SELF-TUNING PID CONTROL STRUCTURES

Peter J. Gawthrop [1]

1 Introduction

Symbol	Meaning
y	System output
u	System input (controller output)
w	Setpoint (possibly prefiltered)
K	Proportional gain
T_i	Integral time constant
T_d	Derivative time constant

Table 1: Notation

There has been much work on self-tuning PID controllers. Some of my own contributions, and extensive references to other work, are given by: Gawthrop (1986b) Gawthrop and Nomikos (1990) Gawthrop (1986a) Gawthrop, Nomikos, and Smith (1990) Gawthrop (1987) Gawthrop (1990). The key points that I want to make are:

- there are many possible structures for PID controllers;
- the concept of PID control can be usefully generalised to make contact with recent methods such as Internal-model control Morari and Zafiriou (1989) and Generalised Predictive Control (Clarke, Mohtadi, and Tuffs 1987a; Clarke, Mohtadi, and Tuffs 1987b; Demircioglu and Gawthrop 1991; Clarke 1994).
- one should not start with a PID controller and then decide how to tune it, but rather one should start with a rational design method and system model from which a (generalised) PID structure will then materialise.
- multiple-model self-tuning PID controllers give a neat way of handling nonlinear or time varying systems.

2 PID Structures and Their Generalisation

A good introduction to PID structures is given by Clarke (1984). In particular, he recommends the structure give in Figure 1. If the limiter is not saturated, the controller transfer function is given as:

$$u = K\frac{1+sT_i}{sT_i}[(w-y)-sT_dy] \tag{1}$$

This is one of the many forms of the PID controller equation. As discussed by Clarke (1984), this structure avoids integral windup when the limiter is saturated.
Figure 2 gives a generalisation of Clarke's structure: $U(s)$ is a ***unit gain, low-pass*** filter and $V(s)$ is a proper transfer function.

- because $U(s)$ has unit gain, the positive feedback loop yields anti-windup integral action;
- $V(s)$ provides a ***dynamic*** derivative time constant giving arbitrary mid and high-frequency loop shaping.

[1]Centre for Systems and Control & Department of Mechanical Engineering, University of Glasgow, Glasgow G12 8QQ, Scotland. Contact: **P.Gawthrop@eng.gla.ac.uk**

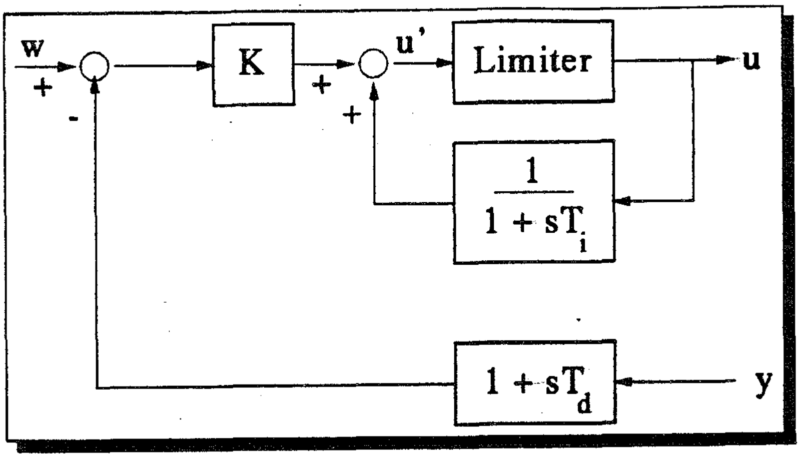

Figure 1: Practical PID Block Diagram

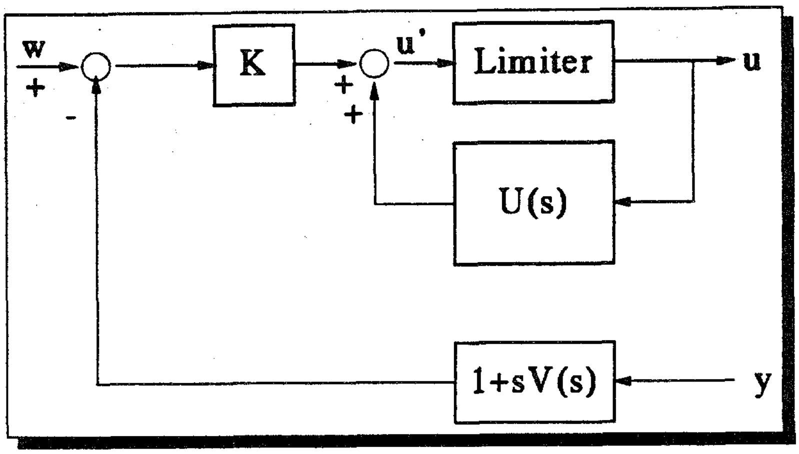

Figure 2: Generalised PID Block Diagram

3 Emulator-based Control

Emulator-based control (Gawthrop 1987) is a unifying approach to many forms of controller including generalised minimum variance control (Clarke and Gawthrop 1975) and generalised predictive control (Clarke, Mohtadi, and Tuffs 1987a; Clarke, Mohtadi, and Tuffs 1987b; Demircioglu and Gawthrop 1991; Clarke 1994). It can also be shown (Gawthrop, Jones, and Sbarbaro 1996) to be equivalent to internal model control.

