Katholieke Universiteit Leuven

Department of Electrical Engineering

ESAT-SISTA/TR 1993-39

A tutorial on PID-control of linear systems[1]

Paul Vanvuchelen[2] and Bart De Moor[3]

December 1992

[1]This report is available by anonymous ftp from *gate.esat.kuleuven.ac.be* in the directory *pub/SISTA/vanvuche/reports/pidcontrol.ps.Z*

[2]ESAT - Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Leuven (Heverlee), Belgium, tel 32/16/22 09 31, fax 32/16/22 18 55, email: *paul.vanvuchelen@esat.kuleuven.ac.be* and *bart.demoor@esat.kuleuven.ac.be.* This research was sponsored by the Flemish Institute for Scientific and Technological Research in Industry (I.W.T.) and the Belgian programme on interuniversity attraction poles (IUAP-50) initiated by the Belgian State - Prime Minister's Office - Science Policy Programming. The scientific responsibility is assumed by its authors.

[3]Bart De Moor is a research associate with the N.F.W.O. (Belgian National Fund for Scientific Research).

Abstract

The problem of tuning the parameters of a PID-controller arises in many engineering applications. Often, different measures of quality must be considered simultaneously, so performance trade-offs are inevitable.
In this work, some tuning procedures for PID-controllers are presented. First, a stabilizing controller is designed, next, the controller parameters are changed to obtain the desired performance.
To allow the designer to work quickly, the interactive design tool *PID-tuner* is provided. At this moment, only the trial-and-error method is implemented: once the designer has entered the control parameters, the closed loop behavior can be observed. To prevent random search strategies, classical and experimental tuning rules are provided in this text. Sophisticated tuning methods, based on multi-performance optimization, will be implemented in the near future.

Chapter 1

Theoretical aspects of control

1.1 Introduction

The purpose of control is to generate input signals to a process (control signals) such that relevant variables in the process behave in a desired way. In output control, only the output variables are relevant (cfr. in state control, not only the outputs but also the system's states are important).
Two main problems are distinguished:

- *The tracking problem:* The aim of tracking control is to let the output and/or the states track a desired trajectory or reference signal as closely as possible.
- *The regulator problem:* The aim is to keep the system's output as closely as possible to a fixed set point (keep the system in an operating point), in the presence of disturbances which act on the system or come from the measurements.

Generically, both problems must be solved simultaneously: we would like to track a reference trajectory/signal as closely as possible (including a good steady state accuracy), even if there are disturbances and process uncertainties.
Two illustrative examples are given below.

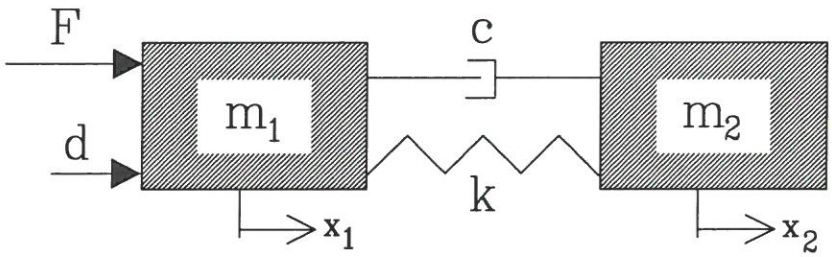


Figure 1.1: *The classical double disk problem.*

Example 1 *The system of interest consists of two masses m_1 and m_2, connected by a spring k and a damping element c. A control force F on the mass m_1 is used to change the position of the smaller mass m_2 (figure 1.1). F is known exactly, since it is delivered by a force*

controlled actuator, while d is an unmeasured, disturbing force. From a controller's point of view, the system has two inputs, F and d, and a single output x_2.
The differential equations to describe the dynamics of this system are:

$$F + d - k(x_1 - x_2) - c\frac{d(x_1 - x_2)}{dt} = m_1\frac{d^2x_1}{dt^2}$$

$$k(x_1 - x_2) + c\frac{d(x_1 - x_2)}{dt} = m_2\frac{d^2x_2}{dt^2}$$

After applying Laplace's transformation, the input-output relation becomes:

$$X_2(s) = P(s)F(s) + P(s)d(s)$$

with $P(s)$ equal to:

$$P(s) = \frac{\frac{1}{m_1}(\frac{c}{m_2}s + \frac{k}{m_2})}{s^2(s^2 + c(\frac{1}{m_1} + \frac{1}{m_2})s + k(\frac{1}{m_1} + \frac{1}{m_2}))}$$

Tracking control requires that the position x_2 of the mass m_2 follows a reference trajectory as closely as possible, while regulator control tries to position the mass m_2 at a fixed position, even in the presence of disturbances.

In the literature, different practical control problems are solved by designing controllers for the equivalent double disk systems:

- *In space technology, geostationary satellites are mounted with receivers on flexible antennnas, to pick up signals from earth (figure 1.2). Once the satellite describes a fixed orbit around the earth, no efforts are required to keep it there, thanks to an equilibrium between the gravity force and the centripetal force. For the same reason, no directioning problems are expected once the satellite is directioned well. However, disturbing sunwinds cause the satellite to spin. The directioning of the receivers towards earth, in the presence of disturbances, is a typical regulator problem that space engineers are faced with.*
- *Another example of a double disk system can be found in robot technology. In industrial assembly environments, most robots are conceived as massive elements, that execute a specific task at a fixed place along the production line. When doing position control of the robot hand at low speeds, their big inertances are advantageous. However, to increase the speed of positioning, lighter robots are required. As a result of this tendency, flexible robots arised. To study the behavior of one flexible link of such a robot, a flexible beam with a payload is connected to a torsional beam, which is itself connected to a direct drive motor (figure 1.3). In most cases, the payload is required to follow a reference trajectory, so a tracking problem needs to be solved.*

Example 2 *Consider a classical disk player. To allow the listener to use the disk player for different types of disks (singles, LPs and oldies), it is important that different rotation speeds can be realized accurately, namely 45 rpm for singles, 33 rpm for LPs and 78 rpm for oldies.*

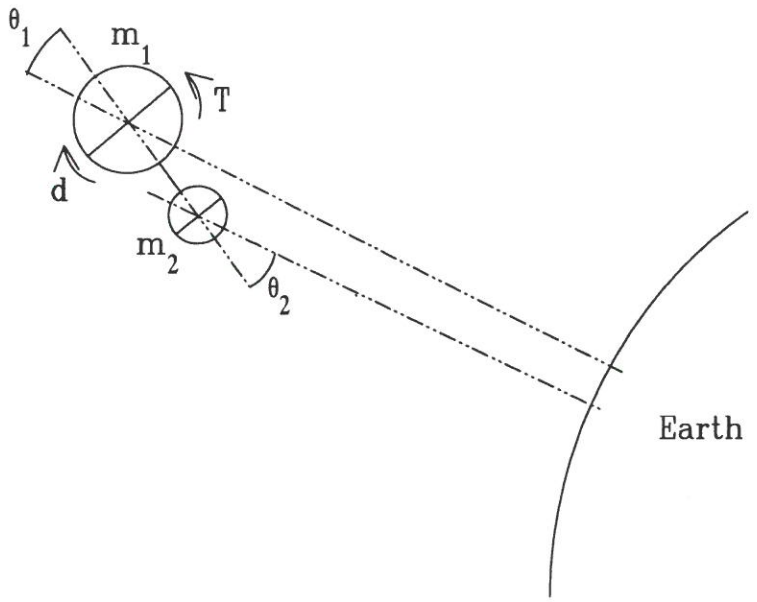


Figure 1.2: *Receiver control of geostationary satellites as a double disk regulator problem: due to the sunwind (disturbance torque d), the satellite m_1 will spin (indicated by the angle θ_1). A receiver m_2 is connected to the satellite by a flexible antenna (k, c). To guarantee a high performance of the system, the deviation angle θ_2 of the sensor must be zero under all circumstances. A torque T, delivered by the satellite motor, is used to compensate for the disturbances.*

In [6], a linear model $P(s)$ is used to describe the relation between the input voltage $v(t)$ (volt) and the angular velocity $\omega(t)$ $(\frac{rad}{s})$:

$$P(s) = \frac{\Omega(s)}{V(s)} = \frac{\pi}{(s+3)(s^2+2s+2)}$$

After the controller increased/decreased the angular velocity to a desired value, in a dynamically acceptable way (a tracking control problem), this speed must be maintained accurately, even in the presence of disturbances (a regulator control problem).

To generate the control signals that give the plants the desired behaviour, a well-thought control system is required. How such a control system can be designed, is explained in the next section.

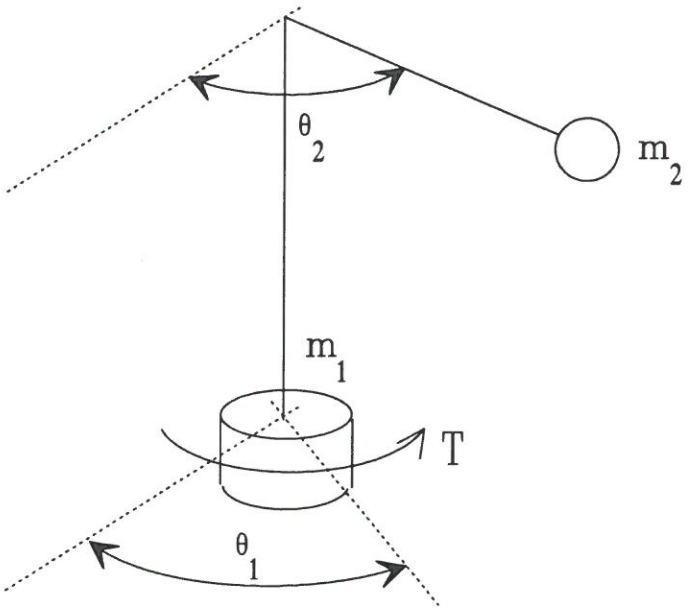


Figure 1.3: *Payload position control as a double disk tracking control problem: it is theoretically verified that a one-link flexible robot can be modelled by a double disk system, if only the first flexible mode is considered (note that a flexible system has an infinite number of eigenfrequencies). Therefore, it is sufficient to solve the tracking problem for the double disk system. In this case, the position θ_2 of the payload m_2 must be controlled by applying a control torque T to change the angular position θ_1 of the motor inertia m_1. The stiffness and damping of the beams is represented by k and c. The uncertainty on the system arises from the neglected dynamics.*

1.2 Development of a control system

A possible procedure to develop a suitable control system is shown in figure 1.4. From this flow-diagram, it follows that different activities need to be done in a well-defined sequence. Although the specific aspects will be explained in the next sections, the various activities are shortly described here:

- *Open Loop Analysis:* what kind of system is considered?

