PID feedback control system design, performance, and tuning for robots.

Omar Osama, Cairo, 1020045@stemegypt.edu.eg

Abstract- **P**roportional-**I**ntegral-**D**erivative feedback control, or PID Controller, is one of the most widely used controllers in the robotics industry. Its ability to effectively regulate a range of processes and dynamic systems while having a very basic structure and easy tuning methods is the foundation of its success. Although it does not perform as well as contemporary control techniques, it is still the greatest place to start when developing the autopilot for an unmanned aircraft. In reality, PID Controls are used in most current attitude control features, whether they are found in open-source or commercial autopilots.

I. Introduction

The Proportional, Integral, and Derivative components work together additively to form the PID Controller. We frequently use P-controllers, PI-controllers, or PD-controllers because not all of them must be present. The PID controller will be discussed in the subsequent sections of this work; any other version may be obtained by omitting the pertinent parts.

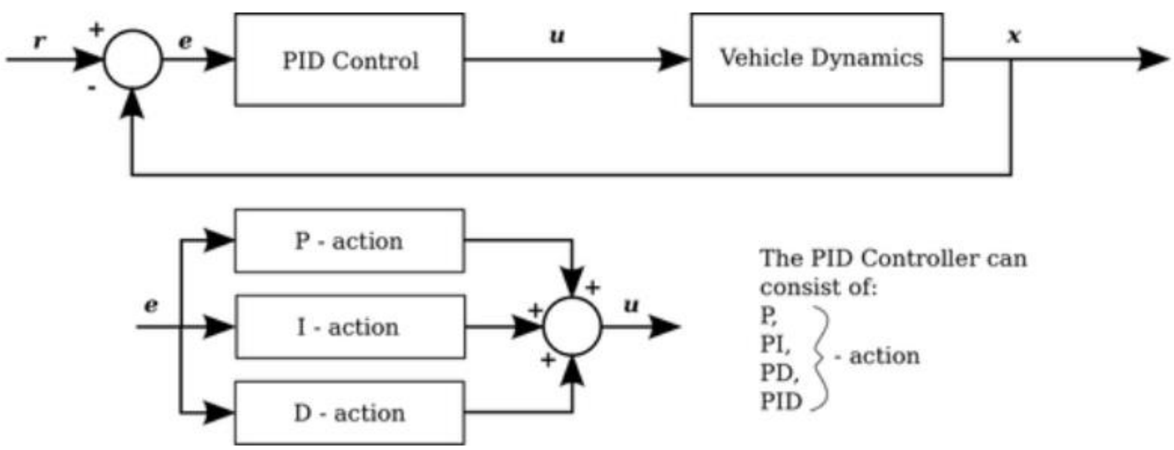

Figure 1 Block diagram of the PID Controller

The "tracking error" e and its three gains KP, KI, and KD serve as the foundation for the PID controller's operation. They result in the control action u when combined, as demonstrated by the following expression:

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

Figure 2 the general equation of the PID controller

The proportional term corresponds to the first term of the expression, the integral action to the second, and the derivative to the last one. Each of these terms plays specific roles in order to shape the transient and steady-state response of the closed-loop system.

II. Basic PID

We'll start by clearing up basic terminology for PID controllers. The output is referred to as the process variable (the measured position), and the reference is referred to as the setpoint (the desired position). Here are a few examples of typical variable name patterns for pertinent amounts.

r(t)	Setpoint	u(t)	Control input
e(t)	Error	y(t)	Output

The error e(t) is r(t) – y(t).

The proportional term drives the position error to zero, the derivative term drives the velocity error to zero, and the integral term accumulates the area between the setpoint and output plots over time (the integral of position error) and adds the current total to the control input. We'll go into more detail on each of these.

III. Proportional:

The P-action is the component mostly relevant with the dominant response of the system. It drives the position error to zero. where K_P is the proportional gain and e(t) is the error at the current time (t).

