**Thursday, December 03, 2015**

Krp h Qh{ v Jruyp Fdhhu Qrwhv Errnv P dqydo IDP #Eorj Korvwdu|

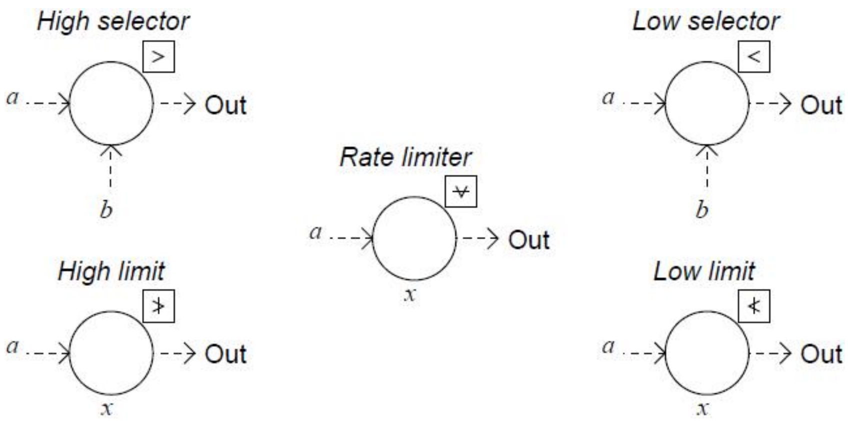
IDP #Vhdufk

Search

# Olp lw/#Vhohfwru/#dqg#R yhuulgh#Frqwurov#

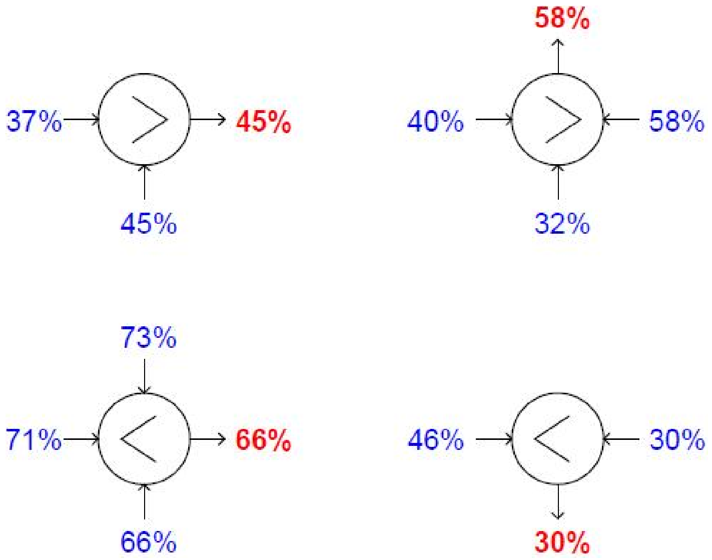
Another category of control strategies involves the use of sig relays or function blocks with the ability to switch between different signal values, or re-direct signals to new pathways. Such functions are useful when we need a control system to choose between multiple signals of differing value in order to make the best control decisions.

The "building blocks" of such control strategies are spe relays (or function blocks in a digital control system) sh here:



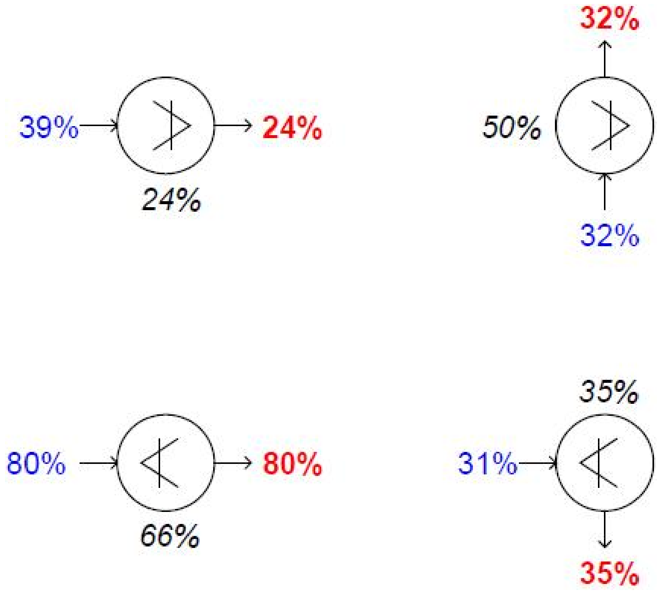
*High-select* functions output whichever input signal has the *greatest* value. *Low-select* functions do just opposite: output whichever input signal has the *least* value. "Greater-than" and "Less than" symbols m these two selector functions, respectively, and each type may be equipped to receive more than two in signals.

Sometimes you will see these relays represented in P&IDs simply by an inequality sign in the middle of large bubble, rather than off to the side in a square. You should bear in mind that the location of the in lines has no relationship at all to the direction of the inequality symbol – e.g., it is not as though a high-se relay looks for the input on the left side to be greater than the input on the right. Note the examples sh below, complete with sample signal values:



*High-limit* and *low-limit* functions are similar to high- and low-select functions, but they only receive input each, and the limit value is a parameter programmed into the function rather than received f another source. The purpose of these functions is to place a set limit on how high or how low a signal valu allowed to go before being passed on to another portion of the control system. If the signal value lies wi the limit imposed by the function, the input signal value is simply passed on to the output with modification.