  Initially, the system to be controlled needs to be analyzed. Since a model of the system is assumed to be available, the system can easily be analyzed in the frequency domain and the time domain.

- *Performance Specifications:* how is the system required to behave?

  Once the system is analyzed, the desired performance must be expressed in terms of the different performance measures that are available. The importance of the various performance measures depends on the type of control problem to solve.

- *Control Configuration:* which signal(s) are used to calculate the control signal?

  Depending on the plant, the desired performance specifications and the allowed complexity of the control system, a control configuration must be chosen. Depending on

the type and the number of input signal(s) to the controller, different configurations are recognized.

- *Control Law:* which algorithm is used to calculate the control signal?

  Once the control configuration is fixed, the control law must be chosen. When the designer is experienced and familiar with a specific control law, it is good practice to choose that law first!

- *Parameter Design:* which are the parameters of the algorithm to calculate the control signal?

  Once a control law is chosen, the different control parameters must be tuned. For each control law, different quantitative techniques and/or qualitative rules are known.

- *Evaluation:* how will the controlled system behave in theory?

  As soon as the controller is tuned, the controlled system must be evaluated. When the specifications are met, the controller can be implemented. In the other case, when not all specifications are met, different actions may be undertaken, respecting the following order:

  - Change the actual parameters of the controller, using the available techniques and rules.
  - Change the control law, for example by adding or changing a controller parameter (e.g. from PI- to PID- or PD-control).
  - Change the control configuration.
  - Relax the design specifications.

- *Implementation* and *Verification:* how will the control system be realized? How behaves the controlled system in practice?

  The controller will be implemented and one will verify if the system is controlled as expected.

From the foregoing analysis, one can conclude that the controller design process is an iterative proces. In the next sections, the different steps are described in detail.

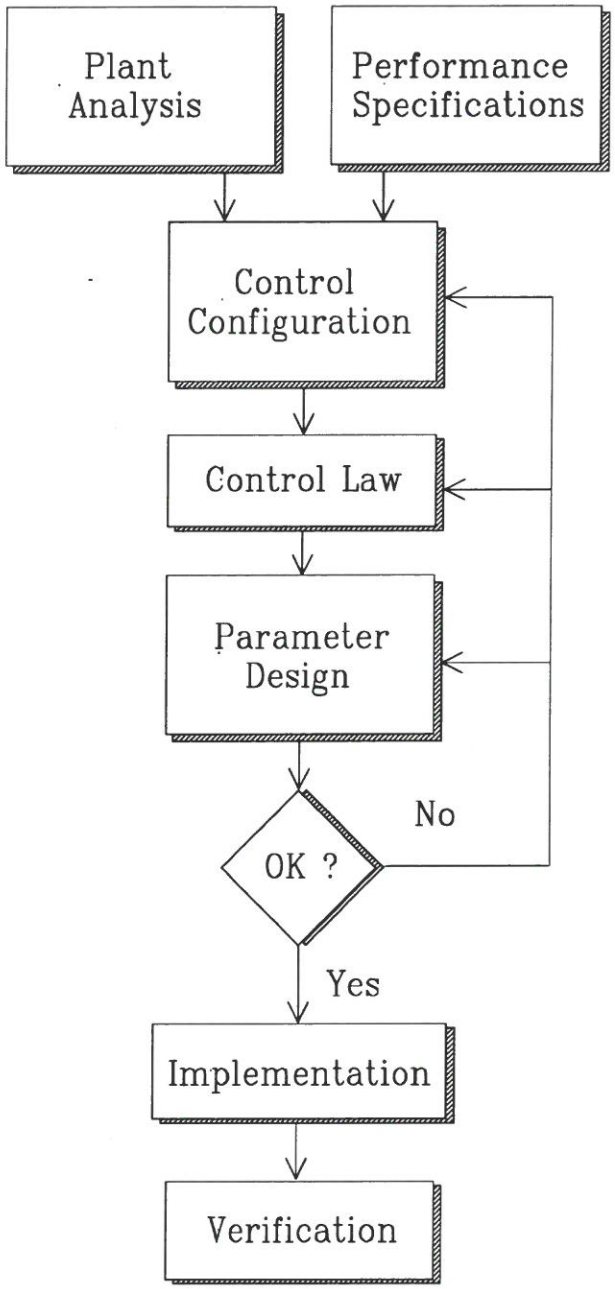


Figure 1.4: *A possible procedure to develop a control system.*

1.3 Plant Analysis

In classical analysis, various characteristics are used to analyze a plant. Depending on the needs, different plots are used to visualize specific properties: pole-zero plots express stability properties, Nyquist and Bode plots can be used to study the influence of sinusoidal inputs, while time responses can be used to characterize the plant's behavior for specific input signals. Although each plot visualizes its related properties quantitatively, it also predicts various other plant properties qualitatively. In fact, the different plots can be seen as complementary information (figure 1.5).

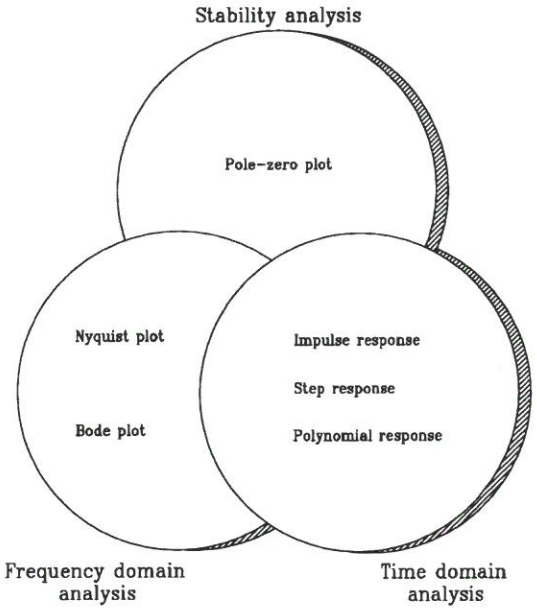


Figure 1.5: *Complementary plots to visualize specific plant properties.*

1.3.1 Pole-Zero analysis

In pole-zero analysis, the positions (in the complex plane) of the plant's poles and the zeros are investigated. It can be shown that many open loop and closed loop properties may be predicted qualitatively and quantitatively by analyzing these poles and zeros.
Let the SISO-system $P(s)$ be:

$$P(s) = \frac{num(s)}{den(s)}$$

The poles p_i are the roots of the polynomial $den(s)$, while the zeros z_j are the roots of the polynomial $num(s)$. Note that the poles and zeros are complex numbers!
To allow quick analysis of some open loop and closed loop behavior, these poles and zeros are drawn in the complex plane. By convention, the poles are indicated by (x)-marks, the zeros by (o)-marks.
By analyzing the positions of the poles, some interesting plant properties can be derived:

- The 'most right' pole (the least stable) determines the stability degree σ of the plant $P(s)$:

$$\sigma = \min_i [-\text{real}(p_i)]$$

  Negative values of σ correspond to an unstable plant, while positive values correspond to a stable one. The absolute stability of a plant grows with σ! When σ decreases, the velocity of the system decreases also.

- Furthermore, the stability ratio σ_r of two stable plants $P_1(s)$ and $P_2(s)$, with respective stability degrees σ_1 and σ_2, is defined as:

$$\sigma_r = \frac{\sigma_1}{\sigma_2}$$

  If σ_r is larger than one, plant $P_1(s)$ is the most stable one, if σ_r is smaller than one, plant $P_2(s)$ is the most stable one.

- Another important property of stable plants, the damping degree δ, is determined by the 'most diagonal' pole (the least damped):

$$\delta = \min_i \mid \frac{\text{real}(p_i)}{\text{imag}(p_i)} \mid$$

  If δ decreases, the velocity of the system increases (positive effect), but the tendency to oscillate increases (negative effect).

But not only the poles determine the plant's properties, the plant's behavior is also influenced by the position of the zeros.
If one or more zeros of the plant are in the right-half plane, the system is said to be non-minimum phase. In all other cases, the system is a minimum phase system. It can be proven that only non-minimum phase systems can give rise to undershoot.

Example 3 *Consider the following plants (normalized DC-gain):*

$$P_1(s) = \frac{6}{(s+3)(s^2+2s+2)}$$

$$P_2(s) = \frac{0.2}{(s+0.1)(s^2+2s+2)}$$

$$P_3(s) = \frac{90}{(s+3)(s^2+2s+30)}$$

$$P_4(s) = \frac{18(s+5)(s+1)}{(s+3)(s^2+2s+30)}$$

$$P_5(s) = \frac{18(s+5)(1-s)}{(s+3)(s^2+2s+30)}$$

$P_1(s)$ has the same dynamics as the previously derived model of the disk player. By modifying the real pole ($P_2(s)$) or the complex conjugated poles ($P_3(s)$), the stability degree and the damping degree are changed.
However, the impulse response and the step response also change by adding zeros. Depending on the positions of the zeros in the complex plane, different properties are observed ($P_4(s)$ has only zeros in the LHP, $P_5(s)$ has a zero in the RHP).
For all plants, the pole-zero plots, together with the corresponding impulse responses and step responses are shown in the figures 1.6, 1.7, 1.8, 1.9 and 1.10. The stability degree, the stability ratio (with respect to $P_1(s)$) and the damping degree are given in the following table.

