Understanding the PID and its Tuning

E. H. Bristol
The Foxboro Co.
Dept. 0346, Bldg. N02-3E
Foxboro MA, 02035, USA
Tel. (508) 549-2019
Fax: (508) 549-3074
email: ebristol@foxboro.com

Classification: D.
Keywords: PID, Series Methods, Time Domain Analysis, Tuning, Eigenfunction Solutions.

Abstract

The PID (Proportional, Integral, Derivative) Controller is the keystone of practical continuous process control. It is simple. And yet it has never been formally understood in an academic framework (in the sense of developing general formulae relating realistic process performance to PID tunings) even for the linear process models that we assume to apply. A more formal understanding is necessary for teaching of the ever-widening range of control novices utilizing controls, and for the development of rigorous theoretical extensions. This paper addresses the basic PID (transient) behavior, with their eigenvalue decomposition, first with Root Locus and partial fraction expansion and then with Taylor series reversion methods, as the basis for a broader PID theory. The focus is on the time domain transients, since this is what a user experiences when actually operating and tuning the controller. The theory also supports more automatic tuning, particularly of the Pattern Recognition type of adaptation.

Introduction

PID control and its associated practices are unmatched in their simplicity and flexibility to address the needs of continuous process control[1] and the product of a long history of trial and error, and of analog based computation efforts. They successfully address all kinds of practical and operational issues. But their subtle, ad hoc solutions run counter to the focus of an increasingly complex process control world whose systems expertise is being stretched over a wider set of digital computer, communications standards, and practice issues. There is a need for a more analytical statement of the PID properties, both to make to it teachable to this wider community of people (who will not have the traditional time and tools to experiment) and to define its strengths (and weaknesses), once and for all, relative to more analytically transparent (but not necessarily superior) techniques.

The basic fact about the PID is that its terms represent the first three terms in a series to any basic controller.[1] As a consequence, any adequate PID theory is, in fact, an adequate theory of any similarly simplified linear controller. Such a theory is also a complete single loop control theory for the range of processes to which the PID is completely effective. The literature is full of experimentally developed prescriptions, relating process model forms to tunings, starting with the Ziegler-Nichols step transients[2], which fail to give uniformly tight and robust control to a predictably wide range of process forms.

In the opposite direction, tuning prescriptions based on adaptive or manual feedback from closed loop performance, including those of Ziegler-Nichols or those used in the original Pattern Recognition controller have been successful.[2-4] Several recent analytic approaches have, at least in experiment, overcome the limitations on getting sound, not loose, tunings from arbitrary process models, sensibly emphasizing compensation with minimum use of zeros, but still with an ad hoc character.[5-7]

A simple mechanism like a PID controller should permit a body of analytic methods which can accurately and rigorously (at least within the traditional linear theory) relate appropriate tunings with closed loop performance for any specified process model in terms of its parameters. It should shed theoretical light on the many traditional practices and questions: Is derivative control fundamentally incompatible with deadtime processes or just of small use? What is the role of complex zeros in PID tuning? What is the tuning role of different parameters?

PID tuning is often motivated by frequency domain arguments[3,5] even though tuning results are demonstrated by plotted transient responses. Tuning is a process of analyzing transients to get progressively better responses. Accordingly, the paper will analyze time domain transients to develop traditional tuning behavior. Analytically this means that the tuner (whether man or machine) recognizes the process from its visible closed loop eigenfunctions,

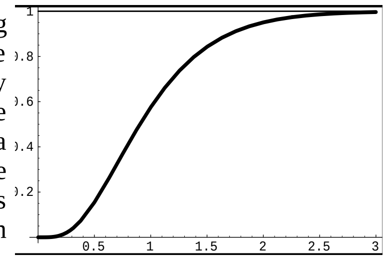
---

1. The practices developed this way for a very good reason: They were a commercial sales necessity (to sell hardware) to venders in their application efforts, for which they could not charge! **[Note: Footnotes will be indicated by numerical superscripts (1, 2, 3, …), whereas references will be indicated by bracketed numerical superscripts ([1], [2,3], [3-5], …).]**

and the affect on these of the appropriate tuning changes. Developing transient behavior in terms of eigenfunction components, rather than overall simulated result, opens up a range of intuitive and analytic possibilities.

The PID controller actually represents a family of different controller structures applied to a variety of process and disturbance forms. For clarity, the paper will take one member of this family and a representative process to illustrate an analysis that can then be extended to fit other forms. The paper will develop some aspects of the tuning transient behavior first with the classic single-loop design tool Root Locus, then include all aspects of the partial fraction expansion of the closed loop transient. This acts as a basis for a more general framework for computing rigorous tunings to achieve any specified response characteristics with a given process. In particular:

- The processes examined will model the common process control process forms having smooth monotonic, S-shaped step response and finite gain (usually modeled as deadtime and lag, but including deadtime and multiple lag). These are processes dominated by many right half plane real poles. Where a large number of small time constant (large pole value) poles dominate these processes approach the sharp deadtime response.[8] Where a small number of large time constant (small pole value) poles predominate, the response will show a more pronounced S shape.[2] The basic example used (at the right) combines large and small time constants to be simple enough to fully analyze, and involved enough to show the general issues. The affect of more analytically realistic processes will be illustrated by the application to a deadtime process. Processes with substantially different structural forms, with resonance or pure integration, need their own similar analyses.



$$\textbf{(1)} \qquad \frac{1}{(0.3s+1)^3\,(0.05s+1)^2}$$

- The analysis will be demonstrated on the basic (unfiltered) PID, expressible in three forms below, the first as the basic form,

$$\textbf{(2)} \qquad \frac{(as^2+bs+c)}{s} = k\frac{(s+\alpha)\,(s+\beta)}{s} = \frac{100}{P_B}(\tau_D s+1)\,\left(1+\frac{1}{\tau_I s}\right)$$

with $a=k$, $b=\alpha+\beta$, $c=\alpha\beta$ *or* $k=100\tau_D/P_B$, $\alpha=1/\tau_D$, $\beta=1/\tau_I$

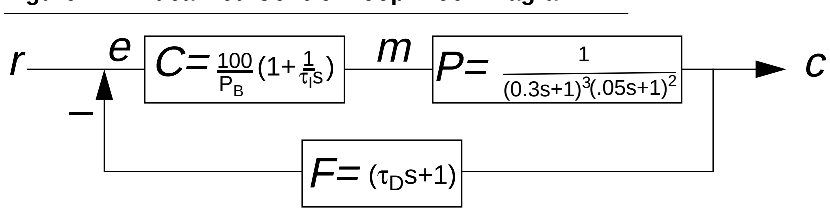
the second factored as in Root Locus analysis, and the third as a more traditional basis for relating to process times constants.[3] Except on the issue of complex zeros they are equivalent; identical, closed-loop responses result from each form with corresponding parameterizations. (Nonetheless their tuning practices might feel quite different.[4]) As with more realistic process forms, the extension of the analysis to include realistic derivative filtering is straightforward but unnecessarily complicating.

- The initial analysis will focus on step setpoint changes. The practical literature has recently emphasized the greater importance of load disturbances.[9] As above, the effect of different disturbance structures calls for altered analysis. The analysis of load changes introduced at the process input will be illustrated later.

