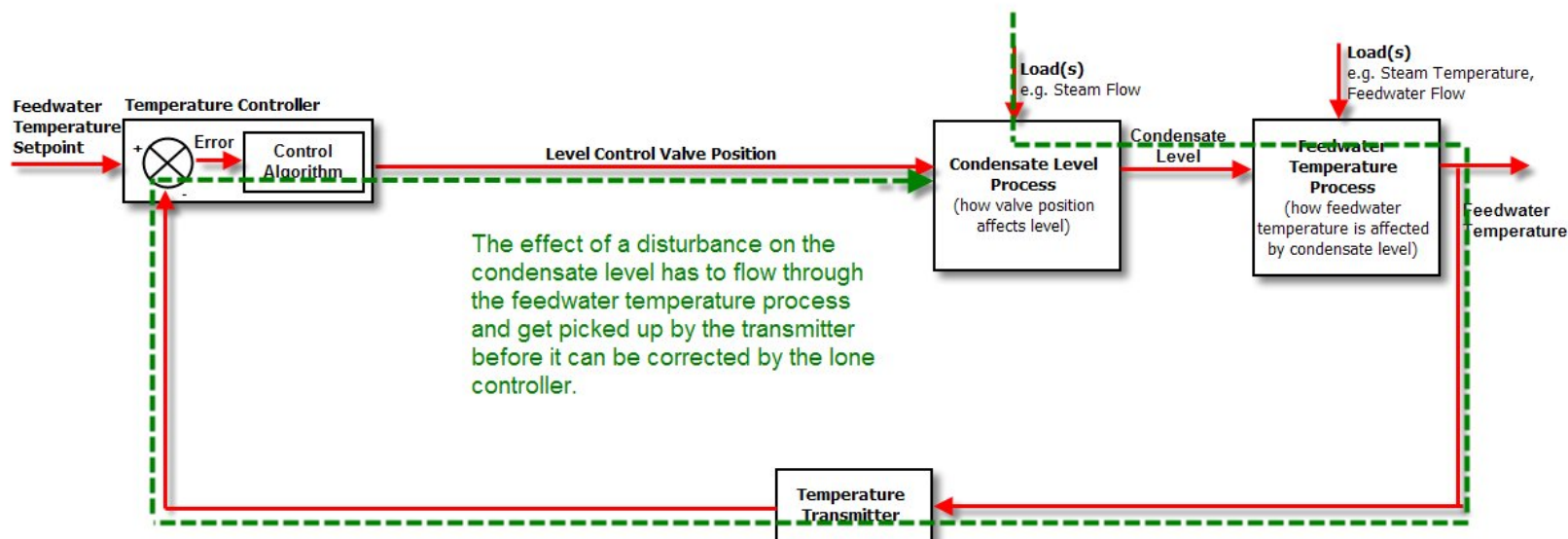


Figure 5 - Handling a level disturbance without cascade

Now let's see how the disturbance is handled with the cascade strategy. I think the picture says it all really:

