Proceedings of the 2025 American Control Conference. ©2025 AACC.

Different Perceptions of PID Control in the Mechatronic and Process Control Worlds

Daniel Y. Abramovitch*

Abstract— **It is not uncommon for graduate students on the mechatronics side of the control world to treat the Proportional plus Integral plus Derivative (PID) controller with a certain amount of disdain. This is not surprising since most control texts from this end of the control world treat PIDs as simple, basic structures, to be quickly replaced by more advanced methods. To that end, these texts devote only a handful of pages to the subject. It seems that – at least in the mechatronics world – PIDs are considered too simple for much interest in academia while practicing engineers do not seem to care why they were working.**

This is a far cry from the treatment of PIDs in the chemical and bio-process control worlds (CPC and BPC, respectively). At this end of the control spectrum, PID controllers are studied in far more depth obtaining entire books or book series. Despite this volumetric expansion of material, it seems that in the latter worlds, many of the issues and concerns one sees in the mechatronic world are treated as obscure corner cases. Depending upon the teaching text, issues of sampling and digital representation may have been completely omitted. There were other surprises. While PIDs were almost universal and standard, they were almost never unified or standardized. Furthermore, what seemed to limit performance was not the structure of the controller itself, but the lack of accurate system/process models based on repeated physical system measurements.

However, the mechatronic and process PID goals and foibles were not that different once one considered the different system, time constant, and measurement constraints. We will discuss these issues with the goal of getting a more unified view of PIDs across our application domains. We will provide a handful of common PID forms and show how they are related, so that we can approach any PID structure with the same analytical approach. We will finally look forward to how PIDs can be used, not only as a fundamental teaching tool for explaining control outside of our research circles, but as a critical component for advanced control methods.

I. Motivation: Framing the paper

Proportional plus integral plus derivative (PID) control is treated very differently in the mechatronic and process control worlds. This can be seen in how the topic is discussed in textbooks from the different areas. In the electromechanical/mechatronic based control books PIDs are typically relegated to a few pages or a small section in one chapter. We refer the reader to the classic texts by Ogata [1] or Franklin, Powell, and Emami-Naeni [2] as well a host of others [3], [4], [5]. It is not even mentioned in [6] or [7]. By a later edition of Ogata's *Modern Control Engineering* [8], it gets a few more pages but most of those are in the problem section of the chapter. In the classic book by Kuo [9] there is a section on PID control, and later on a discussion of discretization (using the backwards rectangular rule). Curiously, their first example is of a process which includes an integrator, and so they choose to only use PD control. In Kuo's book on digital control [10], there are again a few pages on digital PID control. Here the author chooses to discretize the integrator with a trapezoidal rule, but uses the backwards rectangular rule for discretizing the derivative term. Perhaps the deepest treatment (and signs of work to come) are in [11] where the topic gets a full 20 pages, [12] with 18 pages, [13] and [14] where it gets its own 18-page and 25-page chapters, respectively. Two books on writing software for control, specifically of mechanical systems [15], [16] spend only a few pages on PID controllers, despite having a focus on practical implementation of controllers.

*Daniel Y. Abramovitch is with Agilent Technologies, Santa Clara, CA USA (abramovitch@ieee.org)

On the other hand, the process control view often has entire books [17] (or series) [18], [19], [20] on PID controllers. Even when the book is not specifically about PID controllers, but instead about process control, the texts feature multiple chapters on discussions of PID control [21], [22], [23], [24], [25], [26]. Furthermore, the descriptions and use of PID methods permeate through these entire texts. A recent paper by Hägglund and Guzmán went so far as to be entitled, *Give us PID controllers and we can control the world* [27].

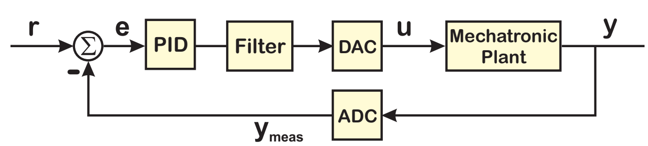

Fig. 1. A mechatronic control loop with a PID and filter implemented in discrete time. The PID handles the rigid body/baseband portion of the plant response while the filter is in place to equalize the resonances and anti-resonances of the physical system.

Looking into these texts, the discrepancy in the treatment of PID controllers goes further. In the process control texts, discretization is rarely discussed while it prominently features in any mechatronic digital control book (if PID controllers are discussed there at all). There are other fundamental discrepancies, such as how the PID parameters are specified and how the plant is characterized. The net result is that anyone trying to learn about PID control for the first time goes down one of two divergent paths. We offer the conjecture that as the field was emerging different dominant models in mechatronic systems versus process systems affected these views. Accepting that, can we return to common principles and see which ones apply in any situation? If we can, then we have a more common framework for discussing

PID control, both amongst ourselves and with scientists and engineers working adjacently to the control field.

For this author – emerging with graduate school training that barely referenced PID controllers – it seemed as if these were relics of the simple past. System identification was to be done with regression on discrete-time transfer function parameters in some sort of ARMA, ARX, ARMAX, etc. form. These would then be mapped to state-space canonical forms where all the modern control tools could be brought to bear. However, the world of industrial research showed that these methods often failed on lightly damped mechatronic systems. Instead, one encountered practical solutions that included some combination of a controller (perhaps a PID or one of its close relatives) plus some filters, as shown in Figure 1. The absence of modern control methods in these applications continues to this day. By the same token, the prevalence of PID controllers in industry should make one ask why such a supposedly under-powered and outdated method still powers a vast majority of industrial control loops.

This paper will explore these differences and propose a historical basis for these very different perspectives. That being done, the aim is to return discussions of PID control back to a more common framework. Within the controls community, the lack of a common framework might seem like different dialects of the same language. We can translate between frameworks with enough math (although often we do not). However, in industry or with co-workers who are not control engineers, a change of framework is often viewed as something fundamental, not simply a "control dialect". (How would a non-expert recognize these differences?) This creates a serious issue in outreach, since many scientists and non-control engineers understand feedback control only as a PID controller (even when the controller itself may be doing far more).

If we wish to restore and even amplify the influence of feedback control principles beyond our community, to be at the center of discussions on automation and machine intelligence, we need to make it easier for scientists and engineers outside our community to embrace the principles. This must be the case even if they cannot fully internalize the math. Unifying inconsistent frameworks on the most ubiquitous control methodology is one start.

II. The Lessons of Phase-Locked Loops

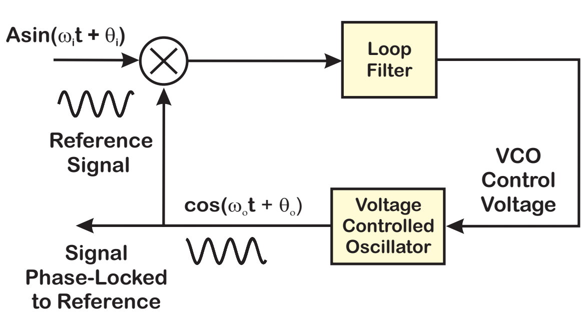

Fig. 2. A classical mixing phase-locked loop (PLL).

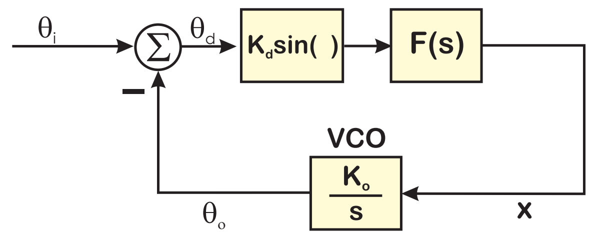

Fig. 3. The baseband model of a classical mixing PLL.

Perhaps the most ubiquitous human-built feedback loop is the phase-locked loop (PLL). This is an electronic synchronization device that has its origins in the 1930s [28] but now ends up in every digital watch, phone, tablet, computer in our lives. It is fundamentally responsible for keeping timing in almost all electronic circuits (both analog and digital) [29]. A classical mixing PLL is diagrammed in Figure 2. With a few trigonometric identities and some low-pass filters not shown in most PLL texts, this can be reduced to the baseband model shown in Figure 3. This is still a nonlinear model, but the sine is a sector 1-3 nonlinearity [29] which is smooth and well behaved. More importantly for our discussion here, the voltage-controlled oscillator (VCO) which converts a voltage into an oscillation frequency is modeled as an integrator (to get from frequency back to phase), shown in (1).

$$P(s) = \frac{K_0}{s} \tag{1}$$

Because the plant is an integrator, one does not need to include an integrator in the controller – in this case the loop filter, $F(s)$ shown in Figure 3. However, it is usually desirable to track slow changes in frequency and this can be viewed as a ramp input to the phase detector. For this PLL designers usually add a second integrator (2),

$$F(s) = \frac{s+b}{s} \Rightarrow OL(s) = \frac{K_d K_o (s+b)}{s^2} \tag{2}$$

which makes the controller a PI controller and the open-loop response into a double integrator at low frequency and an integrator at high frequency. Because the plant can be modeled as an integrator, the control is usually straightforward and the real effort in PLL design is in designing creative phase detectors and low-drift VCOs. A look at most PLL circuit diagrams will also reveal a lot of filters that do not show up in the PLL block diagrams. These are simply considered part of what is needed to turn the loop into one that is suitable for a simple control law.

PLLs teach several immediate lessons about practical feedback loops that generalize to many other loops. The first is that practicing engineers may work with complex systems, but they tend to make the feedback loops in those systems simple. These simple loops maintain intuition and therefore allow one to debug the system. The second lesson is that these simple loops are almost always low order; first or second when possible. In the case of PLLs second order loops, in which the loop filter (controller) contains an integrator are most prevalent for reasons discussed above. This means that – whether they are described as such or not in the PLL texts, the controllers are proportional plus integral (PI) controllers.

The next lessons are a bit more obscure, but there. The third one is that even when the signals are not sinusoids (as in for digital communication, digital circuit timing, or speed control applications), the baseband (modulation domain) analysis eventually looks like an intuitive argument for the sector 1-3 nonlinearity and second-order system analysis of the above PLLs. Our final lesson comes from actual PLL circuits [29] where there are many filters not described in the texts. From a modern control perspective, we might want to consider these filters a response to the full state dynamics of the system, but we see that PLL engineers view them as removing the dynamics from the problem. In other words, even though the actual problem is higher order, these clever engineers, using a divide-and-conquer strategy, "beat" the problem into a second-order model. Finally, anywhere near crossover, the open-loop (OL) response is that of an integrator, which – having infinite gain margin and 90° phase margin, is the easiest plant to control.

These lessons from the PLL are found in many control applications. The need to have something that is robust and can be debugged leads to simplified models (or beating more complicated models into first or second-order via use of "divide-and-conquer filtering). Once the model has been reduced to such a low order the first loop closures are done with simple controllers: lag (PI), lead (PD), lag-lead (PID), and double lead. Higher order dynamics and narrow band disturbance signals – if they are modeled at all – are often filtered before they ever show up in the loop analysis. Physical understanding and intuition are key because they help us debug our system. This means that higher-order discrete-time models are either relegated to modeling and simulation, or only applied when the plant behavior is so benign that the sensitivity of the response to any one physical parameter is minimal. A positive consequence of controlling the simple plant model with a simple controller is that much can be understood about the fundamental behavior of the loop. A negative consequence is that it is often hard to migrate the simple models and designs to more complicated ones in an understandable way.

III. Why We Don't Talk: Historical Musings

A look through the examples in a classic mechatronic control book such as [1] reveals quite a few dynamic models, many for various types of electric motors. While there are many versions of these, their simplified models are of the types shown in (3) and (4).

$$P(s) = \frac{K}{s(s+a)} \tag{3}$$

$$P(s) = \frac{K(s+b)}{s(s+a)} \tag{4}$$

Similarly, the dynamics of voice coil motors as well as that of the motion of a rigid body spacecraft maneuvering in space (along any translational or rotational axis) can be described by the double integrator model of (5):

$$P(s) = \frac{K}{s^2} \tag{5}$$

These electromechanical models share something in common with the PLL plant model of (1) – they all contain an integrator in the forward path. The fact that the plant contains an integrator means that in most cases, we do not need to add it into the controller to achieve zero steady-state error to a step input. Without the need for an integrator, one need not worry about integrator windup and so there is no need to add integrator anti-windup to the controller. Without the need for integrator anti-windup there is less need to separate out the (non-existent) integrator in the controller. This frees us to consider controllers in polynomial forms (as multiplied-out filters or in state space). This means that low-order and higher order plants can be controlled with the same filter structure, simply by adding more taps. This is the form that has been so common in the electromechanical (mechatronic) world.