Like the select functions, limit functions may appear in diagrams with nothing more than the limit syn inside the bubble, rather than being drawn in a box off to the side:

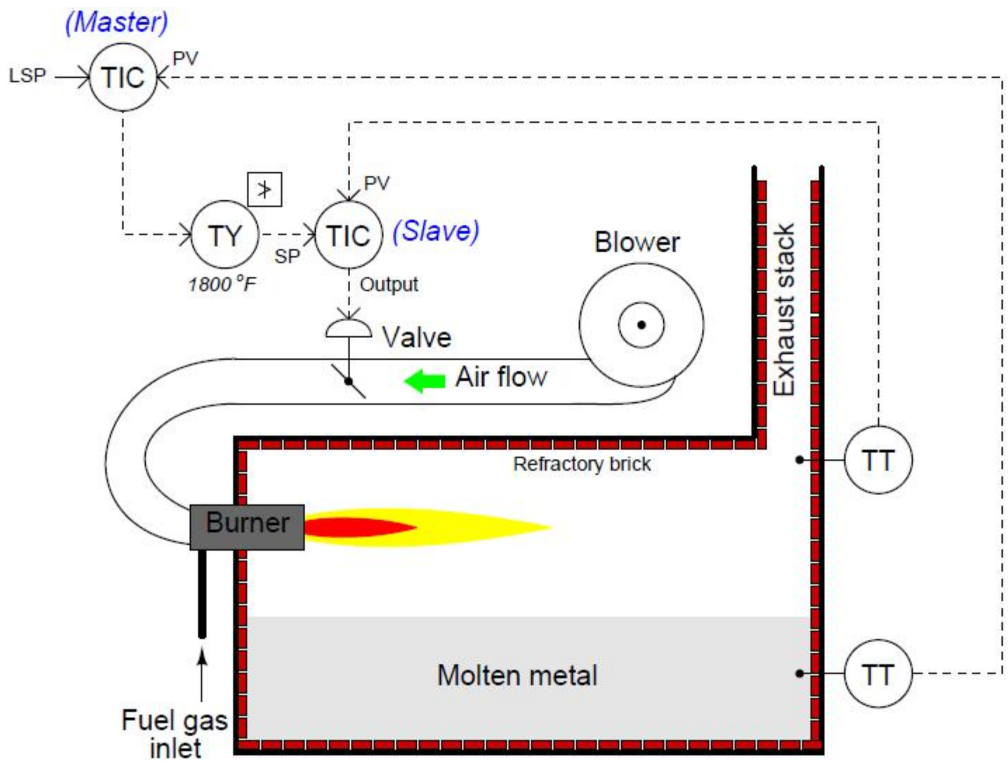


*Rate limit* functions place a maximum rate-of-change limit on the input signal, such that the output sig will follow the input signal precisely until and unless the input signal's rate-of-change over time ( $dx$ exceeds the pre-configured limit value. In that case, the relay still produces a ramping output value, but rate of that ramp remains fixed at the limit $dxdt$value no matter how fast the input keeps changing. After output value "catches up" with the input value, the function once again will output a value preci matching the input unless the input begins to rise or fall at too fast a rate again.

## Limit controls

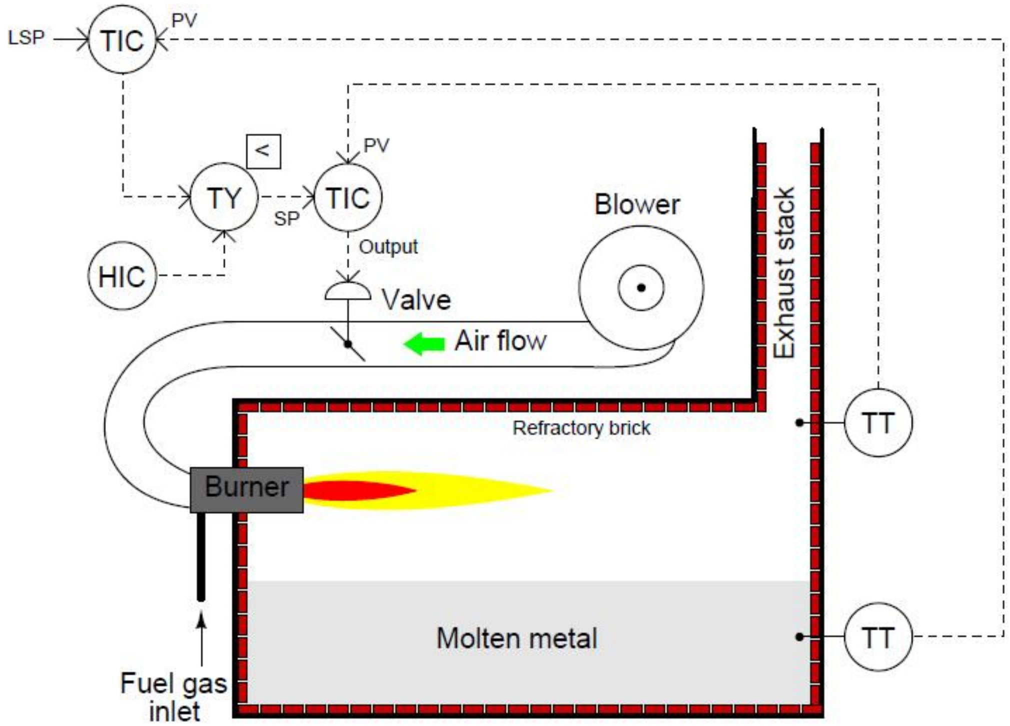
A common application for select and limit functions is in *cascade* control strategies, where the output of controller becomes the setpoint for another. It is entirely possible for the primary (master) controller to for a setpoint that is unreasonable or unsafe for the secondary (slave) to attain. If this possibility exists, wise to place a limit function between the two controllers to limit the cascaded setpoint signal.