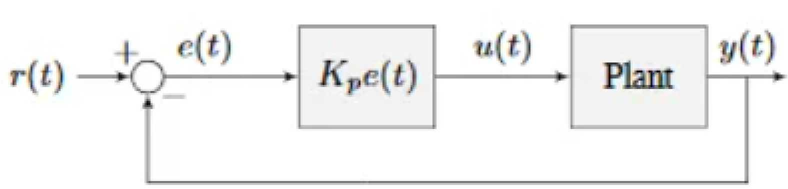

Figure 3 P controller block diagram.

Proportional gains pull the system in the desired direction like "software-defined springs." Recall from your physics classes that we represent springs as follows: (F = -k.x) where (F) is the applied force, k is a proportional constant, and x is the displacement from the equilibrium position. Where (0) is the equilibrium point being another method to express this (F = k (0 – x)). The equations relate one to one if we make the equilibrium point the setpoint of our feedback controller.

$$F = k\,(r-x)$$

$$u(t) = K_P.e(t) = K_P.(r(t) - y(t))$$

As a result, like a spring, the proportional controller pushes the system's output toward the setpoint with a force proportional to the mistake.

Also, increasing the P gain K_P typically leads to shorter rise times, but also larger overshoots. Although it can decrease the settling time of the system, it can also lead to highly oscillatory or unstable behavior.

IV. Derivative:

The velocity inaccuracy is reduced to zero via the derivative term. The derivative action adjusts for the error signal's rate of change, and it mostly affects how the closed-loop system's damping behavior is shaped.

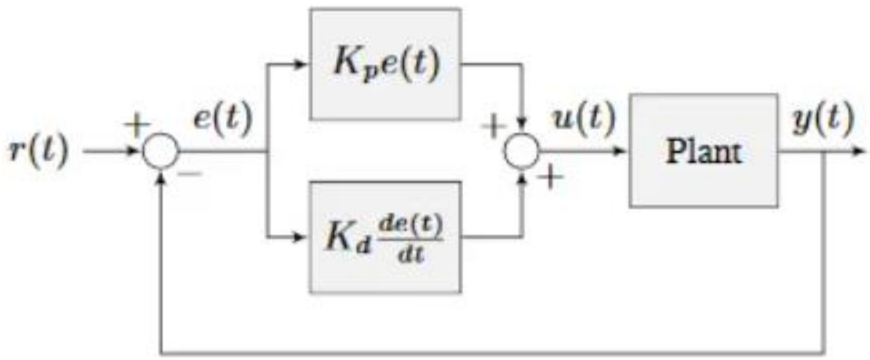

Figure 4 PD controller block diagram.

where K_P is the proportional gain, K_D is the derivative gain, and e(t) is the error at the current time (t).

A proportional controller for position (K_P) and a proportional controller for velocity (K_D) are both components of a PD controller. The way the location setpoint varies over time indirectly provides the velocity setpoint. We shall rewrite the equation for a PD controller to demonstrate this.

$$u(k) = K_P.e(k) + K_D\frac{ek-ek-1}{dt}$$

where u(k) is the control input at timestep k and e(k) is the error at timestep k. e(k) is defined as e(k) = r(k) – x(k) where r(k) is the setpoint and x(k) is the current state at timestep k.

$$u(k) = K_P.(r(k) - x(k)) + K_D\frac{(rk-xk)-(rk-1-xk-1)}{dt}$$

$$u(k) = K_P.(r(k) - x(k)) + K_D\frac{rk-xk-rk-1+xk-1}{dt}$$

$$u(k) = K_P.(r(k) - x(k)) + K_D\frac{rk-rk-1-xk+xk-1}{dt}$$

$$u(k) = K_P.(r(k) - x(k)) + K_D\frac{(rk-rk-1)-(xk-xk-1)}{dt}$$

$$u(k) = K_P.(r(k) - x(k)) + K_D\left(\frac{(rk - xk)}{dt} - \frac{(xk - xk - 1)}{dt}\right)$$

The term K_D slows the system down if it is moving since if the setpoint is constant, the implicit velocity setpoint is zero. Like a "software-defined damper," this operates. These are often found on door closers, and they have linearly increased damping forces.

