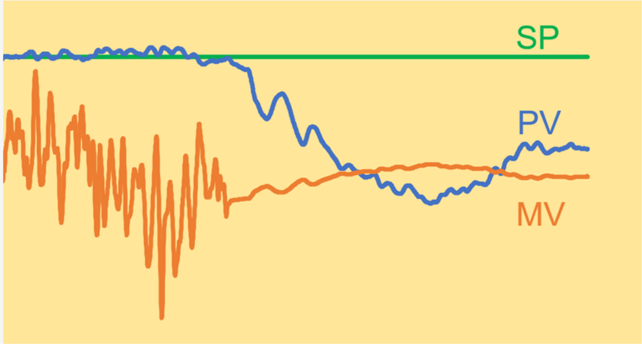
I&EC research
Industrial & Engineering Chemistry Research

Advanced Regulatory Control Techniques for Improved Averaging Level Control Performance

Gustaf Zacharias Gous, Andries Johannes Wiid, Johan Derik le Roux, and Ian Keith Craig*

Cite This: *Ind. Eng. Chem. Res.* 2023, 62, 15578−15587

Read Online
ACCESS | Metrics & More | Article Recommendations



ABSTRACT: Averaging level control is used in liquid processing plants to optimally use the available volume in surge drums and storage tanks to stabilize the feed to downstream equipment. This paper describes three averaging level control techniques, namely, integral gap control (IGC), ramp horizon control (RHC), and a proposed combination of the two techniques called ramp horizon integral gap control (RHIGC). RHIGC is a novel advanced regulatory control (ARC) technique and can be implemented by using standard distributed control system (DCS) functions. The advantage of RHC is that it will keep levels from going over limits when disturbances larger than typical process disturbances occur. However, it does not move the level away from the limit; therefore, the level will violate the limit if a subsequent disturbance occurs. Based on tuning, IGC is able to move the level away from limits very quickly, but it will let the level violate the set limits if a larger than expected disturbance occurs. RHIGC aims to exploit the strengths of both techniques while compensating for individual weaknesses. The techniques are compared in simulations, followed by the implementation of RHIGC on an industrial process plant. RHIGC shows good results in preventing the level from violating the set high and low limits while simultaneously decreasing the variability of the manipulated variable. In process plants, keeping the level between set limits may avoid alarm or trip limits being exceeded, while the decreased variability of the manipulated variable leads to an increased stability in downstream process equipment.

■ INTRODUCTION

Process control engineers strive to keep process variables such as temperatures, levels, and pressures within acceptable ranges in order to produce products that are consistently within specifications. They create controllers that reject process disturbances and move the process from one operating region to another. Process control reduces variability caused by disturbances, which leads to reductions in energy and utility consumption, better quality product, reduced losses, and increased throughput.[1−4] Averaging level control moves the variability in the process from the material flows to the level process variables, making better use of the storage capacity in the plant.

When applying averaging level control in a process plant, the process control engineer will consider the available capacity of storage vessels such as feed drums and will implement controllers that will allow the level to move up and down in order to minimize movement of the manipulated variable (u). The process control engineer will consider what range of the vessel can be used in this way. The upper limit of the level may be limited by process factors such as liquid carryover, as well as alarm or trip limits. The lowest level allowed may be limited by process considerations such as pump cavitation, as well as alarm or trip limits. The engineer will choose limits that are a safe distance away from such constraints and will implement and tune controllers that should not violate these limits.

Averaging level control is important for the stability of processing plants and is widely studied in the literature. Techniques in the literature include proportional-integral-derivative (PID) control,[5−10] piecewise-linear (e.g., gap) control,[6,11,12] range control,[5] nonlinear PID-based control,[6,8,12,13] nonlinear control,[14−16] optimal control,[10,15,17−20] and model-based control.[5,16,21−26]

The process control engineer typically chooses between PID-based control algorithms such as proportional-only, gap and nonlinear PID, or model-based control such as model predictive control (MPC),[24−26] or optimal averaging level control.[20]

Received: May 5, 2023
Revised: August 30, 2023
Accepted: September 6, 2023
Published: September 16, 2023

PID controllers are easier to implement but will allow the level to violate the limits when disturbances are larger than the assumed maximum disturbance used during tuning. When using PID controllers, the upper and lower limits will be the same distance from the desired average value of the level or set point (y^{SP}) as the controllers will attempt to minimize the distance between the current level measurement or process variable (y) and y^{SP}. When implementing model-based controllers, additional hardware and software are required, and they typically execute at a slower execution interval. However, they can be more successful in keeping the level between limits as no assumed maximum disturbance is used during tuning. When using model-based controllers for averaging level control, operators typically set different high and low limits for the level, as opposed to PID controllers that typically use a single value as a set point.

In order to implement optimal averaging level control, additional computer hardware and model predictive software are required. This is typically installed alongside the distributed control system (DCS) or programmable logic controller (PLC) of the plant and connects using communications protocols, such as open platform communications (OPC). Because of this additional complexity, optimal averaging level controllers typically run at a slower execution interval. It is also required that a good model for the level behavior is available,[10] which is either acquired through plant testing and model identification or by calculating the model using drum or tank dimensions. These requirements lead to additional complexity and cost when comparing model-based controllers with DCS or PLC-based controllers.

An alternate approach to optimal averaging level control is to use advanced regulatory control (ARC) techniques, which can be implemented using standard DCS logic blocks. Such techniques can yield results similar to optimal averaging level control, as shown in this paper, without the requirement of additional computer hardware and model predictive software. ARC techniques are often used in industry and are recently receiving increasing attention from the academic literature.[1,5,8,9,11,13,16,27−29]

Both PID- and MPC-based averaging level control techniques work well on large feed tanks holding a few days volume of material or smaller feed drums, which typically hold volumes that represent minutes to hours of throughput. Feed tanks are often upright cylinders in shape, which means that the relationship between the material height in the tank and the p is linear. Feed tanks with an upright cylindrical shape generally exhibit a linear relationship between the level in the tank and the manipulated variable u. In contrast, feed tanks with a horizontal cylindrical shape with domed ends generally exhibit a nonlinear relationship between the level and u. In the case of a nonlinear relationship, process control engineers will typically position the instruments that measure the level in the middle of the drum where the deviation from the set point is approximately linear. In cases where the choice was made to rather use the full capacity of the drum by measuring the level over a wider range, the system can be linearized.[6]

Multiple tank level controllers may occur in series with equipment in between, where the equipment has its own level controllers. For example, a series of tanks may be separated by distillation columns and reactors with their own level controllers. If this happens, applying averaging level control to all tank, drum, and equipment level controllers will have a synergistic effect, where the effect of reducing flow variability upstream will improve the impact of downstream controllers.