Another archetypal second-order mechatronic plant that should not be ignored is the classical spring-mass-damper model in which the primary mass of the system is attached to a fixed surface via spring and mass. This has the characteristic function of

$$P(s) = K\frac{\omega_d^2}{s^2 + 2\zeta_d\omega_d s + \omega_d^2}, \tag{6}$$

where ω_d is the undamped natural frequency of the denominator, ζ_d is the damping factor, and we have chosen a system with no zeros.

At the same time, the types of loops associated with process control systems, flow, pressure, level, temperature [1], [5], [24] are most often characterized with a first order or first order plus time delay (FOPTD) (sometimes referred to as first order plus dead time (FOPDT)) model [30].

$$P(s) = \frac{K}{\tau s + 1}e^{-s\Delta} = \frac{\frac{K}{\tau}}{s + \frac{1}{\tau}}e^{-s\Delta} \tag{7}$$

and τ is the time constant of the system. Sometimes a secondary lag is included which makes the process a second-order process, often called a second order plus time delay (SOPTD) or second order plus dead time (SOPD) but what is notable is the absence of an integrator in the plant model. While it is readily understood that these models are not the actual process [30], they are often considered sufficient for control tasks. These control models are low order and usually well damped (8).

$$P(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}e^{-s\Delta} = \frac{\frac{K}{\tau_1\tau_2}}{(s + \frac{1}{\tau_1})(s + \frac{1}{\tau_2})}e^{-s\Delta} \tag{8}$$

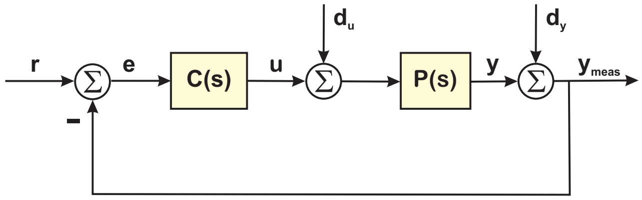

Fig. 4. A simple continuous-time loop.

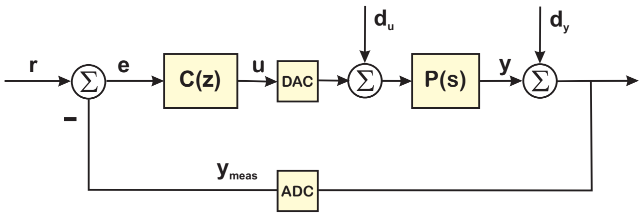

Fig. 5. Discrete-time control of a continuous plant.

IV. Consequences of an Integrator or Lack Thereof in the Plant Model

It is well known in the control community that the desire for having an integrator in the forward path of the control loop stems from the Final Value Theorem, which says that if $E(s)$ has no right-half plane poles or zeros then

$$\lim_{t \longrightarrow \infty} e(t) = \lim_{s \longrightarrow 0} sE(s). \tag{9}$$

Considering Figure 5, with a step input at the reference, r, $R(s) = 1/s$.

$$\begin{aligned} \lim_{s \longrightarrow 0} sE(s) &= \lim_{s \longrightarrow 0} s \left(\frac{1}{1+P(s)C(s)} \right) \frac{1}{s}, & (10) \\ &= \lim_{s \longrightarrow 0} \frac{1}{1+P(s)C(s)}, & (11) \end{aligned}$$

An integrator is required in the plant, P or the controller C to guarantee that the steady state error is 0 [2]. While the PLL example of Section II demonstrates the need for a second integrator to track an input ramp with zero steady-state error, tracking an input step would be more basic, and something we would want every loop to be able to do. Assuming either the P or C contains an integrator, $\tilde{P}(s)\tilde{C}(s)$ has at least one integrator factored out, so that $P\tilde{(0)}$ and $C\tilde{(0)}$ are finite.

$$\lim_{t \longrightarrow \infty} e(t) = \lim_{s \longrightarrow 0} \frac{s}{s + \tilde{P}(s)\tilde{C}(s)} = 0. \tag{12}$$

If P already contains an integrator, it is not needed in C for the closed-loop system to track an input step. With no integrator in P, it must be added to C.

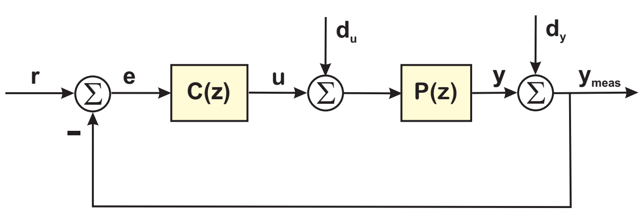

Fig. 6. A simple discrete-time loop.

The place where this argument runs into trouble is with disturbance rejection. The caveat is that different areas model the disturbances as entering at different points in the loop. We can see the significance of this by looking at any of Figures 4–6. Considering the continuous-time example of Figure 4 for simplicity, we see that

$$\begin{aligned} E(s) &= \left(\frac{1}{1+P(s)C(s)} \right) [R(s) - D_u(s)] \\ &+ \left(\frac{P(s)}{1+P(s)C(s)} \right) D_y(s). & (13) \end{aligned}$$

For steps in the reference input, or the output disturbance, d_y, the previous result holds. However, step disturbances at d_u present a different problem. If we factor out all the integrators in the plant, so that $P(s) = \tilde{P}(s)/s^k$ where k is the number of integrators in $P(s)$, then

$$\lim_{s \to 0} \frac{P(s)}{1+P(s)C(s)} = \lim_{s \to 0} \frac{\frac{\tilde{P}(s)}{s^k}}{1 + \frac{\tilde{P}(s)}{s^k} C(s)} \tag{14}$$

$$= \lim_{s \to 0} \frac{\tilde{P}(s)}{s^k + \tilde{P}(s)C(s)} = \frac{\tilde{P}(0)}{\tilde{P}(0)C(0)} = \frac{1}{C(0)}. \tag{15}$$

So, no matter how many integrators are in the plant, the controller needs an integrator to reject step disturbances at d_u. Modeling disturbances as unknown steps at the plant input is a common trait of process control problems. On the other hand, if one already has two integrators in their plant, they might be more inclined to model disturbances at the plant output or reference input, which would remove the need to have a third integrator in the forward open-loop transfer function.

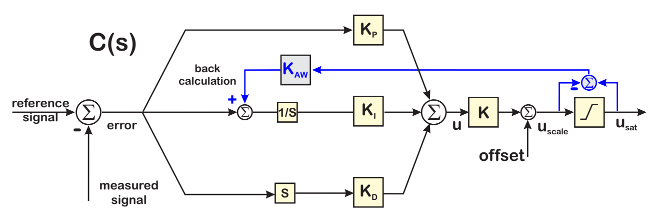

Fig. 7. Continuous-time PID with back calculation anti-windup.

Adding an integrator to a controller that may saturate brings with it the possibility of integrator windup. To avoid this anti-windup methods have become standard practice. These seem to show up very specifically in PID controllers (Figure 7) where the integrator can be treated as a separate element from the rest of the controller. If there is no controller integrator, then there is no need for integrator anti-windup, and we are free to implement our controllers in a more filter centric form. That is, our controller can now take the form of a linear filter, whether it is in a continuous-time, sample-data, or discrete-time formulation (Figures 4, 5, and 6, respectively), e.g.

$$C(z) = \frac{b_0 z^n + b_1 z^{n-1} + \ldots b_n}{z^n + a_1 z^{n-1} + \ldots a_n}. \tag{16}$$

The point is that once we no longer need to separate the integrator for anti-windup, the linear filter structure of the controller is essentially the same for a tenth-order controller as it is for a second-order controller. We spend a lot of time with algebra and linear algebra, searching for roots/eigenvalues of the characteristic equation. We may push them into forms for which we can easily adjust the roots of thees polynomials, but – especially in the digital domain – much of the physical relevance is lost. Still, there are many problems for which only knowing the general properties of the model is sufficient. When the dominant low-order model for control systems is (7), then the incessant need for an integrator in the controller, C, will emphasize anti-windup methods. Furthermore, without higher frequency dynamics, a second-order controller such as a PI or PID should be enough.

V. Regions of Control

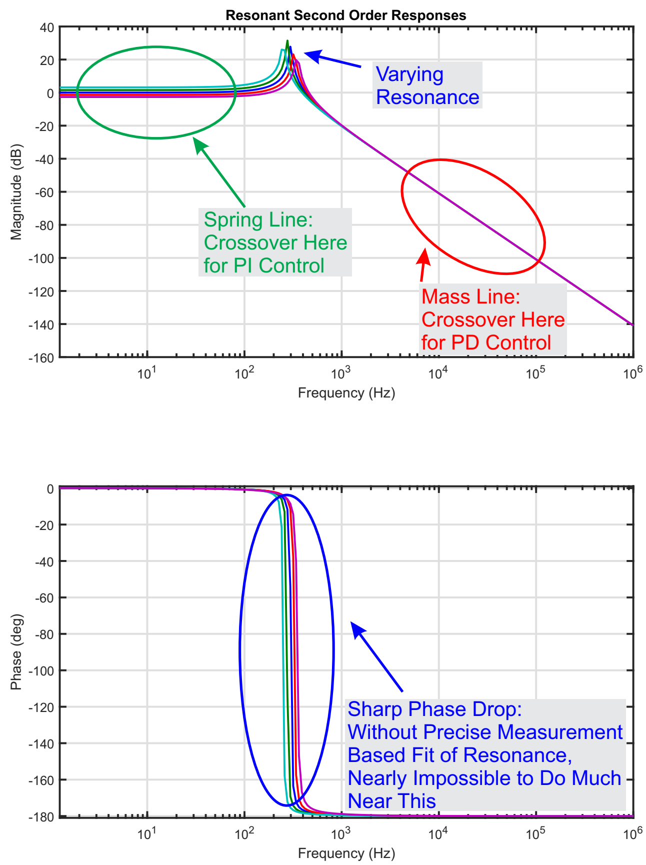

Fig. 8. The typical ranges in a mechatronic system and how they relate to proportional-integral-derivative (PID) control. If we close the loop well below the resonance, we can use PI control. If we close the loop Well above the resonance, we must engage the derivative term to get phase lead. Hence, PD control works, although the integrator is often included for steady state tracking (PID). It is only when we close the loop close to the resonance that we need a very precise model.

Figure 8 shows the ranges of classical spring-mass-damper second-order system, as described in (6). However, if we move out beyond the resonance (or if the resonance is at such a low frequency as to not be in the region of measurement), then we are dominated by the double integrator model of (5). If the loop is closed well below the resonance, a PI controller may be used. If the loop is closed far above the resonance,

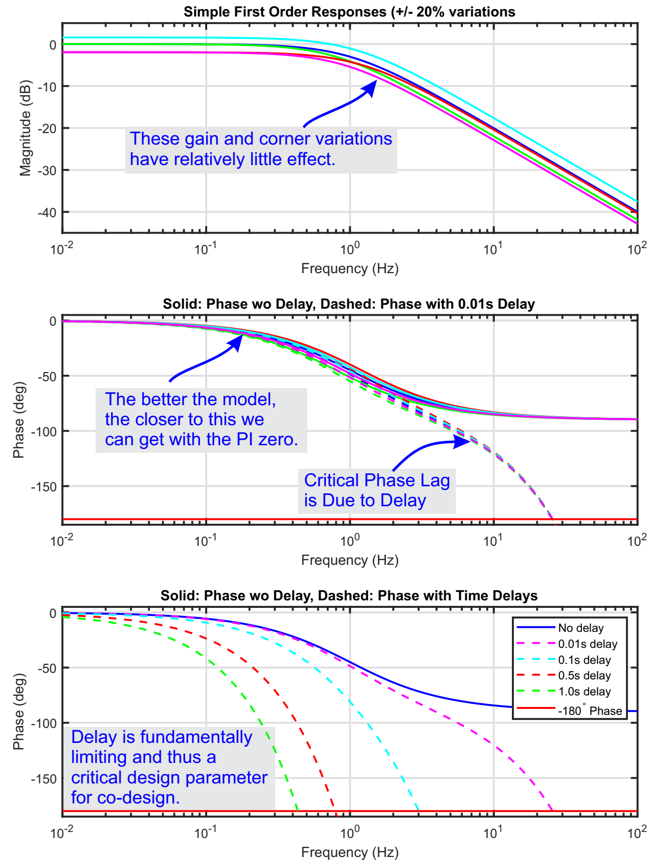

Fig. 9. The typical ranges in a process system – with a first-order-plus-time-delay (FOPTD) base model – and how they relate to proportional-integral-derivative (PID) control. The first-order portion of the plant can be addressed via a PI controller. It is the presence of time delay that brings the derivative term into play, as an attempt to mitigate some of the negative phase incurred by that delay.

a PD controller may be employed. (Even then, we may opt to add integral action to improve the steady-state error.) It is when we try to close the loop in the vicinity of the resonance that we need to carefully use all the PID coefficients. Thus, we need a far more accurate model than the previous two cases. Even a double integrator can be seen as an idealization of the case when the resonant frequency, ω_d, is well below the lowest measurement/plot frequency.

