Seamless PID–MPC Hybrid Control

Norlund, Frida; Hägglund, Tore; Soltesz, Kristian

Published in:
IFAC Proceedings Volumes (IFAC-PapersOnline)

2024

Document Version:
Peer reviewed version (aka post-print)

Link to publication

Citation for published version (APA):
Norlund, F., Hägglund, T., & Soltesz, K. (in press). Seamless PID–MPC Hybrid Control. *IFAC Proceedings Volumes (IFAC-PapersOnline)*.

Total number of authors:
3

Seamless PID–MPC Hybrid Control *

F. Norlund [*,**] **T. Hägglund** [*] **K. Soltesz** [*]

[*] *Lund University, Dept. Automatic Control, Lund, Sweden (email: {frida.norlund, tore.hagglund, kristian.soltesz}@control.lth.se)*
[**] *Boliden AB, Boliden, Sweden*

Abstract: It is common that PID control loops function satisfactorily most of the time, but that they have issues with violating input, state, or output constraints. While MPC solves this problem in principle, it is in practice not straightforward to replace a functioning PID controller with an MPC implementation. This is particularly true for loops that are critical to plant operation, where stops associated with controller replacement and tuning can turn out costly. We propose a novel configuration where an MPC controller is designed based on the PID closed-loop, and provides a feed-forward term to its control signal. This configuration enables a smooth transition between unconstrained PID operation and MPC-dominated control, governed by a single scalar parameter α. We demonstrate the approach using a relevant process industrial application: flotation level control in mineral enrichment. Using a practical example, we show how our approach results in a structure where the PID control loop can be kept, while leveraging the constraint-honoring advantage of MPC control.

Keywords: Proportional-integrating-derivating (PID) control, Model predictive control (MPC), Feed-forward, Mining flotation.

1. INTRODUCTION

In the process industry, plants are traditionally controlled in a decentralized manner using sometimes thousands of PID controllers. These PID controllers are interconnected in an advanced architecture that is often developed over several decades, see Shinskey (1996); Åström and Hägglund (2006). Above this structure, it is becoming more and more common to use optimization routines to optimize the plant's operation. The most common method is to use model predictive controllers (MPC) that provide setpoints to underlying PID controllers. In many cases, these MPC controllers have also been used to replace parts of the control that were previously handled by PID controllers.

The PID controller has the advantage that it is easy to tune manually by plant personnel, and the control becomes good enough if the plant dynamics are not too complicated. A drawback is that it does not manage to handle constraints efficiently, except simple limitations on the controller output using anti-windup, see Åström and Hägglund (2006); Visioli (2006).

The MPC controller has the advantage that it can handle more complicated constraints, not only on the controller output but also on states and other signals in the process. A drawback is that it is more difficult to tune and maintain the MPC controller, and it requires a model of the process, see Rawlings et al. (2017).

In this paper, it is suggested to combine the two controllers, PID and MPC, to utilize the advantages of the two. Rather than letting the MPC govern the setpoint to the PID as mentioned earlier, the MPC controller is connected to a PID loop via a feedforward control signal. The MPC controller considers the closed loop in its optimization and how active the additional control signal is will be determined via a weighting factor. In this way, one can both influence the control in normal operation and utilize the MPC controller's ability to handle advanced constraints. As opposed to the structure where the MPC sets the reference to the PID—and key to control performance—this novel structure gives the MPC direct access to the process input.

2. FLOTATION CONTROL

To introduce our PID–MPC hybrid, we will make use of an industrially relevant control scenario. The process step that separates valuable minerals from waste rock in a copper concentrator is called flotation, see Shean and Cilliers (2011). Flotation utilizes differences in surface properties of minerals to separate them. In the concentrator, ore is milled to a fine sand that is mixed with water to form a slurry. This slurry is pumped to the flotation process that consists of a series of tank cells, see Figure 1. Chemical reagents are added to make the desired minerals water-repellent. This way the desired minerals will float to the surface, attaching to air bubbles generated at the bottom of the cells as they rise through the slurry. The mineral froth formed on top of the cells is collected as it flows over the rim of the cells. This makes level control of the cells a cornerstone to achieve high mineral recovery.

The inflow to the flotation series is mostly slowly varying. This makes PID control satisfactory, and it is the most

* This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. All authors are members of the ELLIIT Strategic Research Area.