The contribution of this paper is the introduction and industrial implementation of a novel ARC technique for averaging level control that combines ramp horizon control (RHC) and an integral gap controller (IGC). RHC sets a future time horizon during which the level is not allowed to go over the high or low limits set by the engineer.[30] After a disturbance occurs, the smallest control moves possible are made, which will reduce the rate of change of the level to zero by the time the limit is reached. As no assumptions regarding maximum disturbances are made during tuning, RHC is better at ensuring that levels do not go over the set limits than PID controllers.[30] Because no attempt is made to move the level away from limits after a disturbance, one weakness of the RHC is that consecutive disturbances in the same direction can result in very large moves or level limit violations.

A common tuning mistake that may lead to a sustained or growing cycle in level control is when too much integral action causes overshoot. IGC is a PID controller that prevents this by using strong integral control action to rapidly move levels away from limits.[31] The integral action is reduced when the level gets closer to the middle of the range and prevents a cycle that could occur otherwise when PI or PI gap controllers are used. As with other PI and PID controllers, the shortcoming of IGC is that disturbances larger than what was assumed during tuning can allow the level to violate the desired limits.

RHC[30] and IGC[31] are combined in this paper in a novel way and termed ramp horizon integral gap control (RHIGC). RHIGC is an ARC technique that combines the benefits of RHC and IGC, and it can be implemented using standard DCS logic blocks. The utility of RHIGC is illustrated through simulation studies and implementation of this technique on an industrial plant. Results show RHIGC is superior to conventional techniques and comparable to MPC in keeping the level between limits while at the same time making smaller control moves. Table 1 summarizes the ARC control algorithm acronyms for ease of reference.

Table 1. Summary of ARC Controllers

controller	acronym	description
integral gap control	IGC	gain scheduling of integral time
ramp horizon control	RHC	use prediction and wait-and-see policy for constraint avoidance
ramp horizon integral gap control	RHIGC	switch between IGC and RHC based on error rate of change

■ AVERAGING LEVEL CONTROL TECHNIQUES

Using PID Control for Averaging Level Control. A PID controller for the level in a vessel will calculate movements in the manipulated variable (represented by u), which is typically a valve opening or the required flow rate, either into or out of the vessel under consideration. The controller implements these moves to bring the level measurement or process variable (represented by y) back to y^{SP}. PID controllers typically run on either a DCS or a PLC.

A typically used PID equation in industry in digital systems is the noninteractive velocity form[6,8,32]

$$\begin{aligned}\Delta u_k &= u_k - u_{k-1}\\ &= K_c\left[e_k - e_{k-1} + \frac{t_s}{\tau_i}e_k + \frac{\tau_D}{t_s}(e_k - 2e_{k-1} + e_{k-2})\right]\end{aligned} \quad (1)$$

where Δu is the change in manipulated variable, K_c is the controller gain, a tuning variable, e_k is the error between y and y^{SP} at the current execution cycle (k), e_{k-1} is the error at the previous execution cycle, e_{k-2} is the error two execution cycles ago, t_s is the execution cycle time of the DCS or PLC, τ_i is the integral tuning constant, and τ_D is the derivative tuning constant.

In industry, averaging level control is typically done without derivative action; i.e., the last term in (eq 1) is neglected. This is because derivative control action can not only result in faster than desired changes in controller moves but also overreact to noise, which will then increase variability in u.

P-Only Control. P-only level control[16,21] is a commonly used form of averaging level control found in industry. Typically, P-only control is done in one of two ways. The first approach is to use a PID controller in a DCS or PLC, where only the proportional action term in (eq 1) is used. The second approach is to map u of a controller to the current value of the level to do P-only level control.[14]

With P-only control, the error will increase as y moves away from y^{SP}, with the control action moving u in proportion to the increasing error. Once u has moved enough to restore the mass balance, the error will stay constant because y does not change. Therefore, no control changes will be made to return y to y^{SP}, and the controller will only reduce the rate of change of the level to zero, called balancing the level.

Nonlinear P-only controllers such as the error squared controller[6] or controllers that set the gain as a function of the absolute value of the error[33] are better suited to prevent y from moving outside the desired range.

PI Control. Adding integral action to a P-only controller to create a PI controller will enable y to return to y^{SP}. This is desirable because consecutive disturbances in the same direction can cause y to go outside of the limits if the controller is designed to keep y only from moving away from y^{SP}. The downside of PI control is an increase in the movement of u to bring y back to y^{SP}.

A common tuning mistake made in industry when using a PI controller is implementing too much integral action that will cause a cycle in the level. When a level is subject to a disturbance, y will start moving away from y^{SP}, causing the error (e_k) to increase if the disturbance moves y upward above y^{SP}. While y is moving away from y^{SP}, the proportional action, $K_c(e_k - e_{k-1})$ and the integral action, $K_c\frac{t_s}{\tau_i}e_k$, will have the same sign. Therefore, while y is moving away from y^{SP}, both proportional and integral actions will work together to reject the disturbance.

When enough control action has been taken to reduce the rate of change of the level to zero, proportional action will be reduced to zero because $(e_k - e_{k-1})$ will be zero. The integral action will still move u in order to reduce e_k to zero, and y will start returning to y^{SP}. During this phase, proportional and integral actions will oppose each other. The integral action will still attempt to reduce the error to zero, while the proportional action will react to the reduction in error by opposing the integral action. The integral action drives the error to zero, while the proportional action prevents y from overshooting y^{SP}. The PI tuning should be such that the proportional control is strong enough to prevent overshoot. If this is not done correctly, a slow cycle in the level will result.

When averaging control is implemented, the dilemma is that weak proportional control is needed to allow y to deviate from y^{SP}, thereby minimizing the movement of u. However, the need to have the integral action to be weaker than the proportional action when y is returning to y^{SP} remains. Therefore, in averaging level control, the integral action must be tuned to be very weak to prevent the level from cycling. Very weak integral action results in the level taking a long time to return to y^{SP}. Depending on the tuning and range allowed for y, it will take much longer to return to y^{SP} than the time it takes to initially balance the level. This may be a problem depending on the frequency of the disturbances.