	$\sigma(P)$	$\sigma_r(P) = \frac{\sigma(P)}{\sigma(P_1)}$	$\delta(P)$
$P_1(s)$	1	1	1
$P_2(s)$	0.1	0.1	1
$P_3(s)$	1	1	0.186
$P_4(s)$	1	1	0.186
$P_5(s)$	1	1	0.186

Notice that complex poles give rise to oscillations and that plants with a smaller damping degree oscillate 'more'. Due to the non-minimum phase character (a zero at $z_1 = 1$), $P_5(s)$ exhibits undershoot.

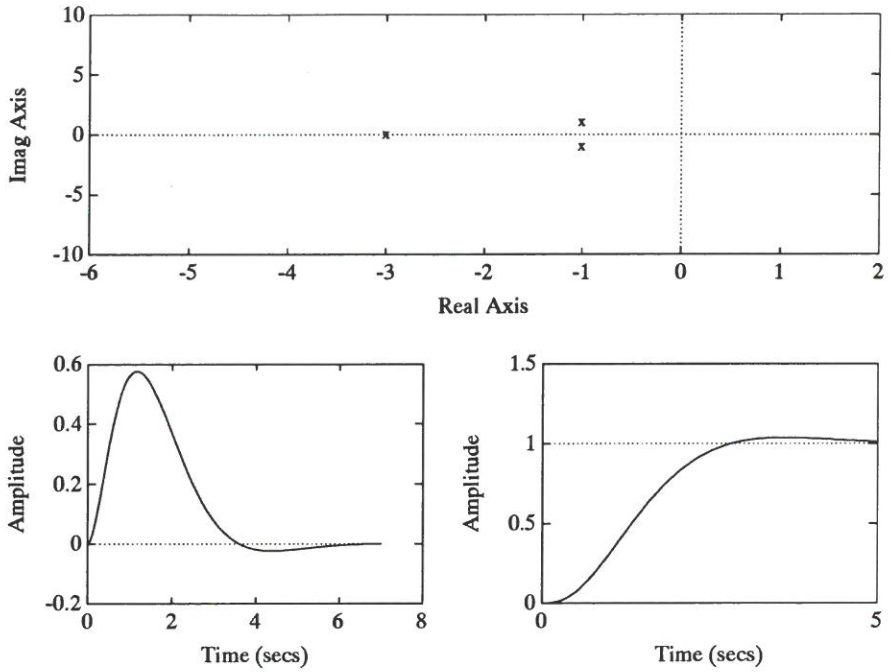


Figure 1.6: *Pole-zero plot, impulse response and step response for the plant* $P_1(s)$.

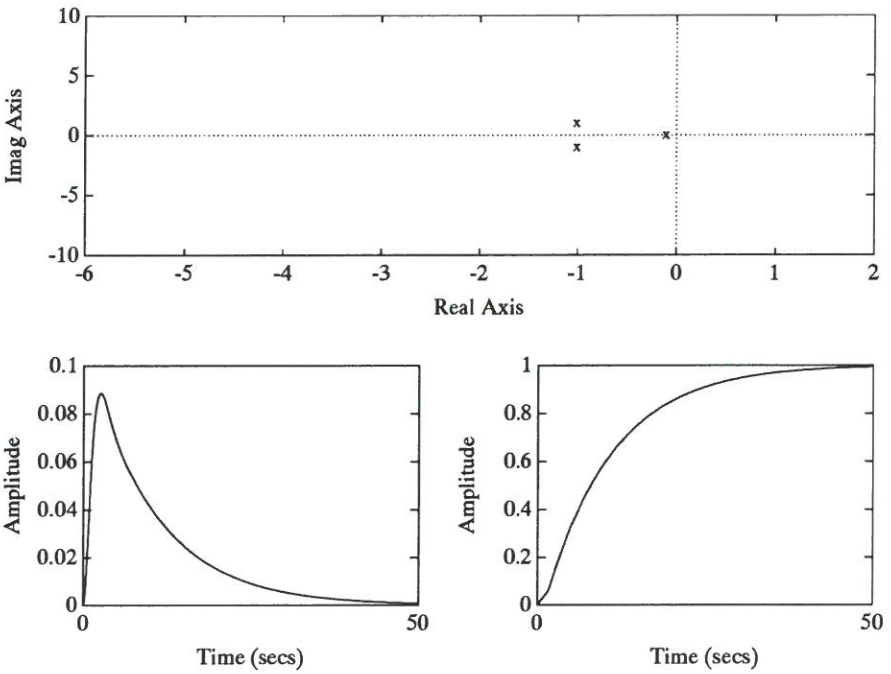


Figure 1.7: *Pole-zero plot, impulse response and step response for the plant* $P_2(s)$.

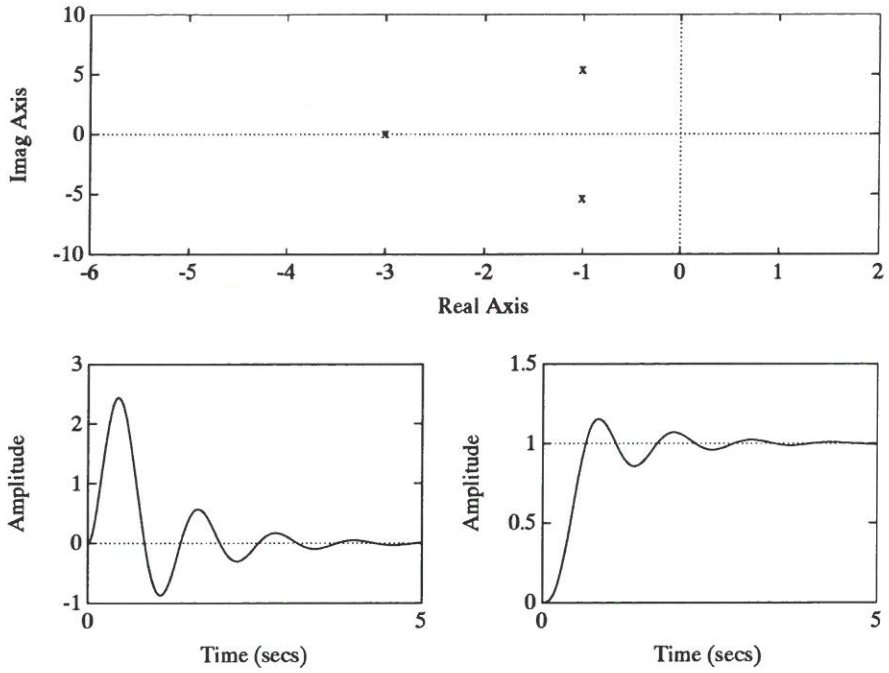


Figure 1.8: *Pole-zero plot, impulse response and step response for the plant* $P_3(s)$.

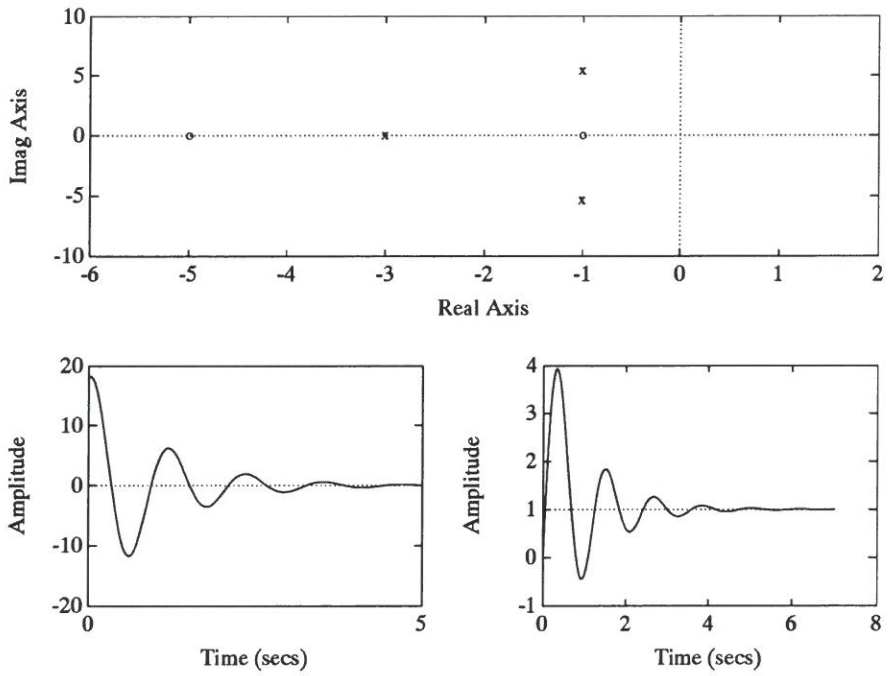


Figure 1.9: *Pole-zero plot, impulse response and step response for the plant* $P_4(s)$.