In the following example, a cascade control system regulates the temperature of molten metal in a furn the output of the master (metal temperature) controller becoming the setpoint of the slave (air temperatu controller. A high limit function limits the maximum value this cascaded setpoint can attain, ther protecting the refractory brick of the furnace from being exposed to excessive air temperatures:



It should be noted that although the different functions are drawn as separate bubbles in the P&ID, it is possible for multiple functions to exist within one physical control device. In this example, it is possible to find a controller able to perform the functions of both PID control blocks (master and slave) and the high limit function as well. It is also possible to use a distributed technology such as FOUNDATION Fieldbus t place all control functions inside field instruments, so only three field instruments exist in the loop: the air temperature transmitter, the metal temperature transmitter, and the control valve (with a Fieldbus positioner).

This same control strategy could have been implemented using a low select function block rather than a hi limit:

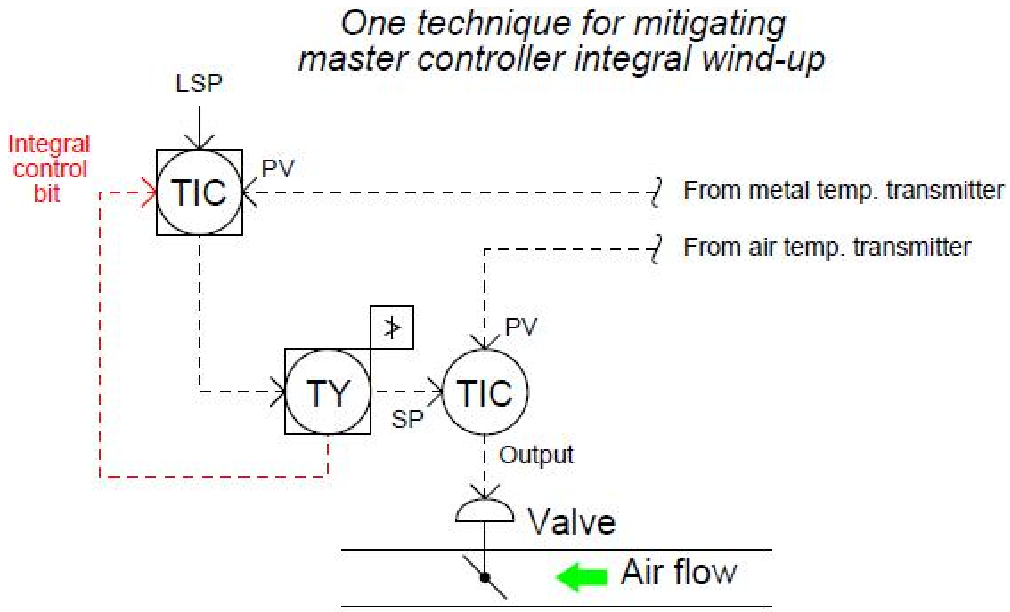


Here, the low-select function selects whichever signal value is lesser: the setpoint value sent by the ma temperature controller, or the maximum air temperature limit value sent by the hand indicating contro (HIC – sometimes referred to as a *manual loading station*).

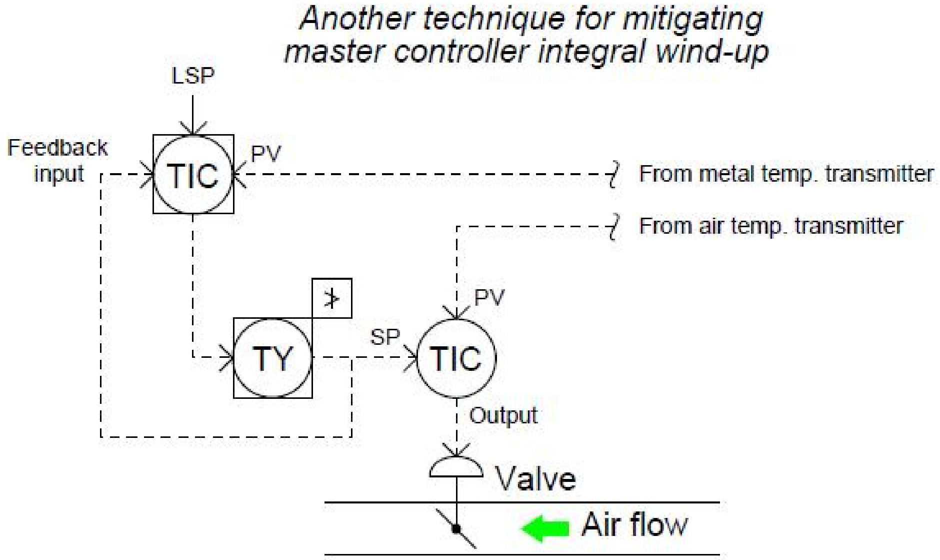
An advantage of this latter approach over the former might be ease of limit value changes. With a configured limit value residing in a high-limit function, it might be that only qualified maintenance pe have access to changing that value. If the decision of the operations department is to have the temperature limit value easily adjusted by anyone, the latter control strategy's use of a manual loa station would be better suited[1].