Optimal Averaging Level Control. Optimal averaging level control[20] uses MPC to predict the trajectory of the level and to calculate control moves that result in the smallest continuous moves in u that will keep the level between limits. These moves will be made until the level limit is reached, and the level is balanced. Optimal averaging level control has been implemented and tested widely, and it is accepted that it improves averaging level control performance.[10,15,17,18,20,22,34−36]

As most DCS-based controllers make control moves that are a function of the error, the PID-based controllers will initially make larger moves after a disturbance, as error will increase quickly. Therefore, PID controllers will initially make large moves that become smaller as time progresses. This gives optimal averaging level control a distinct advantage over PID controllers, as it implements the same move over the full control horizon. This is especially true if the largest move made over the control horizon is used as a measure of success.

Optimal averaging level control requires additional computer and network hardware and software, plant tests, and model identification that increases cost and complexity.

Programmed Imbalance Ramp Control. Commercially available MPC software uses an averaging level control feature known as programmed imbalance ramp control.[24−26] The controller uses a model-based prediction to predict the future trajectory of the level. The process control engineer sets a high and low limit that should not be exceeded, as well as a tuning parameter called the ramp horizon. This is a set time span into the future in which the predicted trajectory of the level should be kept between the high and low limits.

If the level is predicted to cross a limit at a time farther in the future than the ramp horizon, then the controller will not act. As time passes, the level will move toward the limit. Eventually, the level will get close enough to the limit that the current rate of change of the level will cause it to cross the limit in a shorter amount of time than the set ramp horizon. The controller will then make a small move that will decrease the rate of change of the level just enough so that the level will cross the limit at the ramp horizon.

As the level approaches the limit in this way, the controller will keep making control moves that will decrease the rate of change of the level until the rate of change becomes zero, and the level is balanced as the limit is reached.

As with optimal averaging level control, this control method also requires additional computer and network hardware and software, plant tests, and model identification that increases cost and complexity.

Ramp Horizon Control. RHC[30] is a simplification of programmed imbalance control widely used in industry.[24−26] RHC can be seen as a "wait and see" control strategy, where control moves, Δu, are only made if there is imminent danger of the level violating a limit. It can be implemented by using standard DCS functions. A number of execution cycles are selected during which the high and low limits imposed on the level may not be exceeded. This is called the ramp horizon (T_{RH}).

The trajectory of the level is predicted based on the current position and rate of change. Control moves are made only if the predicted trajectory will violate a limit within the ramp horizon. If no violation is predicted, then no control moves are made.

The value of y at T_{RH} is calculated as

$$y_{k+T_{\mathrm{RH}}} = y_k + T_{\mathrm{RH}}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)_k \tag{2}$$

where $y_{k+T_{\mathrm{RH}}}$ is the predicted value of the level at T_{RH} and y_k is the current value of the level. If this predicted value does not violate the high or low limit, no control moves are made; otherwise, a move will be calculated and implemented.

A process slope gain k' is calculated by either using step test data or calculating the volume of liquid between the high and low limits on the drum.[6] The process slope gain is the rate of change in the level that will result when the process input u_{RH} is moved up by one engineering unit

$$k' = \frac{\Delta y}{\Delta u \times \Delta t} \tag{3}$$

where Δy is the change in the output y from $t = 0$ (initial steady state) to $t = \Delta t$ (a selected time after the change in u was made).

By rearranging (eq 3), the maximum size of the move in u that prevents the level limit from being exceeded within the ramp horizon, T_{RH}, can be calculated using the process slope gain

$$\Delta u = \frac{\Delta y}{k' \times \mathrm{T_{RH}}} \tag{4}$$

where Δy is the difference between the value of the active limit and the current value of y.

Consider an example in which the level approaches the upper limit and will exceed the limit within the ramp horizon. Moving u up continuously per cycle will prevent the level from exceeding the limit when the prediction reaches the ramp horizon. However, as the level will still be approaching the limit, during the next execution cycle, the controller will once again predict that the limit will be exceeded. It will implement another controller move to ensure that at the ramp horizon, the level will still be inside the limit. If no other process influences the trajectory of the level, then this process will be repeated until the rate of change of the level reaches zero as it reaches the high limit.

When a relatively small disturbance causes a change on a level that is controlled using RHC, the prediction will initially not show that a limit will be exceeded within the ramp horizon. Initially, no control moves will be made. As time passes, the level will approach the limit, and at some stage, a small violation will be predicted. A control move will be calculated to change the rate of change of the level such that the level will be on the limit at the ramp horizon.

RHC will only ensure the limits are not exceeded and make no attempt to move the level away from the limit. The risk remains that continuous or consecutive disturbances in the same direction may occur once the level reaches the limit. When this happens, the controller will not be able to keep the level at or between the imposed limits. However, if this occurs and the level moves outside a limit, then RHC will return the level to the limit that was exceeded using the same algorithm.

RHC simplifies the calculated response of the level by using gain only. An imprecise control action might result as dynamic behavior is ignored, measurement noise and unmeasured disturbances might be present, and the controller gain might be incorrect. However, in a short execution cycle, the level measurement that is taken as input and the rate of change of the level that is calculated at every execution cycle continuously update the controller error. This compensates for the possible inaccuracies in the measurement and control.

Integral Gap Control. The tuning of PID controllers in most modern DCSs can be changed in real time using function blocks or code based on process or control states. Similar to commonly used proportional gap control,[6] IGC uses this facility to define a gap around y^{SP} of a PID controller where the integral action is decreased. The PID controller then uses less integral action when y is close to y^{SP} and more integral action when y is farther away from y^{SP}. As with PI gap control, this can be implemented by defining a ratio

$$\tau_{\mathrm{i}(r)} = \frac{\tau_{\mathrm{i(gap)}}}{\tau_{\mathrm{i}}} \tag{5}$$

where $\tau_{\mathrm{i}(r)}$ is the ratio between the integral tuning constant used when y is inside the gap ($\tau_{\mathrm{i(gap)}}$) and the integral tuning constant used when y is outside the gap (τ_{i}).

IGC can be defined as

$$\Delta u = \begin{cases} K_{\mathrm{c}}\left[(e_k - e_{k-1}) + \dfrac{t_{\mathrm{s}}}{\tau_{\mathrm{i(gap)}}} e_k\right] & \text{if } \underline{y} < y < \overline{y} \\ K_{\mathrm{c}}\left[(e_k - e_{k-1}) + \dfrac{t_{\mathrm{s}}}{\tau_{\mathrm{i}}} e_k\right] & \text{otherwise} \end{cases} \tag{6}$$

IGC enables more aggressive tuning when y is close to limits while allowing the controller to slow the approach to y^{SP} enough to prevent overshoot. Reducing the integral control action when y is close to y^{SP} while taking more aggressive integral action when y is farther from y^{SP} mitigates the potential overshoot when y returns to y^{SP}.