The paper addresses not some unrealistic best PID structure and performance, but the general PID behavior and the computation of tuning and tuning behavior for any particular PID/process structure. It presents accurate new, analyticly manipulable, series methods for relating general open loop transfer functions and closed loop responses.

Root Locus, Partial Fraction Expansions, and the PID

Figure 1. Idealized Control Loop Block Diagram



The Root Locus[5] was developed as a vehicle for actual design from a known process model. In process control, except in theoretical or tutorial analyses, the process is not that well defined. Thus, the greater need is for an understanding and tool for progressive on-line tuning to the actual process. In this context, the Root Locus provides a good starting basis for understanding the tuning behavior of the PID controller. Figure 1 shows the chosen controller structure with the derivative action applied only to the disturbance. Note that the transfer function denominator and therefore the Root Locus are **not** affected by the placement of the Derivative action or load. The analysis will be parameterized in terms of the third PID form in Equation (2),

---

2. In the spirit of the reference[8] the examples will be normalized to unity gain and unity time constant sum.

3. Following usual practice, the parts of the controller transfer function applied to Setpoint and Measurement will be different, to partially accommodate their differences in affect on the process response. Again the choice will be arbitrary to provide an example for analysis; the optimal will always be dependent on the process/load structure.

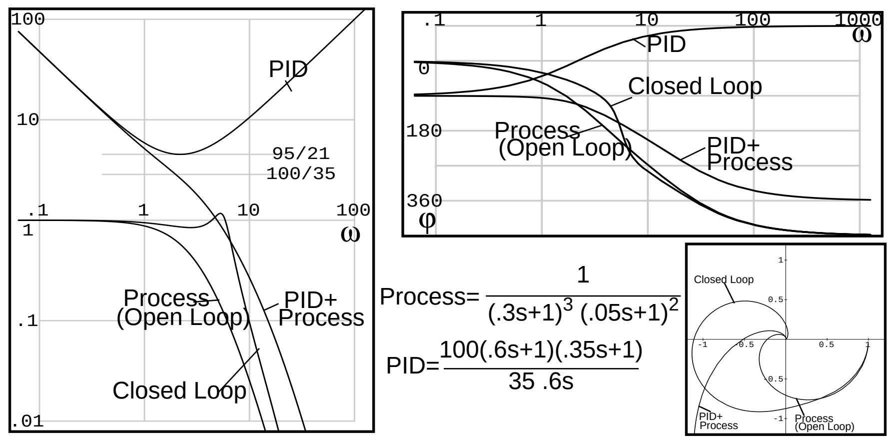
4. In particular, note the possibility of the so-called interactive Integral and Derivative tunings behaving identically (as lead time constants or zeros). With appropriate changes in P_B and accommodation of the different effects of filtering, Integral and Derivative settings can in fact be interchanged! Only convention (and filtering and hardware constraints) makes Derivative the “smaller” action. Practical implications will be elaborated later.

5. The locus of closed loop poles in the complex plane, as the controller gain varies from 0 to ∞.

with the closed loop setpoint change transfer function of Equation (3).[6]

$$\textbf{(3)}\qquad T_r(s) = \frac{\dfrac{\dfrac{100}{P_B}\left(1+\dfrac{1}{\tau_I s}\right)}{(0.3s+1)^3\,(0.05s+1)^2}}{1+\dfrac{\dfrac{100}{P_B}(\tau_D s+1)\left(1+\dfrac{1}{\tau_I s}\right)}{(0.3s+1)^3\,(0.05s+1)^2}} = \frac{100\,(\tau_I s+1)}{P_B \tau_I s\,(0.3s+1)^3\,(0.05s+1)^2 + 100\,(\tau_D s+1)\,(\tau_I s+1)}$$

Figure 2. Frequency, Phase, and Nyquist Plots for Example Case



For comparison purposes, Figure 2 shows frequency curves about the nominal parameter tunings of P=35, τ_I =.6 (β =1.67), and τ_D =.35 (α =2.86).[7] These tunings are chosen experimentally, for convenience, to correspond to the minimum integrated squared error for the setpoint changes. The discussion of transient tuning behavior of a loop is better developed in terms of the Root Locus, as shown in Figure 3. The figure shows the full Root Locus (in the upper left) with the important part for practice, expanded in the main plot. The discussion will justify this shortly. The Root Locus shows the behavior of the closed loop poles (one locus branch for each pole; six in this case) over the full range of controller gain values; the larger black dots show the position of the poles corresponding to the above tunings.

The initial focus is on the tunings about those nominal pole positions. The transient behavior is given from these poles as the partial fraction expansion of eigenfunctions each corresponding to one of the poles.[8] The reference or

$$\textbf{(4)}\qquad T_r(s) = \frac{100\,(\tau_I s+1)}{P_B \tau_I\,(\,(0.3)^3\,(0.05)^2 + 100\tau_D\tau_I)\prod (s+\alpha_i)} = \sum \frac{a_i}{(s+\alpha_i)} \Rightarrow c(t) = \sum a_i e^{-\alpha_i t}$$

the α_i being the roots of the denominator polynomial

setpoint and process step responses for the example case and nominal tunings are shown in Figure 4 [**See Large Figures at the end of the paper.**], with its component eigenfunctions.[9] The Root Locus real axis coordinates of each pole represent the exponent coefficient of its corresponding eigenfunction, and any non-zero imaginary coordinates represent the frequency of a resonant eigenfunction. In the figure, the eigenfunctions and Root Locus branches are labeled (as justified later), associated with the Proportional[10], Integral, and Derivative actions.

Most immediately, notice the negligible magnitude (in this case, enhanced by the fast decay) of the remaining high order eigenfunctions in the response. Conventional analyses assume that this proves that these terms can be ignored. In fact these poles have a major effect in shaping the Root Locus even as they do not show up visibly in the response.[11] The combination of high indirect effect and low direct visibility of the high order poles also explains a fre-

---

6. The following computations and figures were carried out and checked using Mathematica®, a computer mathematics package and useful tool for analysis. Such a tool facilitates the extension of the results to more realistic controller implementations and processes. Mathematica is a registered trademark of Wolfram Research Inc.

7. Of interest, the figure shows the PID behavior as an infinite feedback except for a filter notch about the closed loop resonant frequency