In that sense, increasing the D gain KD, typically leads to smaller overshoot and a better damped behavior, but also to larger steady-state errors.

V. Integration

The integral action is frequently used to optimize the system's steady-state response and control its dynamic behavior. In essence, it gives the system memory. The integral phrase adds the current total to the control input after accumulating the area between the setpoint and output plots over time (i.e., the integral of position error). Integration is the process of adding up the space in between two curves.

$$u(t) = K_P.e(t) + K_I\int_0^t e(\tau)d\tau$$

where K_P is the proportional gain, K_I is the integral gain, e(t) is the error at the current time t, and τ is the integration variable.

The Integral integrates from time 0 to the current time t. we use τ for the integration because we need a variable to take on multiple values throughout the integral.

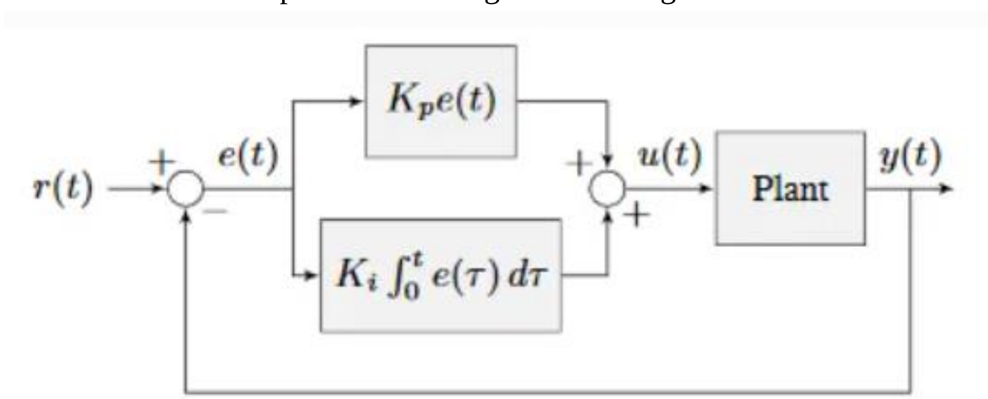

Figure 5 PI controller block diagram.

The proportional term may be insufficient to get the output all the way to the setpoint when the system is near to it in steady-state, in which case the derivative term is zero. The steady-state inaccuracy as indicated in figure 6 can occur from this.