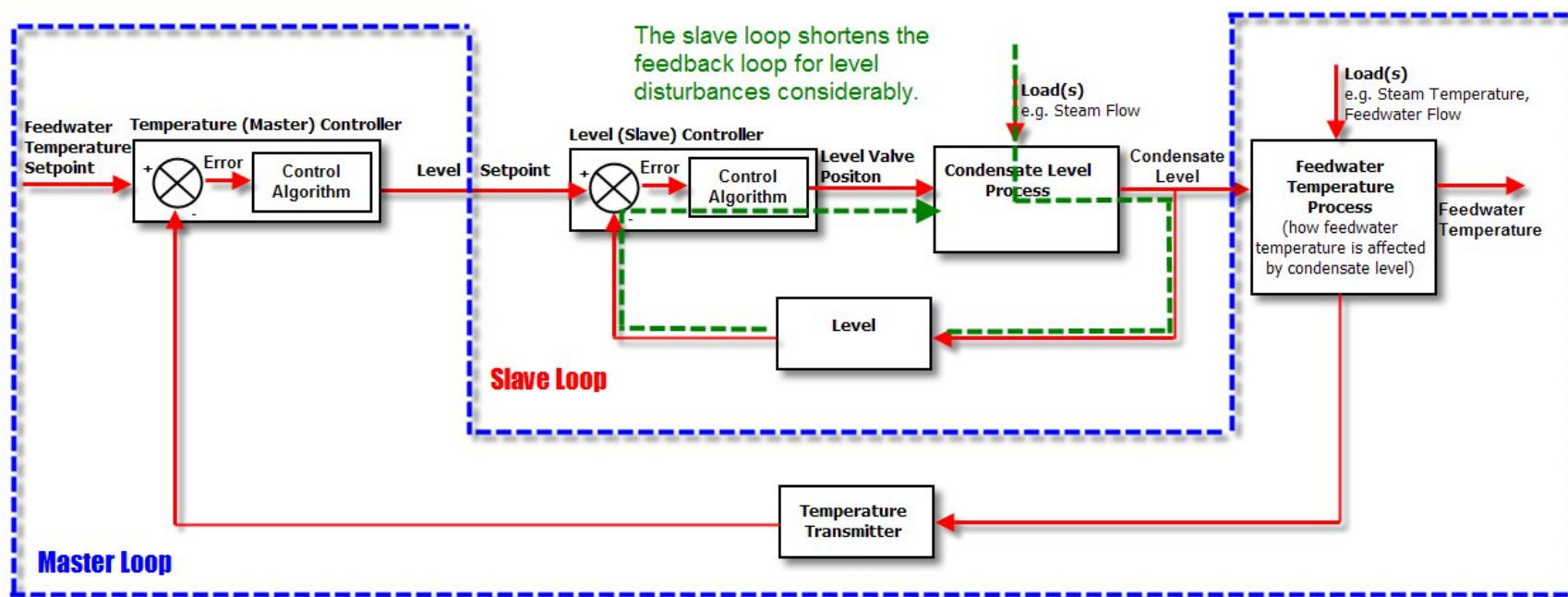


Figure 6 - Handling a level disturbance with Cascade

You can see that the level control disturbance will be handled very quickly here (assuming a well tuned controller!). In fact the thermal lags in the master loop will act as a filter which probably means that any effects of the level disturbance will be almost unnoticeable in the feedwater temperature.

Another point that this diagram makes pretty clear is that cascade is only gonna work well if the inner loop is faster than the outer loop (3 times faster is a good rule of thumb). How do you measure the 'speed' of the loop? Simply measure the open loop time constants (lags) as per the PID Tuning Blueprint (the shorter the lag the faster the loop).

Other practical reasons for cascade

The other nice thing about having a slave loop is that you can put some soft limits on the inner loop. For example if you want upper and lower limits on the level then you can limit the slave controller's setpoint.

Note: Valve positioners

If you have a process whose actuator is a valve driven by a valve positioner, then you may not realise it but this is a cascade loop. A valve positioner is a closed loop position control PID loop. The real bummer with valve positioners is that they tend to be quite slow acting. This means that you are probably violating our rule of thumb that the slave be at least 3 x faster than the master.

This results in a lot of loops with valve positioners having poor performance. If your valve positioner is less than 3x faster than your Master loop then your options are:

- 1) get a faster valve positioner
- 2) ditch the valve positioner and go for open loop valve positioning
- 3) detune your master controller (reduce the P) and have a slower acting loop

Deciding when to use cascade

The example above makes the reasoning for using cascade quite clear:

- 1) Is there a disturbance that affects the variable I am trying to control via an intermediate variable?
- 2) Can I measure the intermediate variable?
- 3) Can I control this variable with a feedback loop that would correct for the disturbance without the disturbance's effect rippling through the main process?

Some other examples of cascade control:

Level control:

Master loop controls level in a tank by controlling a drain valve. Slave loop controls flow of water into the tank with a flow control valve.

Temperature control via heat exchanger:

Master loop controls temperature of the process liquid going through heat exchanger. Slave loop controls the flow of steam through the heat exchanger.

And here's a wacky one that I did a few years ago:

Crystal Growth Control

I once developed a cascade control system to grow wine bottle sized crystals by hanging a rotating 'seed' crystal in a molten cup of secret chemicals.

The Master loop controls the rate of change of crystal diameter. The Slave loop controls speed at which you pull the crystal out of the liquid in mm/day.

Operating a Cascade Control Loop

A single PID loop has 2 modes of operation: Manual and Auto. You just need one Man/Auto switch.

In a cascade loop it gets a little more complicated.

The master loop still has just the 2 modes of operation via a single Man/Auto switch.

The slave loop, too, has a MAN/AUTO switch. And that is fine for manually changing the slave controller output.

But what if you want to run the slave in Auto, but manually change its setpoint? To give the operators this control option, the slave controller has a LOCAL/CASCADE switch for setpoint source selection. Local means the setpoint can be set independently of the Master Controller's Output

We'll see how these three switches are used to operate the cascade control system in both manual and auto next.

How to operate a Cascade Loop: Going from Manual to Fully Automatic Operation

There are many ways to skin this cat. This is my preferred sequence of operation for putting a cascade loop into Automatic without upsetting the PV.

Note that a lot of modern software based control systems will automate many or all of these steps. Even if that is the case with your system, it is always nice to know what the software is doing and why.

A note about the diagrams below.

It can be quite hard to visualise what is going on in a Cascade loop if you are just reading paragraphs of text. If you are like me your brain soon starts to get a bit fuzzy switching between the concept of slaves and masters going from manual to auto to local. And with 2 control loops there are so many signals everywhere it can be hard to keep track of which signals are internal, external, in your control and out of your control.

So I have drawn a diagram for each step of the sequence below. The blue box encapsulates the cascade control loop. All signals external to the cascade loop that bear on its operation can clearly be seen.

The external signals that you can directly manipulate are shown in green.

The external measurement signals that change due to the controller's actions are in yellow.

The disturbances that you have no control over are shown as blue arrows

Cascade Control: The Definitive Guide

You'll see that I've also shown the 3 switches that you have control over, Master AUTO/MAN. Slave AUTO/MAN and Slave LOCAL/CASCADE.

Finally the fat red arrows show you what has changed in each step.

Step 0

Start your cascade loop in Fully Manual mode: Master: Manual, Slave: Manual, Slave: Local

This gives you manual control of everything, and keeps everything as static as possible.