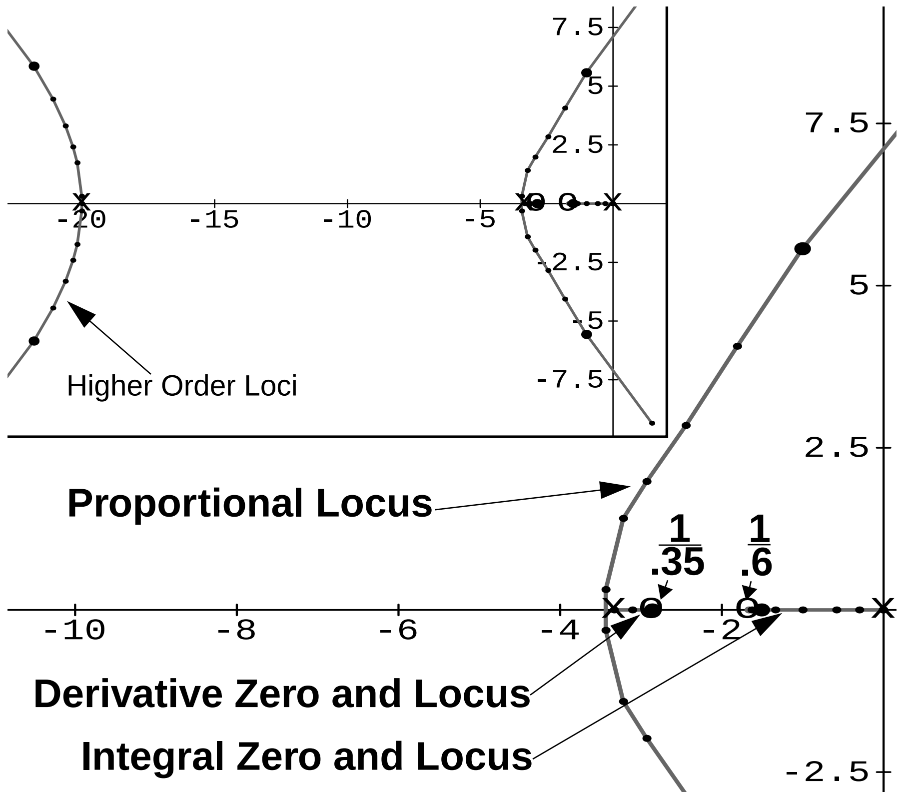
8. The conventional Laplace Transform texts normally express the residue a_i, for an isolated (as normal with Root Locus branch poles) pole α_ι in a transfer function $T(s)$ as $\lim_{s\to\alpha_i}(s-\alpha_i)\,T(s)$. For the later more general purposes this is best expressed as $a_i = R(\alpha_i) = 1/\Phi^{(1)}(\alpha_i)$ where the $\Phi^{(i)}(s)$ is the i*th* derivative of $\Phi(s)=1/T(s)$, and $R(s)$ is called the Residue function.

9. Imaginary terms from complex eigenvalues in the equation (4) cancel out, to leave corresponding damped sinusoids.

10. Note on the Proportional or Percent Proportional Band (P_B) action: Ideally sensors and actuators will each be sized for a natural coverage of the expected process range; this is a simplified basis for arguing that the scaled (digital or electronic) process gain will ideally equal 1. Under these conditions, the controller gain for dynamically simple processes with constant gain will be >1 (P_B<100%). More difficult processes, with dynamic complexity, changing gain, or nonlinearity, will have smaller gain (larger P_B).

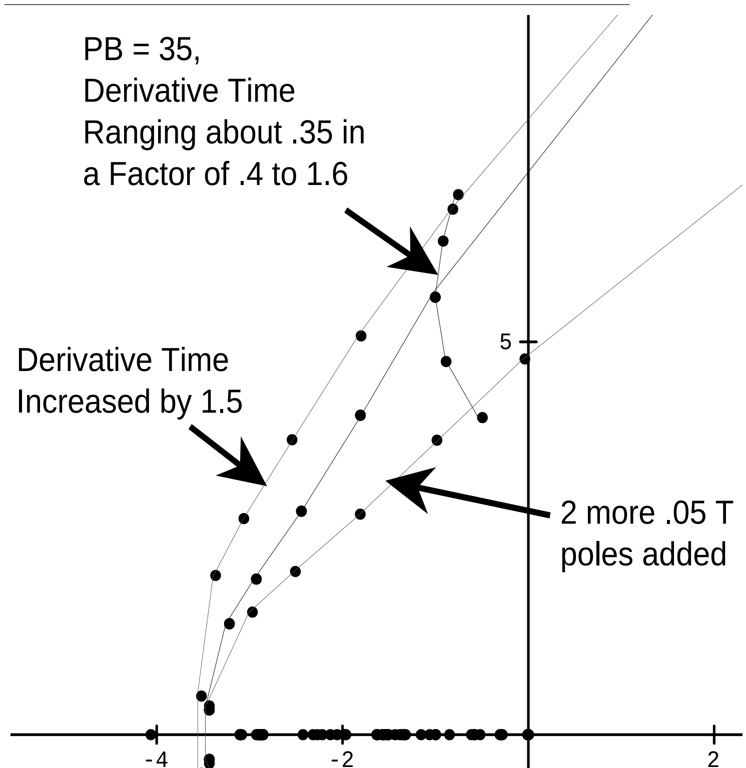
quently observed lack of robustness of model based adaptive algorithms. No identifier will accurately identify these terms from the process record even with their major effect on the Root Locus and on the pole movement under tuning changes.[12]

Figure 3. PID/Process Loop Root Locus



The justification for associating Integral and Derivative terms with Locus branches (and their eigenvalues/functions) is that, as each corresponding open loop zero is moved, any corresponding closed loop pole (in the associated branch) will move proportionately in the same direction; the Integral and Derivative zeros control the time constant of the corresponding closed loop eigenfunctions. In fact, for this case (and similarly tuned S-shaped step response processes), the optimally tuned Integral and Derivative time constants nearly equal the corresponding closed loop eigenfunction time constants. The Proportional setting most obviously affects the “Proportional eigenfunction”.

Figure 5. Root Locus Affected by Derivative Zero Adjustments



The eigenfunctions show the classic Proportional offset as an effect of moving the Integral zero so close to the origin (making the Integral gain 0 or time constant ∞) that its closed loop eigenfunction time constant lengthens correspondingly to the limiting point where it represents an offset. The Derivative zero has always been set conventionally to the left of the Integral zero, making the corresponding closed loop eigenfunction too fast to be as clearly distinguished in the combined step response. As a consequence, after initial recovery, the derivative effects are most visible as secondary effects on the other eigenfunctions. Figure 5 shows the effect of such changes. In this case the least squares, optimum derivative corresponds to the most stable (left-most from the imaginary axis) closed loop Proportional pole possible. At the same time increasing derivative time constants (decreasing derivative zeros) tend to increase the closed loop frequencies.

All of this illustrates the basis for typical tuning practice of this kind of control structure and process (iterated until the performance satisfies):

[11] The classic second order Root Locus extends to infinity perpendicular from the real axis. Higher order processes show Root Loci diverging symmetrically from the center of gravity of the poles until the loci closest to the real axis are close to parallel to it, as in the limiting case of the deadtime process. Thus the major effect of the secondary poles is to determine the general angle of the major Locus branches.

[12] On way of more effectively identifying these large (small time constant) eigenvalues would compute all eigenvalues collectively from transients of several different tunings, directly or indirectly computing the shaping affect on the critical (larger time constant) Root Locus branches.

- Tune the P_B for fastest possible response (with decreasing P_B) up to the point where the Proportional pole is insufficiently damped (or include a safety factor for the affect of the later tuning steps or process changes).
- Tune the Integral time (decreasing it from ∞) to remove the offset, but not so far that it adversely affects (undamps) the Proportional pole.
- Adjust the Derivative tuning so that the resulting closed loop settling time is reduced as much as possible without adversely affecting the stability of the Proportional pole.[13]

As a further aside, note that the combination of all eigenfunctions (as in Figure 4, and later in Figure 8) gives a response that appears more damped than is the actual case, as seen from the Proportional eigenfunction and its Root Locus branch.