Another detail to note in this system is the possibility of *integral windup* in the master controller in the ev that the high setpoint limit takes effect. Once the high-limit (or low-select) function secures the s controller's remote setpoint at a fixed value, the master controller's output is no longer controlling anyth it has become decoupled from the process. If, when in this state of affairs, the metal temperature is still be setpoint, the master controller's integral action will "wind up" the output value over time with absolutely effect, since the slave controller is no longer following its output signal. If and when the metal tempera reaches setpoint, the master controller's output will likely be saturated at 100% due to the time it sp winding up. This will cause the metal temperature to overshoot setpoint, as a positive error will be requ for the master controller's integral action to wind back down from saturation.

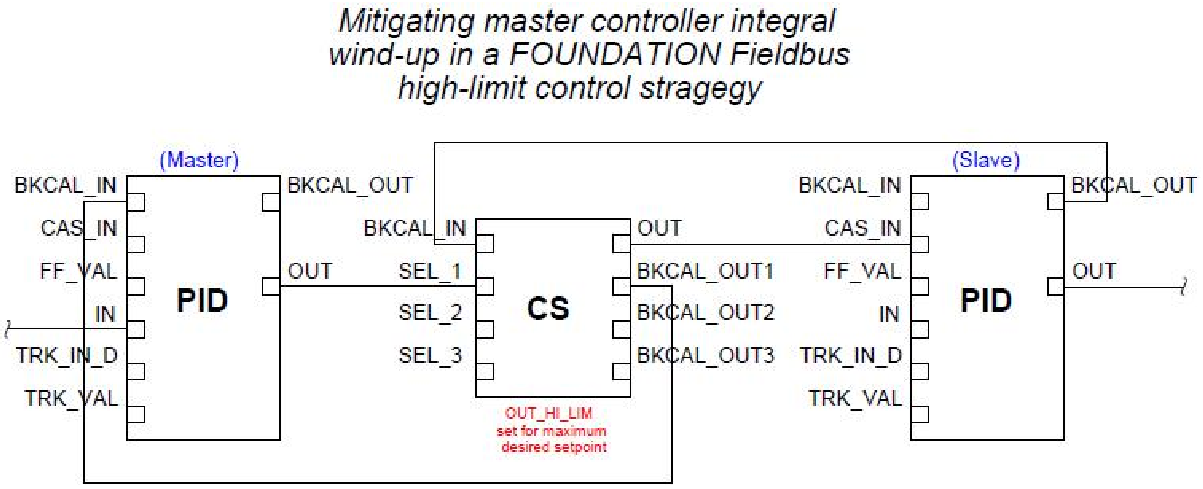
A relatively easy solution to this problem is to configure the master controller to stop integral action w the high limit relay engages. This is easiest to do if the master PID and high limit functions both reside in same physical controller. Many digital limit function blocks generate a bit representing the state of that b (whether it is passing the input signal to the output or limiting the signal at the pre-configured value), some PID function blocks have a boolean input used to disable integral action. If this is the case with function blocks comprising the high-limit control strategy, it may be implemented like this:


One technique for mitigating
master controller integral wind-up
LSP
Integral
control
bit
TIC
PV
From metal temp. transmitter
From air temp. transmitter
PV
TY
TIC
SP
Output
Valve
Air flow

Another method used to prevent integral windup is to make use of the *feedback* input available on some I function blocks. This is an input used to calculate the integral term of the PID equation. In the day pneumatic PID controllers, this option used to be called *external reset*. Normally connected to the outpu the PID block, if connected to the output of the high-limit function it will let the controller know whethe not any attempt to wind up the output is having an effect. If the output has been de-selected by the h limit block, integral windup will cease:


Another technique for mitigating
master controller integral wind-up
LSP
Feedback
input
TIC
PV
From metal temp. transmitter
From air temp. transmitter
PV
TY
SP
TIC
Output
Valve
Air flow

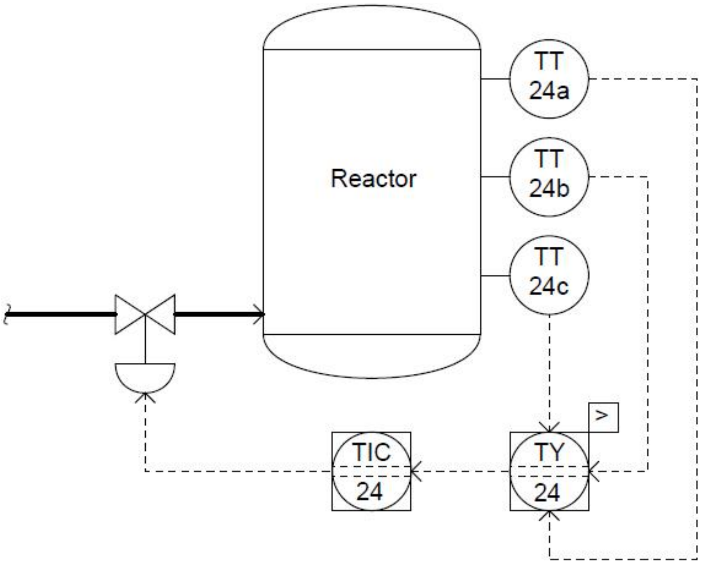
Limit control strategies implemented in FOUNDATION Fieldbus instruments use the same principle, exc that the concept of a “feedback” signal sending information backwards up the function block chain is aggressively-applied design philosophy throughout the FOUNDATION Fieldbus standard. Nearly ev function block in the Fieldbus suite provides a “back calculation” output, and nearly every function b accepts a “back calculation” input from a downstream block. The “Control Selector” (CS) function b specified in the FOUNDATION Fieldbus standard provides the limiting function we need between the ma and slave controllers. The BKCAL OUT signal of this selector block connects to the master controll BKCAL IN input, making the master controller aware of its selection status. If ever the Control Sele function block de-selects the master controller’s output, the controller will immediately know to halt inte action:



## Selector controls

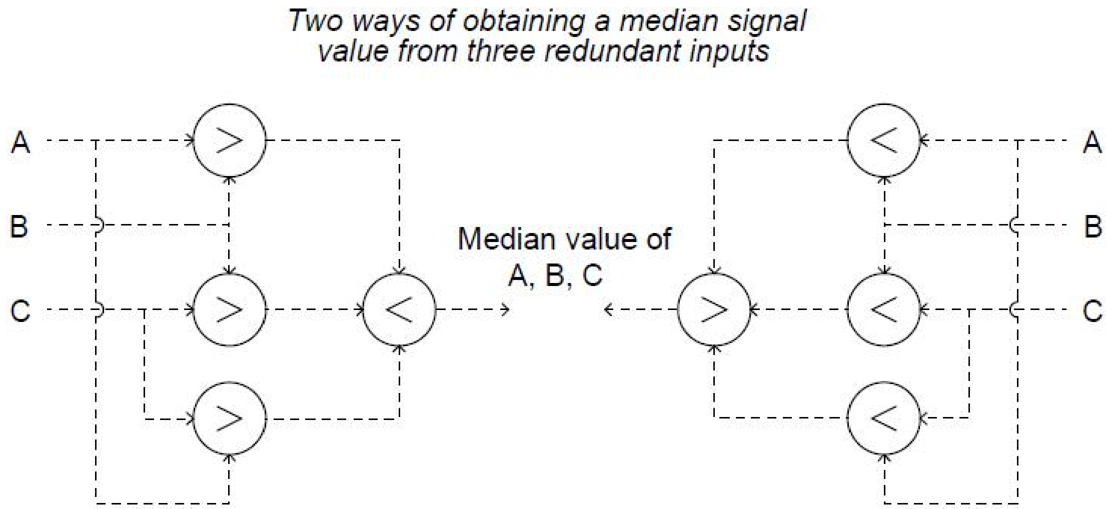
In the broadest sense, a "selector" control strategy is one where one signal gets selected from multiple sig in a system to perform a measurement control function. In the context of this book and this chapter, I use the term "selector" to categorize the automatic selection of a measurement or setpoint signal. Selectio a controller output signal will be explored in the next subsection.

Perhaps one of the simplest examples of a selector control strategy is where we must select a process vari signal from multiple transmitters. For example, consider this chemical reactor, where the control system n throttle the flow of coolant to keep the *hottest* measured temperature at setpoint, since the reaction hap to be exothermic (heat-releasing)[2]:



The high-select relay (TY-24) sends only the highest temperature signal from the three transmitters to controller. The other two temperature transmitter signals are simply ignored. Another use of selector re (or function blocks) is for the determination of a *median* process measurement. This sort of strategy is o used on triple-redundant measurement systems, where three transmitters are installed to measure the e same process variable, providing a valid measurement even in the event of transmitter failure.

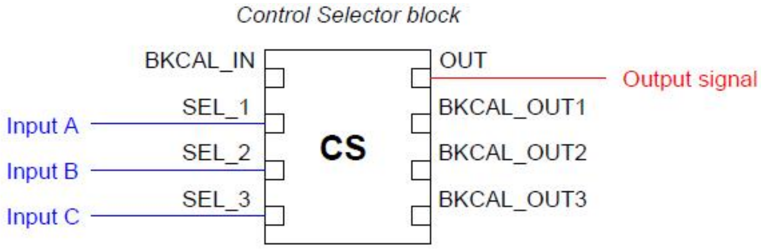
The median select function may be implemented one of two ways using high- and low-select function block

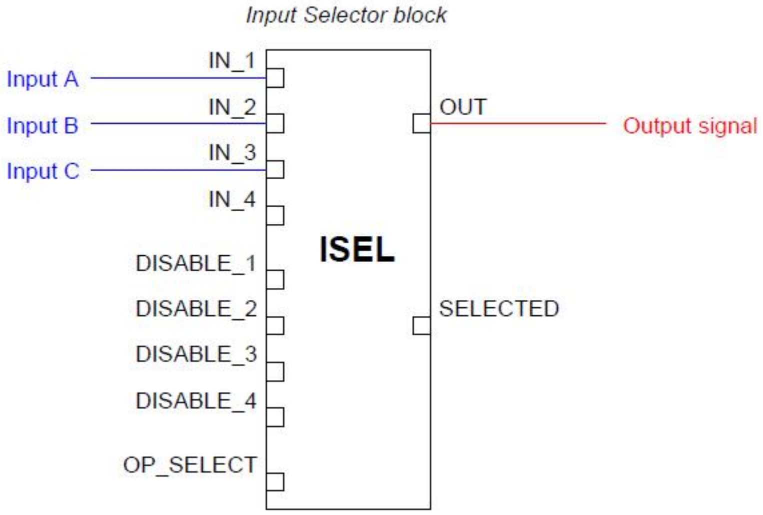

Two ways of obtaining a median signal value from three redundant inputs

The left-hand selector strategy selects the highest value from each pair of signals (A and B, B and C, A C), then selects the lowest value of those three primary selections. The right-hand strategy is exactly oppc – first selecting the lowest value from each input pair, then selecting the highest of those values – but it accomplishes the same function. Either strategy outputs the *middle* value of the three input signals[3].

Although either of these methods of obtaining a median measurement requires four signal selector functi it is quite common to find function blocks available in control systems ready to perform the median se function all in a single block. The median-select function is so common to redundant sensor control syst that many control system manufacturers provide it as a standard function unto itself.