Consider the control of a linear SISO system subject to a constant disturbance d. In Laplace transform terms this can be written as:

$$y(s) = \frac{b(s)}{a(s)}u(s) + \frac{d}{s} = \frac{sb(s)}{sa(s)}u(s) + \frac{da(s)}{sa(s)} \tag{2}$$

The second form of the expression has common denominators and is therefore suitable for emulator-based control design. As discussed by Gawthrop (1996), this leads to a controller equation of the form:

$$w = \frac{F(s)}{C(s)}y + \frac{G(s)}{C(s)}u \tag{3}$$

where $C(s)$, $F(s)$ and $G(s)$ are polynomials in the Laplace operator s. Because of the special structure of Equation 2, it can be shown (Gawthrop 1996) that:

$$\frac{F(0)}{C(0)} = 1;\ \frac{G(0)}{C(0)} = 0 \tag{4}$$

and the degrees of $G(s)$ and $C(s)$ are the same. It follows that:

$$\frac{F(s)}{C(s)} = 1 + s\frac{\tilde{F}(s)}{C(s)}; \; \frac{G(s)}{C(s)} = g(1 - \frac{\tilde{G}(s)}{C(s)}) \tag{5}$$

where $\frac{\tilde{G}(s)}{C(s)}$ has unit gain. Thus Equation 3 can be rewritten in the form:

$$u = \frac{\tilde{G}(s)}{C(s)}u + \frac{1}{g}[w - (1 + s\frac{\tilde{F}(s)}{C(s)})y] \tag{6}$$

With the addition of a saturation function, this form is identical to that of Figure 2 if:

$$K = \frac{1}{g}; \; U(s) = \frac{\tilde{G}(s)}{C(s)}; \; V(s) = \frac{\tilde{F}(s)}{C(s)} \tag{7}$$

Therefore the assumption of a constant disturbance *automatically gives a controller with a generalised PID structure.*

4 Self-tuning Control

Self-tuning can be added to any emulator-based structure; and can therefore give self-tuning (generalised) PID control. As discussed by Gawthrop (1986b), the constant disturbance assumption also leads to self-tuning algorithms which automatically pass data through a low-frequency blocking filter; thus removing any need to estimate the constant disturbance. This is the natural estimation counterpart to the integral action in the controller.
A recent extension of (generalised) PID control arose from research on Artificial Neural networks in general and Local Model Networks (Johansen and Foss 1993) in particular. As discussed by Gawthrop (1996), Local Model Networks lead to arrays of PID controllers suitable for the control of non-linear or time varying systems; the idea can be extended to self-tuning control of such systems (Gawthrop and Ronco 1996).

5 Conclusion

The basic concepts of PID control can be generalised within the same structure but allowing for the control of complicated dynamic systems using advanced control design algorithms. This structure arises naturally from the system description and does not need to be imposed artificially.
Recent advances in Local Model networks give a neat extension of the basic (generalised) PID structure to handle nonlinear or time-varying systems.
The PID concept is alive and well.

References

Clarke, D. (1984, Oct–Dec). Pid algorithms and their computer implementation. *Trans Inst. M.C. 6*(6), 305–316.

Clarke, D. W. (Ed.) (1994). *Advances in Model-based Predictive Control.* Oxford University Press.

Clarke, D. W. and P. J. Gawthrop (1975). Self-tuning controller. *Proceedings IEE 122*(9), 929–934.

Clarke, D. W., C. Mohtadi, and P. S. Tuffs (1987a). Generalised Predictive Control—Part I. The Basic Algorithm. *Automatica 23*(2), 137–148.

Clarke, D. W., C. Mohtadi, and P. S. Tuffs (1987b). Generalised Predictive Control—Part II. Extensions and Interpretations. *Automatica 23*(2), 149–160.

Demircioglu, H. and P. J. Gawthrop (1991, January). Continuous-time generalised predictive control. *Automatica 27*(1), 55–74.

Gawthrop, P. J. (1986a). Self-tuning control — an overview. *EI Technology 1*, 16–19.

Gawthrop, P. J. (1986b). Self-tuning PID controllers: Algorithms and implementation. *IEEE Transactions on Automatic Control AC-31*(3), 201–209.

Gawthrop, P. J. (1987). *Continuous-time Self-tuning Control. Vol 1: Design.* Lechworth, England.: Research Studies Press, Engineering control series.

Gawthrop, P. J. (1990). *Continuous-time Self-tuning Control. Vol 2: Implementation.* Taunton, England.: Research Studies Press, Engineering control series.

Gawthrop, P. J. (1996). Continuous-time local model networks. In R. Żbikowski and K. J. Hunt (Eds.), *Neural Adaptive Control Technology,* World Scientific Series in Robotics and Intelligent Systems, Vol. 15, pp. 41–70. Singapore: World Scientific.

Gawthrop, P. J., R. W. Jones, and D. G. Sbarbaro (1996, August). Emulator-based control and internal model control: Complementary approaches to robust control design. *Automatica 32*(8).

Gawthrop, P. J., P. Nomikos, and L. Smith (1990). Adaptive temperature control of industrial processes: A comparative survey. *Proc. IEE Pt.D 137*(3), 137–144.

Gawthrop, P. J. and P. E. Nomikos (1990, January). Automatic tuning of commercial pid controllers for single-loop and multi-loop applications. *IEEE Control Systems Magazine 10*(1), 34–42.

Gawthrop, P. J. and E. Ronco (1996, June). Local model networks and self-tuning predictive control. In *Proceedings of 4th IEEE Mediterranean Symposium on New Directions in Control and Automation.*

Johansen, T. A. and B. A. Foss (1993). Constructing NARMAX models using ARMAX models. *Int. J. Control 58*, 1125–1153.

Morari, M. and E. Zafiriou (1989). *Robust Process Control.* Englewood Cliffs: Prentice-Hall.