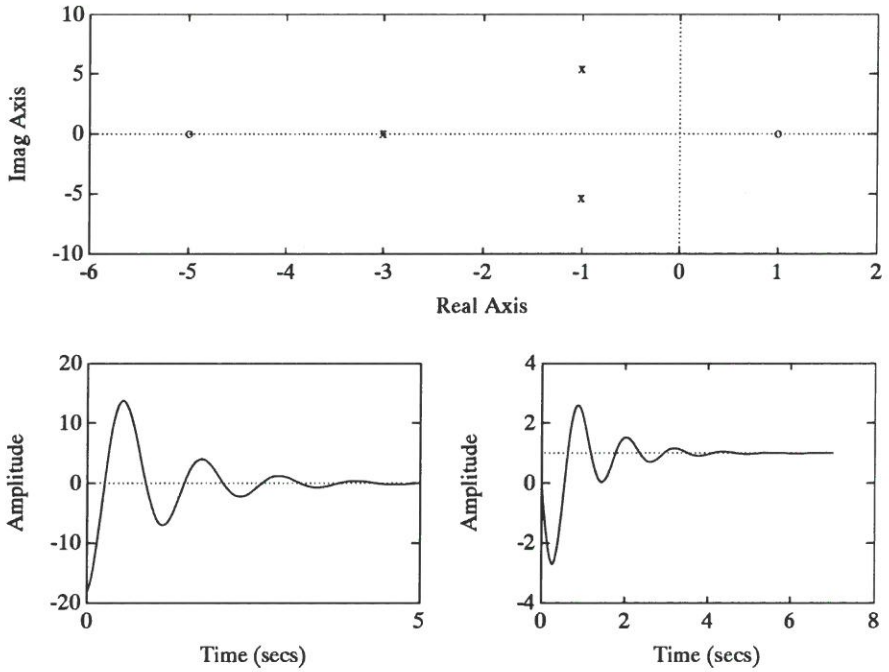


Figure 1.10: *Pole-zero plot, impulse response and step response for the plant* $P_5(s)$.

1.3.2 Frequency domain analysis

The transfer function $P(s)$ of a linear system is a complex-valued, rational function of the complex variable s. This transfer function relates the Laplace transform $U(s)$ of the input signal $u(t)$ to the Laplace transform $Y(s)$ of the output $y(t)$ by:

$$Y(s) = P(s)U(s)$$

For each input signal $u(t)$, the output $y(t)$ can be calculated using the Laplace transform and its inverse.
An equivalent property also holds for the Fourier transform of a plant. For a very broad class of plants, the Fourier transform of the plant $P(s)$ can be calculated as:

$$P(j\omega) = P(s)|_{s=j\omega}$$

Furthermore, it is known that each input signal can be written as a sum of sinusoids (using the Fourier transform). Therefore, it is sufficient to analyze the plant for sinusoidal inputs.
To do the analysis for sinusoidal inputs, consider the Fourier transform $U(j\omega)$ of the input signal $u(t) = sin(\omega^* t)$. It can be proven that $U(j\omega)$ is a sum of two impulses:

$$U(j\omega) = \delta(j\omega - j\omega^*) + \delta(j\omega + j\omega^*)$$

$Y(j\omega)$ can be calculated as:

$$Y(j\omega) = P(j\omega)U(j\omega) = P(j\omega^*)\delta(j\omega - j\omega^*) + P(-j\omega^*)\delta(j\omega + j\omega^*)$$

If $P(j\omega^*)$ is equal to $P_r(j\omega^*) + jP_i(j\omega^*)$, the gain M for the frequency ω^* is:

$$M = \sqrt{P_r^2(j\omega^*) + P_i^2(j\omega^*)}$$

while the phase shift ψ at that frequency is (figure 1.11):

$$\psi = -\arctan(\frac{P_i(j\omega^*)}{P_r(j\omega^*)})$$

Using the inverse Fourier transform, the output $y(t)$ becomes:

$$y(t) = Msin(\omega^* t + \psi)$$

These calculations can be repeated for different frequencies ω. The Nyquist plot and the Bode plot are two plots that visualize $P(j\omega)$ as a function of ω.
The Nyquist plot shows the complex function $P_r + jP_i$ in the complex plane, while the Bode plot shows the gain M and the phase shift ψ as a function of the frequency ω. Examples are given in figures 1.12 and 1.13.
Note that a Bode plot can be used easily to predict some other properties:

- A pole gives rise to a decreasing gain.

  A pole in the left half plane (LHP) corresponds to a decreasing phase shift, a pole in the right half plane to an increasing phase shift.

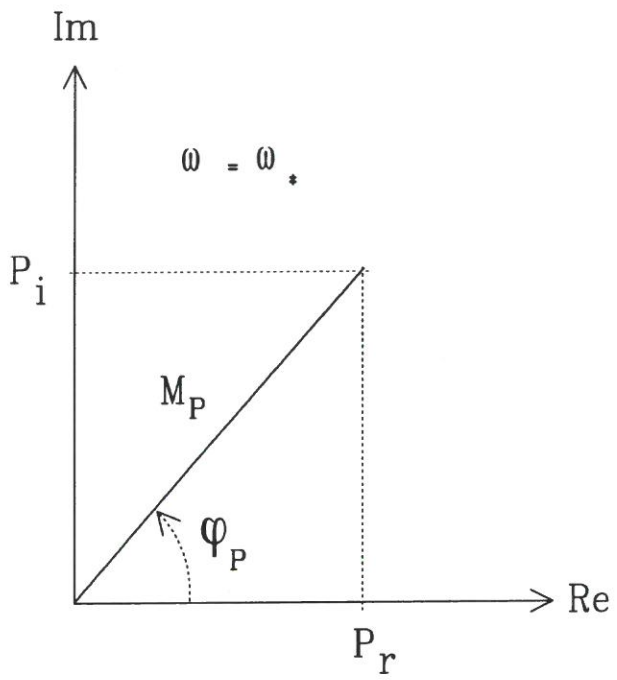


Figure 1.11: *Behavior of the system's transfer function on the imaginary axis.*

- A zero gives rise to an increasing gain.

  A zero in the LHP corresponds to an increasing phase shift, a zero in the RHP to a decreasing phase shift.

For a plant, to be stable, it is required that there is no frequency on the Bode plot where the gain decreases and the phase shift increases!
Further, if there is a frequency where the gain and the phase shift decrease, the plant is guaranteed to be non-minimum phase (thus undershoot is possible!)

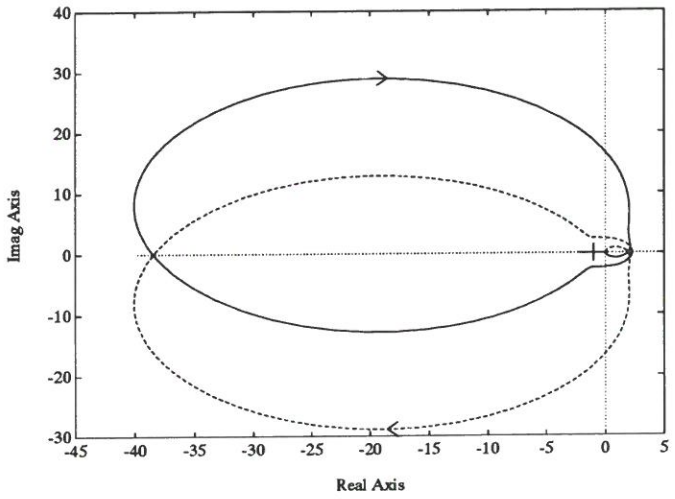


Figure 1.12: *The Nyquist curve of* $P(s) = \frac{1000(s-100)(s+50)(s+10)(s-2)}{(s+500)(s^2+4s+629)(s+70)(s+2)}$. *The full line corresponds to positive frequencies* ω, *the dashed line to negative frequencies.*

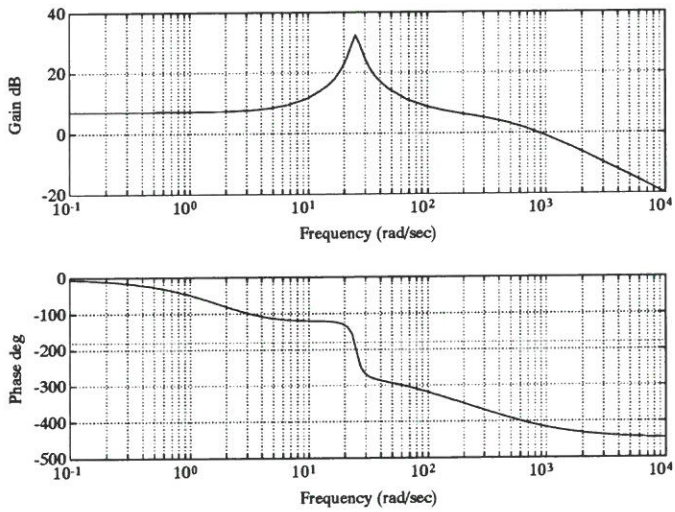


Figure 1.13: *The Bode diagram of* $P(s) = \frac{1000(s-100)(s+50)(s+10)(s-2)}{(s+500)(s^2+4s+629)(s+70)(s+2)}$. *The gain* M *is given in dB (u), the phase shift* ψ *in degrees (b).*

1.3.3 Time domain analysis

Instead of learning about the plant by analyzing sinusoidal inputs, some interesting plant properties may also be observed by studying the effect of an impulse input or a step input. For a unit impulse input signal $u(t) = \delta(t)$ (Fourier transform $U(j\omega) = 1$), the Fourier transform of the output $Y(j\omega)$ is:

$$Y(j\omega) = P(j\omega)U(j\omega) = P(j\omega)$$

To calculate the impulse response $y(t)$, it is sufficient to take the inverse Fourier transform of $P(j\omega)$. Note that:

- Complex conjugated poles $p_{i,i+1} = \alpha \pm j\omega$ give rise to an output term $\exp(\alpha t)\sin(\omega t)$.
- A real pole $p_j = \alpha$ gives rise to an output term $\exp(\alpha t)$

From this, it follows that:

- A decreasing impulse response envelope requires all poles to be in the LHP, while an increasing impulse response envelope means that there is at least one pole in the RHP.
- An impulse response with an oscillating behavior (decreasing, stabilizing or increasing) requires at least one pair of complex conjugated poles.