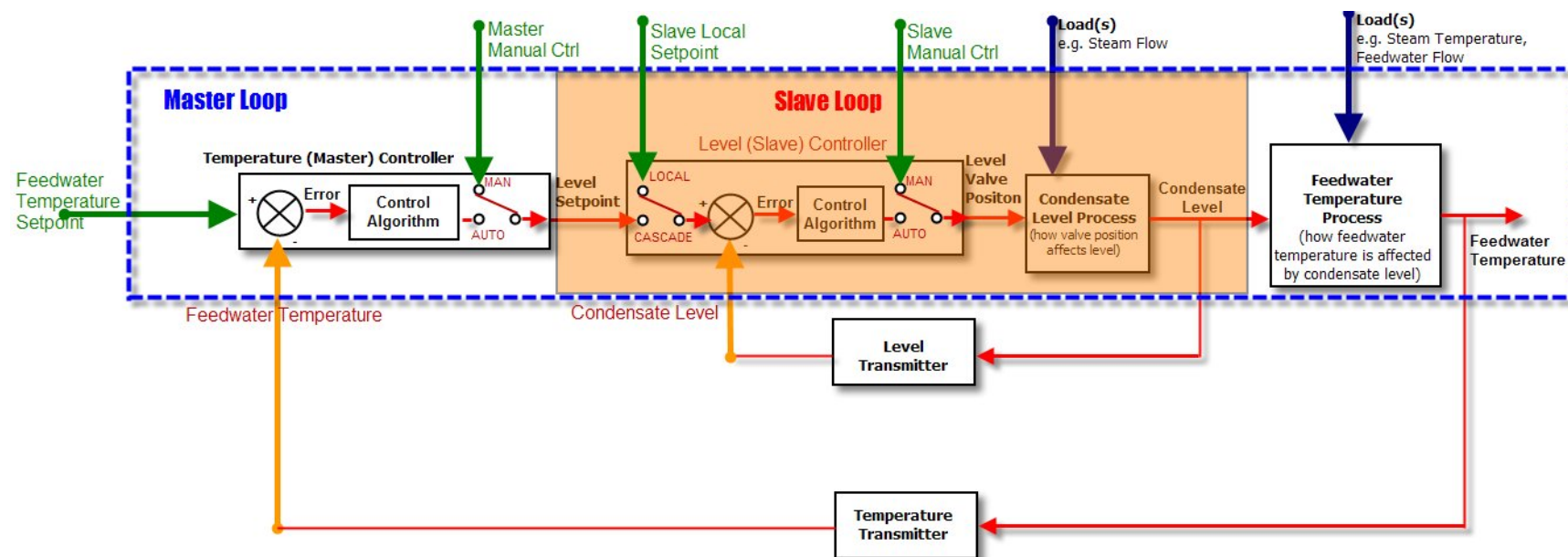


Figure 7 - Cascade Loop In 'Fully Manual' Mode

Cascade Control: The Definitive Guide

Step 1

Switch the Slave Controller from Local to cascade.

Note that your Master Manual trim now determines the Slave SP.

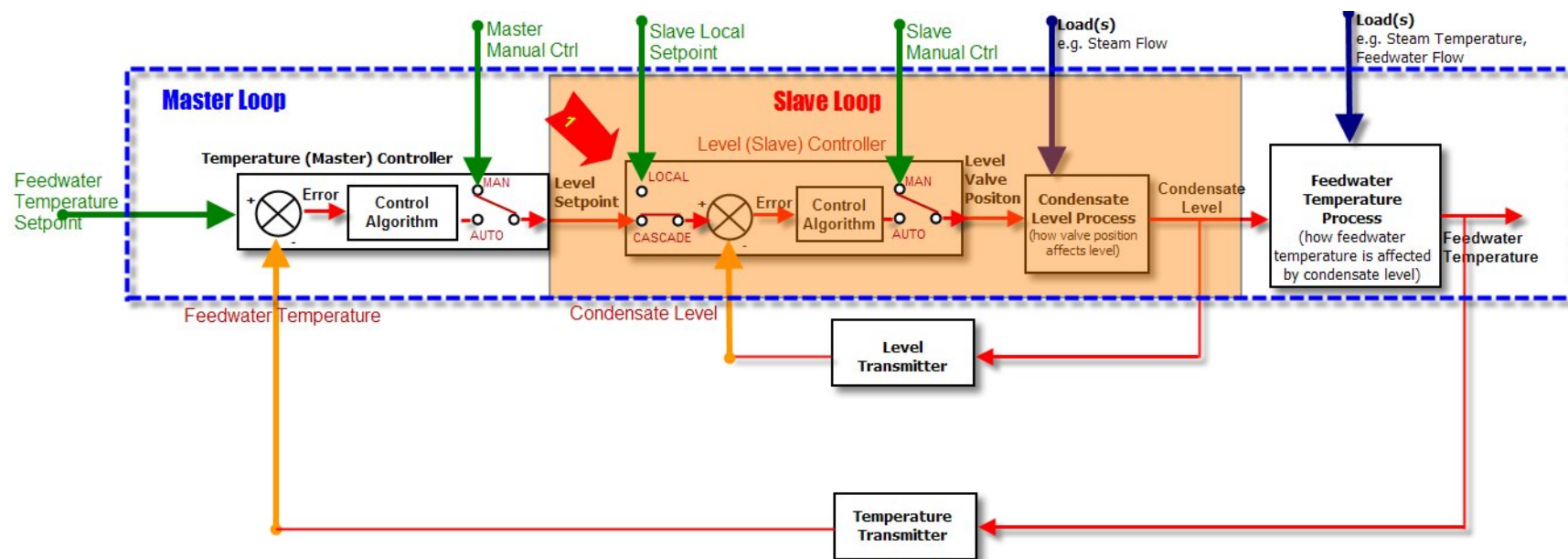


Figure 8 - Step the Slave Controller from Local To Cascade

Cascade Control: The Definitive Guide

Step 2

Use the Master's Manual trim control to set the Slave's Setpoint until it matches the Slave's PV. In this example you want the Level Setpoint to match the level measurement.