When y is moving toward y^{SP} from outside of the gap, proportional control will work against the integral action, slowing down the return to y^{SP}. As the integral action is still tuned aggressively, it should overshadow the proportional action, causing y to quickly return to y^{SP}.

Once y moves back into the gap, decreased integral action will ensure that y will not overshoot y^{SP}. Only small controller moves will be made while y is inside the gap.

By setting the proportional gain, the integral tuning, and the gap correctly, IGC will be able to

- avoid overshoot,
- more aggressively return y to y^{SP} while the error is large, and
- slow down the controller response when the error is small.

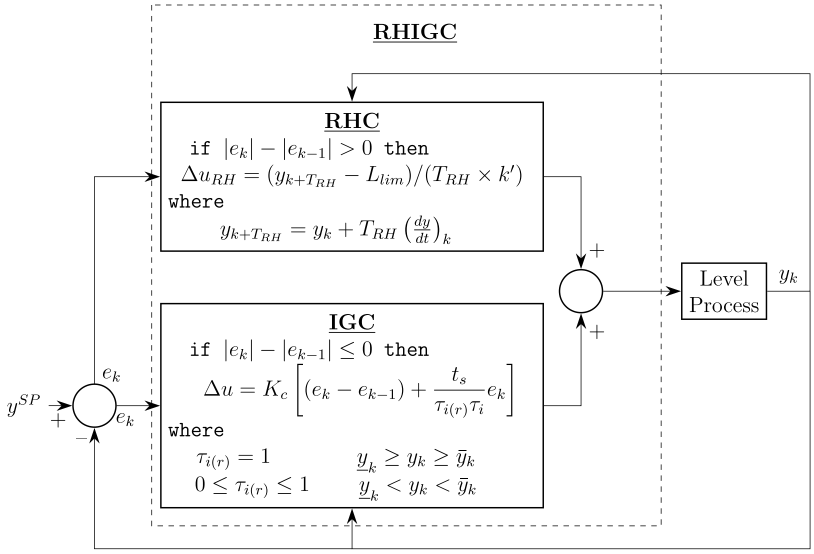


Figure 1. RHIGC.

COMBINING RAMP HORIZON CONTROL AND INTEGRAL GAP CONTROL

This section details the design of the RHIGC, which combines RHC and IGC (Figure 1). In particular, RHIGC benefits from the fact that RHC makes no assumption regarding the maximum disturbance during tuning and that it does not make any changes to u if there is little threat of y going over limits. The IGC part of RHIGC compensates for the main disadvantage of RHC, which does not return the level to y^{SP} after a disturbance.

An advantage of RHIGC over advanced control methods such as MPC is that it can be implemented on a DCS or PLC without the need for additional computer and network hardware and software, plant tests, and model identification.

RHIGC works by switching between the RHC and IGC depending on which controller is best suited to current process conditions. Switching is dependent on whether the error is increasing or decreasing. RHC is used when the error is increasing, i.e., when the current value of the error has the same sign as the rate of change of the level. RHIGC is switched to the IGC part when the error is zero or decreases, i.e., the rate of change of the level becomes zero or while y is moving back toward y^{SP}.

RHC makes the smallest moves possible while the level is moving away from y^{SP} and will prevent y from going outside its limits. IGC will bring y back toward y^{SP} after RHC has rejected the disturbance and balanced y. IGC will move u more aggressively when y is close to the limits, where there is a larger risk that RHC will not be able to keep the level within limits should another disturbance occur. Later, IGC will make smaller moves when y is inside the gap, improving the averaging level control performance and ensuring that y does not overshoot y^{SP}.

It is possible to switch between different inputs and outputs in DCS-based controllers.[28] When y is moving away from y^{SP}, the error increases, and standard DCS functions are used to change the controller to manual mode in order to implement the control moves calculated by the RHC part of RHIGC. This is done by using standard DCS functions to switch the IGC controller to manual and writing the RHC u values directly to the output of the controller. When y is moving toward y^{SP} or stays constant, the error correspondingly decreases or remains constant, and standard DCS functions switch the IGC controller from manual to auto mode, enabling IGC to output its own calculated u value. This is done to enable seamless switching between the two parts of the controller.

When determining whether y is approaching or moving away from y^{SP}, a first-order filtered value of y is used to prevent frequent switching between the IGC and the RHC parts of the controller.

PERFORMANCE METRICS USED

Performance metrics described in this section are used to compare the success of different averaging level control techniques and the relevant tuning parameters used. These metrics penalize u movement, limit violations, and unused buffer capacity.

Performance metrics found in the literature typically only penalize movement of u while assuming that the level does not violate the preset limits. These include the standard deviation of u (σ_u)[19,21,26] and the total variance of u (TV)[37]

$$\sigma_u = \sqrt{\frac{1}{N-1}\Sigma_{k=2}^{N}[(u_k - u_{k-1}) - \mu]^2} \tag{7}$$

$$\mathrm{TV} = \Sigma_{k=1}^{N}|u_{k+1} - u_k| \tag{8}$$

where N is the number of data points in the data set and μ is the mean of u over the data set.

When σ_u is used, a control system that results in a lower standard deviation while keeping the level between the predetermined limits is desired. When using TV, the smaller the maximum rate of change of u, the better the controller is at making smaller rather than larger moves. If TV is divided by the number of samples, i.e., TV/N, a comparison between two or more data sets with differing numbers of samples can be done.

In Horton et al.,[36] the performance metric is normalized by dividing σ_u of the controller being tested by σ_u of a benchmark PI controller. This normalizes the metric to one, with a result smaller than one being better than the benchmark.

■ SIMULATION

Simulation Description. The impact of combining RHC and IGC is illustrated in a simple simulation example of the control of a drum level, as shown in Figure 2. The process simulation is based on the industrial plant example where RHIGC is implemented. The level controller (LIC) cascades to the flow controller (FIC), which manipulates the flow exiting the vessel. In other words, the level (y) is controlled by manipulating the y^{SP} of FIC. The measured flow into the vessel is a disturbance and is measured by FI.