Eigenfunction Tuning Behavior and Tuning Maps

The effects of the different tunings on the different eigenfunctions can be shown graphically in “tuning maps”, as in Figure 6. Each parameter is expressed as a nominal value divided or multiplied by a β run through the value range 0.0 - 1.5. The responsiveness and destabilizing effects of the Proportional setting on the Proportional eigenfunction are shown in the upper left hand map. The insensitivity of the Proportional eigenfunction to (reasonable) Integral tuning (and sensitivity to excessive tuning) is shown in the middle top row map. The offset tightening effect of all settings is shown in the second row. The period tightening effect of the Derivative tuning is shown in the top right map, as is the destabilization when the tuning is moved to either side of optimum.

From a tuning perspective, assuming a representative process, a user’s task is to pick the net response (the sum of (P), (I), and (D) eigenfunctions) which most matches the measured response shape (time scaled appropriately). This effectively identifies the closed loop behavior. He also needs to identify the desired response (as a second sum of eigenfunctions in the tuning map). The ratios of the corresponding settings for the two response shapes, represent the appropriate changes in the controller tunings, necessary to accomplish the desired control. In effect, the Pattern Recognition Adaptation (EXACT®) works this way.

Load Changes

The data so far has emphasized the behavior under set point changes. As shown in Figure 7 and Equation (5), load

(5)

$$T_d(s) = \frac{L(s)}{1+\frac{\frac{100}{P_B}(\tau_D s+1)\left(1+\frac{1}{\tau_I s}\right)}{(0.3s+1)^3(0.05s+1)^2}} = \frac{100(0.3s+1)^3(0.05s+1)^2\tau_I s A(s)}{(P_B\tau_I s(0.3s+1)^3(0.05s+1)^2+100(\tau_D s+1)(\tau_I s+1))B(s)} =$$

$$\frac{100(0.3s+1)^3(0.05s+1)^2\tau_I s A(s)}{P_B\tau_I((0.3)^3(0.05s)^2+100\tau_D\tau_I)\prod(s+\alpha_i)\prod(s+\lambda_i)} = \sum\frac{b_i}{(s+\alpha_i)}+\sum\frac{l_i}{(s+\lambda_i)} \Rightarrow \sum b_i e^{-\alpha_i t}+\sum l_i e^{-\lambda_i t}$$

disturbances have two closed loop effects: they may add additional poles, with time constants, independent of tuning, and outside the control loop; and they alter the amplitude of the eigenfunctions, **but not the Root Locus pole/eigenvalue positions**. The α_i are the same in Equations (4) and (5) but the a_i and b_i are different. The λ_i are independent of the tunings.

Figure 7. Idealized Control Loop with Load Block Diagram

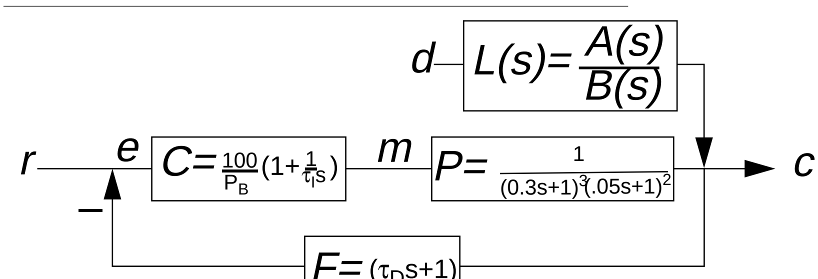
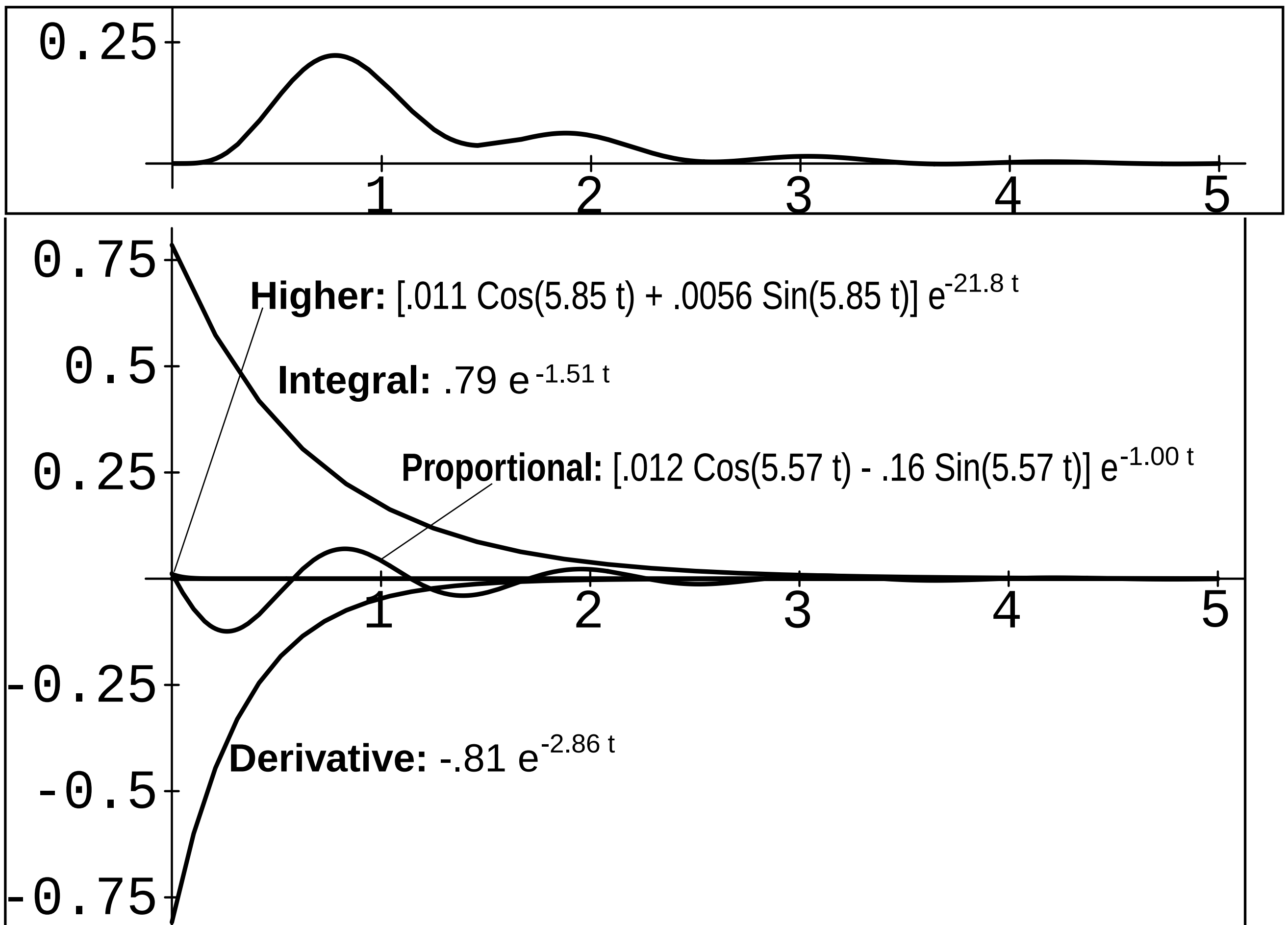


Figure 8 shows the response to a load disturbance introduced at the input to the process, under the same tunings as before with the setpoint changes. In this case, the load poles are cancelled out by the process model; there are no added constant poles.[14] Comparison to Figure 4 shows the change in eigenvalue magnitudes with no change in poles. The Proportional eigenfunction is much

13. This last step is an optimization whereas the first two usually can be seen as adjusting a monotonic characteristic of the response. Thus in the Pattern Recognition Adaptation it was always possible to recognize too much or too little Proportional or Integral action from a single transient. Derivative tuning requires multiple trend transients to recognize the appropriate tuning direction (or an assumed correlation to known process time constants).