Note: This is the standard way to achieve bumpless changeover from Man to Auto. It is bumpless because if the SP = PV then the Error is zero and the controller output won't want to move anywhere. Many digital control systems have a 'setpoint tracking' feature. This simply continuously updates the manual setpoint to match the PV, so that they are always the same value.

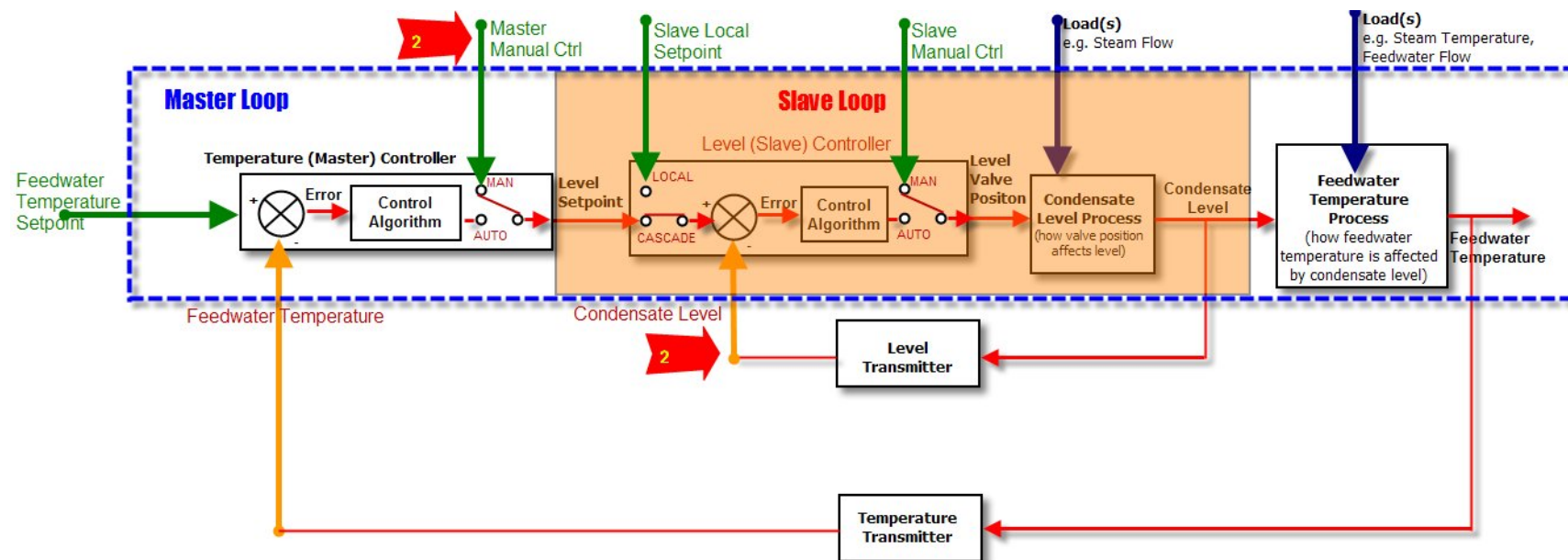


Figure 9 - Match the Slave Setpoint to the Slave PV

Step 3

When the Slave's setpoint equals its measured PV, switch the Slave controller to AUTO. In this example you will get a bumpless transition to automatic level control with the level setpoint being determined by the value of the Master Controller's manual trim.