Using such a diagram can provide intuition about the choices of relative PID gains. The plots of Figure 8 and the second-order model of (6) describe simple problems for many engineers working with mechatronic systems. Simply understanding the regions and which portion of the PID applies gives some valuable intuition. Engineers working with "squishy" dynamics (for example, chemical process control (CPC), bioprocess control (BPC), thermal, and pressure problems) see a see a wholly different pervasive simple model, the first-order-plus-time-delay (FOPTD) [also called first- order-plus-dead-time (FOPDT)] system, with a characteristic transfer function of (7). While the form of the transfer function is not dramatically different from a mechatronic model, the time constants are typically 2–6 orders of magnitude slower than those of electromechanical systems and the time delay in the system, Δ, can become the dominant factor, as demonstrated in Figure 9.

If the dominant model of the system is a first-order, low-pass with effectively no time delay, a PID controller [even just the proportional plus integral (PI) portion] will safely control it. Variations in the magnitude of the model have little real effect on stability. With well-modeled parameters, an ideal design uses a PI controller that makes the open-loop response an integrator, as will be discussed in Section XI. This corresponds to the design arrived at via internal model control (IMC) [25], [31]. Mismodeling the pole location affects the phase margin but does not destabilize it. The open-loop phase may get close to $-180°$ if the integrator gain is too high, but it will never touch it. Consequently, conservative designers tend to keep the integrator gain low so that the integral portion is no longer in effect well before the plant pole location. Again, we feel that such basic diagrams can provide valuable insights for the practicing engineer.

A side note here – so obvious in retrospect that it often gets ignored – is that since the time constants for such systems are typically so slow (on the orders of seconds or minutes), any modern embedded processing system will sample fast enough so that even the conservative nature of the backward rectangular rule discretization so common in PID designs has little effect on performance. Making such simple insights available to practicing engineers working on such problems should be a boon to their work that makes a strong case for applying a bit more theory to everyday problems [32].

For these FOPTD models, a key limit is the time delay, specifically the negative phase associated with the transport delay in the process that limits what any causal controller – including a PID – can do. The phase plots in Figure 9 illustrate how limiting delay can be to open-loop phase margin and therefore to achievable closed-loop performance. IMC uses a Padé approximation of the delay [33], [25] and compensates for some of it by forging a bit of lead from the previously dormant derivative term. This has limits due to the nonminimum-phase (NMP) zeros of most Padé approximants [34] and the fact that the approximation accuracy gets worse for longer delays.

One more caveat that we can share about the differences in these three simple-yet-iconic system types lies in which measurements are practical for system identification. In a mechatronic problem, it is usually reasonable to isolate and stimulate these systems without harm and so frequency response functions (FRFs) (also called empirical transfer function estimates – ETFEs – in the academic literature [35]) are extremely helpful, especially for higher-order, lightly damped dynamics. They are almost unheard of in CPC problems, where the idea of injecting chemical stimulus across a variety of frequencies makes no sense and could only result in a process reactor full of useless waste product. Even for such problems as temperature and pressure regulation, the incredibly slow time constants and lack of lightly damped dynamics dictate a choice between extracting data from operational data and/or step responses.

Tied to this are discussions of loop shaping, which are most easily visualized with Bode plots, such as those in Figures 8 – 9. With FRF measurements, one can visualize the effects of controller design on the measured plant response, but as we avoid those measurements with the CPC problems and their relatives, loop-shaping concepts become a bit more strained. The important exception may be if one wants to use IMC to derive the parameters of a PID controller for the FOPTD problem. In the absence of time delay, with a perfect model of the plant, and doing the analysis in continuous time, IMC gives parameters for a PI controller that result in an open-loop response that is that of an integrator [25]. This is exactly a loop-shaping result, but the result is more analytic than based on a direct measurement (Section XI-B). Consequently, the quality of the loop shaping once again depends upon the quality of the plant model.

VI. Specifying PID Parameters

Another area of difference between the mechatronic and process worlds is in how the PID parameters are specified. In the mechatronic control world, it is hard to find a common specification for the parameters, but four basic versions of analog PID control equations show up in the mechatronic control literature and in commercial PID controllers [36], [37]. In the time domain representation those forms are:

$$u(t) = K_P e(t) + \frac{K_I}{T_I}\int_0^t e(\tau)d\tau + K_D T_D \dot{e}(t), \quad (17)$$

$$u(t) = K_P e(t) + K_{I,i}\int_0^t e(\tau)d\tau + K_{D,i}\dot{e}(t), \quad (18)$$

$$u(t) = K_P e(t) + \frac{K_I}{T_I}\int_0^t e(\tau)d\tau + K_D T_D \dot{x_1}(t), \quad (19)$$

$$u(t) = K_P e(t) + K_{I,i}\int_0^t e(\tau)d\tau + K_{D,i}\dot{x_2}(t), \quad (20)$$

where $e(t)$ error input to the controller, $u(t)$ is the controller output, and

$$\dot{x_1} = \dot{e} - \frac{a_1}{T_D}x_1 \quad \text{and} \quad \dot{x_2} = \dot{e} - a_1 x_2. \quad (21)$$

In the frequency domain the four forms for $C(s) = \frac{U(s)}{E(s)}$ are:

$$C(s) = K_P + \frac{K_I}{T_I s} + K_D T_D s, \quad (22)$$

$$C(s) = K_P + \frac{K_{I,i}}{s} + K_{D,i}s, \quad (23)$$

$$C(s) = K_P + \frac{K_I}{T_I s} + K_D\frac{T_D s}{T_D s + a_1}, \quad (24)$$

$$C(s) = K_P + \frac{K_I,i}{s} + K_{D,i}\frac{s}{s + a_1}. \quad (25)$$

For ease of explanation, we will keep to the frequency domain forms. In 24 and 25 we have chosen the derivative filter gain so that – in combination with the derivative – it has a high frequency gain of 1. We could also have chosen to a filter with DC gain of 1. The four forms are chosen by picking two options:

- explicit time specification and

- differentiator filtering.

Explicit time specification simply refers to whether the T_I and T_D terms are present, or whether they are absorbed into K_I and K_D, respectively. It is perfectly legitimate to have

$$K_{I,i} = \frac{K_I}{T_I} \quad \text{and} \quad K_{D,i} = K_D T_D, \tag{26}$$

where $K_{I,i}$ and $K_{D,i}$ can be considered "implicit time" versions of the integral and differential gains. Alternately, the designer can easily go from explicit to implicit time simply by setting $T_D = T_I = 1$. However, leaving the T_I and T_D terms in the equation give the designer some flexibility *and* also allow these terms to drop out when the discrete-time PID is generated. In particular, for the backward rule equivalent of an ideal PID controller with the sample period, $T = T_I = T_D$, the time terms drop out of the equation, making it appear much simpler.

The second option is differentiator filtering. We know that any practical analog differentiator will eventually roll off. It should make sense to explicitly include this in the controller design, but this is not common. Perhaps designers are expecting the plant dynamics and/or circuits to provide low pass behavior. Still, one might wonder why use of a low-pass derivative filter is not a standard practice. This author's best guess is that the most common implementation of a PID controller is a backward rule discrete equivalent approximation. This equivalent puts in its own low pass filter on the differentiator. The typical over-conservatism of the backwards rule equivalent saves the casual designer the trouble and will tend to behave well, especially at low frequencies.

Understanding these four basic forms are useful to a user that has purchased a system that includes a PID controller *e.g.* the controller of a motion control system. Invariably, the user trying to model these systems will find that one of these forms has been used without it being documented in the product literature. Likewise, technical papers on PID controllers will often default to one of these forms without any discussion about the particular choice. Because of this, it is pretty common to see PID gain ranges that vary all over the place, even for the same basic controller.

In the process control world, there is a standard form for PID controllers.

$$u(t) = K\left(e(t) + \frac{1}{\tau_I}\int_0^t e(\lambda)d\lambda + \tau_D \dot{e}(t)\right), \tag{27}$$

$$C(s) = K\left(1 + \frac{1}{\tau_I s} + \tau_D s\right). \tag{28}$$

The parameterization of (27) – time domain – and (28) – transfer function – is often referred to as the ISA (for International Society of Automation) form [38], and commonly presented in the time-domain equivalent. It is typically – but not always – associated with process control applications, temperature, and pressure control [17], [19], [20]. There is an overall controller gain, but the relative gains of the three parts are adjusted via the terms T_I and T_D, which nominally are meant to refer to the integration time (time over which the integral takes place) and differentiation time (time over which the derivative takes place), respectively. Even here, the terminology is confusing, since as written, the integral and differentiator take place over all time. These terms take the place of the integrator and differentiator gains. One would naturally assume that integration and differentiation times (or time constants) would be characteristics of the device or a measurement parameter, rather than a control gain.

It is fairly easy to go from (22) or (23) to the ISA form of (28 by setting $\frac{K_I}{K_P T_I} = \frac{1}{\tau_I}$ and $\frac{K_D T_D}{K_P} = \frac{1}{\tau_D}$ in the former or by setting $\frac{K_I T}{K_P} = \frac{1}{\tau_I}$ and $\frac{K_D T}{K_P} = \frac{1}{\tau_D}$ in the latter.

While the ISA parameterization has the advantage that it is standard in the process control world, there are two annoying features. The first is that in place of integral and derivative term gains, it uses integration and differentiation time parameters. This is counter-intuitive because one usually thinks of these as device properties, not tunable gains. The second is that there are some very nice properties of the explicit time, no derivative filtering form under discretization. These will be discussed in Section VII.

VII. Discretization Differences and Issues

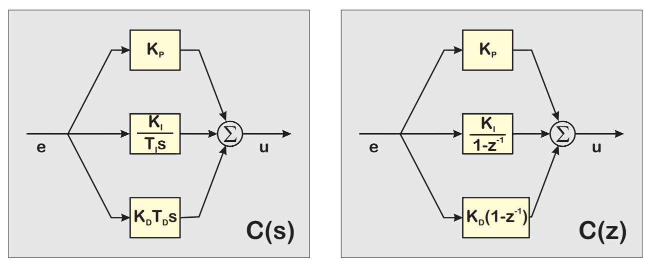

Fig. 10. Structurally similar analog and digital proportional-integral-derivative (PID) controllers. A parallel form topology of a simple analog PID controller (left). When the structure on the left is discretized using the backwards rectangular rule and the sample period (T_S) matches the integration time (T_I) and the differentiation time (T_D), the digital PID controller (right) shows much of the same structure.

Another major area of difference between mechatronic and process treatments of PID control is in discussions of discretization. Part of this difference is that the time constants in most process systems are so slow – relative to any modern real-time compute engine – that "sampling fast", i.e. 100–1000 or more times the fastest time constant in the plant is relatively straightforward. In such an environment, when the the control action is a frequency decade or more below the Nyquist frequency, the concerns of the effects of discretizing the controller are often secondary to other concerns. On the other hand, mechatronic systems feature much smaller time constants (usually several orders of magnitude), which means that the Nyquist frequency and the plant dynamic frequencies are much closer to each other. Consequently, mechatronic control engineers worry about real-time computation a lot more than process control engineers (or at least the former should) [32].

Furthermore, while consequential time delay in simple process system models are generally associated with the process itself, the delays in mechatronic systems are often recognized from the input, output, and computation chains

[32]. As we add more dynamics into our system model, we are more likely to use a controller in the form of (16). Even if we choose to structure the controller as diagrammed in Figure 1, we are most likely to recognize that we must implement those filters digitally. One other detail about this structure is that the PID block is a placeholder for PID-ish controllers: PID, PI, PD, lag-lead, lead-lag, double lead, PIDA, etc.

Furthermore, while many digital control designs might be implemented by doing direct digital design on a discretized model of the plant [39], [40], the design of PID controllers are almost universally done in the continuous-time domain, and then the discretization is done after the fact. By far the most common discretization method for PID controllers is the backwards rectangular rule,

$$s \longrightarrow \frac{1 - z^{-1}}{T_S} = \frac{z-1}{T_S z} \tag{29}$$

[18], [19], [20]. The trapezoidal rule

$$s \longrightarrow \frac{2}{T_S}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) = \frac{2}{T_S}\left(\frac{z-1}{z+1}\right) \tag{30}$$

is sometimes discussed, [20], although this requires that the derivative term include the derivative filtering of (20) and (25) or the PID controller will have an oscillatory pole at $z = -1$ [37]. Readers should appreciate the uniqueness of using the backwards rule here as it does not show up anywhere in Matlab except for the PID toolbox. It is not an option in the c2d() function that is so commonly used by control engineers. The use of the backwards rule effectively chooses a low pass filter for the designer (one with pole at $z = 0$). Recalling that the backwards rule is not used on anything but the PID, the filter design and discretization are almost always done separately, which gives the structure of Figure 1. Again, those extra filters are typically not in process control systems and hence the need to consider discretization is also postponed in most texts.