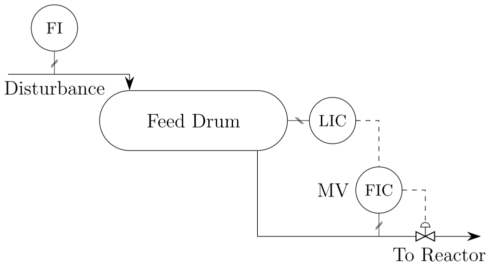


Figure 2. Feed drum process flow diagram.

The simulation was done to show typical responses of the different controllers rather than quantitative analysis. It shows how the PI controller has a typical response, how the IGC can move u faster when error is large, as well as how it can move u slower when error is small. It also shows how RHC can prevent the level from going over the limit during the first disturbance only and how RHIGC could prevent it during the second disturbance as well because the IGC part of the controller moves the level away from the limit.

The simulation was conducted on a standard DCS using a typical PI controller as well as IGC, RHC, and RHIGC controllers to control the feed drum level. The level was simulated by using standard filter blocks that calculate the level changes based on changes in u and the disturbances. The simulation starts with y at y^{SP}, i.e., the level of the tank is at 50%. The upper hard constraint is a level of 70%. Two consecutive steps of 5 m^3/h each were made, with an interval of 120 min between them. The duration of the interval between the two steps was chosen to allow enough time for the controllers either to reduce the rate of change of y to zero or to start returning y to y^{SP}. The tuning parameters of the controllers are listed in Table 2.

The model of the process in Figure 2, as used in the simulation, represents the response of the flow set point to the drum level

$$G(s) = -1.43\frac{0.19s + 1}{0.54s^2 + s}e^{-0s} \tag{9}$$

where the units for $G(s)$ are $\frac{\text{level }\%}{\text{m}^3/\text{h}}$.

Simulation Results. Figure 3 shows how the PI controller, IGC, RHC, and RHIGC responded to the two step changes in the disturbance. The PI controller is able to reject the first step disturbance successfully but fails to keep the level below the high limit of 70% when the second disturbance occurs.

IGC increases its integral control action when the level moves higher than the gap, which is why it goes over the limit by a smaller margin than the PI controller.

RHC can keep the deviation over the high limit smaller than both the PI and IGC controllers. However, it does make very large moves during the period that the level is over the limit, and once the level turns, no attempt is made to stop the level from drifting through the range between the high and low limits.

Table 2. Tuning Parameters for Simulation

controller	parameter	value	units
PI	K_c	0.2	$\%^{-1}$
	τ_i	128	min
	t_s	1	s
IGC	K_c	0.19	$\%^{-1}$
	τ_i	96	min
	$\tau_{i(gap)}$	256	min
	gap	10	%
	t_s	1	s
RHC	K_G	18.75	$\%^{-1}$
	T_{RH}	900	s
	t_s	1	s
RHIGC	K_c	0.19	$\%^{-1}$
	τ_i	96	min
	$\tau_{i(gap)}$	256	min
	K_G	18.75	$\%^{-1}$
	T_{RH}	900	s
	t_s	1	s

The RHIGC can keep the level from going over the high limit. This is possible because the IGC component of the controller moves the level away from the limit once the RHC component of the combined controller brings the rate of change of y to zero.

The simulation results clearly illustrate the principle of RHIGC. As can be seen from Figure 3, the RHC part of the controller, which is active from $t = 5$ min to $t = 74$ min, as well as from $t = 116$ min to $t = 157$ min where $|e_k| - |e_{k-1}| \geq 0$, prevents the level from exceeding limits. The IGC part of the controller, which is active from $t = 75$ to $t = 115$ min as well as from $t = 158$ to $t = 295$ min where $|e_k| - |e_{k-1}| \leq 0$, moves the level away from the limits to prevent consecutive disturbances from forcing the level over a limit.

Comparison with Model Predictive Control. The simulated plant shown in Figure 2 was used to compare the performance of RHIGC to a traditional MPC algorithm. Typical plant data of a stream flowing into a drum on an actual process were used as the disturbance for a 24 h simulation. The low and high hard level constraints are chosen as 30 and 70%, respectively.

The MPC can be represented as

$$\min_{u_k, u_{k+1}, \cdots, u_{k+N_C-1}} \sum_{j=1}^{N_P} (e_{k+j}^2 Q_R + S_{k+j} Q_S) + \sum_{j=0}^{N_C-1} (\Delta u_{k+j})^2 R \tag{10}$$

subject to

$$x_{k+j} = Ax_{k+j-1} + Bu_{k+j-1} \ \forall\ j = 1, \cdots, N_P \tag{11a}$$

$$y_{k+j} = Cx_{k+j} + Du_{k+j} \ \forall\ j = 1, \cdots, N_P \tag{11b}$$

$$\underline{y} \leq y_{k+j} \leq \overline{y} \ \forall\ j = 1, \cdots, N_P \tag{11c}$$

$$\underline{u} \leq u_{k+j} \leq \overline{u} \ \forall\ j = 0, \cdots, N_C - 1 \tag{11d}$$

$$\Delta\underline{u} \leq \Delta u_{k+j} \leq \Delta\overline{u} \ \forall\ j = 0, ..., N_C - 1 \tag{11e}$$

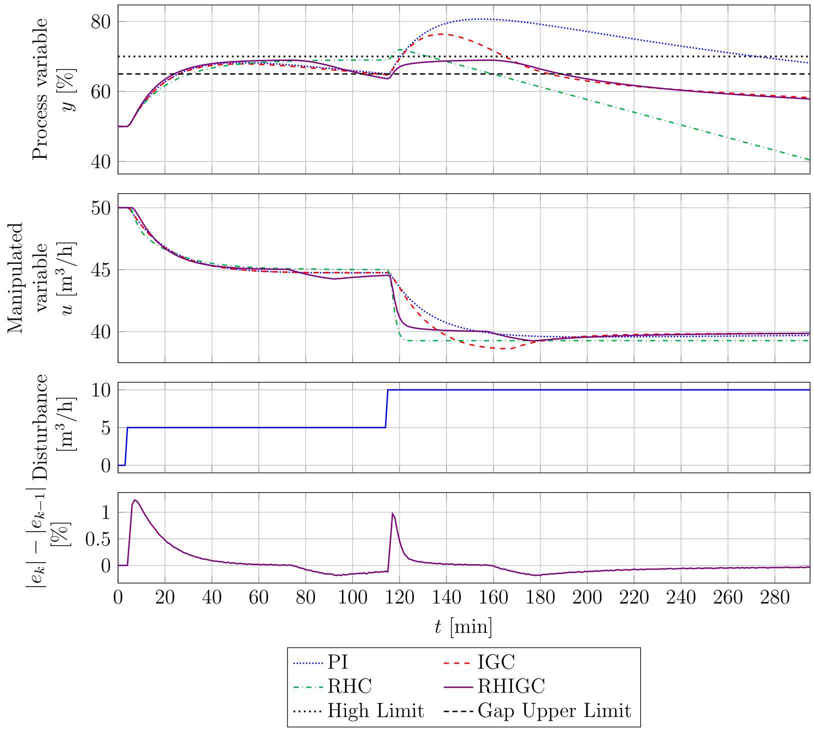


Figure 3. Simulated response of the PI, IGC, RHC, and RHIGC controllers to two step disturbances.