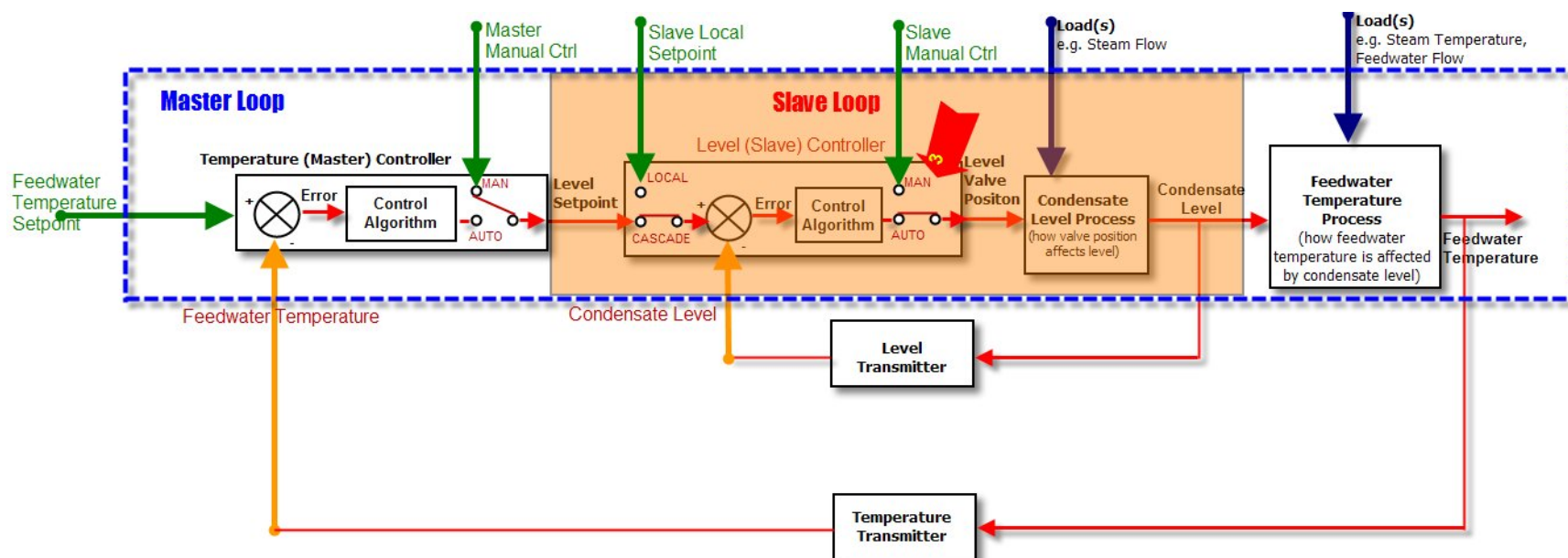


Figure 10 - Switch the slave to AUTO

Cascade Control: The Definitive Guide

Step 4

Adjust the Master Controller's setpoint to match the Master Loop's measured PV. In this example, adjust the temperature SP to match the measured feedwater temperature.

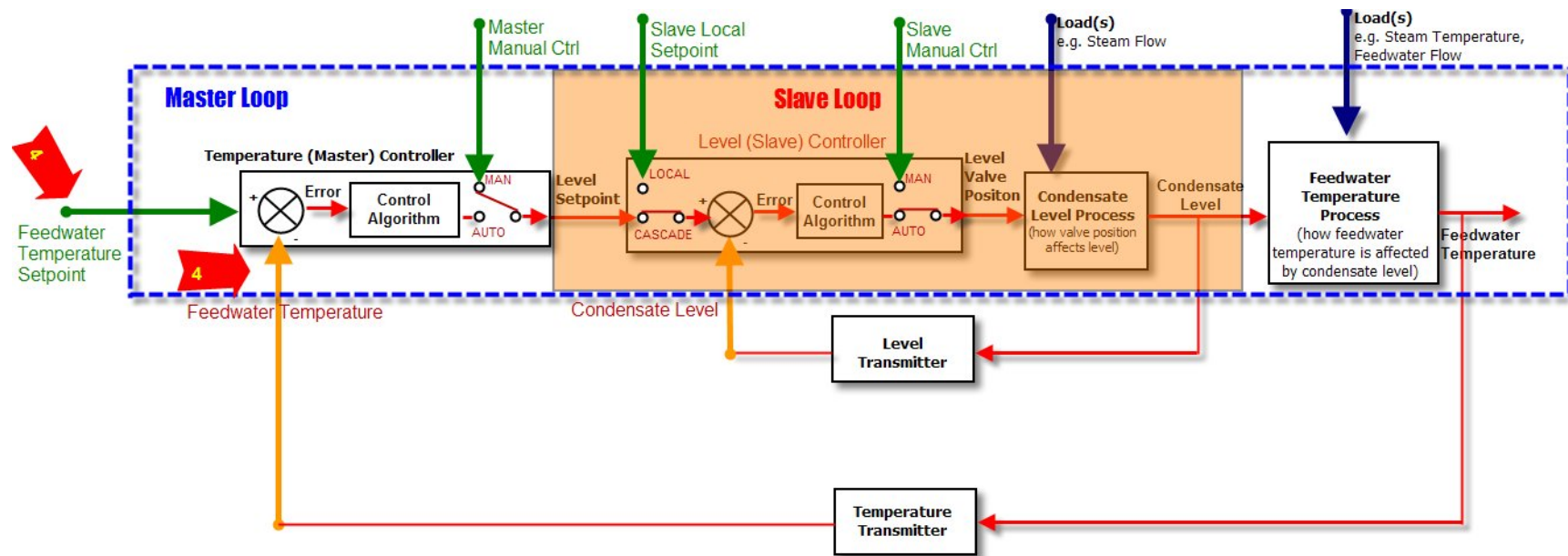


Figure 11 - Match the Master SP to the Master PV

Step 5

When the Master SP equals the Master PV, switch the master to AUTO. The cascade control system is now in fully automatic operation.