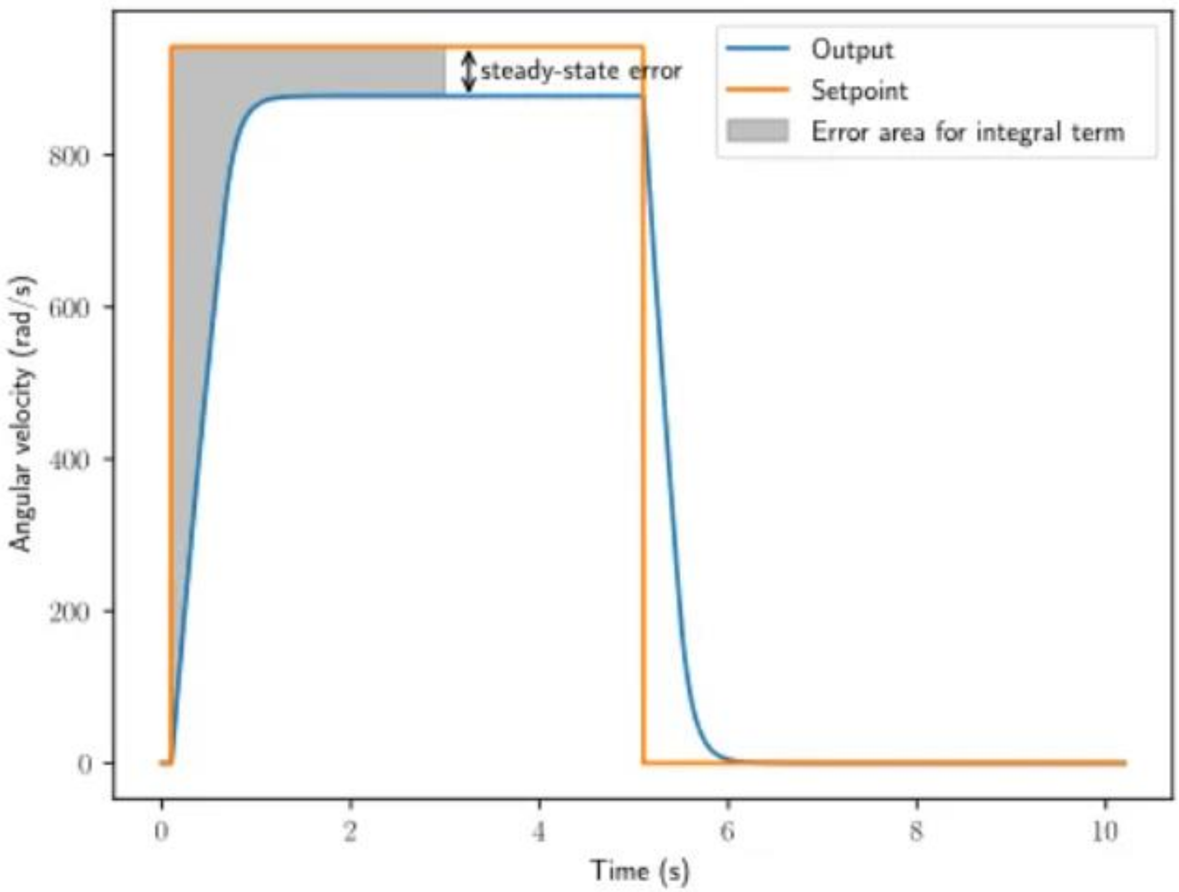

Figure 6 P controller with steady-state error.

Integrating the error and adding it to the control input is a popular technique for getting rid of steady-state error. The control effort is increased as a result until the system converges. Adding an integrator to the flywheel controller removes steady-state error for a flywheel, as shown in figures 6 and 7. Figure 8 illustrates how an excessive integral gain might cause overshoot.

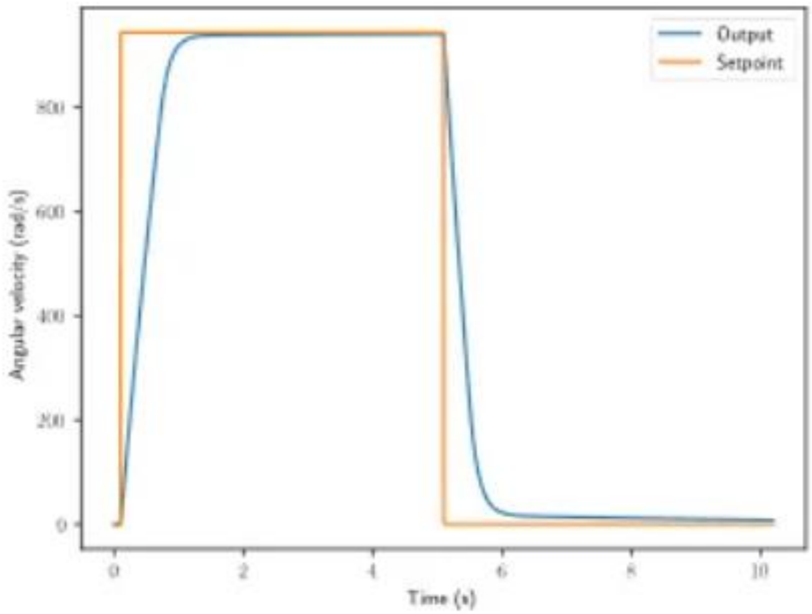

Figure 7 PI controller without steady-state error.