This is certainly true in the FOUNDATION Fieldbus standard, where two standardized function blocks capable of this function, the CS (Control Selector) and the ISEL (Input Selector) blocks:


Control Selector block


Input Selector block

Of these two Fieldbus function blocks, the latter (ISEL) is expressly designed for selecting transmitter sigr whereas the former (CS) is best suited for selecting controller outputs with its "back calculation" facili designed to modify the response of all de-selected controllers. Using the terminology of this book section, ISEL function block is best suited for *selector* strategies, while the CS function block is ideal for *over* strategies (discussed in the next section). If receiving three "good" inputs, the ISEL function block will out the middle (median) value of the three. If one of the inputs carries a "bad" status[4], the ISEL block outp the averaged value of the remaining two (good) inputs. Note how this function block also possesses indivic "disable" inputs, giving external boolean (on/off) signals the ability to disable any one of the transmi inputs to this block. Thus, the ISEL function block may be configured to de-select a particular transmi input based on some programmed condition other than internal diagnostics.

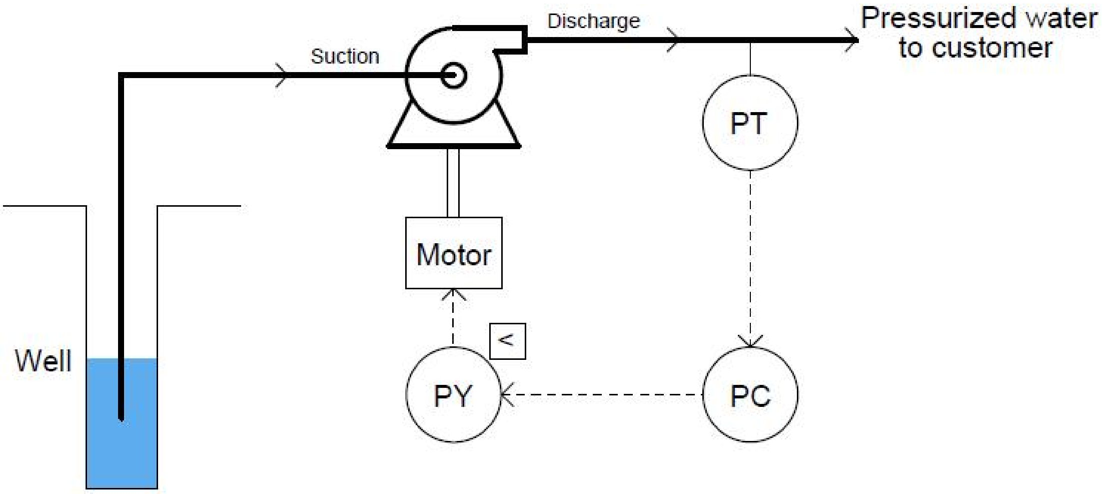
If receiving four "good" inputs, the ISEL function block normally outputs the average value of the two mic (median) signal values. If one of the four inputs becomes "bad" is disabled, the block behaves as a nor three-input median select.

A general design principle for redundant transmitters is that you *never* install exactly two transmitter: measure the same process variable. Instead, you should install three (minimum). The problem with hav two transmitters is a lack of information for "voting" if the two transmitters happen to disagree. In a th transmitter system, the function blocks may select the median signal value, or average the "best 2 out of If there are just two transmitters installed, and they do not substantially agree with one another, i anyone's guess which one should be trusted[5].

## Override controls

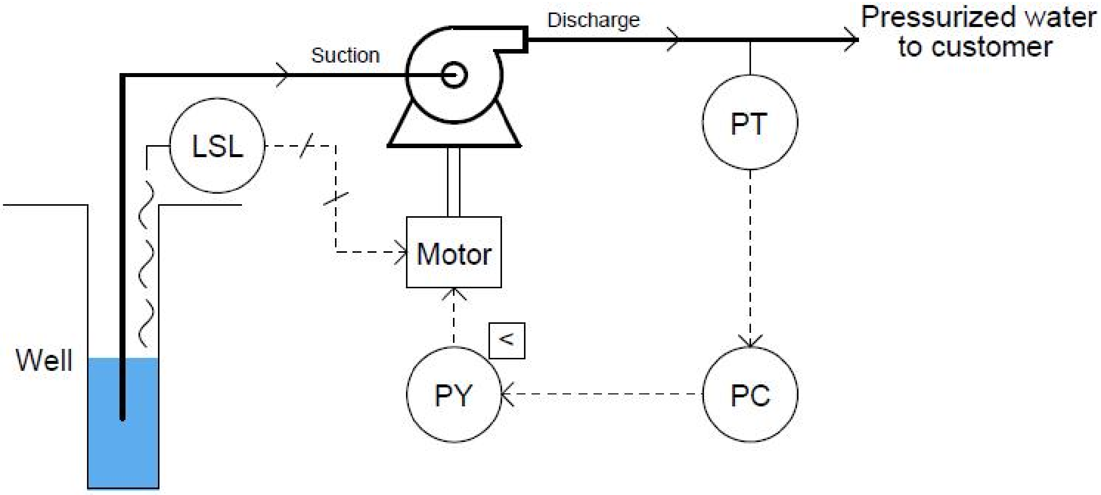
An "override" control strategy involves a selection between two or more controller *output* signals,where c one controller at a time gets the opportunity to exert control over a process. All other "de-select controllers are thus *overridden* by the selected controller.