Fig. 1. Photo of a part of the flotation process in the Aitik concentrator, located near Gällivare, Sweden. Photo credit: Jonas Westling.

common choice for flotation level control. However, there are occasional variations caused by for example brief stops of the upstream milling lines. These abrupt changes can cause the levels in the cells to deviate so much from their references that the froth layer is destroyed, disrupting the yield. It may also cause flotation cells to overflow, causing a lot of tailing material to end up in the next flotation step.

The constraint-handling capability of MPC could resolve the aforementioned issues. However, the flotation process is critical to plant operation, and replacing a functioning PID control solution with an MPC is in practice associated with temporary performance degradation during a tuning phase and will require model maintenance to keep a good performance. The robustness of the PID controller is hence an advantage during normal operating conditions, while the constraint-handling properties of the MPC controller are desirable during abnormal operating conditions.

3. COMBINING PID WITH MPC

3.1 MPC feed-forward to the PID loop

Based on section 2, a smooth transition between PID and MPC, where the performance of the PID is kept while the MPC ensures the honoring of constraints, would be valuable. Here we present the, to our knowledge, first method that achieves this. Furthermore, this smooth transition is governed by a single scalar parameter α, that the user can dial to seamlessly transition between PID and MPC-dominated control. Particularly, this smooth transition can also be used when the end goal is to replace a PID loop with an MPC controller. The transition can be done gradually and the PID does not have to be removed to leave place for the MPC controller, making the introduction of the new controller less critical for operation.

Instead of designing an MPC for the process model P, the basic idea underlying our approach is to design an MPC for the closed-loop G, shown as a gray box in Figure 2, where r is the reference, C is the PID controller with control signal v, P is the process to be controlled and y is the process output. Loads are modeled by the state disturbance d and may, or may not, be measured (or estimated).

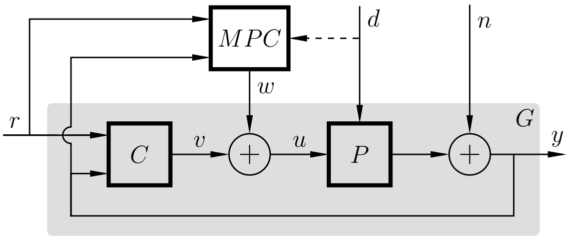

Fig. 2. Block diagram illustrating a feedback SISO loop with PID controller C, process (model) P, reference r, process load disturbance d, measurement noise n, and process output y. The process input $u = v + w$ consists of the PID control signal v and the MPC control signal w. The closed-loop system in the gray box, denoted G, is the process considered in the MPC design. Note that the PID controller is an internal component in G.

Instead of providing a control signal that acts as the process input u, the MPC controller in our approach outputs a control signal w, that is added to the nominal control signal v, provided by the PID controller.

If w is penalized so that its signal energy (2-norm) is small compared to that of v, the total control signal $u = v + w$ will be similar to that from the PID controller, and there will essentially only be activity in w when the MPC predicts constraint violations. The extreme case of $w = 0$ corresponds to the nominal PID loop with no constraint honoring. If, instead, the energy in w is constrained less, the MPC will dominate u, and in effect override the PID controller.

3.2 Mathematical formulation

The mathematical formulations in this paper will be done in continuous time. However, for practical implementation, a corresponding discrete time formulation is needed. The code generating all examples in this paper uses a 1 s sampling period, and is available through Norlund, F. (2023).

To arrive at a system model for the closed-loop G, we start by expressing its components on state space form[1]. The plant model is expressed in state space form as

$$P : \begin{cases} \dot{\boldsymbol{x}}_p = A_p \boldsymbol{x}_p + B_p u + B_{pd} d, & \text{(1a)} \\ y = C_p \boldsymbol{x}_p + D_p u + D_{pd} d, & \text{(1b)} \end{cases}$$

where the disturbance terms corresponding to any non-measurable disturbances are omitted from the MPC design.