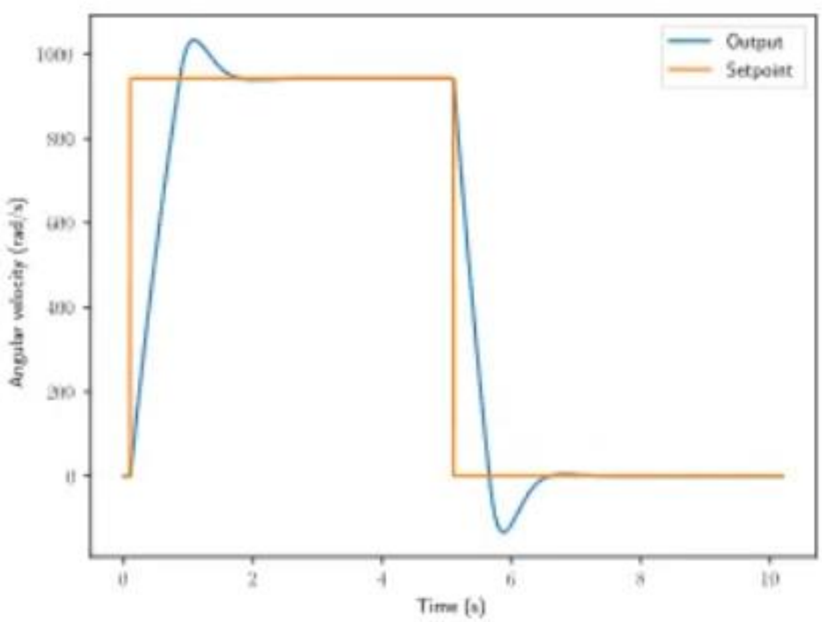

Figure 8 PI controller with overshoot due to higher KI gain.

Increasing the I gain K_I, leads to reduction of the steady-state error (often elimination) but also more and larger oscillations.

VI. PID collection

When these terms are combined, one gets the typical definition for a PID controller.