A simple, but very useful observation is that if one uses the parameterization of (22) and sets $T_I = T_D = T_S$ then the backwards rectangular rule discrete equivalent means that:

$$C(s) = K_P + \frac{K_I}{T_I s} + K_D T_D s \Longrightarrow \tag{31}$$

$$C(z) = K_P + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1}) \tag{32}$$

$$= K_P + K_I \frac{z}{z-1} + K_D \frac{z-1}{z}. \tag{33}$$

Note that the discrete differentiator of (33) has a pole at $z = 0$. This simple choice means that the discrete-equivalent PID is intuitively very close to the continuous-time PID, especially if the sample period, T_S, is 1 second. Unless there is a specific reason to change it, most of the examples in the rest of this paper will use the explicit-time, no-filtering PID structure of (22) or its digital equivalent of (33)

VIII. Anti-Windup

How much we think about discretization also affects how we implement anti-windup schemes. This section will discuss that area, although a deeper dive will be presented in [41]. Since most of the PID centric literature is from the process world, and the treatment there is almost entirely a continuous-time one, it is logical that the most commonly discussed form of anti-windup would be the back calculation scheme diagrammed in Figure 7 [42], [43], [44], [45], [46]. Back calculation acts to zero out the input to the integrator once the PID is saturated by feeding back the difference between the saturated and unsaturated controller outputs to the input of the integrator. However, since the controller signal has three components, matching the exact amount that goes to the integrator is a matter of tuning a feedback gain, K_{AW}. A key advantage of this scheme is that it can be implemented in continuous time or discrete time, as shown in Figure 11. Note that since we cannot instantaneously feed back the cancellation term, we require a time delay on the path of K_{AW}. Back calculation shows up in a lot of the PID literature, although many examples seem limited to PI controllers.

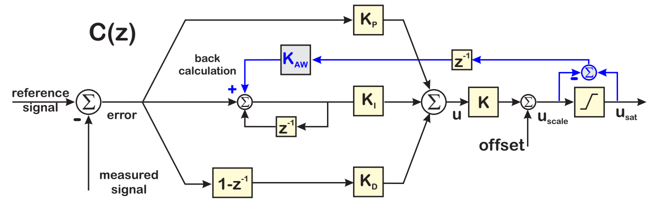

Fig. 11. Discrete-time PID with back calculation anti-windup.

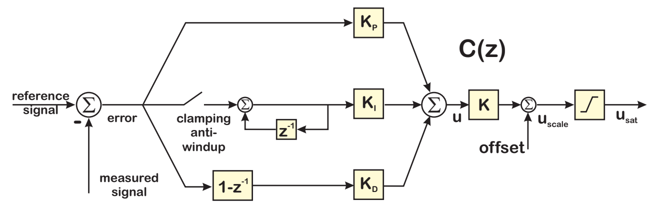

Fig. 12. Discrete-time PID with integrator clamping anti-windup.

However, if the goal is simply to zero out the input to the integrator while the controller is saturated, embracing a fully digital method, such as the integrator clamping – also called conditional integration – shown in Figure 12 accomplishes that [42], [43], [45]. The key is the use of conditional logic, which is far simpler to implement in digital logic. Integrator clamping zeros the input to the integrator when the controller is saturated, clamping the integrator at its previous level. Because it is in a computer, one can apply extra intelligence to bleed off the integrator value when the sign of the error signal is the opposite of the sign of the contents of the integrator. Essentially, conditional integration does what back propagation promises to do, but without the worry of selecting K_{AW}.

IX. Derivative Filtering

Another area that seems pushed to the background is that of derivative filtering. The premise is that we cannot have a pure differentiator in an actual system, and so at some point some sort of low-pass mechanism must be employed. There seem to be five basic methods for this:

The Analog Ignore It: The basis for this method is that the analog circuitry for the controller, or the interface circuits, or the plant itself are necessarily low pass and so the need for derivative filtering will be handled somewhere else without need for the designer to be overly concerned. (This is analogous to all of those filter circuits that the PLL designers fail to mention.) While this is feasible, it leaves the choice of the filter up to someone other than the control designer. Often that works, but it does remove a design option from the control engineer.

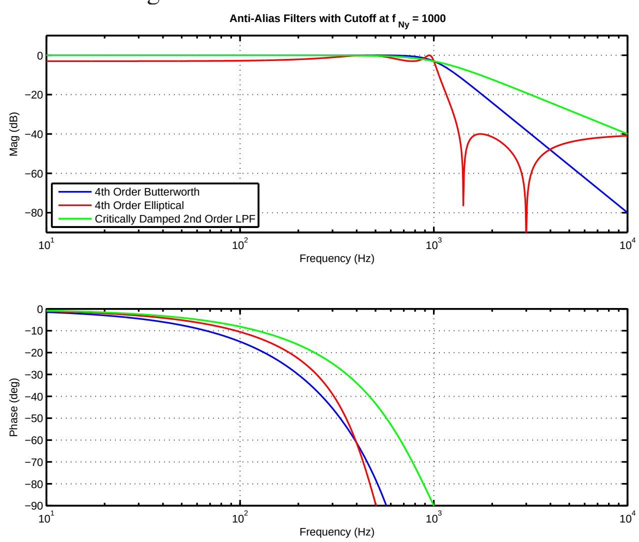

Fig. 13. Frequency responses of various anti-alias filters. All filters have a DC gain of 1, with the passband ending at the Nyquist Frequency ($f_{Ny} = f_S/2$ = 1kHz here). Under an assumption that the sample frequency is 10 or 20× the open loop crossover frequency, we can examine the filter phase response, which can significantly degrade the phase margin of the system, as documented in Table I. The frequency axis is normalized by the Nyquist frequency.

Analog The Whole Enchilada: In this form, we wrap a low-pass filter around the entire controller, e.g. if we are using (22) then

$$C(s) = \left(K_P + \frac{K_I}{T_I s} + K_D T_D s\right)\left(\frac{a}{s+a}\right)^n. \quad (34)$$

This does provide the needed filtering on the D-term and has the simplicity that it does not change the zeros of the PID controller. However, the PI portion of the PID was not the problem area and likely did not need the filtering. One can argue that since the PI action happens at low frequency and the low pass happens at higher frequencies, that these do not interfere with each other. However, the anti-alias filter example used in [47] shows that the phase effects of even simple low-pass filters can be severe, as seen in Figure 13 and Table I. All this is to provide an example that when the cutoff frequency of a low pass filter is close to the dynamic frequencies of the control system, the former can cost a substantial phase penalty.

Analog Just the D: This is the method applied in (24). It has the advantage that it limits the gain of D term at high frequency while not affecting the PI region of the controller. However, it does change the overall algebra of the controller equations, complicating analysis [48].

Filter	**Phase at $f_{Ny}/10$**	**Phase at $f_{Ny}/5$**	**Attenuation at $10f_{Ny}$**
4^{th} Order Butterworth	$-15.0276°$	$-30.1223°$	-80.1201 dB
4^{th} Order Elliptical	$-10.5523°$	$-22.8086°$	-40.8932 dB
2^{nd} Order Butterworth	$-8.1486°$	$-16.4211°$	-40.0605 dB

TABLE I

PHASE PENALTY OF REPRESENTATIVE ANTI-ALIAS FILTERS. THE CORNER FREQUENCY IS CHOSEN TO BE AT THE NYQUIST FREQUENCY, HALF THE SAMPLE FREQUENCY, $f_{Ny} = f_S/2$. COMPARISONS ARE MADE WITH RESPECT TO THE NYQUIST FREQUENCY AT IT IS CONSIDERED THE LIMIT OF INTENTIONAL DIGITAL CONTROL ACTION. THE TWO BUTTERWORTH FILTERS ARE FLAT IN THE PASSBAND, BUT INCUR A LARGER PHASE PENALTY RELATIVE TO THE ELLIPTICAL FILTER FOR STOPBAND GAIN ATTENUATION THEY PROVIDE. ON THE OTHER HAND, THE ELLIPTICAL FILTER HAS UP TO 3 DB MAGNITUDE DISTORTION IN THE PASSBAND. ALL OF WHICH IS TO SAY THAT THE CHOICE OF ANTI-ALIAS FILTER STRUCTURE SHOULD NOT BE SEPARATED FROM THE AVAILABLE SAMPLE RATE OPTIONS, NOR THE ROBUSTNESS OF THE SYSTEM TO GAIN AND PHASE DISTORTIONS.

Digital Ignore It: This one stems from using the backwards rectangular rule for computing the discrete equivalent, which conveniently puts a low pass filter on the digital D-term at $z = 0$.

However, the most popular form of limiting the noise from the D-term is **Set K_D = 0**. That is, while we talk about PID controllers, any informal survey of control engineers using PID controllers reveals that the vast majority of these designs have zeroed out the D term and are really PI controllers.

X. DIFFERENCES IN MEASUREMENTS AND MODELING

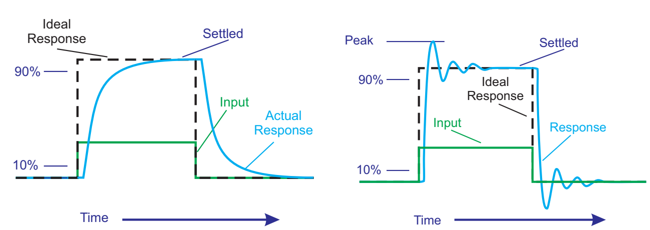

Fig. 14. Idealized responses of first (left) and second (right) order models.

The differences in dominant low order models also affect how these models are identified. We will stay with our assertion that the dominant models in the process control worlds are the first-order-plus-time-delay (FOPTD) with the occasional second-order-plus-time-delay (SOPTD) model. Key to each of these is that they are open-loop stable and do not contain any integrators [21]. Furthermore, the SOPTD model has real poles, rather than taking the form of an underdamped resonance. These simple models typically lack any zeros in the response, which makes them amenable to identification via step response methods – often referred to as process reaction curves in the process control world [20], diagrammed in Figure 14. If the system is dominated by one of the poles, then relay tuning or Ziegler-Nichols Tuning (or some similar method) may be used [20] Because these models are first or second order, even discrete-time, time-domain methods can retain some physical intuition; something that is much harder for higher order models.

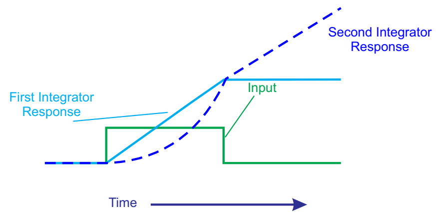

Fig. 15. Idealized step response of single and double integrator.

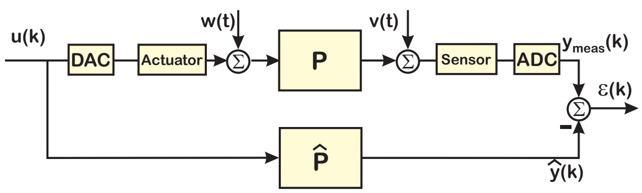

Fig. 16. Time domain, discrete-time model, system identification diagram.

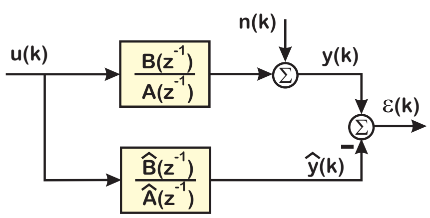

Fig. 17. Simple ARMA plant regression model.

When the dominant low-order model contains an integrator then a step response does little good. For example with the ubiquitous double integrator the first integrator turns the input into a ramp and the second into a quadratic (Figure 15). Since the plant is not open-loop stable, there is no settle time and we cannot extract the gains simply from a ratio of steady state output to steady state input. Closing the loop to produce a step response of a stable system requires at least some lead, and so some nominal PD controller must be used (assuming there are not higher order dynamics of concern). We can put in a square wave function and try to discern the gain constant from there.

Neither of these provide a strategy for dealing with other dynamics in the model. Since step response methods are not overly useful, these plants suggest input signals without DC content: either random (pseudo-random) or oscillatory input signals. We also start considering more more generalized structures for the plant model. Putting these together we get to the time-domain regressions on discrete-time parametric models, as diagrammed in Figure 16, or frequency-domain, non-parametric methods [35], [49]. The former suffer from a lack of of connection between the discrete-time model coefficients and the physical parameters. The diagram illustrates how components of the input-output response of the plant, the sensors, actuators, ADCs, and DACs, all get subsumed into the digital model. While this works for some systems, the loss of physical intuition often makes it much more difficult to debug a real system. The non-parametric, frequency domain methods suffer from the need to extract model parameters from the frequency response function (FRF) [50], [51], [52].