The PID controller is expressed in state space form as

$$C : \begin{cases} \dot{\boldsymbol{x}}_c = A_c \boldsymbol{x}_c + B_{cr} r + B_{cy} y, & \text{(2a)} \\ v = C_c \boldsymbol{x}_c + D_{cr} r + D_{cy} y. & \text{(2b)} \end{cases}$$

[1] In this paper attention is delimited to single-input single-output (SISO) processes, but not necessarily with a scalar state. We will therefore write $\boldsymbol{x}_p$ instead of x_p, when introducing the methodology, to indicate that the state could be a vector. Throughout we use boldface for vectors, capital roman letters for matrices, and lowercase roman letters for scalars. When not needed for unambiguity, we omit the time argument, and for example write y instead of $y(t)$.

Combining (1), (2), and that, from Figure 2,

$$u = v + w, \tag{3}$$

we get

$$\begin{aligned} \dot{\boldsymbol{x}}_p &= A_p\boldsymbol{x}_p + B_p(v+w) + B_{pd}d, & \text{(4a)}\\ \dot{\boldsymbol{x}}_c &= A_c\boldsymbol{x}_c + B_{cr}r + B_{cy}y, & \text{(4b)}\\ v &= C_c\boldsymbol{x}_c + D_{cr}r + D_{cy}y, & \text{(4c)}\\ y &= C_p\boldsymbol{x}_p + D_p(v+w) + D_{pd}d. & \text{(4d)} \end{aligned}$$

Multiplying (4c) by D_p and adding (4d) one arrives at

$$y - D_pD_{cy}y = C_p\boldsymbol{x}_p + D_pC_c\boldsymbol{x}_c + D_pD_{cr}r + D_pw + D_{pd}d. \tag{5}$$

Introducing

$$E_c = (I - D_pD_{cy})^{-1}, \tag{6}$$

(5) can be written as

$$y = E_cC_p\boldsymbol{x}_p + E_cD_pC_c\boldsymbol{x}_c + E_cD_pD_{cr}r + E_cD_pw + E_cD_{pd}d. \tag{7}$$

Similarly one can multiply (4d) by D_{cy}, add (4c) and introduce

$$E_p = (I - D_{cy}D_p)^{-1} \tag{8}$$

to arrive at

$$\begin{aligned} v =&E_pD_{cy}C_p\boldsymbol{x}_p + E_pC_c\boldsymbol{x}_c + E_pD_{cr}r + E_pD_{cy}D_pw\\ &+ E_pD_{cy}D_{pd}d. \end{aligned} \tag{9}$$

Inserting (7) and (9) in (4a) and (4b), and introducing the combined closed-loop state vector

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_p\\ \boldsymbol{x}_c\end{bmatrix} \tag{10}$$

one arrives at the state space representation for G:

$$\begin{aligned} \dot{\boldsymbol{x}} =& \underbrace{\begin{bmatrix} A_p + B_pE_pD_{cy}C_p & B_pE_pC_c\\ B_{cy}E_cC_p & A_c + B_{cy}E_cD_pC_c \end{bmatrix}}_{A}\boldsymbol{x}\\ &+ \underbrace{\begin{bmatrix} B_pE_pD_{cr}\\ B_{cr} + B_{cy}E_cD_pD_{cr} \end{bmatrix}}_{B_r} r + \underbrace{\begin{bmatrix} B_p + B_pE_pD_{cy}D_p\\ B_{cy}E_cD_p \end{bmatrix}}_{B_w} w\\ &+ \underbrace{\begin{bmatrix} B_{pd} + B_pE_pD_{cy}D_{pd}\\ B_{cy}E_cD_{pd} \end{bmatrix}}_{B_d} d, \end{aligned} \tag{11a}$$

$$\begin{aligned} y =& \underbrace{\begin{bmatrix} E_cC_p & E_cD_pC_c \end{bmatrix}}_{C}\boldsymbol{x} + \underbrace{E_cD_pD_{cr}}_{D_r} r + \underbrace{E_cD_p}_{D_w} w\\ &+ \underbrace{E_cD_{pd}}_{D_d} d, \end{aligned} \tag{11b}$$

which can be compactly written as

$$G : \begin{cases} \dot{\boldsymbol{x}} = A\boldsymbol{x} + B_rr + B_ww + B_dd, & \text{(12a)}\\ y = C\boldsymbol{x} + D_rr + D_ww + D_dd. & \text{(12b)} \end{cases}$$

This state space representation can now be used to design the MPC controller that generates the control signal component w.