$$u(t) = K_P.e(t) + K_I\int_0^t e(\tau)d\tau + K_D\frac{de}{dt}$$

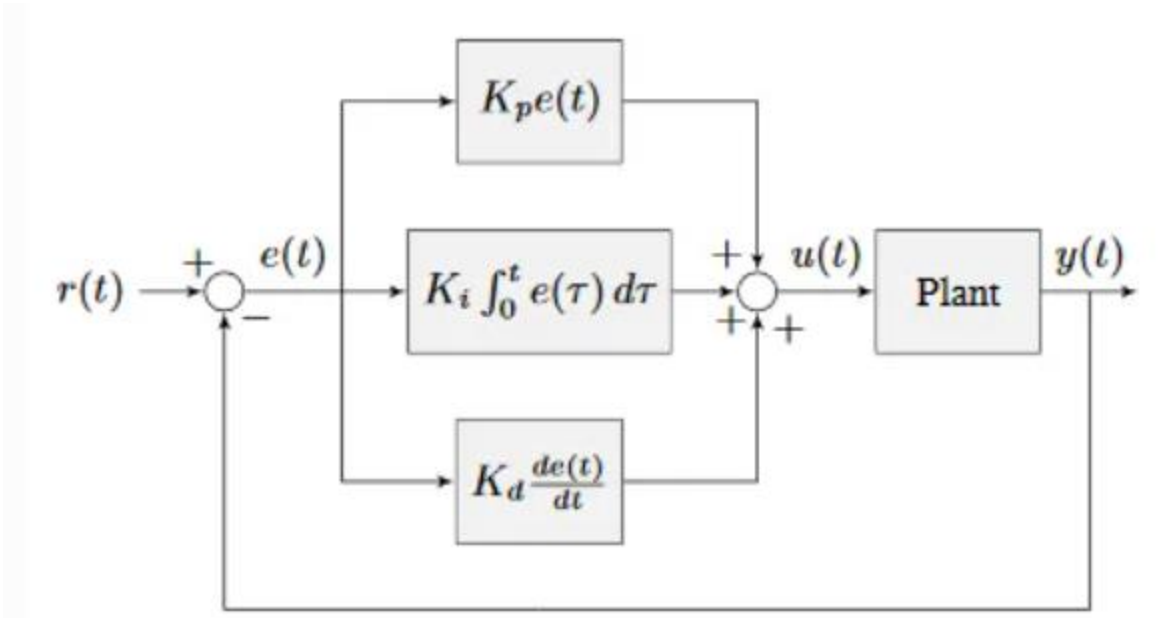

Figure 9 Block diagram of PID controller.

It is clear from this brief explanation of the function of each PID component action that the three separate gains cannot be individually tuned. Each of them really seeks to provide a desirable response characteristic (such as quicker response, damped and smooth oscillations, and close to zero steady-state error), but at the same time has a drawback that must be offset by fine-tuning a different gain. As a result, PID tuning is an iterative, highly connected process.

VII. Summary of tuning tendencies

CL Response	Rise Time	Overshoot	Settling TIme	Steady-State Error
KP	Decrease	Increase	Small change	Decrease
KI	Decrease	Increase	Increase	Eliminate
KD	Small change	Decrease	Decrease	No change

Figure 10 PID effect on responses.

VIII. Response Types

Underdamped, overdamped, and critically damped responses are the three main types of responses that a PID controller-driven system often exhibits. Figure 11 displays them.

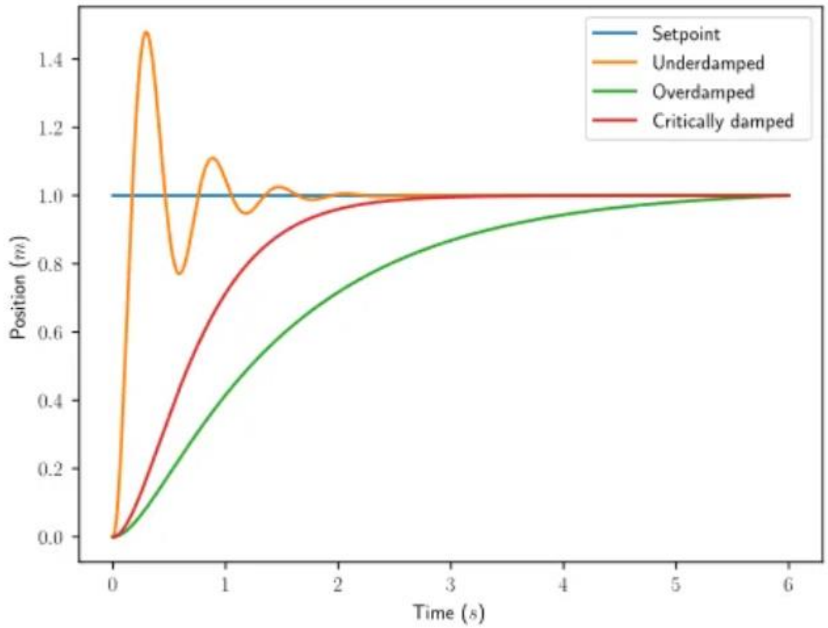

Figure 11 PID response types.

Rise time for step responses refers to how long it takes the system to reach the reference following the application of the step input. The system's time to reach the reference after the step input has been applied is known as the settling time.

Before settling, an underdamped response oscillates about the reference. An overdamped reaction rises slowly and doesn't exceed the reference point. The response that rises the fastest without exceeding the reference is one that is critically damped.

IX. Tuning PID

1. Set K_P, K_I, and K_D to zero.
2. Increase K_P until the output starts to oscillate around the setpoint.
3. Increase K_D as much as possible without introducing jittering in the system response.