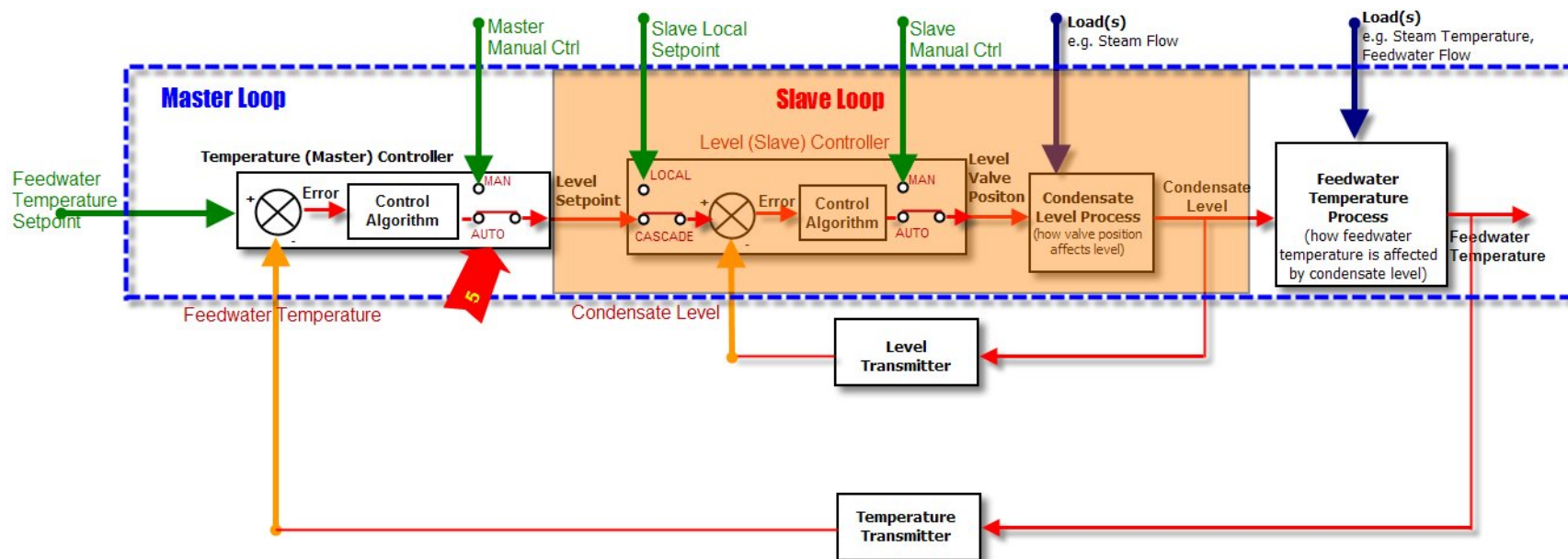


Figure 12 - Switch the master to Auto

To put a Cascade control system into Manual from fully Auto, just reverse this sequence.

Tuning a Cascade Control System

1. Tune the Slave: Put the system into fully manual mode (Master: MAN, Slave: MAN, Slave: LOCAL) as per Figure 7. Then tune the Slave loop.

Tune the Slave exactly as per the PID Tuning Blueprint. Don't miss out any steps they are all relevant for the slave loop.

When you get to Step 8 of the Blueprint, perform the open loop step testing just as if the slave loop was a single loop. The diagram below shows the slave loop without any of the other loop confusing the picture:

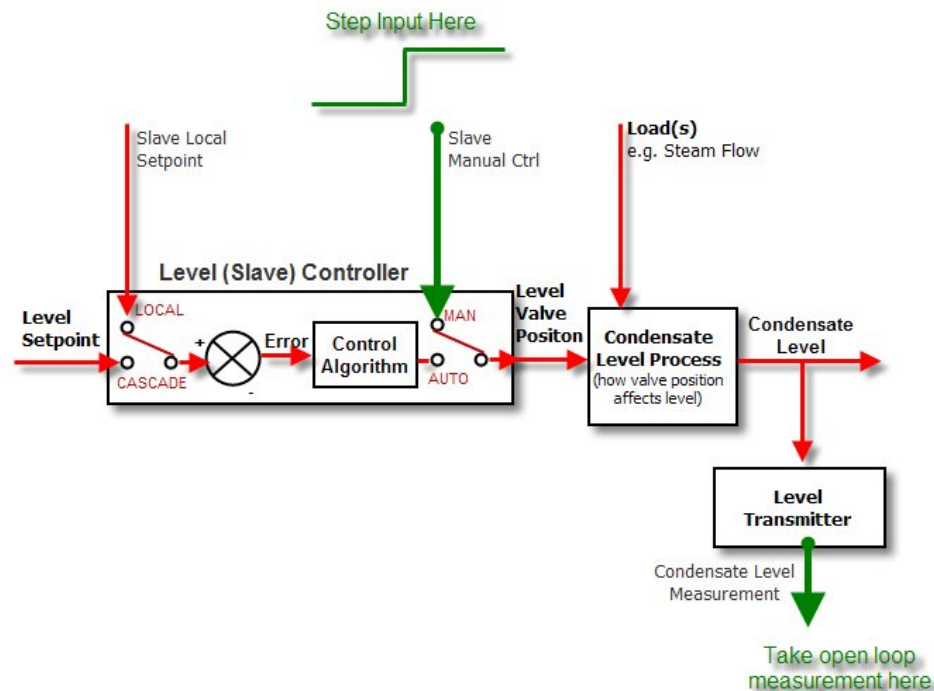


Figure 13 - Tuning the Slave

When you are happy that the slave loop has good closed loop performance, it is time to move on to the Master Loop's tuning.

2. Tune the Master Loop

When we are tuning the Master Controller, we need treat the Slave loop as just another part of the process, and make believe that we are just tuning a single loop in isolation. In our example, we would just follow the blueprint as if we were tuning a single temperature control loop.