In any event, we can readily see that the switch between the stable FOPTD and SOPTD models prevalent in the process control world and the marginally stable models so prevalent in the mechatronic world not only change which measurement methods are used, but also affect the physicality of the extracted model parameters from those measurements.

There is more evidence for this in examining the operation of *i-pIDtune*, which is an interactive tool for system identification and PID control design [53]. While this is more of a teaching tool than an industrial application system (as it does not generate test vectors for export to physical devices nor import the measured responses resulting from those test vectors), it provides an interesting view of what some of the leading researchers in the use of PID controllers in process control consider important. The models considered for PID tuning are either first or second-order plants without integrators, although with the possibility of non-minimum phase (NMP) zeros. The recent update to the tool, *i-pIDtune 2.0* [54] expands the set of plants to include second-order plants with integrators. In both versions of the tools, zero mean signals are used as a stimulus (either Pseudo-Random Binary Signals – PBRS or multi-sines) is different from the normal step-response methods. This frees the identification step from needing to specify a model structure in advance. The authors use an AutoRegressive with eXternal input(ARX) model structure to match a general discrete-time transfer function model and then do model reduction to extract the parameters for one of their supported models. There is an option in Version 2.0 to remove known integrators from the simulation *a priori*. However, even this advanced teaching tool illustrates that the presence or absence of an integrator in the plant structure changes how we try to identify it. Thinking about how this might apply in practice, we could easily see how mechatronic and process control engineers would have a vastly different view of plant identification, based upon their experience in dealing with plants with and without integrators.

Another difference between mechatronic and process control system identification is the relative importance of time delay. For many mechatronic systems, the amount of delay relative to the time constants of the system are small enough that issues of high frequency dynamics (e.g. resonances and anti-resonances) dominate. The time delay may not be included in the model if small enough.

On the other hand, when the dominant models for a set of problem always have "plus time delay" in their names, it seems that we are less likely to ignore those. This creates an issue for typical time-domain identification with linear, discrete-time models because one often represents the delay with a Padé approximant [33]. The most accurate of these have at least one non-minimum phase (NMP) zero. Adding this to the identification model begs the question as to whether it should just be considered another order of the model or whether the effects of delay on the data can be separated out (as we can in the frequency domain).

XI. Different Views of Loop Shaping

An oft-quoted story about Mahatma Gandhi is that when a journalist asked, "What do you think of Western civilization?" he responded with, "I think it would be a good idea." Similarly, loop shaping is often considered a good idea for control design, but how it is viewed with respect to the use of PID controllers is again different in the mechatronic and process control worlds. More examples from this discussion can be found in [55].

In the mechatronic control world, loop shaping is often described and evaluated in terms of the frequency domain. It is possible to work directly from frequency response function (FRF) measurements of the plant, adjusting controller components until the measured (or projected) open-loop response looks favorable. A common theme proposed by this author [56], [57] and others [58], [59], [60] is to shape the open-loop response towards that of an integrator. This can be done in a variety of ways, but results in a response that can be easily evaluated and optimized for its closed-loop properties. Following this philosophy, a combination of PID-like simple controllers and filters can be added and adjusted until the open-loop frequency response looks like an integrator. The key limitation here is time delay which can be mitigated some by lead filters, but eventually eliminates the phase margin (PM) and limits the open-loop gain crossover frequency.

From an industry perspective, or the perspective of someone trying to transfer a methodology into an industrial environment, making the open loop an integrator is "explainable" loop shaping.

- There is no need to convert to state space or use H_∞, (but we could).
- It is so intuitive that it can be done without using to optimization package (but we could).
- It can be understood in continuous or discrete time.
- It can be accomplished algebraically and/or graphically.
- The FRFs provide an easy-to-understand visualization.
- It enables simple computations for evaluating against open and closed-loop constraints.

All of this means that using this tuning philosophy makes it relatively easy transfer to industry without much pushback. After all, we're not taking away their PID; we're showing them how to tune it up. A side benefit is that success in transferring improvements to industry PIDs in this manner opens the door to other things we might want to add to an industry control system.

Notably, for portions of the plant response that can be modeled with stable, minimum phase, proper filters, an inverse filter corrects the response. It is not practical to directly invert pure integrators, since pure differentiators are rarely practical, whether in continuous or discrete time. Instead, we are often left with compensating the integrator(s) with lead elements. These have a built-in corner frequency for the zero and so the amount of lead compensation is limited.

A. Loop Shaping in the Frequency Domain

Frequency domain measurements are often favored over time-domain measurements for lightly damped mechatronic systems because they can often clearly resolve the key dynamics from those that are below the noise floor and establish what is feasible. However, frequency domain identification results in FRFs, a set of ordered pairs of frequencies and complex response, which do not immediately fit into an analytic transfer function or state-space model.

Frequency domain loop shaping is very visual, most commonly working from Bode plots. The advantage of this in systems that are well suited to frequency domain measurements, is that the design can be done working directly from the measurements, without having to reduce the measurement to an analytic model. In the days before widespread use of powerful digital computing tools, this was a very practical way of designing controllers, particularly for single-input, single-output (SISO) systems. Three examples are particularly illustrative here, in part because we can compare them to doing the same design using Internal Model Control (IMC) in Section XI-B. These are:

- Loop shaping for a first-order, plus time delay (FOPTD) model provides two examples. We will look at the PID controller we need for a first order without time delay and then see how that changes when we have time delay.
- Loop shaping on double integrator. This model is far more common in the mechatronics world, and on the simpler side of what is typically done in the frequency domain, but it is still illustrative of how we configure a PID for such systems.

1) First Order Plus Time Delay (FOPTD): The model for a FOPTD plant was given in (7). For $\Delta = 0$, this reduces to a simple first order model,

$$P(s) = \frac{K}{\tau s + 1} = \frac{\frac{K}{\tau}}{s + \frac{1}{\tau}} = \frac{Ka}{s+a} \tag{35}$$

If we take the objective of our controller as making the open-loop response resemble that of an integrator, then we expect

$$C(s) = \frac{K_C(s+a)}{s} = K_C + \frac{K_C a}{s} = K_P\left(1 + \frac{K_{I,i}a}{K_P s}\right). \tag{36}$$

Here, we are using the implicit time form from (25) just to simplify the text. This yields an open-loop response of:

$$P(s)C(s) = \frac{KK_C a}{s}. \tag{37}$$

In the absence of time delay, a well matched PI controller turns the no-delay curve of Figure 9 into a straight integrator curve with a phase of $-90°$, resulting in infinite gain margin

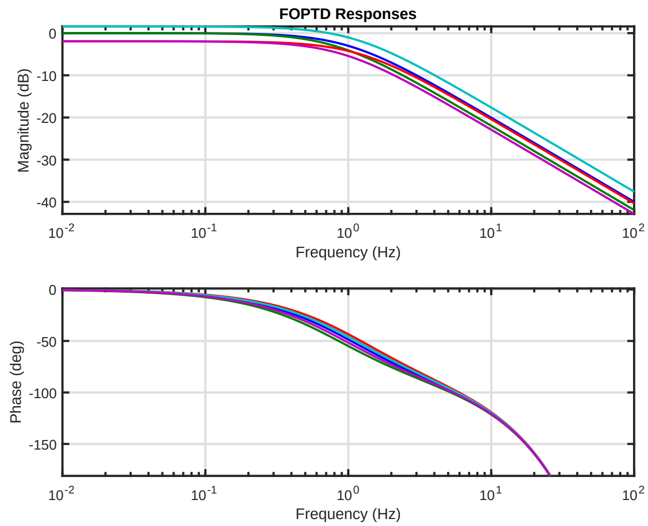

Fig. 18. Bode plot for first order plus time delay plant plant, but with varying gains.

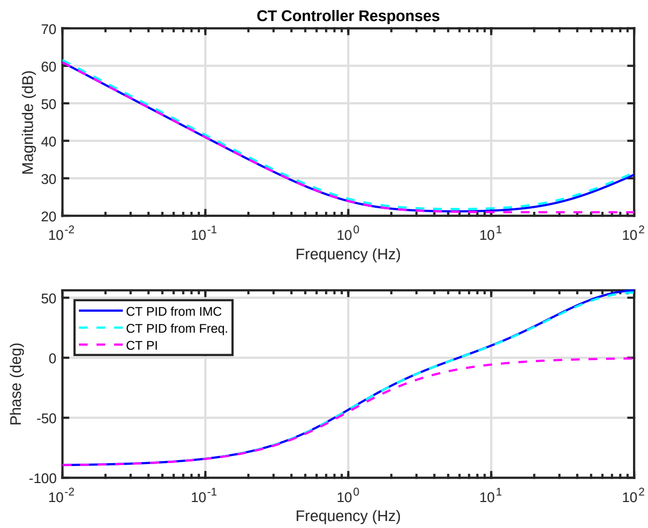

Fig. 19. Bode plot for three continuous-time (CT) controllers. One is a PID, designed as a PI plus a filtered PD, in series. The value for the low pass filter, a_2, was picked, and then the values for b, a_1, and K_C were picked from the IMC calculations in (79). The second is a PID designed from adjusting the b, a_1, and K_C parameters until the open-loop and closed-loop responses looked reasonable. The third is a PI controller where the PI zero matches the first order pole, and the gain is adjusted to match the open-loop response of the other two.

and $90°$ phase margin. As mentioned earlier, it is the time delay that eats away the phase margin, as seen in Figure 9. The only element of a PID that can restore any of the phase is the D-term. We can plot the effect of $e^{-\Delta s}$ directly in frequency responses since this results in a pure phase component, but to work with any transfer function (or state-space) we would want to approximate this, typically with a Padé approximation. In most applications, a first order Padé approximant [33] is used:

$$e^{-\Delta s} \approx \frac{\frac{2}{\Delta} - s}{\frac{2}{\Delta} + s}. \tag{38}$$

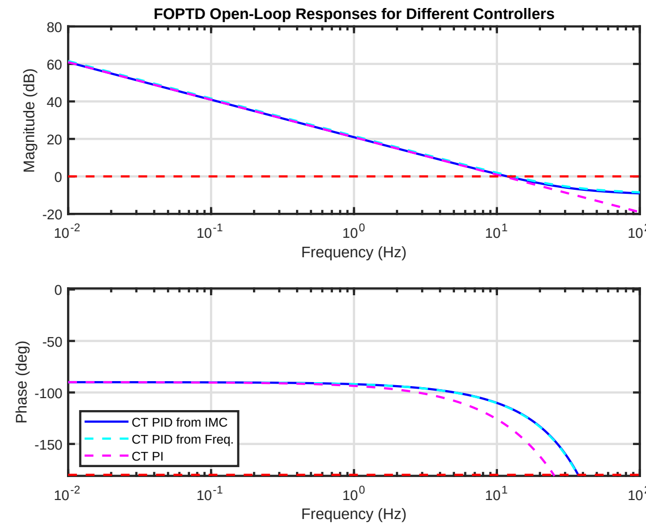

Fig. 20. Open-loop responses for nominal plant from Figure 18 and the three different controllers of Figure 19.

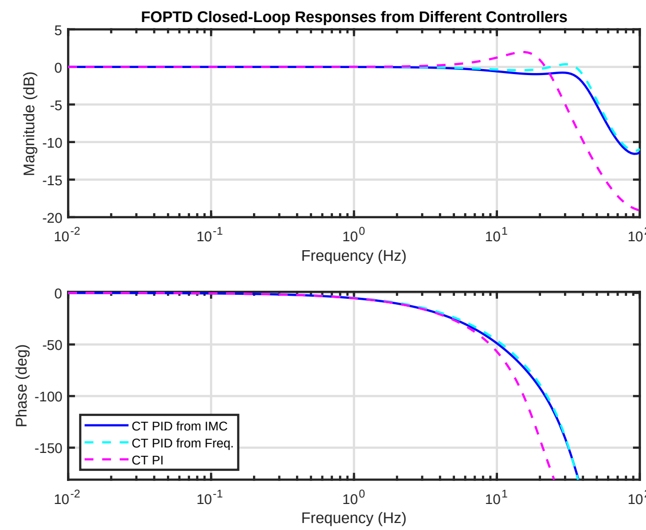

Fig. 21. Closed-loop responses for nominal plant from Figure 18 and the three different controllers of Figure 19.

Although the first-order Padé (38) is commonly used as a rational approximation of time delay in controller design, it is worth noting that it is bounded below by $-180°$, while the phase of $e^{-\Delta s}$ decreases without bound towards $-\infty$. We should be cautious about using this when Δ is close to the sample period or larger.