14. $L(s)=P(s)$ in Figure 7. Pole structure is similarly unchanged with any other disturbance introduction directly between process input and output.

Figure 8. Response and Component Eigenfunctions under Process Input Load Step



smaller (reflecting the smoothing action of the load disturbance). The Integral eigenfunction is much larger (because Integral action takes a larger role in load recovery) and of opposite sign (because the compared disturbance and set-point recoveries are in opposite directions). The Derivative eigenfunction is more or less unchanged (slightly reduced).

The higher order eigenfunctions are much filtered/reduced like the Proportional eigenfunction, becoming almost invisible. Because of the reduced Proportional eigenfunction, the load response appears to show Proportional offset, and calls for moving the Integral zero to the left in the Root Locus plot (decreasing the Integral time constant). This is in fact a legitimate tuning correction. It gives a relatively less stable tuning than before but this is less apparent with the load disturbance since the influence of the less stable Proportional eigenfunction is correspondingly less.

But compared to the normal or model based intuitions about Derivative acting to anticipate load effects, it is paradoxical. In fact, Integral and Derivative actions are both represented by zeros, having similar effects; tuning the Derivative can also reduce the offset. The Integral tuning may be more effective because it starts with a larger initial value (closer to the disturbance time scale). In either case, the load tuning will generally be less stable than the set-point change tuning but this will be acceptable because the amplitude of the resonant eigenfunction will be smaller.

To summarize, the role of the Root Locus in the new analysis is to demonstrate the qualitative structure of the control behavior and the dominant role of the visible eigenfunctions in the response. The analytical development of the response behavior can then be developed from the dominant poles computed as a function of the tunings and of the full transfer function. The qualitatively structural issues turn out to be largely independent of the process detail. Thus the dominant poles of the example process behave similarly under tuning to the corresponding poles in other processes with monotone S-shaped step responses.

The Extreme Monotone: Pure Deadtime

As indicated, significantly different process, controller, and load structures will each call for their own variant of the analysis. One such common case, which is at the limit of the general S-shaped step response form previously addressed is the pure deadtime process. It illustrates the accommodation of the Root Locus discussion to the more realistic infinite order models. Figure 9 shows the experimentally developed, integrated least squared optimal, PI setpoint step response for a deadtime process, compared to the sum of its first six eigenfunctions. Among other things, this tuning results in a response only 12% worse than the best possible control, something observed by several authors previously. The lower figure shows a break-out of these eigenfunctions and the residual of all remaining eigenfunctions. In this case higher order terms are needed to give accurate responses but the qualitative affects can still be developed using the three dominant eigenfunctions.

The visibility of the effects of the higher order eigenfunctions corresponds to an easier identification and adaptation of deadtime in processes generally familiar to practitioners. But, within the methods described here, these eigenfunctions could be associated with higher order tuned controller pole or zero time constants, capable of differentiating between the normal integration step response ramp and the stair step response counterpart behavior seen in Smith Predictor style controllers. These additional terms would lend themselves to series solutions as described below. The visibility of these higher order eigenfunctions is a consequence of the sharp response discontinuities; introduction of process lags or smooth loads would restore the earlier behavior.

Derivative is left out of the tunings. For the step responses it is clear that the Derivative of the step response discontinuities would introduce infinite terms in the integrated square response. The question rises if load or other filtered responses would allow Derivative benefit. But a more basic problem occurs: Figure 10a shows the complex poles with or without a τ_D of .01. The low order eigenfunctions are unchanged. But at the level of the 31st complex pair of poles, the Derivative action results in instability. The issue becomes transparent using the Nyquist plot in Figure 10b. This behavior is an extreme case of a phenomena common to less extreme processes: Derivative action can make higher eigenfunctions more resonant and visible.

A Program for a PID Style Controller Theory

This section outlines a strategy for formal analysis of PID tuning from its transients, based on Taylor series between tunings and transient characteristics. As indicated earlier, the example case is only one of several distinct cases important to process control. Each case would be separately developed.

The analytic descriptions of the transients, for each case, can be developed according to the following strategy:

- An initial analytic study (like the above) of the class of process determines which eigenfunctions operate visibly, associating them with tunable open loop poles or zeros.
- For the chosen control and disturbance structure, define the transfer function with the process in general form *P*(s). For example:

$$T_r(s) = \frac{100(1+\tau_I s)P(s)}{100P(s) + \{P_B\tau_I + 100\tau_D P(s) + 100\tau_I P(s) + 100\tau_D\tau_I P(s)s\}s} \qquad \text{or}$$

$$T_d(s) = \frac{100\tau_I sL(s)}{100P(s) + \{P_B\tau_I + 100\tau_D P(s) + 100\tau_I P(s) + 100\tau_D\tau_I P(s)s\}s}$$

- Develop the terms in the characteristic equation of form $1+k\,K(s)$ where k is the loop gain (= $100/P_B$ above):

(6)
$$K(s) = \frac{(\tau_D s+1)(\tau_I s+1)}{\tau_I s}P(s)$$

- Develop analytic expressions for each of the dominant closed loop poles by expanding the characteristic equation for loop gain (or its inverse) as described later either about the open loop, controller based, poles or zeros, in a power or Taylor series, and invert this series to get a series for the pole in terms of the loop gain. The result, at the chosen level of truncation, is analytic, not only as a function of gain, but as a function of the Integral and Derivative parameters, which are included in the expressions for the open loop poles and the derivatives of *K*(s). This makes it appropriate, not only for human use but for inclusion in simple adaptive PID tuners, like the Pattern Recognition adaptation, where it can be used both for the identification and the tuning. This computation is simplest when the expansion takes place about a controller determined (analytic) open loop pole or zero, as with the Integral or Derivative poles. A variant strategy must be used for the Proportional poles.
- Compute the residues from the residue function for the process and disturbance ($R(s)$), as in Footnote 8.
- The sum of the dominant eigenfunctions now constitutes an expression for the controlled variable transient which is conveniently analytic and generally accurate. **No numerical methods have been applied to the process.**
- Once a general solution has been identified, representative sets of tuning values can be examined in detail to ensure that there are no "gotchas" like the earlier deadtime zero effects. Rigorous series convergence can also be addressed.

Generally the resonant eigenfunctions will determine the stability and sensitivity properties, whereas, the exponential eigenfunctions (with real poles) will shape the responses relative to the resonant eigenfunctions. Generally the resonant poles will be calculated as a trade-off between the amplitude of the resonance and its time constant (a faster and a smaller resonance both being preferred). Generally the exponential eigenfunctions will be computed as a trade-off between their time constant and any impact on the resonant eigenfunction; once the exponential eigenfunction's time constant gets smaller than a half cycle of the resonant eigenfunction it ceases to matter. The combined expressions are subject to all of the usual analytic manipulations and can then be used to:

- Shape or optimize the transients;
- Compute sensitivities;
- Explain, validate and refine traditional rules and concerns; and
- Generalize adaptive controllers for different process classes.