Important: Ensure that the Slave loop is in AUTO and CASCADE modes for the whole tuning process. For all open loop step testing, the Master loop will need to be in MANUAL.

Here's the setup for tuning the master:

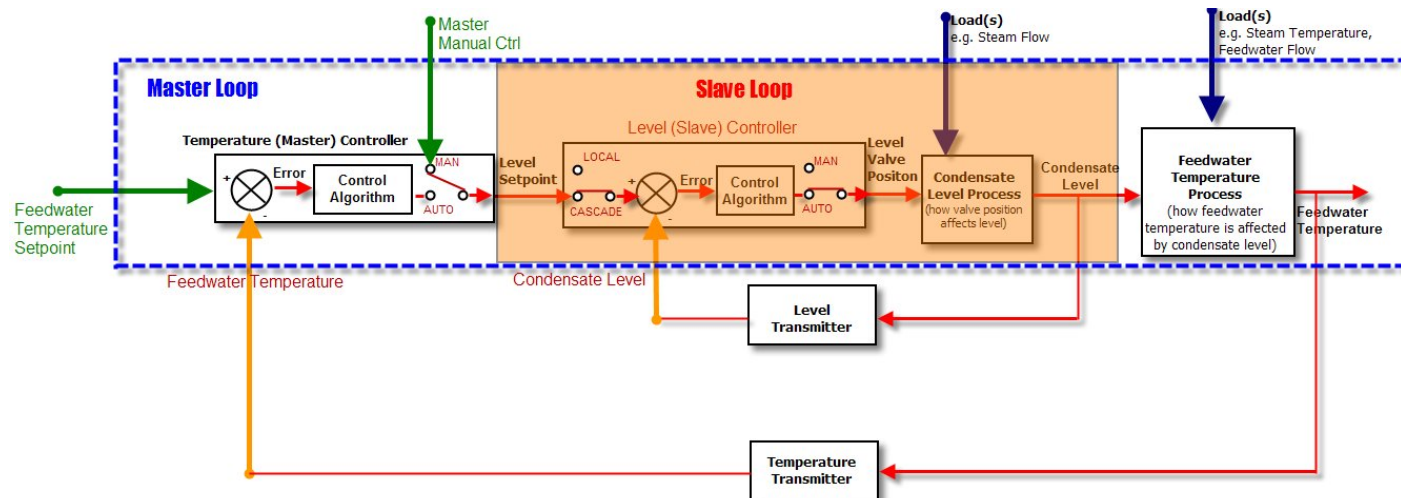


Figure 14 - Tuning the Master

To make it even simpler, the entire Slave loop should just be treated as a black box:

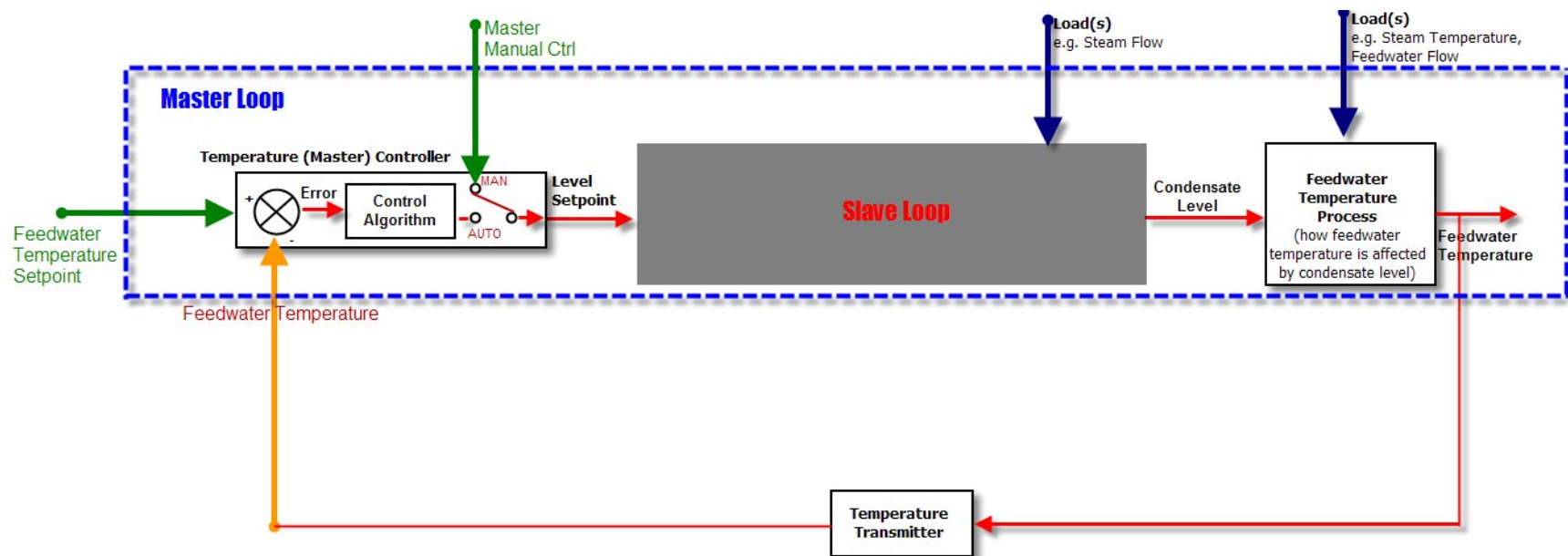


Figure 15 - Simplifying the system

Nah, still too many boxes in that diagram! Let's simplify it even further.