N_P is the prediction horizon, and N_C is the control horizon. The objective is to keep the level between soft limits with the slack variable weight Q_S and to slowly move the level toward y^{SP} with the penalty Q_R so that sequential disturbances in the same direction can be rejected. Flow stability is influenced through the input move size weight R.

The tank level is modeled as $\Delta y/\Delta t = 1/a(Q_{in} - Q_{out})$, where a is the area of the drum, Q_{in} is the flow into the drum, and Q_{out} is the flow out of the drum. Therefore $A = 0$, $B = 1/a$, $C = 1$, and $D = 0$.

S_{k+j} is a slack variable used to penalize level limit violations before hard constraints are reached and is defined as

$$S_{k+j} = \begin{cases} y_{sh} - y; & y \ge y_{sh}, \\ y - y_{sl}; & y < y_{sl}, \\ 0; & y_{sl} \le y \le y_{sh}, \end{cases} \qquad (12)$$

where y_{sl} and y_{sh} are the soft low and high level limits, respectively. The soft limits are chosen to be 3% away from the hard limits. $\underline{y}$ and $\overline{y}$ are y low and high limits, $\underline{u}$ and $\overline{u}$ are u low and high limits, and $\Delta\underline{u}$ and $\Delta\overline{u}$ are u move size low and high limits, respectively. The controller calculates a vector of future control moves up to N_C, and the first control move in the vector is implemented at every execution cycle.

Set point tracking, soft limits for y, and minimizing u move sizes are included as optimization objectives in (eq 11a). Process constraints are met using hard constraints for the range of y (eq 11c), the range of u (eq 11d), and the maximum u move size (eq 11e).

The MPC and RHIGC tuning parameters for the simulation are listed in Table 3. The process model used for the simulation as well as the model used for control by the MPC is shown in (eq 9).

Table 3. Tuning Parameters for Comparison of RHIC to MPC

controller	parameter	value	units
MPC	N_P	40	min
	N_C	15	min
	Q_R	9	$\%^{-1}$
	Q_S	3	$\%^{-1}$
	R	0.1	$\%^{-1}$
	t_s	60	s
RHIGC	K_c	0.19	$\%^{-1}$
	τ_i	96	min
	$\tau_{i(gap)}$	256	min
	K_G	18.75	$\%^{-1}$
	T_{RH}	900	s
	t_s	1	s

To enable a fair performance comparison, data was sampled at 1 min intervals for both controllers, even though t_s is 1 s for RHIGC and t_s is 1 min for MPC.

The performance comparison of the controllers is shown in Table 4, and the typical behavior of the controllers is shown by displaying data for the last 4 h of the 24 h simulation of each controller in Figure 4. The disturbance added to the simulations on the inlet flow is typical process data.

As shown in Table 4, the performance of RHIGC and MPC is similar. The σ_u performance metric indicates that RHIGC has the advantage of requiring less movement in u. The

Table 4. Comparison between RHIGC and MPC for the 24 h Simulation

	$\overline{L}$	$\underline{L}$	σ_u	TV/N
RHIGC	68.0	31.4	0.079	0.039
MPC	68.3	31.4	0.085	0.039

https://doi.org/10.1021/acs.iecr.3c01506
Ind. Eng. Chem. Res. 2023, 62, 15578−15587

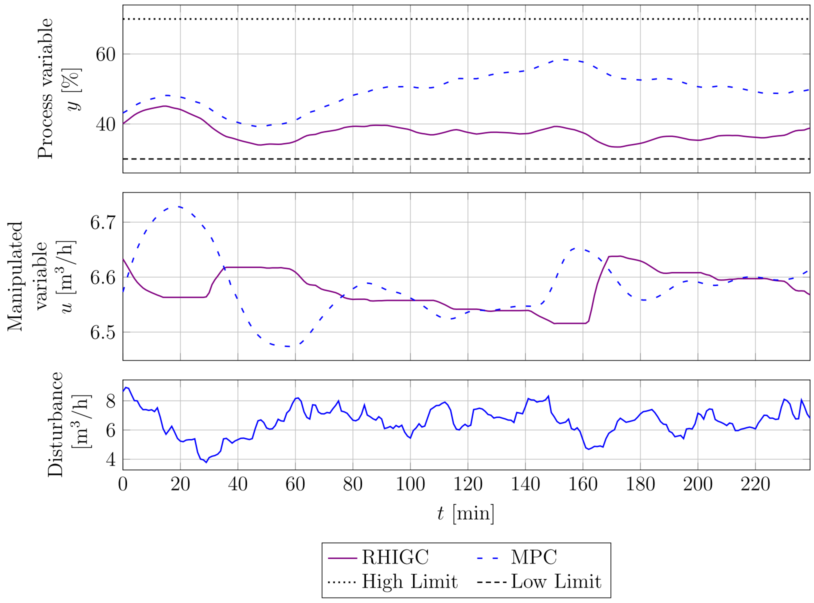


Figure 4. Typical RHIGC and MPC behavior with flow disturbance.

RHIGC has a faster execution interval and simplified process model running natively on a DCS. The MPC has a slower execution frequency and a higher fidelity model that includes the dynamic model (eq 9) running on additional computer hardware and communicates via an open process control (OPC) protocol.

■ PLANT TEST RESULTS

RHIGC was implemented on a standard DCS instrument used in the petrochemical industry. This process is also used in the simulation that compares RHIGC with MPC, with the only difference being that the FI reading is not available in the real process, although it is used in the simulation for information purposes.