If we want to compensate for this, we need a lead circuit

$$C(s) = \frac{K_C(s+a)}{s}\left(\frac{s+b}{s+a_1}\right). \tag{39}$$

As before, the first term serves to compensate for the plant pole. The second factor is a phase lead if $0 < b < a_1$. If we want to extract our PID parameters from this, we can compare it to one of our earlier formulations. For simplicity we will choose the implicit time with derivative filtering form of (25). If we put that form over a common denominator, we

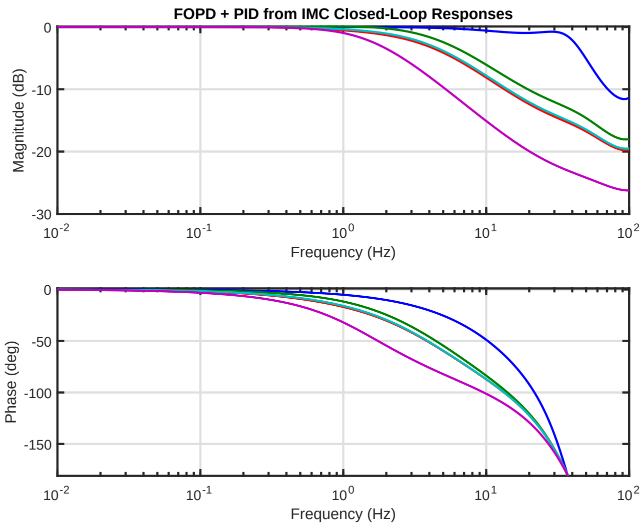

Fig. 22. Closed-loop responses for plants of Figure 18 and IMC designed PID controller of controller of Figure 19. The variation in responses is due to the variation in the plant responses from Figure 18.

have:

$$C(s) = \frac{K_P s(s+a_1) + K_{I,i}(s+a_1) + K_{D,i}s^2}{s(s+a_1)}, \quad (40)$$

$$= \frac{(K_P + K_{D,i})s^2 + (K_P a_1 + K_{I,i})s + K_{I,i}a_1}{s(s+a_1)}, \quad (41)$$

$$C(s) = \frac{K_C s^2 + K_C(a+b)s + K_C ab}{s(s+a_1)}. \quad (42)$$

We can extract our PID coefficients from equating the numerator terms of (41) and (42). We note that in (42), the only free parameters are b, a_1, and K_C. The first two determine when the lead action starts and ends. The last one is the overall gain of the system which is usually set from open-loop gain and phase margin considerations. Matching the numerator terms of (41) and (42), we get

$$K_C = K_P + K_{D,i}, \quad (43)$$

$$K_C ab = K_{I,i} a_1, \text{ and} \quad (44)$$

$$K_C(a+b) = K_P a_1 + K_{I,i}. \quad (45)$$

Finally, from these, we get the PID gains:

$$K_P = \frac{K_C}{a_1}\left[(a+b) - \frac{ab}{a_1}\right], \quad (46)$$

$$K_{I,i} = K_C \frac{ab}{a_1}, \text{ and} \quad (47)$$

$$K_{D,i} = K_C - K_P. \quad (48)$$

The FOPTD plant responses for five slightly varying plants are plotted in Figure 18. PI and PID controllers are plotted in Figure 19, where one set of PID parameters was generated from the IMC loop shaping of (79), while another was tuned by visually adjusting the open-loop response. The PI controller was adjusted to match most of it's open-loop response to the other two. The resulting open and closed-loop responses are shown in Figures 20 and 21. Note that the action of both PID designs in Figure 19 is to add the desired phase lead to the controller not seen in the PI controller. This extra lead gives more phase margin (Figure 20) allowing increased closed-loop bandwidth with less peaking (Figure 21. Figure 22 shows variations in the closed-loop response when the IMC designed PID is used on mismatched plant models.

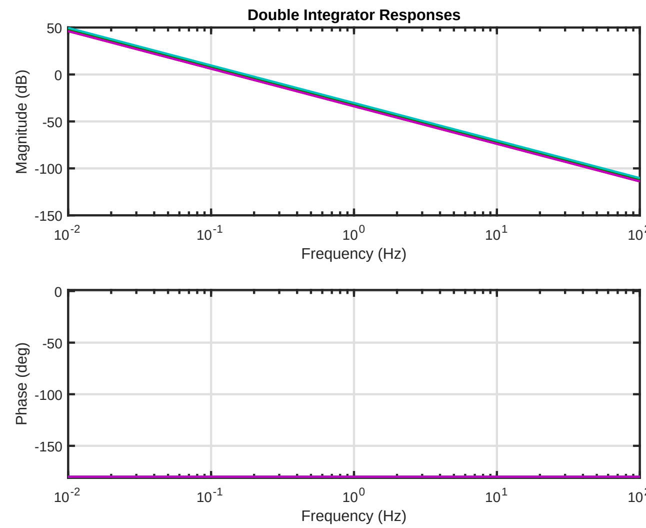

Fig. 23. Bode plot for double integrator plant, but with varying gains.

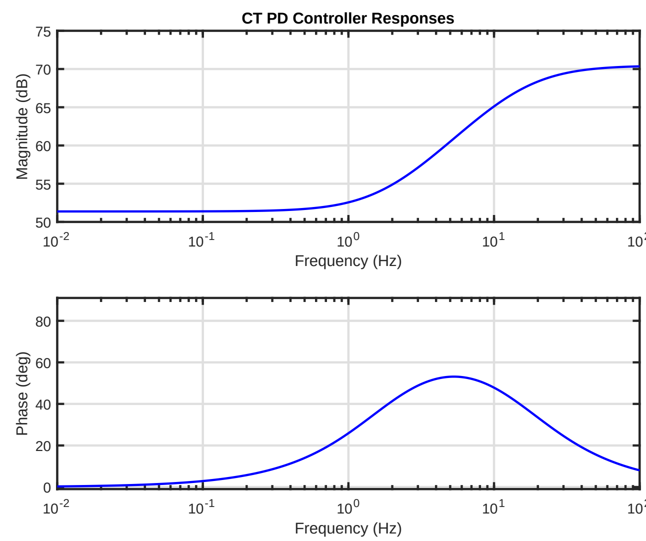

Fig. 24. Bode plot for a continuous-time (CT) proportional plus derivative (PD) controller. The value for the low pass filter, a_1, was picked, and then the values for b and K_C were picked from the IMC calculations in (93).

2) Double Integrator: If the FOPTD of (7) is the iconic model for process control systems, then the double integrator of (5) is the iconic model for mechatronic systems. If we ignore for a moment, the possibility of step disturbances at the plant input as discussed in Section IV, our instinct is that this system does not need another integrator in the controller. With the plant phase already at $-180°$, what it needs is some phase lead to stabilize the system. The only way to get this

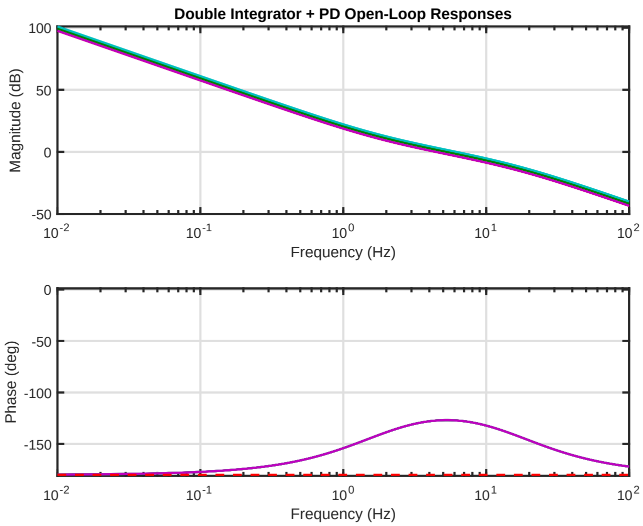

Fig. 25. Open-loop responses for plants of Figure 23 and controller of Figure 24.

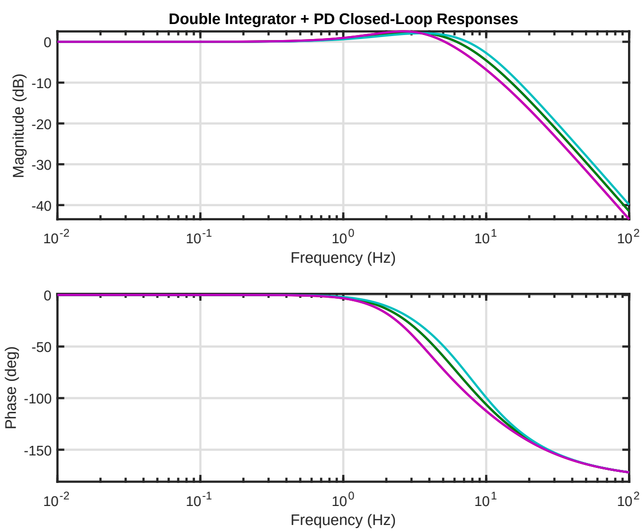

Fig. 26. Closed-loop responses for plants of Figure 23 and controller of Figure 24.

with a PID is using the D-term:

$$C(s) = K_P + K_{D,i}\left(\frac{s}{s+a_1}\right), \quad (49)$$

$$= \frac{(K_P + K_{D,i})\left[s + \frac{K_P a_1}{K_P+K_{D,i}}\right]}{s+a_1}, \quad (50)$$

$$C(s) = \frac{K_C\,(s+b)}{s+a_1}. \quad (51)$$

This always produces a lead, since $\frac{K_P}{K_P+K_{D,i}} < 1$ for any positive values of K_P and $K_{D,i}$. How much lead we have is determined by the relative sizes of K_P and $K_{D,i}$. If one accepted the simple model as truth, one might be tempted to make b as small as possible and a_1 as large as possible, but we know practically that the double integrator behavior at low frequency helps minimize steady-state error to a step or a ramp, and noise and unmodeled time delay often put a limit on a_1. From these, we get the PID gains:

$$K_C = K_P + K_{D,i} \quad \text{and} \quad (52)$$

$$b = \frac{K_P a_1}{K_P + K_{D,i}}.. \quad (53)$$

From here we get the PD coefficients (there is no I-term):

$$K_P = K_C\frac{b}{a_1} \quad \text{and} \quad (54)$$

$$K_{D,i} = K_C\left(1 - \frac{b}{a_1}\right). \quad (55)$$

The double integrator plant response, PD controller, open-loop, and closed-loop responses are plotted in Figures 23–26. The plant gain is set to 1 and the low pass filter corner, $a1$ is set to 100. From there, the rest of the gains were adjusted using the formulas from the IMC loop shaping below, particularly (93). This results in the PD controller of Figure 24. We can see the effect of the practical lead in Figure 25, producing close to 60° of phase margin. The use of parameters from IMC results in the magnitude crossing $0dB$ near the frequency of maximum phase margin and produces very clean closed-loop responses with minimal peaking (Figure 26).

B. Loop Shaping Using IMC

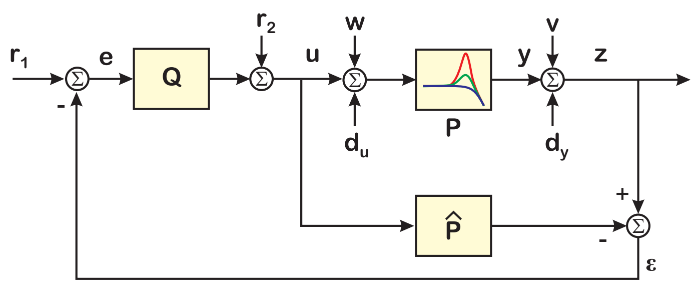

Fig. 27. Internal Model Control (IMC)

A common diagram for Internal Model Control (IMC) [31] is shown in Figure 27, although this one has been augmented with more noise and disturbance inputs than are typical in the standard process control centric diagrams. We can see that the formulation uses a model, $\hat{P}$, of the plant, P, to construct the controller. With a bit of foreshadowing, one might note that this method will depend on how closely $\hat{P}$ – which can be considered an observer – approximates P, but for now we will go through the algebra.

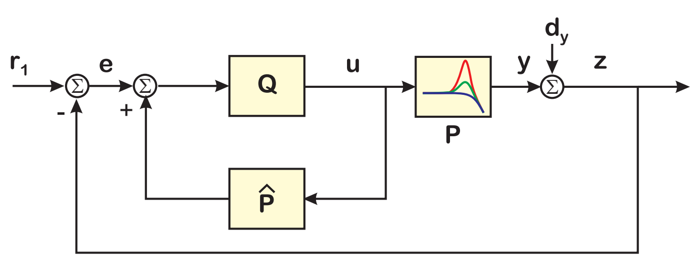

Fig. 28. Internal Model Control (IMC) rewritten to isolate controller.