Consider this water pumping system, where a water pump is driven by a variable-speed electric moto: draw water from a well and provide constant water pressure to a customer:



Incidentally, this is an excellent application for a variable-speed motor as the final control element ra than a control valve. Reducing pump speed in low-flow conditions will save a lot of energy over t compared to the energy that would be wasted by a constant-speed pump and control valve. A poter problem with this system is the pump running "dry" if the water level in the well gets too low, as m happen during summer months when rainfall is low and customer demand is high. If the pump runs for long with no water passing through it, the seals will become damaged. This will necessitate a complete sl down and costly rebuild of the pump, right at the time customers need it the most.

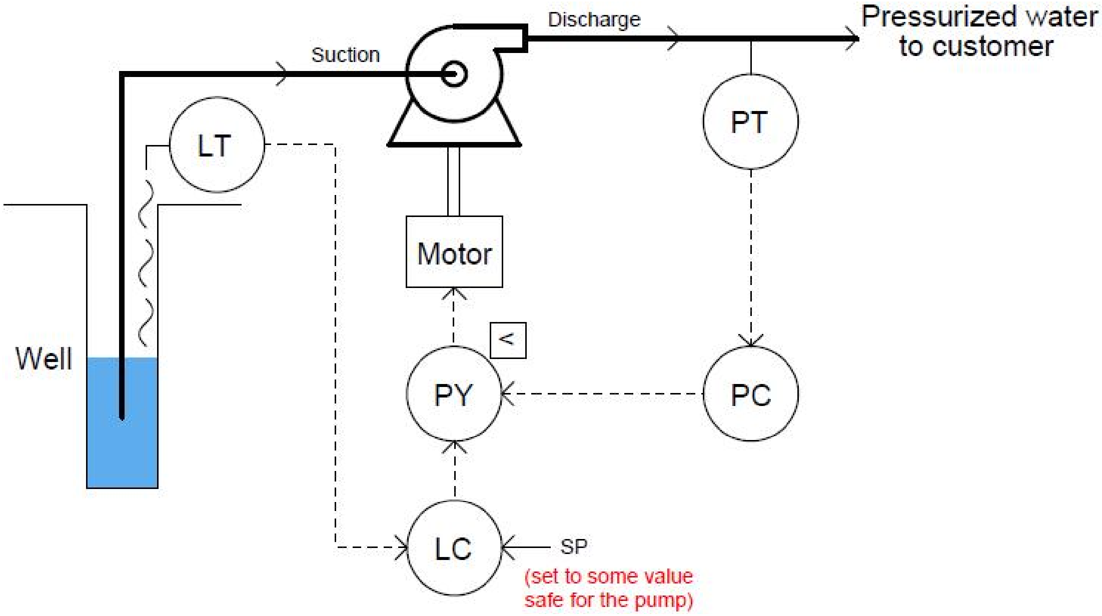
One solution to this problem would be to install a level switch in the well, sensing water level and shutting the electric motor driving the pump if the water level ever gets too low:



This may be considered a kind of “override” strategy, because the low-level switch over-rides the press controller’s command for the pump to turn. It is also a crude solution to the problem, for while it prot the pump from damage, it does so at the cost of completely shutting off water to customers. One way describe this control strategy would be to call it a *hard override* system, suggesting the uncompromi action it will take to protect the pump.

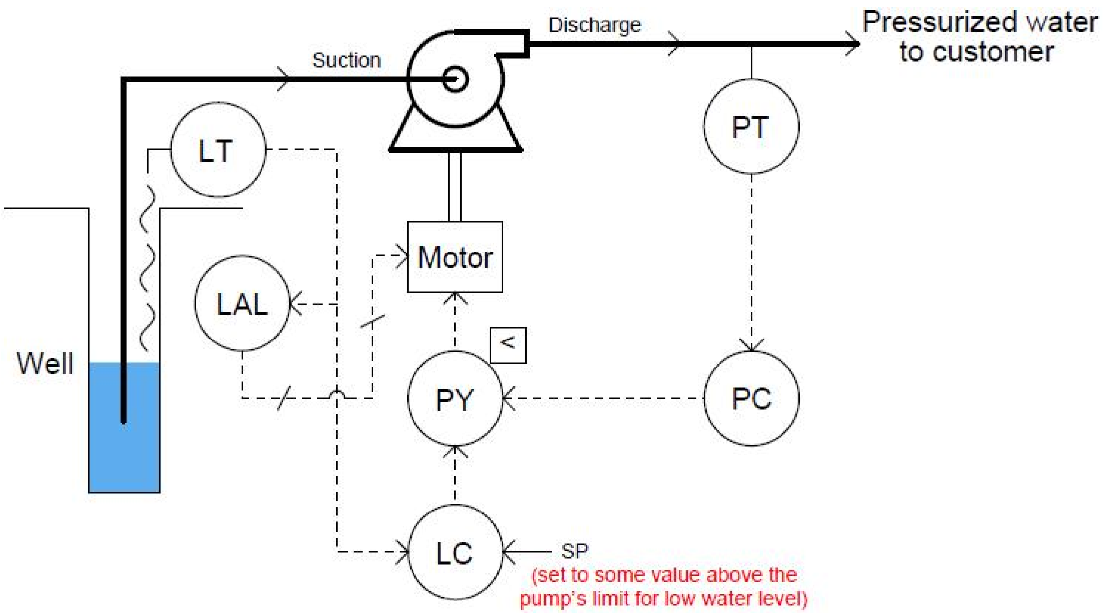
A better solution to the dilemma would be to have the pump merely slow down as the well water l approaches a low-level condition. This way at least the pump could be kept running (and some amoun pressure maintained), decreasing demand on the well while maintaining curtailed service to customers still protecting the pump from dry-running. This would be termed a *Soft override* system.