When tuning the Master, to all extents and purposes the level control loop is just a part of the Feedwater temperature process. And the feedwater temperature process is just a system which takes the Master (Temperature) controller's MV as its input and converts that to a feedwater temperature as its output.

When you draw the master loop like this, it becomes straightforward to tune it using the Blueprint's process:

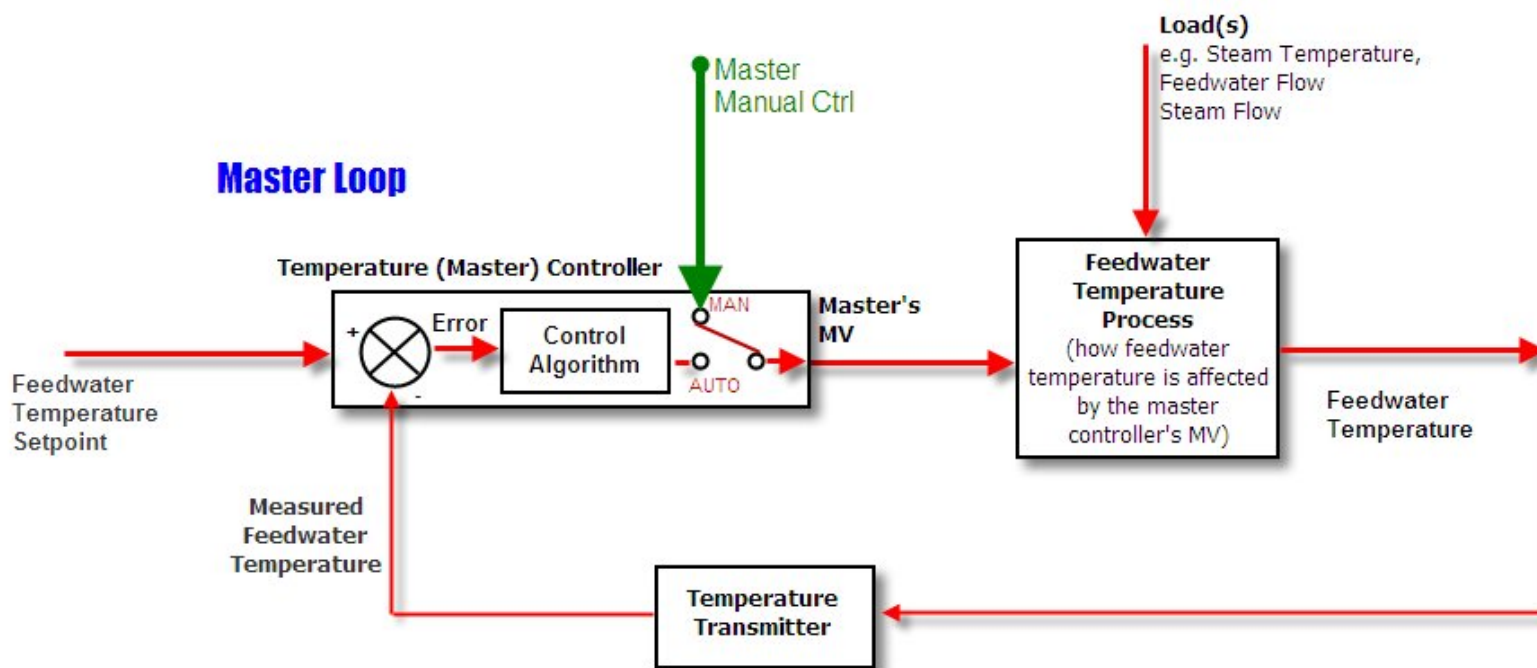


Figure 16 - Think of tuning the Master just like any other single loop

Stop Winding Me Up!

A cascade loop winds up when the Master controller continues to ramp its output even though the slave can go no further. It's kinda like flogging a dead horse. If the slave's actuator is fully open, there is no point asking it to open some more.

The problem with windup in a cascade loop is that if your master keeps increasing its output even though the slave can go no further, when the time comes to go in the other direction, there will be absolutely no effect on the process until the master has 'wound down' to the point at which the slave starts moving again.

The fix for this is dead simple in a digital system. First you ensure that a Slave MV of 0% coincides with the lower limit of the actuator, and 100% drives the actuator to its upper limit. Then you simply add logic that says:

"If the Slave MV hits 0% don't let the Master MV get any lower and if the Slave MV hits 100% don't let the Master MV get any higher"

TIP: One good practical way to achieve this is to disable the Master's Integral action when the slave actuator hits an upper or lower limit. (If you are scratching your head at why this works, have another read of the chapter on Integral Action in the 'Idiot's Guide to the PID Algorithm').

Another windup situation

There is one more situation where cascade loops can wind up. Remember that windup is defined as the Master continuing to change its output even though the Slave can't move any further.

Well, if the Slave is switched to Manual and the Master stays in Auto, we can get this situation too. The simple solution is an interlock that forces the Master to manual whenever the Slave goes into Manual.