3.3 PID controller model

A common formulation of the PID controller is

$$v(t) = K\left(\beta r(t) - y(t) + \frac{1}{T_i}\int_0^t r(\tau) - y(t)\ dt - T_d\dot{y}(t)\right), \tag{13}$$

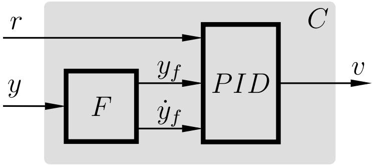

Fig. 3. Block diagram of the controller C. A filter F, with measurement (process output) y as input; filtered measurement y_f and its derivative $\dot{y}_f$ as outputs. The PID controller takes the reference value r as input as well as the outputs from the filter. Its output is the control signal v.

where K, T_i, T_d and β are controller parameters. See Hägglund (2023) for details concerning (13) and its parameters. The input to the controller is usually not the measurement y itself, but rather the filtered signal y_f. Here, a filter with Laplace domain representation

$$Y_f(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}Y(s) \tag{14}$$

is employed. Being second-order, the filter provides high-frequency roll-off even with derivative action present in the controller. With the state selection

$$\begin{bmatrix}\dot{y}_f\\ \ddot{y}_f\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega^2 & -2\zeta\omega\end{bmatrix}\begin{bmatrix}y_f\\ \dot{y}_f\end{bmatrix} + \begin{bmatrix}0\\ \omega^2\end{bmatrix}y, \tag{15}$$

the filter will output both the filtered measurement y_f, and the filtered derivative $\dot{y}_f$. The filter outputs are then fed as inputs to the PID controller as demonstrated in Figure 3. Choosing the state vector

$$\boldsymbol{x}_c(t) = \left[\int_0^t (r(\tau) - y_f(\tau))\ d\tau \quad y_f(t) \quad \dot{y}_f(t)\right]^\top, \tag{16}$$

the combined state space representation of the filter in (15) and the PID controller in (13) can be written as

$$\dot{\boldsymbol{x}}_c = \underbrace{\begin{bmatrix}0 & -1 & 0\\ 0 & 0 & 1\\ 0 & -\omega^2 & -2\zeta\omega\end{bmatrix}}_{A_c}\boldsymbol{x}_c + \underbrace{\begin{bmatrix}1\\ 0\\ 0\end{bmatrix}}_{B_{cr}} r + \underbrace{\begin{bmatrix}0\\ 0\\ \omega^2\end{bmatrix}}_{B_{cy}} y, \tag{17a}$$

$$v = \underbrace{K\begin{bmatrix}\frac{1}{Ti} & -1 & -T_d\end{bmatrix}}_{C_c}\boldsymbol{x}_c + \underbrace{K\beta}_{D_{cr}} r. \tag{17b}$$

Note that this representation of the controller is consistent with (2), with $D_{cy} = 0$, and it can hence be used to represent C in G of Figure 2 when designing the MPC.

3.4 MPC design

In our setting, the MPC cost J over a horizon, h, is expressed as a trade-off, defined by the scalar parameter $\alpha \in [0, 1]$, between penalizing control error (state error) and energy of the MPC feed-forward signal w:

$$J(\alpha) = \int_0^h (1-\alpha)\left(r(t) - y_f(t)\right)^2 + \alpha w^2(t)\ dt. \tag{18}$$

The control signal is then determined by minimizing J over h, subject to the system dynamic and the constraints. In our example, we will utilize only a state constraint. However, constraints on the control signal u, rate of change of the control signal, and indeed any linear combination of

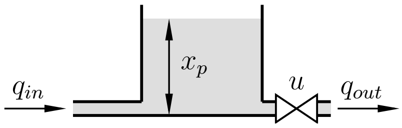

Fig. 4. Schematic illustration of the level control process in a single flotation cell. The slurry level x_p constitutes the sole state variable, and it is controlled by u that varies the position of an outflow valve, to compensate for changes in the (not directly controlled) volumetric inflow rate q_{in}.

states and other signals internal to G could be imposed on the solution.

The control horizon, h_c, specifies until which time w is allowed to change value. This parameter is usually set shorter than h to keep the minimization problem small.