In greater detail: the task is now to compute the closed loop poles and corresponding residues of $T(s)$ as in Equation (4), developing the transient as the sum of the significant eigenfunctions.[15] Normally we would use some variant of Newton's method. But our program requires analytic results. These are achieved by Taylor series methods: The poles are the roots of the characteristic equation, which can be expressed (in terms of the above characteristic component transfer function $K(s)$ and variable gain k) as $\frac{1}{k} + K(s) = 0$ (in the region about $k=\infty$, and the corresponding open loop zeros) or (in terms of $\Phi_K(s) \equiv \frac{1}{K(s)}$) as $k + \Phi_K(s) = 0$ (in the region about $k=0$ and the open loop poles). The two forms can be re-expressed as equations expressing k and $1/k$ as functions of s. Either one of these can be expressed analytically as a Taylor series about $\underline{s}$, the appropriate open loop zero ($k \approx \infty$) or pole ($k \approx 0$):

$$\frac{1}{k} = -K^{(1)}(\underline{s})(s-\underline{s}) - \frac{K^{(2)}(\underline{s})}{2!}(s-\underline{s})^2 - \dots \text{ or } k = -\Phi_K^{(1)}(\underline{s})(s-\underline{s}) - \frac{\Phi_K^{(2)}(\underline{s})}{2!}(s-\underline{s})^2 - \dots .$$

The series for k and $1/k$ can be inverted, where each of the derivatives of the inverse is expressible in terms of the derivatives of the direct function.[10] Inversion of a series is sometimes referred to as reversion. The corresponding inverse series (about $\underline{k}$, usually set to, in the first case, to ∞ ($1/k=0$) and, in the second, to 0) are:

$$s = \underline{s} - \frac{\frac{1}{k} - \frac{1}{\underline{k}}}{K^{(1)}(\underline{s})} - \frac{K^{(2)}(\underline{s})\left(\frac{\frac{1}{k} - \frac{1}{\underline{k}}}{K^{(1)}(\underline{s})}\right)^2}{2K^{(1)}(\underline{s})} - \dots \text{ and } s = \underline{s} - \frac{k - \underline{k}}{\Phi_K^{(1)}(\underline{s})} - \frac{\Phi_K^{(2)}(\underline{s})\left(\frac{k - \underline{k}}{\Phi_K^{(1)}(\underline{s})}\right)^2}{2\Phi_K^{(1)}(\underline{s})} - \dots .$$

The first series is usually most appropriate for computing the (real axis) Integral and Derivative poles. Whichever series is appropriate, it will normally converge rapidly, giving good results even from the linear terms alone. While only the dominant poles need be computed, the critical effects of higher order dynamics are fully included in the $K^{(i)}$, $\Phi_K^{(i)}$ terms. Table 1 shows the calculated pole solutions for the closed loop real poles in the original example.

Table 1: Series Solutions to the Example Closed Loop Real Poles

Which Pole	0th Term	+1st Term	+2nd Term	"Exact" Solution
Integral	-1/.6=1.66667	-1.51962	-1.50962	-1.50986
Derivative	-1/.35=-2.85714	-2.86228	-2.8621	-2.86221

The end result, for such a process, is Equation (7) for the process response in terms of the three dominant sets of eigenfunctions, where, for example, E_I is given in Equation (8), truncated to three terms.

$$c(t) = E_P(t) + E_I(t) + E_D(t) \quad \textbf{(7)}$$

The complex (Proportional) poles can be computed analytically, extending the methods usually used to compute Root Locus asymptotes. In this case, the characteristic equations are expanded about the point at infinity on the two branches of the Root Locus closest to the positive real axis. The poles at infinity are converted to poles at zero in

15. When the poles in this substitution are complex ($s_i = \sigma_i \pm i\omega_i$), the corresponding complex residues ($a_i = \alpha_i \pm i\beta_i$) and eigenfunctions can be combined $a_i e^{p_i t} + \bar{a}_i e^{\bar{p}_i t} = 2e^{\sigma_i t}[\alpha \cos(\omega t) + \beta \sin(\omega t)]$, eliminating all complex terms and computing any resonant eigenfunction.

terms of a new variable $w = \frac{1}{s}$. As the limit of the Root Locus asymptotes, the expansion is now taken about a point having repeated poles.

(8)

$$E_I(t) = R\left(\frac{-1}{\tau_I} - \frac{\frac{P_B}{100}}{K^{(1)}(\frac{-1}{\tau_I})} - \frac{K^{(2)}(\frac{-1}{\tau_I})\left(\frac{\frac{P_B}{100}}{K^{(1)}(\frac{-1}{\tau_I})}\right)^2}{2K^{(1)}(\frac{-1}{\tau_I})}\right)e^{\left(\frac{-1}{\tau_I} - \frac{\frac{P_B}{100}}{K^{(1)}(\frac{-1}{\tau_I})} - \frac{K^{(2)}(\frac{-1}{\tau_I})\left(\frac{\frac{P_B}{100}}{K^{(1)}(\frac{-1}{\tau_I})}\right)^2}{2K^{(1)}(\frac{-1}{\tau_I})}\right)t}$$

Instead of expanding series for functions solving for k or $1/k$, we must expand the appropriate nth root of these functions (n being the number of asymptotes or repeated roots). Because the resulting expansion computes a pole (in s) close to the origin from a pole at infinity, the computation may take many terms for adequate accuracy. A slight cheat expands instead about a natural finite controller gain, perhaps computed by Newton's method. In this case, the expansion is now about a single pole in the region of interest and either of the simpler above inverse series may be used. From this starting point a rapidly convergent analytic approximation is achievable, based on the computed constant starting values of the pole and gain. For example, the position on the Proportional locus, for which neutral stability is achieved, can be computed by solving $\mathrm{Im}[\Phi(\underline{s})] = 0$ for $\underline{s}$ (on the imaginary axis, with no real part), and then solving $\underline{k} + \Phi(s) = 0$ for the corresponding $\underline{k}$. This inverse series solution is a natural basis for study of the closed loop Ziegler-Nichols methods.

The development of series with analytical terms allows the examination of causes in the same way as conventional exact analysis. For example, where a low order term gives accurate results and depends only on $K^{(0)}$ or $\Phi_K^{(0)}$ this proves a property being effectively dependent on process gain alone. Similarly, results involving only $K^{(1)}$ or $\Phi_K^{(1)}$ involve only first order dynamics (the sums of all cascaded deadtimes and lag time constants minus all lead time constants), and so on.[16] Occasionally these results will have to be buttressed by graphical methods or plotted simulation results, to highlight relationships or identify new ones that are not available analytically. In this way, the earlier Root Locus discussion pointed the way to the proper choice of poles for series expansion.

Summary

The PID is the basic control element of regulatory process control. It has been the basis of many techniques, applied intuitively by experts with the necessary experience and judgement. The modern era stretches the available expertise thin.This requires automatic techniques for specifying (and then compiling) designs in terms of standardized user objectives.

Formalisms are needed to better guarantee the automation and to provide the equivalent of the old control engineers experience, for the modern engineer who must master so many other aspects of a digital control world. As with his predecessors, he needs to know what to look out for, even as he has less time to look out for it. A PID analysis is essentially based on approximation, directly acknowledging that real process models are approximate. Without a theory of the PID, we in fact lack a valid theory for any continuous process control.