However, for many practical plants, it is impossible to apply a unit impulse, but it is easy to apply a step input. In that case, the response $y(t)$ for a unit step $u(t)$ (Laplace transform $\frac{1}{s}$) will be observed. The Laplace transform $Y(s)$ is:

$$Y(s) = P(s)\frac{1}{s}$$

From the finite value theorem, the output value for $t \to \infty$ can be calculated:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} P(s)$$

From this, it follows that:

- The output of a plant with one or more poles at $s = 0$ (integrating terms) goes to ∞ for a step input.
- The output of a plant with one or more zeros at $s = 0$ (differentiating terms) goes to zero for a step input.
- To have no steady state error ($y(\infty) = u(\infty) = 1$), it is required that $\lim_{s\to 0} P(s) = 1$.

1.4 Performance specifications

Once the plant properties are known, the control specifications must be written down. In practice, various quality measures are used to indicate a control system's quality. The most frequently used measures for a classical feedback controlled system (figure 1.14) are given below. By specifying 'desired' values for the different quality measures, the designer expresses the control specifications in an objective way.

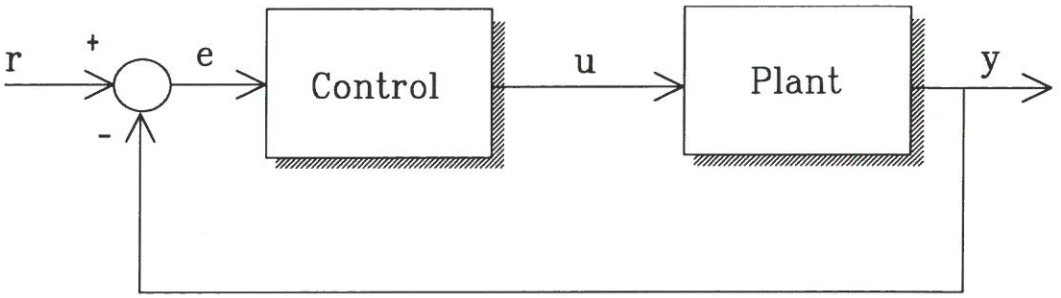


Figure 1.14: *Classical feedback control: in this scheme, y denotes the plant output or sensor signal, u denotes the plant input or actuator signal, r denotes the reference or command input signal and e denotes the error signal.*

The plant's transfer function is given by $P(s)$, while the transfer function of the controller is given by $C(s)$. The open loop transfer function $L(s)$ is:

$$L(s) = P(s)C(s)$$

the closed loop transfer function $H(s)$ (from r to y) is:

$$H(s) = \frac{L(s)}{1 + L(s)}$$

1.4.1 Closed-loop stability

To guarantee *absolute stability* of the closed loop system $H(s)$, the controller's parameters need to be chosen such that all poles of $H(s)$ are in the LHP. In practice, the stability degree σ of the closed-loop system is required to be positive.
On the other hand, if the loop transfer $L(s)$ itself is stable, the *relative stability* of the closed loop system can be expressed by three different measures:

- The *positive gain margin* (pgm) is the smallest increase in loop gain that results in an unstable closed loop system. A positive gain margin smaller than 0 dB means that the design yields an unstable closed loop system. In practice, a pgm of 2 dB is required in order to guarantee sufficient stability.
- The *negative gain margin* (ngm) is the smallest decrease in loop gain that results in an unstable closed loop system. A negative gain margin smaller than 0 dB means that the design yields an unstable closed loop system. In practice, a ngm of 2 dB is required in order to guarantee sufficient stability.

- The *phase margin* (*pm*) is the smallest decrease or increase in loop phase that results in an unstable closed loop system. A phase margin lower then zero degrees means that the design yields an unstable closed loop system. In practice, a *pm* of 45 degrees is required in order to guarantee sufficient stability.

Although it is sufficient to check either phase margin or gain margin, it is advisable to meet both specifications. These measures can be visualized on the Nyquist plot and the Bode plot (figure 1.15).

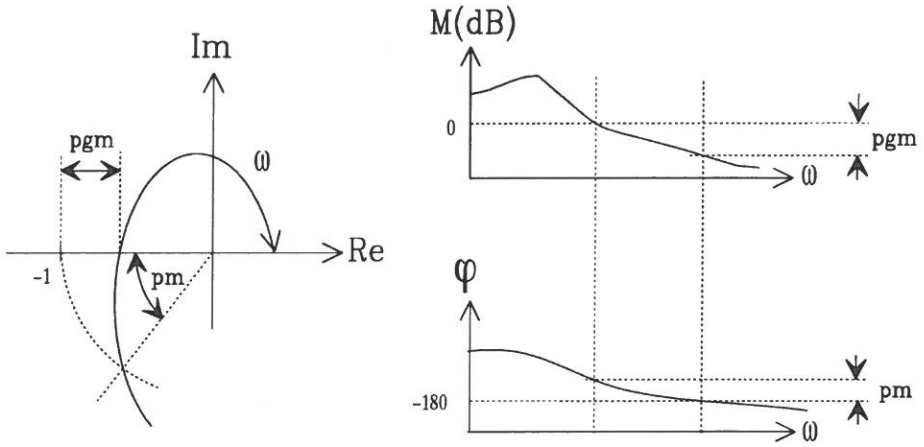


Figure 1.15: *Visualizing the gain and phase margin on the Nyquist plot (l) and the Bode plot (r).*

1.4.2 Frequency domain performance

The *bandwidth* (*bw*) of a system is the smallest frequency at which the closed loop gain is less than -3 dB. In practice, the *cross-over frequency* ω_{co} (the frequency for which the loop gain equals 0 dB) is a good measure for the closed loop system's bandwidth (figure 1.16).

- *Tracking properties* can be derived by observing

$$H(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)}$$

For frequencies below the bandwidth, $\mid L(j\omega) \mid \gg 1$ and $1 + L(j\omega) \approx L(j\omega)$, so:

$$H(j\omega) \approx \frac{L(j\omega)}{L(j\omega)} = 1$$

thus the system is able to track these signals. For frequencies above the the cross-over frequency, $\mid L(j\omega) \mid \ll 1$ and $1 + L(j\omega) \approx 1$, thus:

$$H(j\omega) \approx L(j\omega) \ll 1$$

which implies that these higher frequencies are not 'followed' anymore: the system seems to be 'insensitive' for them.

- *Output noise compensation* can be studied by observing:

$$H(j\omega) = \frac{1}{1 + L(j\omega)}$$

For frequencies below the bandwidth, $\mid L(j\omega) \mid \gg 1$ and $1 + L(j\omega) \approx L(j\omega)$, so:

$$H(j\omega) \approx \frac{1}{L(j\omega)} \ll 1$$

thus the system suppresses these noise components. For frequencies above the the cross-over frequency, $\mid L(j\omega) \mid \ll 1$ and $1 + L(j\omega) \approx 1$, thus:

$$H(j\omega) \approx 1$$

which implies that these higher frequencies are not 'suppressed' anymore.

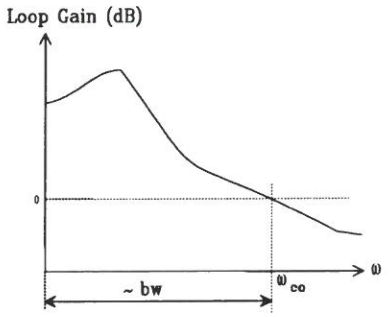


Figure 1.16: *Quality measures for frequency domain performance.*

1.4.3 Time domain performance

Different quality measures are used to evaluate the time-domain performance of a control system. The step response quality is expressed by the following measures (figure 1.17):

- The *steady state error* e_{ss} is equal to $\frac{1}{|1+L(0)|}$, which is the difference between one and the closed loop transfer function at $DC(s = 0)$. It is a measure for the steady state accuracy of a system.

  e_{ss} is also equal to the DC sensitivity, i.e., for small changes in the open loop DC gain, it gives the percentage change in closed loop DC gain divided by the percentage change in the open loop DC gain. For example, if e_{ss} equals 0.1, a 10% change in open loop DC gain will cause a change in the closed loop DC gain of about 1%.

  Notice that the e_{ss} is zero when $L(s)$ contains an integral term.

- The *rise time* T_r is the smallest time at which the step response is at least 90% of its asymptotic value (which is the closed loop DC gain). It can be verified experimentally that this time is proportional to the inverse of the cross-over frequency ω_{co}: a system with a high cross-over frequency will also have a high *response speed.*

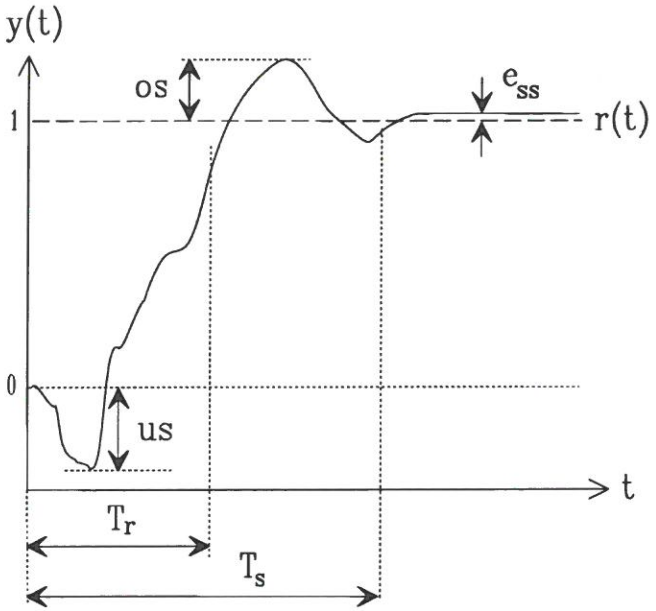


Figure 1.17: *Quality measures for time domain performance, using the step response.*

- The *settling Time* T_s is the smallest time after which the step response stays within 5% of its asymptotic value.
- The *overshoot os* is the maximum amount by which the step response ever exceeds its asymptotic value, expressed as a percentage of the asymptotic value.
- The *undershoot us* is the maximum amount the step response dips below zero, expressed as a percentage of the asymptotic value of the step response.