4. FLOTATION EXAMPLE

Level control of a single flotation cell will serve as an example for the control structure introduced in section 3. The process is schematically depicted in Figure 4. The state is the slurry level, denoted x_p [cm]. The volumetric outflow rate q_{out} [cm^3/s], is controlled by a valve with control signal u. The inflow, q_{in} [cm^3/s], is the outflow from the previous cell and is hence not controlled by the current cell, making q_{in} a disturbance, d, in (1). The process model in this example will be a linearized flotation cell, and it can be described as[2]

$$\begin{cases} \dot{x}_p = -0.0218x_p + 0.0521u - 3.54 \cdot 10^{-6}d & \text{(19a)} \\ y = x_p, & \text{(19b)} \end{cases}$$

which is consistent with (1). For further details on the process modeling, see Norlund, F. (2022).

The controller parameters for the PID controller are chosen according to the parameters for the PI controllers that operate the real plant, making C a PI controller with $K = 0.9$, $T_i = 87$ s, $T_d = 0$ s, and $\beta = 0.7$. The filter parameters are designed to have critical damping, $\zeta = 1/\sqrt{2}$, and $\omega = 100 \cdot 2\pi/T_i$.

When designing the MPC, we introduce the constraint $x_p \geq 450$ cm as a lower limit. The upper limit is, of course, more relevant in practice, but since violating it results in overflowing the cell, the effects are harder to visualize, therefore the example will only focus on the lower limit. The MPC control signal is then determined by solving

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & J(\alpha), \\ \text{subject to} \quad & G, \\ & 450 \leq x_p. \end{aligned} \tag{20}$$

The flotation cell has a typical settling time of about 150 s, and this is chosen as the prediction horizon, h. The control horizon, h_c is chosen to 50 s.

When it comes to choosing α in the MPC controller, different choices will give different controller behaviors. Three different tunings will be highlighted in the examples, $\alpha = \{1, 0.33, 0.1\}$. These three α-values were chosen since the resulting controllers showcase three distinct behaviors. When $\alpha = 1$, the MPC only penalizes the control signal w. It will hence only act when absolutely necessary to avoid violating constraints. When $\alpha = 0.1$, the MPC controller has a significantly bigger weight on the output deviation and will be active to control the output even for small deviations. Choosing $\alpha = 0$ would seem like the natural choice to demonstrate the extreme case, but it results in an extremely noise sensitive controller, since it does not penalize control signal activity at all. We found that for our example, $\alpha = 0.33$, results in a balanced behavior in-between those for $\alpha = 1$ and $\alpha = 0.1$.

To demonstrate the different controller behaviors, a step disturbance in the inflow to the flotation cell is simulated. Figure 5 shows the case when the disturbance is measurable. The vertical line in the figure indicates when the inflow changes, it decreases by 25 %. All hybrid controllers keep the level within the allowed range, but how fast it returns to its reference after the disturbance differs. When $\alpha = 1$, the MPC is only active when absolutely necessary to keep the level within the bounds. For $\alpha = 0.33$, it becomes active as soon as the disturbance appears to reduce the deviation, and then as the state returns to its reference, the MPC contribution returns to 0. Both for $\alpha = 1$ and $\alpha = 0.33$, the MPC uses relatively small contributions from its control signal, w, to reduce the disturbance effects. When w is not needed, it is not used, and C controls the plant alone. When $\alpha = 0.1$, w is used to follow the level reference, and for this controller, w is a large part of the total control signal u. With this tuning, the PI-controller instead contributes less to the total control signal.

In the real plant, however, the inflow disturbance is not measured. This case is shown in Figure 6. The behavior is similar to that shown in Figure 5, but with some differences: The level is not kept within its bounds by all the MPC-tunings. The constraint is violated when $\alpha = 1$. When the disturbance is unknown, this controller will not act on it until the constraint is hit, at which point it is too late to entirely avoid a constraint violation.

So far, no noise has been present in the simulations. Figure 7 shows the same step disturbance in the inflow as in Figure 5 and Figure 6, but a representative measurement noise, n, has been added to y. This noise model is driven by white noise and it is further described in Norlund, F. (2022). The controller behavior is very similar to the one in Figure 6 when $\alpha = 0.33$, while the control signals for $\alpha = 1$ and $\alpha = 0.1$ are much more responsive to noise.