RHIGC was implemented on a real process plant where a varying feed rate into a feed drum causes the level to vary continuously, similar to the process flow diagram shown in Figure 2. There was an existing PI gap controller implemented on the process, with existing tuning, as shown in Table 5. RHC was implemented in the process to improve the control by replacing the existing PI gap controller. The control was subsequently further improved by replacing RHC with RHIGC.

The RHC part of RHIGC was tuned to keep the level between 30 and 80%. The IGC part of the RHIGC was tuned by inspection of plant data to identify a typical and maximum feed disturbance magnitude. Averaging level tuning rules proposed by King[6] were used to calculate K_c and τ_i for the IGC part of RHIGC. The RHC part of the controller was given a horizon of 100 min based on trial and error. The controller tuning parameters for the real plant implementation are given in Table 5.

Table 5. Tuning Parameters for Plant Test

controller	parameter	value	units
PI gap	K_c	0.95	$\%^{-1}$
	τ_i	180	min
	K_r	0.625	
RHC	K_G	20	$\%^{-1}$
	T_{RH}	100	min
RHIGC	K_c	0.6	$\%^{-1}$
	τ_i	180	min
	$\tau_{i(gap)}$	247	min
	K_G	20	$\%^{-1}$
	T_{RH}	100	min

Figure 5 shows the level and outlet flow rate set by u, each for a different 24 h time span of typical operation, using the original PI gap controller, RHC, and RHIGC, respectively.

As can be seen, all controllers are well tuned and reasonably successful in keeping the level in the range of 30−80% while making minimum control moves.

The comparison of the controllers according to the performance metrics in (eqs 7 and 8) is shown in Table 6, along with the highest value ($\overline{L}$) and the lowest value ($\underline{L}$) of y during the time each controller was used. The PI gap controller was tuned very well as it kept the level between limits within a small margin. RHC was able to successfully minimize u movement but left y close to limits, increasing the risk that subsequent disturbances may push the level over the limits. Both σ_u and TV/N show that RHIGC moved u much less than the PI controller, with associated improvements downstream due to increased stability.

If averaging level control is implemented on a plant where multiple disturbances that move the level in the same direction can occur, then RHIGC will bring the level back to y^{SP} after a disturbance. This will ensure that the control system is in a better position to reject consecutive disturbances. This ability was demonstrated on a typical process plant with a significant improvement in performance, as shown in Table 6.

■ CONCLUSIONS

This paper described three advanced regulatory control techniques for averaging level control, namely, integral gap control (IGC) and ramp horizon control (RHC), and a combination of the two techniques called ramp horizon integral gap control (RHIGC). RHIGC combines the strengths of RHC and IGC to improve the averaging level control in a process plant. RHIGC is able to keep y within limits in the

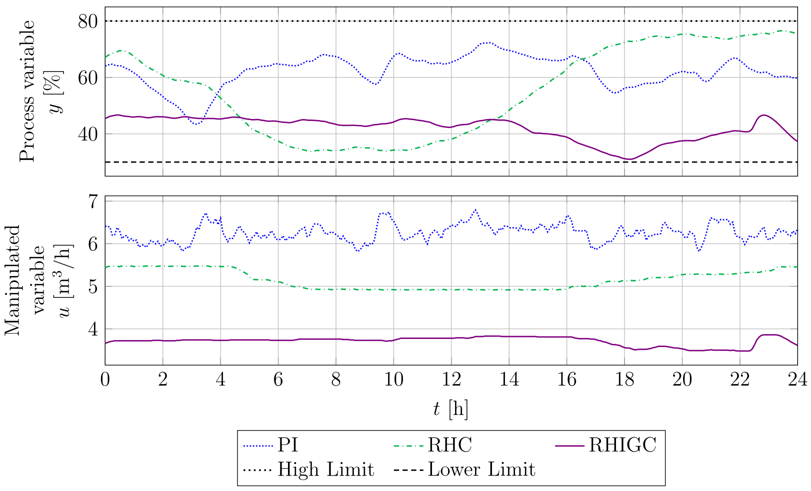


Figure 5. Behavior of the PI gap, RHC, and RHIGC controllers on the plant.

Table 6. Comparison of RHIGC and PI Gap Control

	$\overline{L}$	$\underline{L}$	σ_u	TV/N
PI gap controller	84.7	36.7	0.33	0.15
RHC	77.6	30.6	0.057	0.043
RHIGC	78.2	29.4	0.052	0.013

presence of large disturbances due to the RHC component. RHIGC is able to return y to y^{SP} in a manageable time frame after frequent disturbances due to the IGC component. Additionally, RHIGC aims to move u as little as possible in light of frequent and large disturbances.

RHIGC is able to achieve similar results when compared to MPC in a simulation environment and has the advantage of being implementable by using standard DCS blocks. This allows RHIGC to be executed at high sampling frequencies and does not require specialized software and hardware.

RHIGC was implemented on a real process plant and showed good improvements when compared with an existing PI gap controller and RHC. The control techniques were compared by using the standard deviation and total variance as performance metrics.

RHIGC shows good results in preventing the level from going outside of set high and low limits while simultaneously decreasing the variability of the manipulated variable. In process plants, keeping the level between set limits may avoid alarm or trip limits being exceeded, while the decreased variability of the manipulated variable leads to an increased stability of the downstream process equipment.

AUTHOR INFORMATION

Corresponding Author

Ian Keith Craig − *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa;* orcid.org/0000-0001-7437-3983;
Email: ian.craig@up.ac.za

Authors

Gustaf Zacharias Gous − *Analyte Control, Pretoria 0165, South Africa; Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa*

Andries Johannes Wiid − *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa*

Johan Derik le Roux − *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa;* orcid.org/0000-0002-6306-1949

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.iecr.3c01506

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number: 137769).

REFERENCES

(1) Muller, C. J.; Craig, I. K. Energy reduction for a dual circuit cooling water system using advanced regulatory control. *Appl. Energy* **2016**, *171*, 287−295.

(2) Bauer, M.; Craig, I. K. Economic assessment of advanced process control−a survey and framework. *J. Process Control* **2008**, *18*, 2−18.

(3) Bauer, M.; Craig, I. K.; Tolsma, E.; de Beer, H. A profit index for assessing the benefits of process control. *Ind. Eng. Chem. Res.* **2007**, *46*, 5614−5623.

(4) Rambalee, P.; Gous, G.; de Villiers, P. G. R.; McCulloch, N.; Humphries, G. Control and stabilization of a multiple boiler plant: An APC approach. *IFAC Proc. Vol.* **2010**, *43*, 109−114.