A basic theory can be developed, based on transient responses, which identifies the three or four key poles or eigenvalues and their eigenfunctions in the response, and permits accurate low order approximation for these eigenfunctions, accurately including the effect of higher order poles, and the effect of disturbances. The resulting series approximations is analytic in process and controller transfer functions to permit study of causes. There is still the need for empirical methods: simulation for the final design tests of novel complex designs, and appropriate graphics plotted to further visualize and extend the analytical methods. The combination makes the results of the analysis accessible to the working engineer.

Bibliography

[1] E.H.Bristol, "Even the PID Needs Theory/Software Advances", 1994 International Control Engineering Exposition, 3rd In-

[16]. These observations extend earlier methods used to demonstrate some of the counter-intuitive relations between steady state RGA values and closed loop dynamics.[11] The RGA results are easily explained as being derived from the real axis loci or from the linear approximations, thus dependent on gain alone.

ternational Control Engineering Conference, McCormick Place, Chicago, Mar. 14-16, 1994.

[2] J.G. Ziegler and N.B. Nichols, “Optimum Settings for Automatic Controllers”, Transactions of the ASME, Vol. 64, Mar. 1942, pp. 759-768.

[3] K.J. Åström and T. Hagglund, “PID Controllers, Theory, Design, and Tuning”, 2nd Edition, ISA, Research Triangle Park, NC, 1995.

[4] E.H. Bristol, “Process Systems Adaptation Beyond Math Models”, Workshop on Adaptive Control Strategies, Lake Kananakis, Alberta (organized by Dept. Chemical Engineering, University of Alberta), June 20-22, 1988, pp. 336-354.

[5] P.D. Hansen, “Controller Structure and Tuning for Unmeasured Load Rejection”, 1998 ACC, Philadelphia, June 1998, Vol.1, pp104-106.

[6] P.D. Hansen, “Robust Adaptive PID Controller TUning for Unmeasured Load Rejection”, To be published.

[7] S. Majhi and D. P. Atherton, “A New Smith Predictor And Controller for Unstable and Integrating Processes with Time Delay”, 37th IEEE Conference on Decision And Control. Tampa, Dec. 16-18, 1998, pp. 1341-1345.

[8] H. Paynter, “On an Analogy between Stochastic Processes and Monotone Dynamic Systems”, Regelungstechnik, Moderne Theorien und ihreVerwendbarkeit, R. Oldenbourg, Munich, 1957, pp.243.250.

[9] F.G. Shinskey, “How Good are our Controllers in Absolute Performance and Robustness?”, Measurement and Control, Vol. 23, May 1990, pp. 114-121.

[10]C. Carathéodory (Translator: F. Steinhardt), Theory of Functions of a Complex Variable, Chelsea Publishing Company, N.Y. 1958.

[11]E.H. Bristol, “Recent Results on Interaction in Multivariable Process Control”, Seminar on Recent Developments in Control and Estimation, AiChE 71st Annual Meeting, Miami Beach, Nov., 1978.

Large Figures (4, 6, 9, 10)