Another most relevant imperfection to consider when working with model-based controllers, such as MPC, is model errors. In Figure 8, the process model P, in G of Figure 2, is altered by introducing control signal (u) gain errors as scalar factors of 0.5 and 2 that multiply the plant input matrix B_p in (1). The effect of the model error is most visible when $\alpha = 1$, where the response in w to the disturbance is affected. When $\alpha = 0.33$ and $\alpha = 0.1$, the effect of the model error is barely visible.

[2] In this representation, x_p is really the froth thickness, which is the measurement given in the real system. However, in the results we will instead visualize the slurry level, given by the cell height minus the froth thickness, since it is more intuitive for a reader who is not familiar with the process.

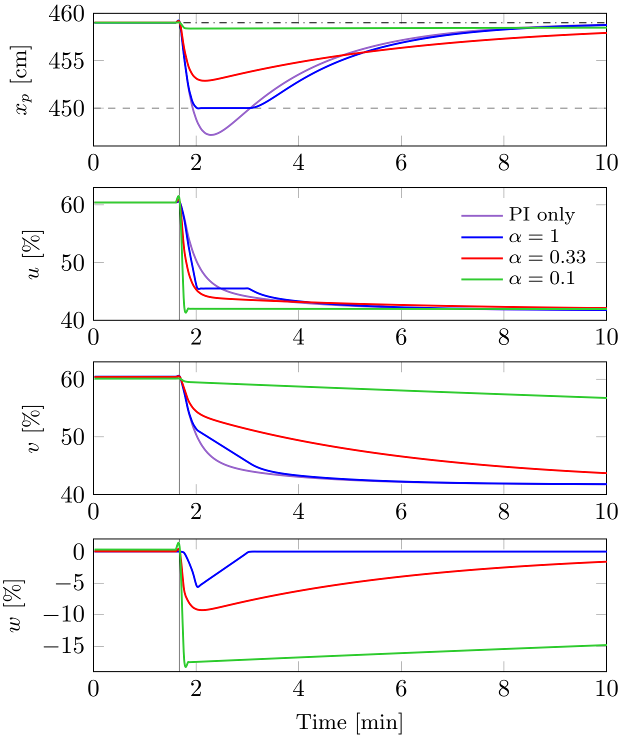

Fig. 5. Simulation of load disturbance response where the penalty for state deviations in the MPC is varied. The load disturbance is a step-shaped inflow disturbance that reduces the inflow by 25 %. This event is marked with the black vertical line. The dash-dotted and dashed horizontal lines in the topmost subplot indicate the the reference value, and the level constraint for x_p, respectively. The disturbance is measured and used as an input to the MPC controller.

5. DISCUSSION

The main difference between Figure 5 and Figure 6 appears for $\alpha = 1$, corresponding to the only controller that hits the constraint. With known disturbance (Figure 5), the corresponding MPC contribution w just keeps the level within the constraints, and the MPC starts acting before the constraint is hit. When the disturbance is unknown (Figure 6) and $\alpha = 1$, the MPC does not start acting until the constraint is hit, leading to a slight constraint violation. With $\alpha = 1$, the MPC controller does not drive the state to its reference, but only acts to satisfy constraints with minimal energy expenditure in w.

The other extreme is represented by $\alpha = 0.1$. It has a relatively small energy penalty on w, and will hence drive the process state to its reference faster. Consequently, the PID controller is less active. However, the low penalty on w also makes this tuning more sensitive to measurement noise, as seen in Figure 7.

The balanced tuning $\alpha = 0.33$ results in a practically more viable alternative, with comparable penalties on state error $r - x_p$ and MPC control signal w. The controller drives the state to its reference, while also taking control signal effort into account. This is demonstrated in Figure 7, where the MPC with $\alpha = 0.33$ barely reacts to measurement noise, while allowing the PID to act on smaller state errors.