We can use loop manipulations to redraw Figure 27 to that of Figure 28, ignoring for now most of the disturbance inputs. We can now see that the controller, $C(s)$ is constructed with the positive feedback loop:

$$C(s) = \frac{Q(s)}{1 - \hat{P}(s)Q(s)} \quad (56)$$

This is often recognized as the Youla-Kucera parameterization [61], [62], although the latter typically assumes that P and $\hat{P}$ are stable. For our purposes, this means that there are no unstable poles or integrators in either, which right away is a difference from models so typical in the mechatronic world. One can wrap the unstable/marginally stable plant in a stabilizing feedback loop, but that clouds the IMC discussion. Our goal is to point out once again that IMC is most typically applied to plants that are already stable and lack an integrator.

The a key feature of the Youla-Kucera parameterization is that Q is stable and proper. As it is used in IMC, Q will also be used to invert the plant, P. If the plant is stable and strictly proper, with a pole-zero excess of m, then IMC usually adds in a low-pass filter (LPF) of the form:

$$F(s) = \frac{1}{(\tau_F s + 1)^m} = \left(\frac{\frac{1}{\tau_F}}{s + \frac{1}{\tau_F}} \right)^m, \quad (57)$$

so that $Q(s) = \tilde{Q}(s)F(s)$ is proper.

$$OL(s) = P(s)C(s) = \frac{P(s)Q(s)}{1 - \hat{P}(s)Q(s)} \quad (58)$$

At this point in most IMC descriptions, there is a slight of hand in which the model, $\hat{P}$, is replaced by the plant, P. In other words, there is an assumption of perfect modeling, so that $\hat{P}(s) = P(s)$ and

$$OL(s) = P(s)C(s) = \frac{P(s)Q(s)}{1 - P(s)Q(s)}. \quad (59)$$

Now if $\hat{P}(s) = P(s)$ is invertible (i.e. no RHP zeros), then pick

$$Q(s) = \tilde{Q}(s)F(s) = \hat{P}^{-1}(s)F(s) = P^{-1}(s)F(s). \quad (60)$$

This gives:

$$Q(s)\hat{P}(s) = P^{-1}(s)P(s)F(s) = F(s) \quad (61)$$

so that

$$OL(s) = \frac{P(s)\tilde{Q}(s)F(s)}{1 - P(s)\tilde{Q}(s)F(s)} = \frac{F(s)}{1 - F(s)}. \quad (62)$$

If the pole-zero excess of P is one, as is the case for the FOPTD plant so common in process control systems, then $m = 1$ and

$$OL(s) = \frac{\frac{1}{\tau_F s+1}}{1 - \frac{1}{\tau_F s+1}} = \frac{1}{\tau_F s}, \quad (63)$$

In its most ideal form, and for the most common plant model in the process control world, IMC turns the open loop frequency response into an integrator. This ideal response depends upon a close match between $\hat{P}(s)$ and $P(s)$. That is, the benefits of IMC truly depend on having a good model. (It's in the name, after all.)

In the particular case of the FOPTD model of (7) with no time delay ($\Delta = 0$), $\tilde{Q}(s)$ would have a single zero to cancel the plant pole. The controller, $C(s)$ ends up being a PI controller:

$$C(s) = \frac{P^{-1}(s)F(s)}{1 - F(s)}, \quad (64)$$

$$= \frac{\frac{\tau s+1}{K(\tau_F s+1)}}{\frac{\tau_F s}{\tau_F s+1}} = \frac{\tau s + 1}{K\tau_F s}, \quad (65)$$

$$C(s) = \frac{\tau}{K\tau_F}\left(1 + \frac{1}{K\tau_F s}\right). \quad (66)$$

This matches our previous loop shaping example using PID on a FOPTD model with no delay, but is arrived at just through algebra.

When we do have delay, then the D term of the PID becomes important. For those in the mechatronic world, this is logical as it is the only part of a PID that can give us phase lead. However, the unpopularity of the use of the frequency domain in the process control world means that this bit of intuition must be arrived at differently. In this case, the desire to use algebraic methods means that the time delay gets represented by a Padé approximation of (38) [33]. Critical to this is the choice of the many approximations to use, but for the classic FOPTD model it seems that a first order approximant with a single stable pole and non-minimum phase (NMP) zero. If we are using only a PID, then the question becomes how to use the D term to compensate for some of the excessive negative phase of the NMP zero.

1) First Order Plus Time Delay (FOPTD): To do repeat the loop shaping of Section XI-A.1 using IMC, we apply the Padé approximation [33] of (38) to (7), to yield

$$P(s) = \frac{Ka}{s+a}\left(\frac{\frac{2}{\Delta} - s}{\frac{2}{\Delta} + s}\right). \quad (67)$$

In this case, we compute:

$$\tilde{Q}(s) = \frac{(s+a)(\frac{2}{\Delta} + s)}{K\frac{2}{\Delta}}, \quad F(s) = \frac{a_2^2}{(s+a_2)^2}, \quad (68)$$

which means:

$$Q(s) = \frac{a_2^2\Delta(s+a)(\frac{2}{\Delta} + s)}{2K(s+a_2)^2}, \quad (69)$$

and

$$\hat{P}(s)Q(s) = \left(\frac{a_2^2\Delta}{2}\right)\frac{(\frac{2}{\Delta} + s)}{(s+a_2)^2}. \quad (70)$$

Finally,

$$C(s) = \frac{\frac{a_2^2\Delta(s+a)(\frac{2}{\Delta}+s)}{2K}}{1 - \left(\frac{a_2^2\Delta}{2}\right)\frac{(\frac{2}{\Delta}+s)}{(s+a_2)^2}}, \quad (71)$$

$$= \frac{\frac{a_2^2\Delta}{2K}(s+a)(\frac{2}{\Delta} + s)}{(s+a_2)^2 - \frac{a_2^2\Delta}{2}(\frac{2}{\Delta} + s)}, \quad (72)$$

$$C(s) = \frac{\frac{a_2^2\Delta}{2K}(s+a)(\frac{2}{\Delta} + s)}{s\left(s + a_2\left(2 + \frac{\Delta a_2}{2}\right)\right)}. \quad (73)$$

Looking at (73), we recognize the PI control portion of $\frac{s+a}{Ks}$ that we had before. That handles the first order dynamics. We also know from our frequency domain intuition that the rest of the controller needs to be a lead circuit in order to counteract the negative phase of the time delay. That means that we need the zero from the $\frac{2}{\Delta}$ to happen before the pole from the $a_2(2+\frac{\Delta a_2}{2})$ term.

$$a_2\left(2+\frac{\Delta a_2}{2}\right) > \frac{2}{\Delta}, \quad (74)$$

$$4a_2+\Delta a_2^2 > \frac{4}{\Delta}, \quad (75)$$

$$\Delta a_2^2+4a_2-\frac{4}{\Delta} > 0, \quad (76)$$

$$\frac{a_2}{\Delta}^2+4\frac{a_2}{\Delta}-\frac{4}{\Delta^2} > 0,. \quad (77)$$

If we complete the square, we end up with:

$$\left(a_2+\frac{2}{\Delta}(\sqrt{2}+1)\right)\left(a_2-\frac{2}{\Delta}(\sqrt{2}-1)\right) > 0. \quad (78)$$

Our free parameter here is a_2. Equation 78 will hold if a_2 is very positive or very negative. Since we want a causal filter, we need $a_2 > 0$, which means our requirement is $a_2 > \frac{2}{\Delta}(\sqrt{2}-1)$ for our controller to have a lead. We can interpret this as needing the low pass filter on the D-term to be far enough out to leave room for some lead action.

With $C(s)$ defined in (73), our open loop response becomes:

$$P(s)C(s)=\frac{\frac{a_2^2\Delta}{2}\left(\frac{2}{\Delta}-s\right)}{s\left(s+a_2\left(2+\frac{\Delta a_2}{2}\right)\right)}. \quad (79)$$

This is the best that the IMC design can do, and we know from (73) that it has provided some phase lead. Comparing this to the lag-lead controller of (39), we see that

$$a_1=a_2\left(2+\frac{\Delta a_2}{2}\right), b=\frac{2}{\Delta}, \text{ and } K_C=\frac{a_2^2\Delta}{2K}. \quad (80)$$

We see that the IMC design has determined most of our parameters, except for the final selection of a_2 which determines a_1. From these values, we can use (46)– (48) to calculate the PID gains.

2) Double Integrator: To do repeat the loop shaping of Section XI-A.2 using IMC, we have:

$$\tilde{Q}(s)=\frac{s^2}{K}, \quad F(s)=\frac{a_2^2}{(s+a_2)^2}, \quad (81)$$

which means:

$$Q(s)=\frac{a_2^2}{K}\frac{s^2}{(s+a_2)^2}, \quad (82)$$

and

$$\hat{P}(s)Q(s)=\frac{a_2^2}{(s+a_2)^2}. \quad (83)$$

Finally,

$$C(s) = \frac{\frac{a_2^2}{K}\frac{s^2}{(s+a_2)^2)}}{1-\frac{a_2^2}{(s+a_2)^2)}}=\frac{\frac{a_2^2}{K}s^2}{(s+a_2)^2-a_2^2}, \quad (84)$$

$$C(s) = \frac{\frac{a_2^2}{K}s}{s+2a_2}. \quad (85)$$

This controller is a lead, but with a pure derivative numerator. If we want the PD controller from Section XI-A.2, we need to pick a different $F(s)$ filter. Using the same $\tilde{Q}(s)$ from (81), but

$$F(s)=\frac{s+b}{(s+a_2)^3}\left(\frac{a_3^3}{b}\right), \text{ so that} \quad (86)$$

$$Q(s)=\frac{s^2}{K}\left(\frac{a_2^3}{b}\right)\frac{s+b}{(s+a_2)^3}, \quad (87)$$

and

$$\hat{P}(s)Q(s)=\left(\frac{a_2^3}{b}\right)\frac{s+b}{(s+a_2)^3}, \quad (88)$$

Finally,

$$C(s) = \frac{\frac{s^2}{K}\left(\frac{a_2^3}{b}\right)(s+b)}{1-\left(\frac{a_2^3}{b}\right)\frac{s+b}{(s+a_2)^3}}, \quad (89)$$

$$= \frac{\frac{s^2}{K}\left(\frac{a_2^3}{b}\right)(s+b)}{(s+a_2)^3-\left(\frac{a_2^3}{b}\right)(s+b)}, \quad (90)$$

$$C(s) = \frac{\frac{s^2}{K}\left(\frac{a_2^3}{b}\right)(s+b)}{s^3+3a_2s^2+a_2^2\left(3-\frac{a_2}{b}\right)s} \quad (91)$$

To get the lead of Section XI-A.2, we need

$$a_1>0 \text{ and } a_1=3b, \quad (92)$$

which simplifies (91) to

$$C(s)=\frac{\frac{3}{K}a_2^2(s+\frac{a_2}{3})}{s+3a_2} \quad (93)$$

To get to the form of (51), we match coefficients:

$$K_C = K_P+K_{D,i}=\frac{3}{K}a_2^2 \quad (94)$$

$$b = \frac{a_2}{3} \quad (95)$$

$$a_1 = 3a_2, \quad (96)$$

$$\frac{K_P}{K_P+K_{D,i}}a_1 = \frac{a_2}{3}, \quad (97)$$

$$\frac{K_P}{K_P+K_{D,i}}3a_2 = \frac{a_2}{3}, \quad (98)$$

$$\frac{K_P}{K_C} = \frac{1}{9}. \quad (99)$$

Finally,

$$K_P=\frac{a_2^2}{3K} \quad \text{and} \quad K_{D,i}=\frac{8a_2^2}{3K}. \quad (100)$$

Note that IMC gives a very specific set of relationships in order to generate the needed, practical lead controller this problem requires. Even though the double integrator is not asymptotically stable, the Youla-Kucera parameterization [61], [62] still yields good parameters.

C. Loop Shaping Summary

The examples of this section have tried to show how – within reason – the graphical loop shaping in the frequency domain favored by mechatronics oriented control engineers and the algebraic loop shaping in the transform domain favored by the process oriented control engineers, are largely working towards the same control design. The IMC examples above relied on the plant model, $\hat{P}(s)$, matching the true plant, $P(s)$, but the method robust to model inaccuracies is one of the foci of IMC research not covered here. In the case of the frequency domain methods, the robustness or lack thereof is very visual, showing up in the Bode plots. Still, what has been demonstrated is that we can draw a lot of intuition from the frequency domain to apply to our IMC, and we can draw a lot of parameter specifications from IMC to test in the frequency domain.