1.4.4 Conclusions

From the foregoing analysis, the following design criteria for a closed loop control system can be derived:

- In order to guarantee absolute stability, all closed loop poles are required to be in the LHP. If the loop transfer $L(s)$ is stable, a sufficiently high phase margin (pos. or neg.) and gain margin guarantee a certain degree of stability (relative stability).
- To obtain a high steady-state accuracy, $L(s)$ should have a large DC-gain. By adding integrating elements, this gain becomes ∞.
- To obtain a high bandwidth (good tracking, good disturbance compensation) and a high response speed, the cross-over frequency must be high.

It is clear that these design criteria result in design conflicts, so performance trade-offs must be accepted.

1.5 Control configurations

Before discussing a specific control system into detail, we present an overview of the most current control configurations. Depending on the type and the number of input signals to the controller, different configurations are recognized.
The number of input signals to the controller determines the degree-of-freedom of the control configuration: in one-degree-of-freedom control (1DOF), only one signal enters the controller. In two-degree-of-freedom control (2DOF), two signals are entering the controller.

1.5.1 One-degree-of-freedom control

Feedforward control

Open loop control or feedforward control occurs when the desired plant output (y_{sp}) is the only input signal to the controller (figure 1.18).

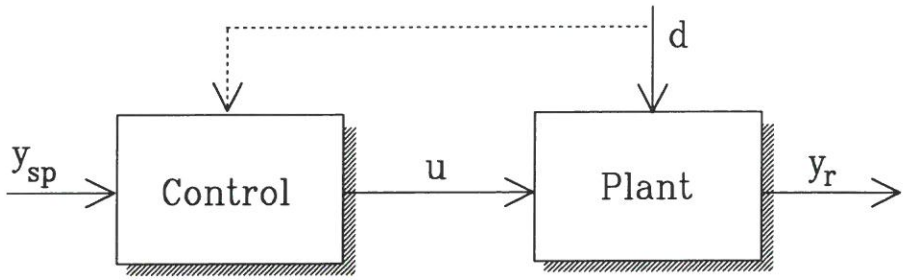


Figure 1.18: *Feedforward control.*

It can be used if the process model allows calculation of the input signal u necessary to obtain $y_r = y_{sp}$. It is easy to understand that the inverse of the system is the best feedforward controller for a given system. In that case, the real output y_r and the desired output y_{sp} are exactly the same, provided that the process model describes the process accurately. However, any unexpected disturbance of whatever origin will deteriorate the performance. On the other hand, if the disturbance is measurable or predictable, the control input can be changed accordingly, to compensate for the disturbance.
In order to compute the feedforward signal, it is necessary to differentiate the reference signal once or even more times (n). This is only possible if the reference trajectory is sufficiently smooth, i.e. the reference trajectory has finite derivatives up to order n. If the reference trajectory meets this requirement, perfect tracking can be achieved.

Feedback control

On the other hand, closed loop or feedback control occurs when the control signal depends only on the error e, the difference between the desired output y_{sp} and the real output y_r (figure 1.19).
This type of control has the advantage of relative insensitivity for disturbances, uncertainties in the process and process parameter changes. If the process parameters change due to a measurable signal, the controller parameters may be adapted to have an augmented performance.

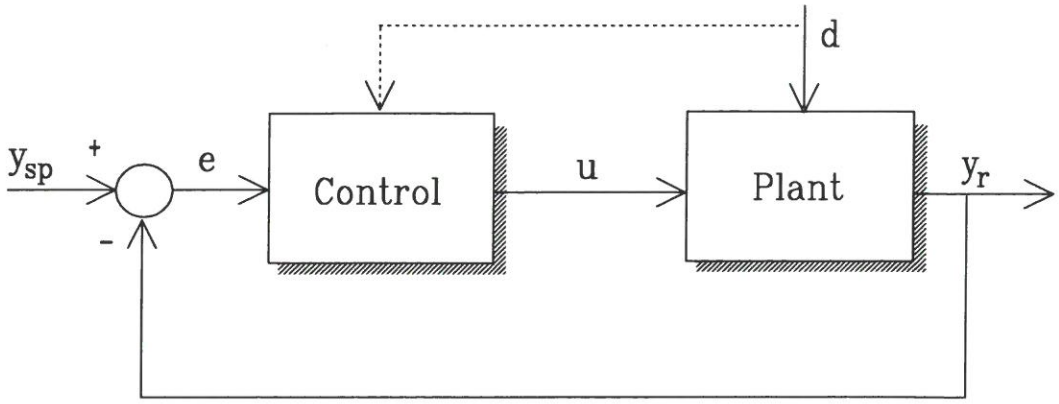


Figure 1.19: *Feedback control.*

1.5.2 Two-degree-of-freedom control (2DOF)

In two-degree-of-freedom control, feedforward control and feedback control are combined. For stabilization and to change the system's dynamics, a feedback controller is required. However, feedback control only is not sufficient to avoid tracking errors, so feedforward control must be added.
Two configurations can be used to combine these methods. For both configurations, it should be noticed that the controller has two input signals, namely the desired output y_{sp} (feedforward) and the error signal e (feedback).

2DOF control using explicit feedforward

In this configuration, the feedforward control signal u_{ff} and the feedback control signal u_{fb} are added explicitly (figure 1.21).

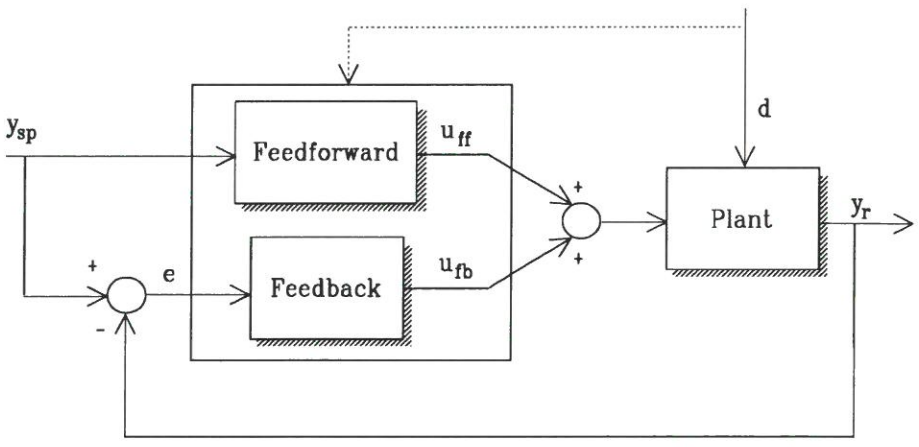


Figure 1.20: *Explicit feedforward.*

2DOF control using prefiltering as feedforward

In this configuration, the feedback controller is designed first. As described before, the best feedforward controller (realized by the prefilter) for the resulting closed loop system is the inverse of the closed loop system (figure 1.20).
However, it is not always possible to design such a prefilter, particularly in the case when the closed loop transfer function contains 'unstable' zeros (zeros in the RHP), since these zeros become unstable poles in the prefilter.

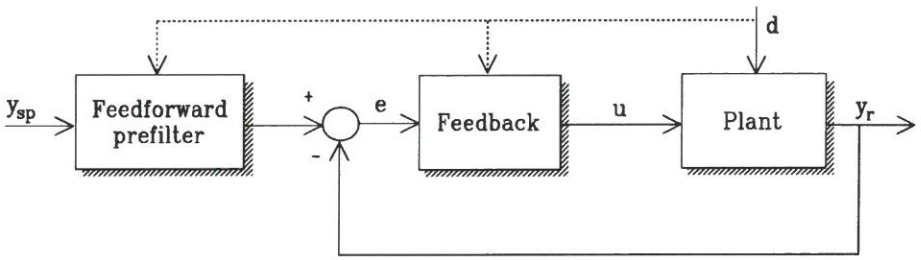


Figure 1.21: *Prefiltering technique.*

1.6 The PID-control law

In industrial environments, the PID-controller is still the most favourite feedback controller: a vast amount of practical experience learns that the PID-controller has a structure which is sufficiently versatile to cope in an excellent way with a multitude of process control problems. Other advantages of the PID-law are the low number of design parameters and the fact that the controller parameters can easily be related to performance measures.
The PID-controller is a linear controller. It transforms the error signal $e(t)$ into the control signal $u(t)$. Since it is a linear controller, it can be studied in the time domain and in the frequency domain.
In the time domain, the ideal control signal looks like:

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int e(t) + T_d\frac{d}{dt}e(t))$$

After Laplace transformation, this becomes:

$$U(s) = K_p(1 + \frac{1}{sT_i} + sT_d)E(s)$$

Basically, the controller consists of three terms, each with a typical function. The first term is proportional to the error signal, the second term is proportional to its integral and the third term is proportional to its derivative.

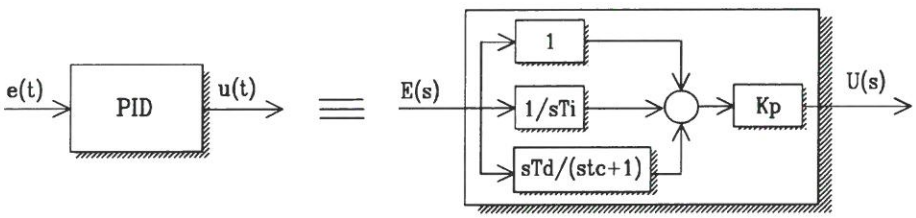


Figure 1.22: *The PID-control law.*

However, to be practical, this algorithm needs to be modified, especially for the implementation of the derivative action, since pure derivation is a practical nonsense, as it has unbounded amplification at high frequencies, which is physically impossible and even unwanted. Indeed, the slightest high frequency noise, superposed on the error signal, would be enormously amplified. Furthermore, the controller output at a step in the desired output would change annoyingly abrupt (in theory, an impulse is required).
'Smoothing' can be introduced by processing the error signal with a low-pass filter, before taking the derivative. In practice, both operations are implemented in one filter. The transfer function of the PID-controller is then given by:

$$C(s) = K_p(1 + \frac{1}{sT_i} + \frac{sT_d}{st_c + 1})$$

where t_c must be small, compared to T_d, to have a derivative action. When t_c is chosen too big, the third term becomes a proportional one, even for low frequencies. A possible PID-scheme is given in figure 1.22.