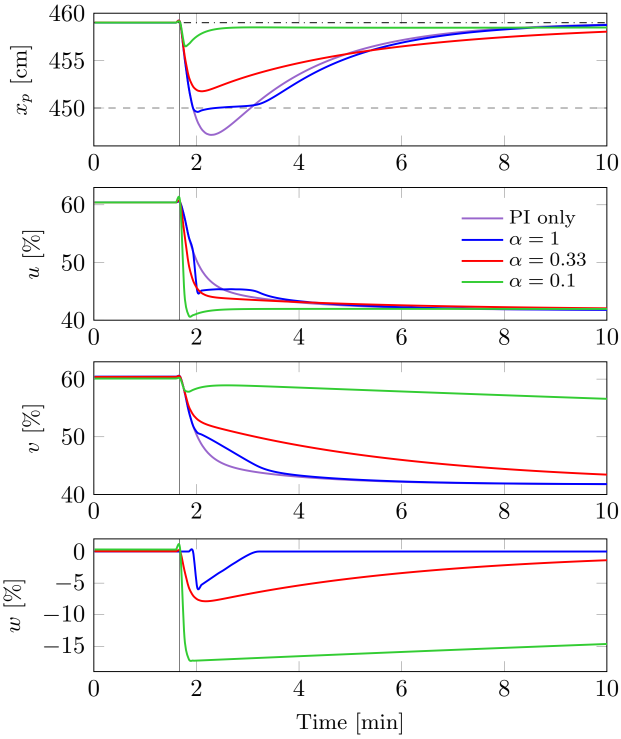

Fig. 6. Variation of the simulation of Figure 5, where the inflow disturbance is not measured, and hence not known by the MPC.

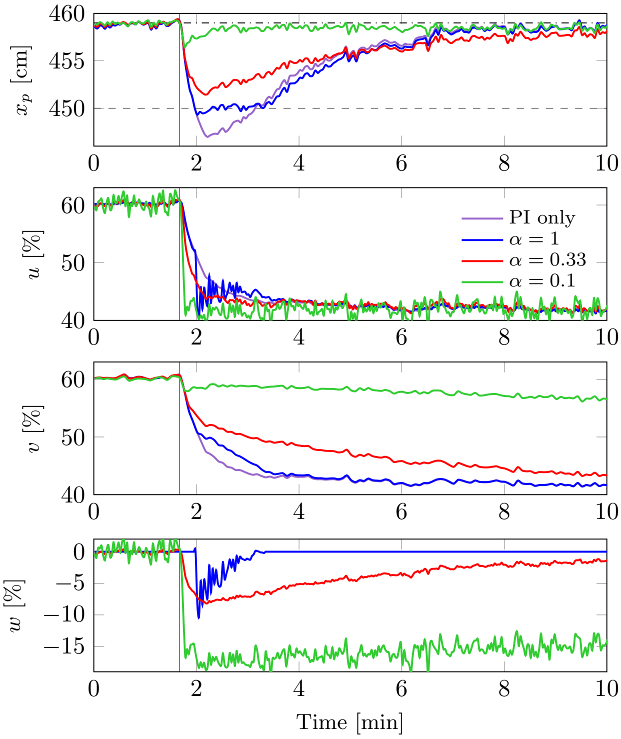

Fig. 7. Variation of the simulation in Figure 6, where the inflow disturbance is not measured and where a representative measurement noise n has been added to the process output.

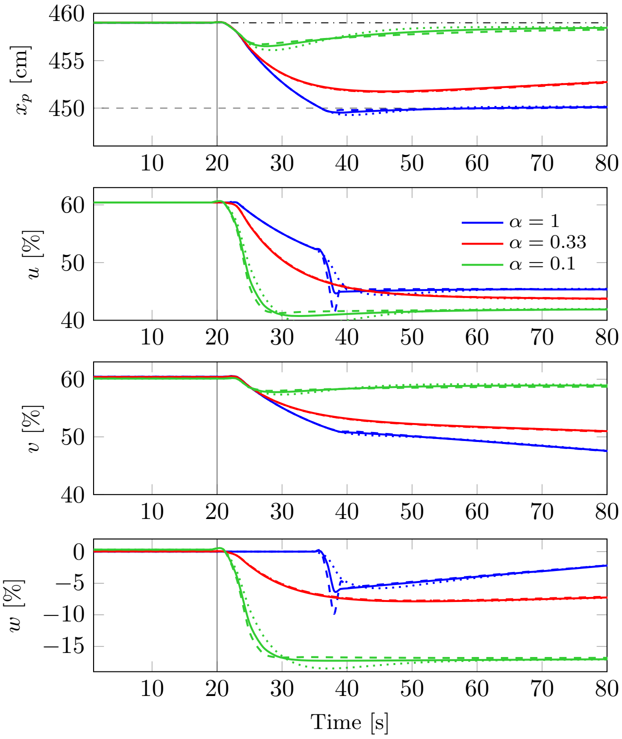

Fig. 8. Variation of the simulation in Figure 6, where the inflow disturbance is not measured and where the process model in the MPC differs from the actual process. The plant input gain B_p of (1) is reduced to half (dashed), and doubled (dotted). The model-error-free case is shown in solid.