Figure 4. Setpoint Step Response and Component Eigenfunctions for Example Case

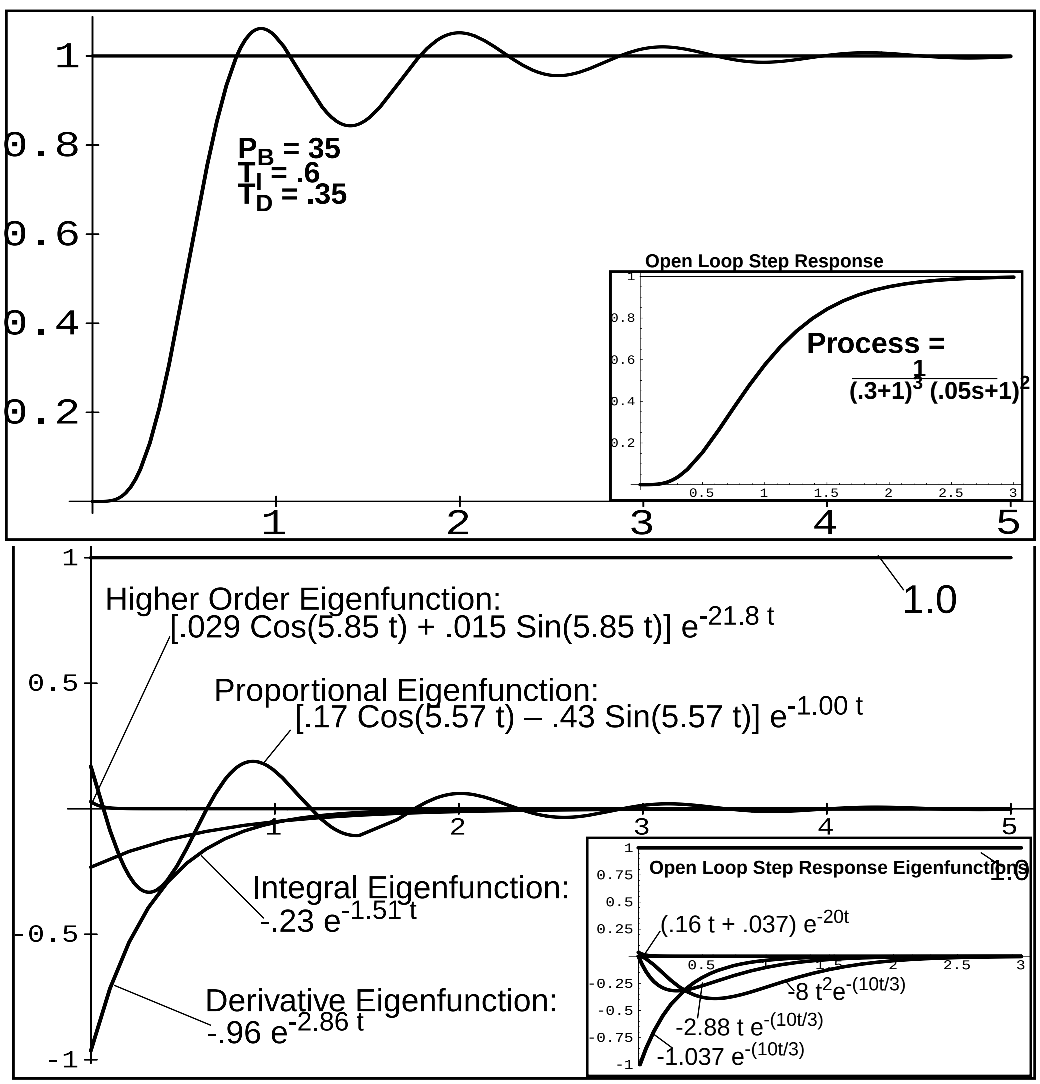


Figure 6. Tuning Maps for the Example Case

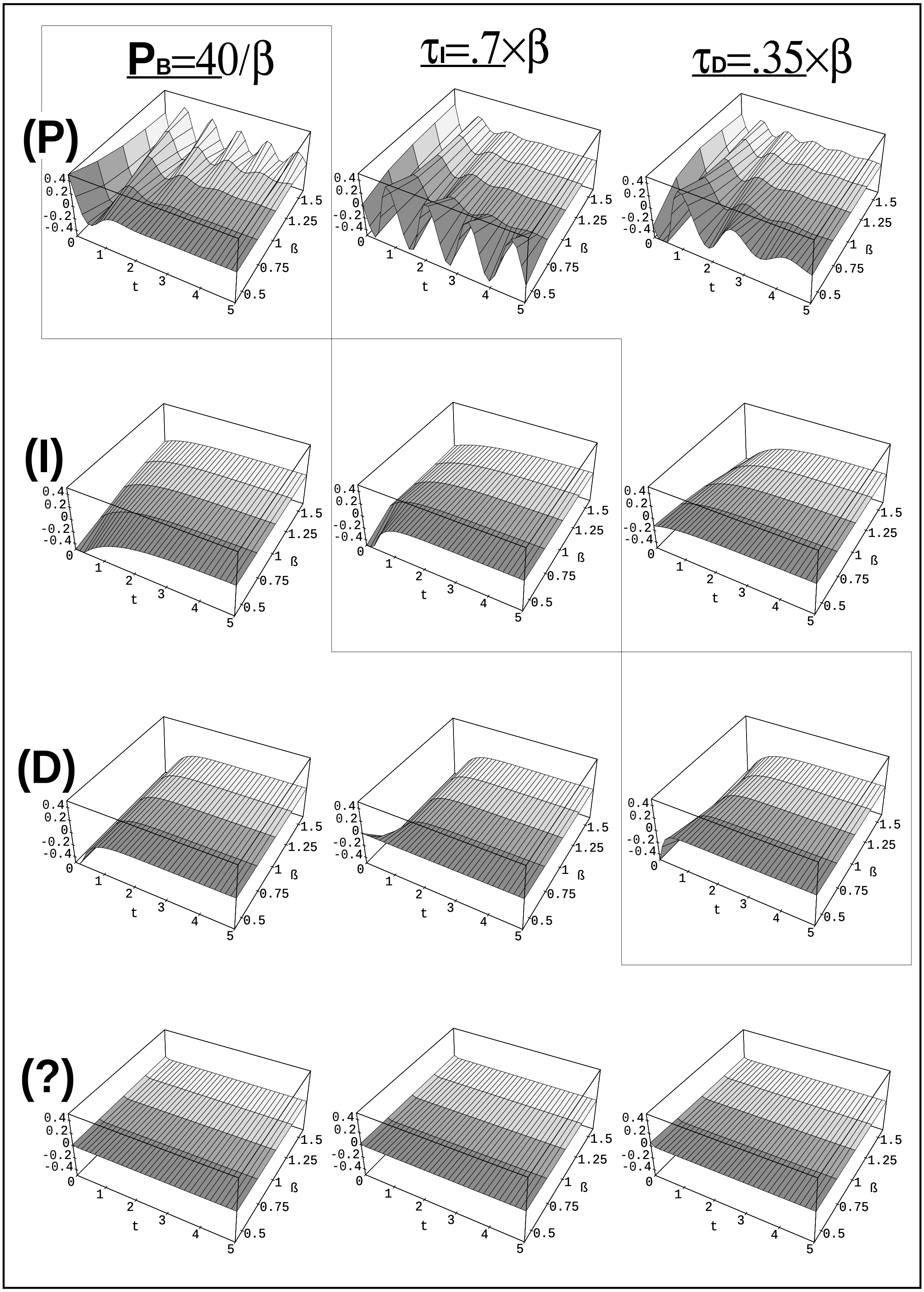


Figure 9. Deadtime Setpoint Step Response and Eigenfunctions

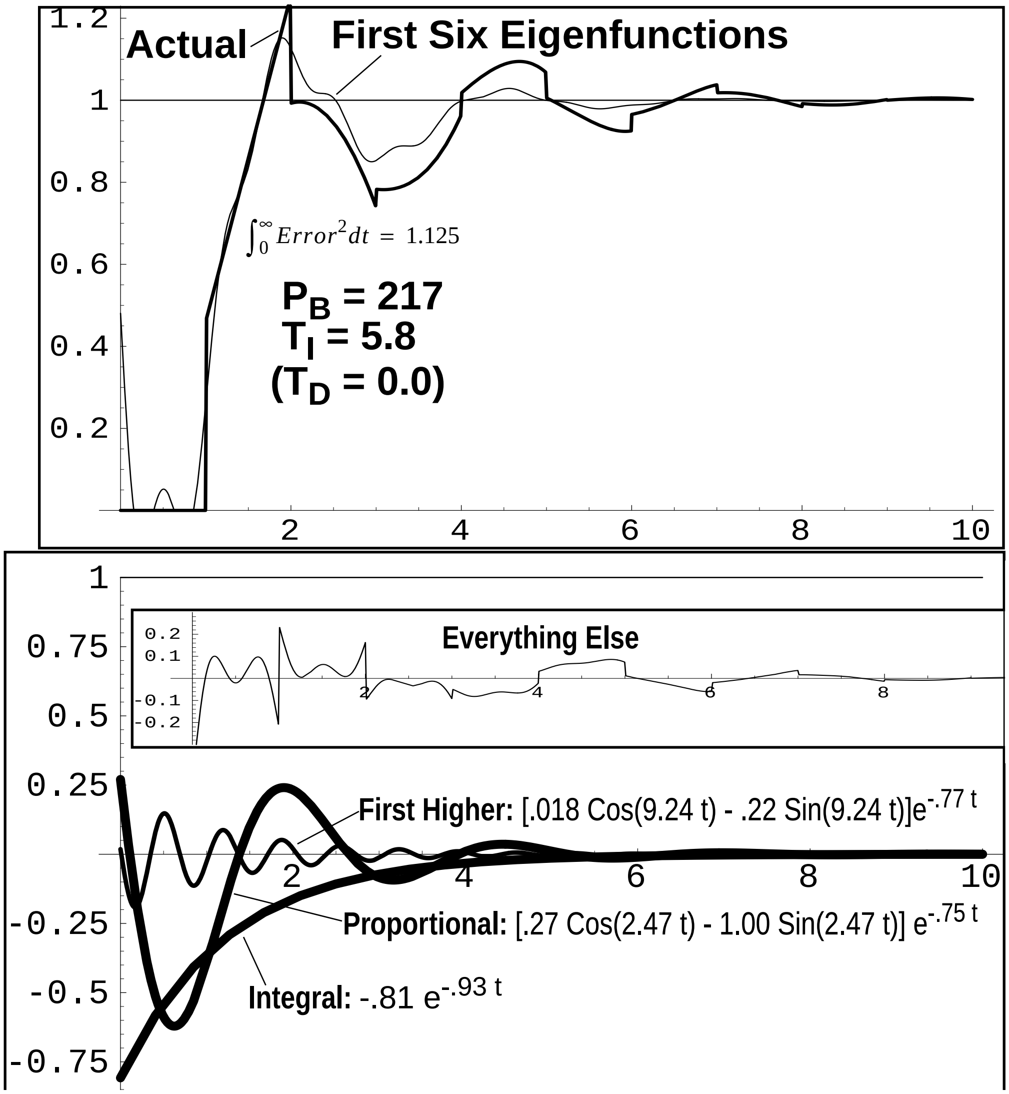


Figure 10. Deadtime with .01 Derivative: Pole and Nyquist Plots