1.7 Closed Loop Design: PID-tuning

Four parameters can be used to tune a PID.

- *The gain K_p*
- *The integration time or reset time T_i*
- *The derivation time or lead time T_d*
- *The time constant t_c of the low-pass filter*

In this section, various rules and techniques to tune a PID-controller (used in a 1DOF configuration) are described. In the next section, these rules and techniques are used to tune a PID for velocity control of a disk player.

1.7.1 Classical tuning

In this subsection, the classical tuning rules of Ziegler and Nichols are presented. Up to now, these experimental rules-of-thumb are the most applied rules for tuning industrial PID-controllers, because it is not necessary to have a complete, dynamic model of the plant. Only a few, fairly simple experiments are required to find suitable controller parameters.
Since these rules are empirical rules, it should be recognized that problems can occur. However, in many cases, these rules give rise to a control system with very acceptable performance trade-offs.

Transient response method

To apply this method, it is required that the plant's step response has a similar shape as the one given in figure 1.23. From this step response, the time delay or lag L and the steepest slope or reaction rate $R = \frac{K}{\tau}$ can easily be derived (after a delay of L seconds, the output increases K units in τ seconds).

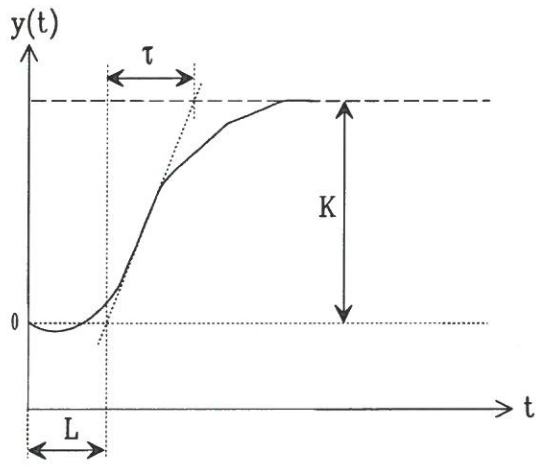


Figure 1.23: *The time delay L and the reaction rate $R = \frac{K}{\tau}$ are easily calculated from a step response.*

Once these parameters are computed, the controller parameters are selected from the following table.

	K_p	T_i	T_d
P-control	$\frac{1}{RL}$		
PI-control	$\frac{0.9}{RL}$	$3.3L$	
PID-control	$\frac{1.2}{RL}$	$2L$	$\frac{L}{2}$

However, if the plant's step response differs drastically from that in figure 1.23, this method is not applicable anymore, since L and R can't be measured/computed anymore.

Stability limit method

First, the plant is controlled using proportional control only ($T_i = \infty, T_d = 0$). Once a stable design is found (start with a very small value for the gain K_p), K_p must be increased slowly until a permanent oscillation occurs. At this point, the gain K_p and the oscillation period P_o are recorded (figure 1.24).

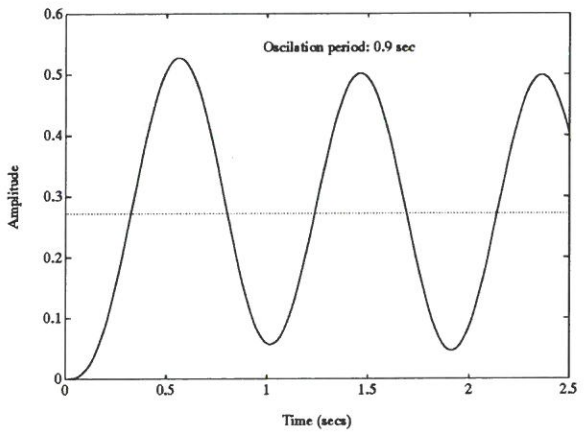


Figure 1.24: *The oscillation period P_o for the stability limit method. The ultimate controller gain K_p, responsible for this permanent oscillation, is denoted as K_o.*

If they are denoted by K_o and P_o, the controller parameters are determined from the following table.

	K_p	T_i	T_d
P-control	$0.5K_o$		
PI-control	$0.45K_o$	$\frac{P_o}{1.2}$	
PID-control	$0.6K_o$	$\frac{P_o}{2}$	$\frac{P_o}{8}$

Design rule:

1. *Do some simple experiments and determine the characteristics that are required for the method you want to apply.*
2. *Use the apropriate table to calculate the controller parameters.*

However, the performance of modelling and identification techniques increased enormously in the last decades. Therefore, it is very reasonable to assume that a suitable linear model of the plant is available. In the next subsections, model-based tuning techniques and design rules are presented.

1.7.2 Tuning for stabilization

One important reason why a feedback loop is used in many control problems, is the ease with which stabilization can be obtained. To design a stabilizing PID-controller, different techniques are available.
Two different approaches can be followed: one can design a controller by asking for absolute stability (all closed loop poles are explicitly required to be in the LHP) or one can design a controller by asking for relative stability (sufficiently high phase and gain margins are required). In both cases, the resulting controller will stabilize the system.

Asking for absolute stability

If only one parameter needs to be chosen (K_p or K_i or K_d), the root-locus method can be used to analyze the stability of the closed loop system in terms of the free parameter. The free parameter of the controller $C(s) = K\frac{N_C(s)}{D_C(s)}$ is denoted by K (K_p for P-control, $\frac{K_p}{T_i}$ for I-control and K_pT_d for D-control). If the plant's model is given by $P(s) = \frac{N_P(s)}{D_P(s)}$, the loop transfer function $L(s)$ can be rewritten as:

$$L(s) = KL_f(s) = K\frac{N_P(s)N_C(s)}{D_P(s)D_C(s)}$$

The poles of the closed loop system can be calculated in terms of the free parameter K, by solving:

$$1 + KL_f(s) = 0$$

for various values of K. The values of K for which all roots are in the LHP correspond to a stabilizing controller.
Notice that for K equal to zero, the roots correspond to the poles of the loop transfer function, while for K going to ∞, they correspond to the zeros of the loop transfer function. For non-minimum phase plants (some zeros in the RHP), the last observation shows explicitly that the free parameter K is limited by the requirement of closed loop stability.

Design rule:

1. *Choose an initial value of K and calculate the closed loop poles.*
2. *If they are all in the RHP, decrease K till they are in the LHP. Once they are all in the LHP, the corresponding K-value stabilizes the plant. When they are all 'deep' in the LHP, K may be increased without destabilizing the system.*

Although it is possible to repeat this method in situations where more then one parameter must be tuned, other techniques are preferred, since root-loci are not practical anymore.
In general, explicit stabilization conditions in terms of the controller parameters are derived. Once these conditions are given, an (in)equality solver can be used to determine a stabilizing design (notice that multiple solutions are possible).
If the loop transfer function is given by $L(s) = \frac{N_L(s)}{D_L(s)}$, the closed loop denominator (the critical polynomial) becomes:

$$l(s) = D_L(s) + N_L(s)$$

Necessary and sufficient conditions for $l(s)$ to have only roots in the LHP may be obtained by applying the Routh-Hurwitz criterion to the polynomial $l(s)$.
However, applying Routh-Hurwitz for high order polynomials requires a lot of computation and gives rise to highly non-linear conditions on the controller parameters.
In these situations, some easy-to-derive conditions are preferred. As a possibility, the coefficient rule can be applied: all coefficients of the polynomial $l(s)$ must be strictly positive, to have a chance that the polynomial has only roots in the LHP. Ifone of the coefficients is negative, there will be at least one root in the RHP. It should be noticed that the resulting conditions are only necessary for stability, not sufficient.

Design rule:

1. *Calculate the loop transfer function* $L(s) = \frac{N_L(s)}{D_L(s)}$ *and construct the polynomial* $l(s) = D_L(s) + N_L(s)$.
2. *Apply the Routh Hurwitz criterium or the coefficient rule.*
3. *Solve the resulting set of inequalities (in the design parameters).*

Asking for relative stability

As indicated before, a sufficiently high gain margin and/or phase margin guarantee closed loop stability in the case of a stable loop transfer. In that case, (frequency domain) loop-shaping techniques can be used to calculate a stabilizing controller.

Design rule (for PI-control):

1. *Find the frequency where the original system has a phase shift of* $-(180 - pm - pl)$. *Choose this frequency as the new cross-over frquency* ω_{co}.
2. *Select* T_i *such that the contribution of the PI-controller to the phase shift at* ω_{co} *is equal to* pl. *For PI-control, this results in:*

$$T_i\omega_{co} = \tan(90 - pl)$$

3. *Calculate* K_p *such that the gain at* $\omega = \omega_{co}$ *becomes* 1*:*

$$L(j\omega_{co}) = 1$$

This method guarantees a phase margin pm. An additional phase lag of pl at the new cross-over frequency ω_{co} is allowed. For the other PID-like controllers, similar procedures can easily be derived.

1.7.3 Performance-based tuning

Once a stabilizing design is found, it is human's nature to ask for other stabilizing designs, of higher quality then the previous design. To express a system's quality, different performance measures may be considered. Two main categories of specifications can be distinguished: specifications in the frequency domain and specifications in the time domain.
The influence of the PID-controller parameters on the closed loop performance measures will be discussed now.