We may create just such a control strategy by replacing the well water level switch with a level *transmi* connecting the level transmitter to a level controller, and using a low-select relay or function block to se the lowest-valued output between the pressure and level controllers. The level controller’s setpoint will be at some low level above the acceptable limit for continuous pump operation:



If ever the well’s water level goes below this setpoint, the level controller will command the pump to s down, even if the pressure controller is calling for a higher speed. The level controller will have *overridden* pressure controller, prioritizing pump longevity over customer demand. Bear in mind that the concept low-level switch completely shutting off the pump is not an entirely bad idea. In fact, it might be pruden integrate such a “hard” shutdown control in the override control system, just in case something goes wr with the level controller (e.g. an improperly adjusted setpoint or poor tuning) or the low-select function.

With two layers of safety control for the pump, this system provides both a “soft constraint” provic moderated action and a “hard constraint” providing aggressive action to protect the pump from operation:


Discharge
Pressurized water to customer
Suction
PT
LT
Motor
LAL
<
Well
PY
PC
LC
SP
(set to some value above the pump's limit for low water level)

In order that these two levels of pump protection work in the proper order, the level controller's (: setpoint needs to be set to a higher value than the low level alarm's (LAL) trip point. A very import consideration for any override control strategy is how to manage integral windup. Any time a controller v any integral (reset) action at all is de-selected by the selector function, the integral term of the controller have the tendency to wind up (or wind down) over time. With the output of that controller de-coupled f the final control element, it can have no effect on the process variable. Thus, integral control action – purpose of which being to constantly drive the output signal in the direction necessary to achieve zero e between process variable and setpoint – will work in vain to eliminate an error it cannot influence. If when control is handed back to that controller, the integral action will have to spend time "winding" other way to un-do what it did while it was de-selected.

Thus, override controls demand some form of integral windup limits that engage when a controller is selected. Methods of accomplishing this function are discussed in an earlier section on limit controls.

[1] I generally suggest keeping such limit values inaccessible to low-level operations personnel. This is especially true in cases as this where the presence of a high temperature setpoint limit is intended for the longevity of the equipment. There is a st tendency in manufacturing environments to "push the limits" of production beyond values considered safe or expedient by engineers who designed the equipment. Limits are there for a reason, and should not be altered except by people with understanding of and full responsibility over the consequences!

[2] Only the coolant flow control instruments and piping are shown in this diagram, for simplicity. In a real P&ID, there woul many more pipes, valves, and other apparatus shown surrounding this process vessel.

[3] In order to understand how this works, I advise you try a "thought experiment" for each function block network whereby arbitrarily assign three different numerical values for A, B, and C, then see for yourself which of those three values becomes output value.

[4] In FOUNDATION Fieldbus, each and every signal path not only carries the signal value, but also a "status" flag declaring be "Good," "Bad," or "Uncertain." This status value gets propagated down the entire chain of connected function blocks, to dependent blocks of a possible signal integrity problem if one were to occur.

[5] This principle holds true even for systems with no function blocks "voting" between the redundant transmitters. Perhaps installation consists of two transmitters with remote indications for a human operator to view. If the two displays substant disagree, which one should the operator trust? A set of *three* indicators would be much better, providing the operator enough information to make an intelligent decision on which display(s) to trust.

Go Back to Lessons in Instrumentation Table of Contents

## Frp p hqw#+5,

Vyevfuleh#wr#wklv#frp p hqw*v#ihhg#

Hhhthvw#dttuhfldwlrq

{ulwhq#e|#Deld#Rnywlq|dqk/#Myd|#47/#5446#

Myvw#wr#vd|#p |#wkdqnv#iru#wklv#{ rqghuiyo#{ uLwh0yt/#L#kdyh#vhdufkhg#iru#vrp hwklqk#vr#vlp tch#todlq#dqg#{ lwk#hdv|# wr#yqghuvwdqg#dqkydkh#{ lwk#w|tlfdo#h{dp tohv#dqg#|ry#nyvw#vdyhg#p h2#

L#{ lvk#|ry#fryog#fryhu#p ruh#wrtlfv/#lw#{ lo#uhdoo|#eh#ri#khoe#dv#Lp #nyvw#d#|ryqk#lqvwuyp hqwdwlrq#hqklqhhu#{ runlqk#lq#d HTF#ilup #orrnlqk#iru#d#fudvk#fryuvh#lq#frqwuro#dqg#dywrp dwlrq2#

Xkdqnv#rqfh#dkdlq2#

+1

wkdqnv#iru#wklv#{ rqghuiyo#{ ulwh0yt
{ ulwwhq#e|#dvrn/#Dykyvw#4:/#5446#
wrqv#ri#wkdqnv#iru#wklv#{ rqghuiyo#{ ulwh0yt#

+1

## Z ulwh#frp p hqw

**Name**

**Email**

**Website**

**Title**

**Comment**

vp dohu#¡#elkkhu

☑ Vyevfuleh#ylb#hp dlo#+uhklvwhuhg#yvhuv#rqo|,#

□ #L#kdyh#uhdg#dqg#dkuhh#wr#wkh#Xhup v#ri#Xvdkh2

Z ulwh#wkh#glvtod|hg#fkdudfwhuv

Add Comment

Kdv#Xyuelqh#Xywruldo

Lqvwuyp hqwdwlrq#Xyelqk#dqg#wkhlu#
Frqqhfwlrqv

R lo#dqg#Kdv#Turgyfwlrq#Kdqgerrn