(5) Reyes-Lúa, A.; Backi, C. J.; Skogestad, S. Improved PI control for a surge tank satisfying level constraints. *IFAC-PapersOnLine* **2018**, *51*, 835−840.

(6) King, M. *Process Control: A Practical Approach*; John Wiley & Sons: United Kindom, 2016.

(7) Wade, H. Tuning Level Control Loops. In *Instrument Engineers' Handbook: Process Control and Optimization*; Liptak, B. G., Ed.; CRC Press, 2005; Vol. 2, pp 432−441.

(8) Seborg, D. E.; Edgar, T. F.; Mellichamp, D. A.; Doyle, F. J., III *Process Dynamics and Control*; John Wiley & Sons, United Kingdom, 2016.

(9) Cheung, T.-F. Liquid-level control in single tanks and cascades of tanks with proportional-only and proportional-integral feedback controllers. *Ind. Eng. Chem. Fundam.* **1979**, *18*, 15−21.

(10) Lakerveld, R.; Benyahia, B.; Heider, P. L.; Zhang, H.; Braatz, R. D.; Barton, P. I. Averaging level control to reduce off-spec material in a continuous pharmaceutical pilot plant. *Processes* **2013**, *1*, 330−348.

(11) Cheung, T.-F.; Luyben, W. L. Nonlinear and nonconventional liquid level controllers. *Ind. Eng. Chem. Fundam.* **1980**, *19*, 93−98.

(12) Wu, K.-L.; Yu, C.-C.; Cheng, Y.-C. A two degree of freedom level control. *J. Process Control* **2001**, *11*, 311−319.

(13) Sausen, A.; Sausen, P. S.; de Campos, M. The error-squared controller: a proposed for computation of nonlinear gain through Lyapunov stability analysis. *Acta Sci., Technol.* **2014**, *36*, 497−504.

(14) Sanchis, R.; Romero, J. A.; Martín, J. M. A new approach to averaging level control. *Control Eng. Pract.* **2011**, *19*, 1037−1043.

(15) Rosander, P.; Isaksson, A. J.; Löfberg, J.; Forsman, K. Practical control of surge tanks suffering from frequent inlet flow upsets. *IFAC Proc. Vol.* **2012**, *45*, 258−263.

(16) Taylor, A. J.; La Grange, T. G. Optimize surge vessel control. *Hydrocarbon Process.* **2002**, *81*, 49−52.

(17) Ye, N.; Mcavoy, T. J.; Kosanovich, K. A.; Piovoso, M. J. Optimal averaging level control for the Tennessee Eastman problem. *Can. J. Chem. Eng.* **1995**, *73*, 234−240.

(18) Sbarbaro, D.; Ortega, R. Averaging level control: An approach based on mass balance. *J. Process Control* **2007**, *17*, 621−629.

(19) Lee, M.; Shin, J. Constrained optimal control of liquid level loop using a conventional proportional-integral controller. *Chem. Eng. Commun.* **2009**, *196*, 729−745.

(20) McDonald, K. A.; McAvoy, T. J.; Tits, A. Optimal averaging level control. *AIChE J.* **1986**, *32*, 75−86.

(21) Rosander, P.; Isaksson, A.; Löfberg, J.; Forsman, K. *Performance Analysis of Robust Averaging Level Control*; Linköping University Electronic Press: Sweden, 2011.

(22) van der Burg, M. W.; Djavdan, P. Model predictive averaging level control using disturbance prediction. *IFAC Proc. Vol.* **1995**, *28*, 219−224.

(23) Campo, P. J.; Morari, M. Model predictive optimal averaging level control. *AIChE J.* **1989**, *35*, 579−591.

(24) Qin, S. J.; Badgwell, T. A. A survey of industrial model predictive control technology. *Control Eng. Pract.* **2003**, *11*, 733−764.

(25) Gupta, Y. P. Control of integrating processes using dynamic matrix control. *Chem. Eng. Res. Des.* **1998**, *76*, 465−470.

(26) Brooks, K.; Burchell, J.; Pieterse, F. Model predictive control of an exothermic pressure leach circuit. *IFAC-PapersOnLine* **2017**, *50*, 10232−10237.

(27) Reyes-Lúa, A.; Skogestad, S. Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *J. Process Control* **2020**, *91*, 1−11.

(28) Reyes-Lúa, A.; Skogestad, S. Systematic design of active constraint switching using classical advanced control structures. *Ind. Eng. Chem. Res.* **2020**, *59*, 2229−2241.

(29) Zotică, C.; Forsman, K.; Skogestad, S. Bidirectional inventory control with optimal use of intermediate storage. *Comput. Chem. Eng.* **2022**, *159*, No. 107677.

(30) Gous, G.; de Vaal, P.; van Coller, G. Implementation of averaging level control using native DCS functions. Hydrocarbon Processing, 2021 https://www.hydrocarbonprocessing.com/magazine/2021/may-2021/process-controls-instrumentation-and-automation/implementation-of-averaging-level-control-using-native-dcs-functions.

(31) Gous, G.; de Vaal, P. Integral vs. proportional gap for averaging level control. Hydrocarbon Processing, 2021 https://www.hydrocarbonprocessing.com/magazine/2021/july-2021/bonus-report-the-digital-plan/integral-vs-proportional-gap-for-averaging-level-control.

(32) Corripio, A. B.*Digital control techniques, Series A*; AIChE: Washington, 1982.

(33) Bequette, B. W. *Process Control: Modeling, Design, and Simulation*; Prentice Hall Professional Technical Reference: NJ, 2003.

(34) Ogawa, S.; Allison, B.; Dumont, G.; Davies, M. In *A New Approach to Optimal Averaging Level Control with State Constraints*, Proceedings of the 41st IEEE Conference on Decision and Control; IEEE, 2002; pp 1952−1957.

(35) Sbarbaro, D.; Ortega, R. In *Averaging Level Control of Multiple Tanks: A Passivity Based Approach*, Proceedings of the 44th IEEE Conference on Decision and Control; IEEE, 2005; pp 7384−7389.

(36) Horton, E. C.; Foley, M. W.; Kwok, K. E. Performance assessment of level controllers. *Int. J. Adapt. Control Signal Process.* **2003**, *17*, 663−684.

(37) Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. *J. Process Control* **2003**, *13*, 291−309.

https://doi.org/10.1021/acs.iecr.3c01506
Ind. Eng. Chem. Res. 2023, 62, 15578−15587