Plot the position setpoint, velocity setpoint, measured position, and measured velocity. The velocity setpoint can be obtained via numerical differentiation of the position setpoint (i.e., v desired, k = $\frac{rk-rk-1}{\Delta t}$). Increase K_P until the position tracks well, then increase K_D until the velocity tracks well.

One can raise K_I such that the controller reaches the setpoint in an acceptable length of time if it settles at an output that is either above or below the setpoint. Integral control is greatly favored over a steady-state feedforward, though (especially for PID control).

Be careful since integral windup may happen if K_I is too large. The integral term can accrue an error greater than the maximum control input after a significant change in setpoint. As a result, the system overshoots and keeps growing until the built-up mistake is eliminated.

X. Conclusion

Both the PID controller's potent performance and its simplicity contribute to its success. Modern techniques are available now to best adjust such control regulations. However, the actual installation of PID controllers is a far more complex procedure. The following significant concerns, among others, must be taken into account while building flight control features utilizing PID controllers:

- The PID controller must be developed to take into consideration the limitations of the aerial vehicle's control margins.
- Due to the frequently critically unstable or steady qualities displayed by unmanned aircraft, the integral term requires extra consideration.
- Aerial vehicles are nonlinear systems, with the exception of hovering or trimmed flight. Given that the PID is a controller, it follows that it will not always behave correctly over the system's whole flight envelope. To deal with this reality, a number of approaches are used, such as Gain scheduling.

Finally, PID tuning is primarily a form of engineering that cannot just rely on automated methods but also requires the designer's skill.

XI. References

[1] K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and

Tuning. Research Triangle Park, NC: Instrument Soc. Amer., 1995.

[2], “PID Control,” in The Control Handbook, W. S. Levine,

Ed. Piscataway, NJ: IEEE Press, 1996, pp. 198–209.

[3], Benchmark Systems for PID Control, pp. 165–166, 2000.

[4], “The future of PID control,” Control Eng. Pract., vol. 9, no. 11,

pp. 1163–1175, 2001.

[5] C. Bohn and D. P. Atherton, “An analysis package comparing PID antiwindup strategies,” IEEE Control Syst. Mag., vol. 15, no. 2, pp. 34–40,

Apr. 1995.

[6] R. Cao and T. McAvoy, “Evaluation of a pattern recognition adaptive

PID controller,” Automatica, vol. 26, no. 4, pp. 797–801, Jul. 1990.

[7] G. Chong and Y. Li, “Trajectory controller network and its design

automation through evolutionary computing,” in Proc. EvoWorkshops

2000: Real-World Applications of Evolutionary Computing, Edinburgh,

U.K., Apr., pp. 139–146.

[8] (1998, May) Control Engineering—Single-Loop Controllers Dominate Marketplace. [Online] http://www.manufacturing.net/ctl/article/

CA189 397

[9] F. G. Shinskey, Feedback Controllers for the Process Industries. New

York: McGraw-Hill, 1994.

[10] (2004, Jun.) Patents—Patent Full-Text and Full-Page Image Databases.

[Online] http://ww.uspto.gov/patft

[11] H. J. Versteeg, H. J. Jansma, and K. Turner, “Evaluation of commercially

available adaptive controllers,” J. A, vol. 27, no. 3, pp. 120–126, 1986.

[12] L. Wang, T. J. D. Barnes, and W. R. Cluett, “New frequency-domain

design method for PID controllers,” Proc. Inst. Elect. Eng. D—Control

Theory Appl., vol. 142, no. 4, pp. 265–271, 1995.

[13] C. Wilson and C. Callen. (2004, Jan.) Close Process Control Translates To Quality Heat Treated Parts. [Online] http://www.industrialheating.com/CDA/ArticleInformation/coverstory/BNPCoverStoryItem/

0,2830,116383,00.html

[14] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” Trans. ASME, vol. 64, pp. 759–768, 1942.

[15] ABB. (2004, May) The ABB Group. [Online] http://www.abb.com

[16] ACT. (2004, May) ACT GmbH Homepage. [Online] http://www.actcontrol.com