When the load disturbance occurs, and the deviations become larger, the MPC acts to compensate for it. The behavior for $\alpha = 0.33$ is similar for the cases with and without noise, shown in Figure 6 and Figure 7.

For $\alpha = 0.33$ a small error in x_p remains once the state has been brought back close to its reference. In the example of Figure 6, the PID contribution v continues to decrease while its MPC counterpart w increases at the same pace, reducing the control signal deviation for w. Once $w = 0$ again, the small x_p error is also controlled to 0.

Model errors were modeled as multiplicative input gain errors, since input gain is the component of the real plant model with the largest uncertainty. Even with B_p differing by -50 % and $+100$ % in the model compared to the process, the controllers perform well, as shown in Figure 8. The difference is barely visible for $\alpha = 0.33$ or $\alpha = 0.1$. For $\alpha = 1$, w is affected by the model error, but its effect on x_p remains small.

There are also some conceptual aspects of the hybrid design that we find worth discussing. The fact that MPC action can be gradually introduced provides a means to deploy an MPC without disrupting operations, otherwise associated with changing from PID to MPC. A related advantage is that even with the MPC in place, the PID controller remains. In for example production plants, it is common that the PID is a local controller, whereas the MPC would be implemented in a remote computer. Should the connection to this computer be lost, the hybrid architecture of Figure 2 provides a fallback to PID control.

6. FUTURE WORK

In the near future, we are expecting to evaluate the solution outlined in this paper in production at Boliden's Aitik mine. This comes with a few practical considerations. For example, integrator anti-windup of the PID controller needs to be explicitly addressed. This can be done by also imposing constraints on $u = w + v$, using the same methodology that led to constraining $\boldsymbol{x}_p$ in section 3.

As long as the plant model and controller are both modeled by linear and time-invariant (LTI) systems, the methodology introduced above can be readily applied, and extension from the herein demonstrated single-input single-output (SISO) process model case to the multiple-input multiple-output (MIMO) case, is straightforward.

Currently, there is a successful pilot project in which an entire flotation bank is under linear-quadratic (LQ) control. This would serve as a suitable demonstrator for the hybrid strategy in a MIMO scenario, where the controller and process model are still LTI.

Another natural extension consists of applying the hybrid strategy in scenarios where the full (process) state cannot be measured. In such scenarios, some type of state observer would be required to operate the MPC, with the Kalman filter constituting a natural candidate in the LTI case.

It is of course also possible to employ the hybrid strategy in cases where the process model or the "inner" closed-loop controller is nonlinear. In such applications, the MPC problem will become nonlinear, and not take on the form of a quadratic program (QP), as in the LTI case.

We furthermore anticipate to apply the solution in contexts outside of mining. PID controllers are found in a multitude of applications, and as in industrial process control, state and input constraints often pose practical challenges that are commonly addressed by de-tuning controllers, trading performance for practical feasibility.

REFERENCES

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, Research Triangle Park, North Carolina.

Hägglund, T. (2023). *Process Control in Practice*. De Gruyter, Berlin, Germany.

Norlund, F. (2022). *Comparison of Level Control Strategies for a Flotation Series in the Mining Industry*. Master Thesis Report TFRT-6178. Lund University, Department of Automatic Control.

Norlund, F. (2023). pid-mpc. https://github.com/fridanorlund/pid-mpc. Commit: 8a0fe9f.

Rawlings, J., Mayne, D., and Diehl, M. (2017). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Santa Barbara, California.

Shean, B. and Cilliers, J. (2011). *A Review of Froth Flotation Control*. International Journal of Mineral Processing, 100(3), 57–71.

Shinskey, F.G. (1996). *Process-Control Systems. Application, Design, and Tuning*. McGraw-Hill, New York, fourth edition.

Visioli, A. (2006). *Practical PID Control*. Springer, Berlin.