XII. Unifying Views of PID Controllers

Perhaps no single control technology would be embraced more universally by engineers and scientists both inside and adjacent to our field than higher performance, more universal PIDs. We make a few suggestions for how to get there:

- Embrace the digital. Don't obscure the relationships between CT and DT parameters.
  - We know the controller will be digital; let's let practicing engineers know how we handle it.
- Standardize simple model extraction for PID parameters from measurements.
  - Build these methods directly into our real-time controllers on parallel hardware.
  - Connect measurements, CAD tools (modeling & design), and implementation, so as to make iteration far more painless. The mantra here should be: "Connect, connect, connect."
- A common parameterization mostly that makes sense in both analog and digital helps. This paper has tried to show the utility of the continuous-time, no derivative filtering model of (22):

$$C(s) = K_P + \frac{K_I}{T_I s} + K_D T_D s,$$

  and sets $T_I = T_D = T_S$ then the backwards rectangular rule discrete equivalent means that:

$$C(z) = K_P + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1}).$$

- Do not be cavalier about the filtering. Respect the phase effects of noise and derivative filtering. That being said, be open to using filters for loop shaping to help make the open loop an integrator.
- The loop shaping discussion of Section Section XI showed that loop showed that frequency domain and IMC based loop shaping should not be viewed as opposites, but should be used complimentarily. The IMC gives a great starting point for picking controller parameters, while the frequency domain plots provide instant intuition about the margins (and therefore robustness) of the resulting controller design. On the other hand, it is the frequency domain that instantly informs us of the "need for lead", which guides how we construct our IMC problem.

XIII. Closing: Remember the Audience

This paper has tried to bridge the gap between the mechatronic and process control views of PID controllers. For those of us who spend a lot of time doing control research, PID may be viewed as:

- the main controller,
- a placeholder until we can insert our more sophisticated controller,
- a fundamental part of our overall, sophisticated controller,
- something for beginners, or
- all of the above.

However, for our colleagues in adjacent fields or for folks with little control background, PID *is* feedback control. It seems that we often miss opportunities to have the controls community lead in the machine intelligence world, because we do not spend enough effort reaching out to these people. If we did, then those working in machine intelligence would know that sometimes they need to call those folks who care about how their algorithms interact with the physics of dynamic systems.

PIDs are the "gateway drug" for these colleagues to embrace more feedback principles. Speaking with more commonality may reduce the number of published papers but will greatly improve our interaction with the larger world. Even if we in the the controls community have our own dialects, we should be willing to provide a "Rosetta Stone" of PIDs for those outside – but adjacent to – our field.

References

[1] K. Ogata, *Modern Control Engineering*, 3rd ed., ser. Prentice-Hall Instrumentation and Controls Series. Englewood Cliffs, New Jersey: Prentice-Hall, 1970.

[2] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 2nd ed. Menlo Park, CA: Addison-Wesley, 1990.

[3] K. Ogata, *Discrete-Time Control Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[4] R. C. Dorf, *Modern Control Systems*, 4th ed. Menlo Park, CA: Addison-Wiley, 1986.

[5] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 1st ed. Menlo Park, California: Addison-Wesley, 1986.

[6] G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 1st ed. Menlo Park, California: Addison-Wesley, 1980.

[7] D. J. Inman, *Vibration with Control*. West Sussex, England: Wiley, 2006.

[8] K. Ogata, *Modern Control Engineering*, 5th ed., ser. Prentice-Hall Instrumentation and Controls Series. Englewood Cliffs, New Jersey: Prentice-Hall, 2010.

[9] B. C. Kuo, *Automatic Control Systems*, 4th ed. Englewood Cliffs, NJ 07632: Prentic-Hall, Inc., 1982.

[10] ——, *Digital Control Systems*. Englewood Cliffs, NJ 07632: Prentic-Hall, Inc., 1980.

[11] K. J. Åström and B. Wittenmark, *Computer Controlled Systems, Theory and Design*, 2nd ed. Englewood Cliffs, N.J. 07632: Prentice Hall, 1990.

[12] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Upper Saddle River, New Jersey: Prentice Hall, 2006.

[13] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*, 1st ed. Prentice Hall, 2001.
[14] K. M. Moudgalya, *Digital Control*. John Wiley & Sons, 2007.
[15] D. M. Auslander and C. H. Tham, *Real-Time Software for Control: Program Examples in C*. Prentice Hall, 1989.
[16] D. M. Auslander, J. Ridgely, and J. Ringgenberg, *Control Software for Mechanical Systems: Object-Oriented Design in a Real-Time World*. Prentice Hall, June 27 2002.
[17] M. A. Johnson and M. H. Moradi, Eds., *PID Control: New Identification and Design Methods*. London: Springer-Verlag, 2005.
[18] K. J. Åström and T. Hägglund, *Automatic Tuning of PID Controllers*. ISA Press, 1988.
[19] ——, *PID Controllers: Theory, Design, and Tuning*. ISA Press, 1995.
[20] ——, *Advanced PID Control*, ser. Oxford Series on Optical and Imaging Sciences. Research Triangle, NC USA: ISA Press, August 15 2006.
[21] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*, 1st ed. Wiley, 1989.
[22] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, *Process Dynamics and Control*, 4th ed. Wiley, 2016.
[23] B. A. Ogunnaike and W. H. Ray, *Process Dynamics, Modeling, and Control*. Oxford, UK: Oxford University Press, 1994.
[24] F. Shinskey, *Process Control Systems*, 4th ed. McGraw-Hill, 1996.
[25] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Prentice Hall, 2006.
[26] T. Hägglund, *Process Control in Practice*. De Gruyter, 2023.
[27] T. Hägglund and J. L. Guzmán, "Give us PID controllers and we can control the world," *IFAC-PapersOnLine*, vol. 58, no. 7, pp. 103–108, June 2024, 4th IFAC Conference on Advances in Proportional-Integral-Derivative Control. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319305269
[28] H. De Bellescize, "La réception synchrone," *L'onde électrique*, vol. 11, pp. 225–240, May 1932.
[29] D. Y. Abramovitch, "Phase-locked loops: A control centric tutorial," in *Proceedings of the 2002 American Control Conference*, AACC. Anchorage, AK: IEEE, May 2002.
[30] R. R. Rhinehart, *Nonlinear Model-Based Control: Using First-Principles Models in Process Control*. ISA, 2024.
[31] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control 4 PID controller design," *Ind. Eng. Chem. Process Des. Dev.*, vol. 25, pp. 252–265, 1986.
[32] D. Y. Abramovitch, S. B. Andersson, K. K. Leang, W. S. Nagel, and S. D. Ruben, "A tutorial on real-time computing issues for control systems," in *Proceedings of the 2023 American Control Conference*, AACC. San Diego, CA: IEEE, May 31–June 2 2023, pp. 3751–3768.
[33] Wikipedia. (2018) Padé approximant. [Online; accessed June 6, 2018]. [Online]. Available: https://en.wikipedia.org/wiki/Pade_approximant
[34] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "Fitting discrete-time models to frequency responses for systems with transport delay," in *ASME 2011 International Mechanical Engineering Congress and Exposition*, ASME. ASME, 2011.
[35] L. Ljung, *System Identification: Theory for the User*, ser. Prentice-Hall Information and System Sciences Series. Englewood Cliffs, New Jersey 07632: Prentice-Hall, 1987.
[36] D. Y. Abramovitch, "A unified framework for analog and digital PID controllers," in *Proceedings of the 2015 Multi-Conference on Systems and Control*, IEEE. Sydney, Australia: IEEE, September 2015, pp. 1492–1497.
[37] ——, *Practical Methods for Real World Control Systems*. Self, December 1 2022.
[38] V. VanDoren, "Understanding PID control and loop tuning fundamentals," *Control Engineering Magazine*, 2023, [On line: Accessed September 30, 2022]. [Online]. Available: https://www.controleng.com/articles/understanding-pid-control-and-loop-tuning-fundamentals/
[39] J. R. Ragazzini and G. F. Franklin, *Sampled-Data Control Systems*. New York, N. Y.: McGraw-Hill Book Company, 1958.
[40] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Menlo Park, California: Addison Wesley Longman, 1998.
[41] D. Abramovitch, "A more unified view of anti-windup methods," in *Proceedings of the 2025 American Control Conference*, AACC. Denver, CO USA: IEEE, July 2025.
[42] K. J. Åström and L. Rundqwist, "Integrator windup and how to avoid it," in *Proceedings of the 1989 American Control Conference*, AACC. Pittsburgh, PA USA: IEEE, 1989, pp. 1693–1698.
[43] L. Rundqwist, "Anti-reset windup for PID controllers," in *Proceedings of the IFAC 11th Triennial World Congress*, IFAC. Tallinn, Estonia, USSR: IFAC, August 1990, pp. 453–458.
[44] K. J. Åström. (2002) Control system design lecture notes for me 155a. [On line; accessed June 9, 2017]. [Online]. Available: https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom.html
[45] A. Bemporad. (2010) Anti-windup techniques. [On line; accessed June 9, 2017]. [Online]. Available: http://cse.lab.imtlucca.it/~bemporad/teaching/ac/pdf/AC2-09-AntiWindup.pdf
[46] A. Ghoshal and V. John, "Anti-windup schemes for proportional integral and proportional resonant controller," in *Proceedings of the National Power Electric Conference*, India, 2010, department of Electrical Engineering, Indian Institute of Science, Bangalore, Email: anirban@ee.iisc.ernet.in, vjohn@ee.iisc.ernet.in.
[47] D. Y. Abramovitch, "Thoughts on furthering the control education of practicing engineers," *IEEE Control Systems*, vol. 43, no. 1, pp. 64–88, February 2023.
[48] ——, "Using feedback control principles as guiding metaphors for business processes," in *Proceedings of the 2022 American Control Conference*, AACC. Atlanta, GA: IEEE, June 2022, pp. 3088–3093.
[49] L. Ljung and T. Glad, *Modeling of Dynamic Systems*. Upper Saddle River, NJ: Prentice Hall, 1994.
[50] J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, 2nd ed. New York, NY: John Wiley & Sons, 1986.
[51] E. Levy, "Complex-curve fitting," *IRE Transactions on Automatic Control*, vol. AC-4, pp. 37–43, 1959.
[52] J. L. Adcock, "Curve fitter for pole-zero analysis," *Hewlett-Packard Journal*, vol. 38, no. 1, pp. 33–37, January 1987.
[53] J. L. Guzmán, D. E. Rivera, M. Berenguel, and S. Dormido, "i-pidtune 2.0: An updated interactive tool for integrated system identification and PID control," *IFAC-PapersOnLine*, vol. 45, no. 3, pp. 146–151, March 28–30 2012, 2bd IFAC Conference on Advances in Proportional-Integral-Derivative Control. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016310151
[54] ——, "i-pidtune 2.0: An updated interactive tool for integrated system identification and PID control," *IFAC-PapersOnLine*, vol. 58, no. 7, pp. 61–66, June 2024, 4th IFAC Conference on Advances in Proportional-Integral-Derivative Control. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896324007225
[55] D. Abramovitch, "A more unified view of loop shaping for PIDs," in *Proceedings of the 2025 American Control Conference*, AACC. Denver, CO USA: IEEE, July 2025.
[56] D. Y. Abramovitch, S. Hoen, and R. Workman, "Semi-automatic tuning of PID gains for atomic force microscopes," in *Proceedings of the 2008 American Control Conference*, AACC. Seattle, WA: IEEE, June 11–13 2008.
[57] D. Y. Abramovitch, "Trying to keep it real: 25 years of trying to get the stuff I learned in grad school to work on mechatronic systems," in *Proceedings of the 2015 Multi-Conference on Systems and Control*, IEEE. Sydney, Australia: IEEE, September 2015, pp. 223–250.
[58] R. A. Mueller, "Evolution of a simulator pilot force-feel system," in *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Modeling, Simulation, and Technologies (MST) Conference*, AIAA. Toronto, Ontario, Canada: AIAA, August 2–5 2010.
[59] ——, "Optimizing the performance of the pilot control loaders at the NASA vertical motion simulator," *American Institute of Aeronautics and Astronautics (AIAA) Journal of Aircraft*, vol. 47, no. 2, pp. 682–693, March-April 2010.
[60] ——, "Optimizing the performance of the pilot control loaders at the NASA vertical motion simulator," in *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Modeling, Simulation, and Technologies (MST) Conference and Exhibit*, AIAA. Honolulu, HI: AIAA, August 18–21 2008.
[61] D. C. Youla, H. A. Jabr, and J. J. Bongiorno, "Modern Wienr-Hopf design of optimal controllers: Part II," *IEEE Transactions on Automatic Control*, vol. 21, pp. 319–338, 1976.
[62] V. Kucera, "Stability of discrete linear feedback systems," in *Proceedings of the 6th IFAC World Congress*, IFAC. Boston, MA USA: IFAC, 1975.