P-control

In case of proportional control, the controller transfer function is just a gain:

$$C(s) = K_p$$

and the control signal is proportional to the control error. From the Bode plot of a proportional controller (figure 1.25), it follows that the loop gain will be multiplied by K_p, while the phase shift is not influenced at all.
The purpose of proportional control is to increase the loop gain, in order to obtain a higher accuracy (DC-gain) and a large cross-over frequency. However, due to stability requirements, K_p is limited. For plants with a tendency to oscillate, one can easily verify that the rise time T_r decreases, but that the settling time T_s and the overshoot os often increase.

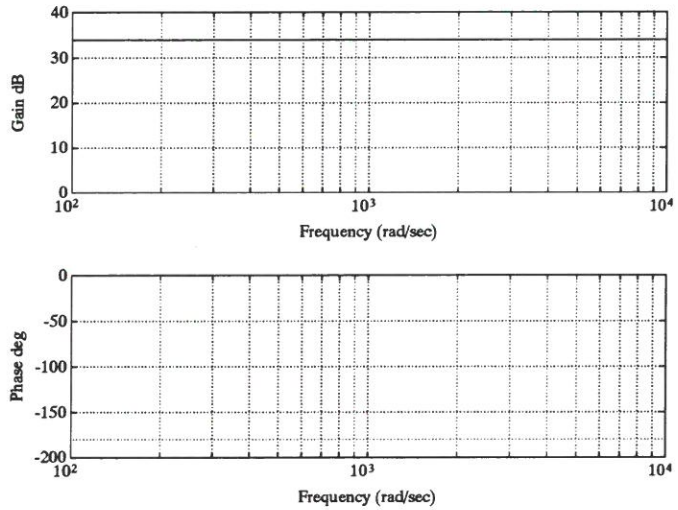


Figure 1.25: *The Bode plot of a P-controller* ($K_p = 50$).

I-control

In case of integral control, the controller transfer function is just an integrator:

$$C(s) = \frac{K_p}{sT_i}$$

and the control signal is proportional to the integral of the control error. From the Bode plot of an integral controller (figure 1.26), it follows that the loop gain will be multiplied by $\frac{K_p}{T_i}$ at $\omega = 1$ and that an additional phase shift of -90 degrees will be introduced.

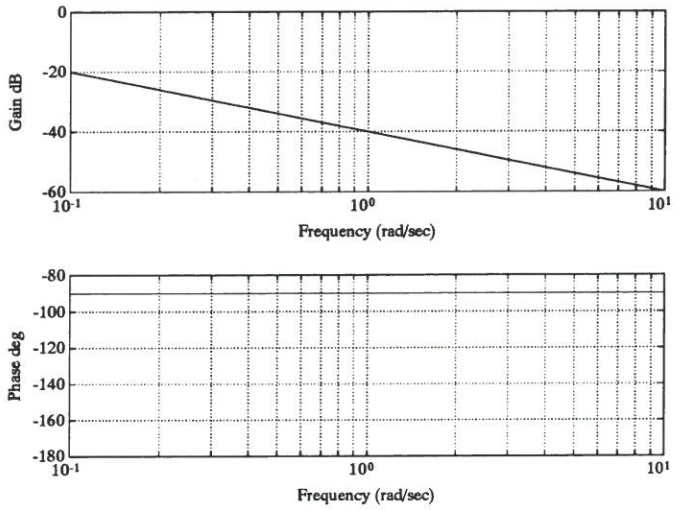


Figure 1.26: *The Bode plot of a I-controller* ($K_p = 1, T_i = 100$).

The purpose of integral control is to make the loop gain at DC equal to ∞, in order to obtain a zero steady state error. However, the additional phase shift forces the designer to decrease the bandwidth, to ensure relative stability (so $\frac{K_p}{T_i}$ is limited).

D-control

In case of proportional control, the controller transfer function is just a derivative term:

$$C(s) = sK_pT_d$$

and the control signal is proportional to the derivative of the control error. From the Bode plot of a derivative controller (figure 1.27), it follows that the loop gain will be multiplied by K_pT_d at $\omega = 1$ and that an additional phase shift of $+90$ degrees will be introduced.
The purpose of derivative control is to increase the response speed (decrease T_r). However, due to the amplification at high frequencies (noise sensitivity), the derivative action must be limited. In theory, it would be sufficient to place an extra low-pass filter $\frac{1}{st_c+1}$ before the differentiator, to 'remove' the high frequency components in the error signal. In practice, the low-pass filter and the derivative action are implemented by a single filter. In that case, the controller transfer function is:

$$C(s) = \frac{sK_pT_d}{st_c + 1}$$

PI-control

In PI-control, the advantages of P- and I-control are combined. P-action is added for bandwidth and speed, the I-action is added to guarantee a zero steady state error.

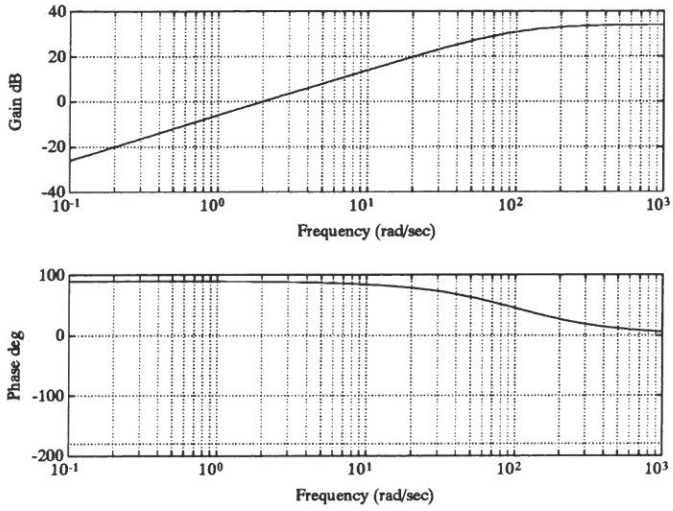


Figure 1.27: *The Bode plot of a D-controller* ($K_p = 1, T_d = 0.5, t_c = 0.01$). *The derivative action is bounded for frequencies* $\omega \geq \frac{1}{t_c} = 100$.

From the Bode plot of a PI-controller (figure 1.28), it follows that a high loop gain is introduced for lower frequencies, while the additional phase shift vanishes for higher frequencies. The positive and the negative effects of PI-control become more outspoken by making the integration time T_i smaller: the high gain for low frequencies increases, but the phase margin reduces.

PD-control

In PD-control, the advantages of P- and D-control are combined. P-action is added for bandwidth and speed, the D-action to further decrease the rise time (while P is limited by stability requirements).
From the Bode plot of a PD-controller (figure 1.29), it follows that a high phase lead and a high gain are introduced for higher frequencies: the speed of response increases, together with the sensitivity to noise. The positive and the negative effects of PD-control become more outspoken by making the integration time T_d larger: the phase lead for high frequencies increases, but the noise sensitivity increases also!

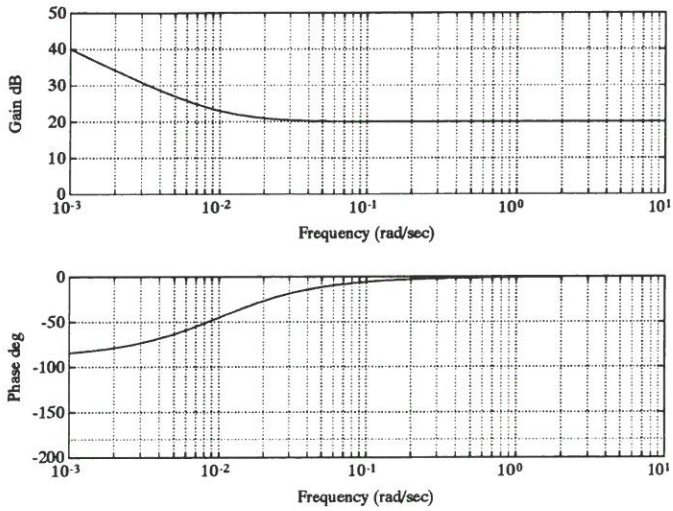


Figure 1.28: *The Bode plot of a PI-controller* ($K_p = 10, T_i = 100$). *The integral action is higher for frequencies* $\omega \leq \frac{1}{T_i} = 0.01$.

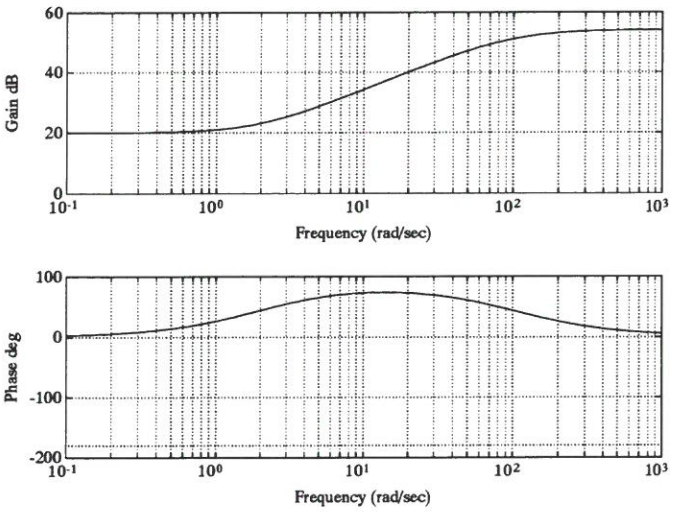


Figure 1.29: *The Bode plot of a PD-controller* ($K_p = 10, T_d = 0.5, t_c = 0.01$). *The derivative action is bounded for frequencies* $\omega \leq \frac{1}{T_d} = 2$ *or* $\omega \geq \frac{1}{t_c} = 100$.

PID-control

An attempt to combine all the previous advantages, while accepting the corresponding disadvantages, is the PID-controller, which is described in a previous section. It is easy to understand that the relative sizes of the different controller parameters determine which effects become more/less important. By analyzing the Bode plot of the PID-controller (figure 1.30), it is possible to do some predictions of the closed loop behavior.

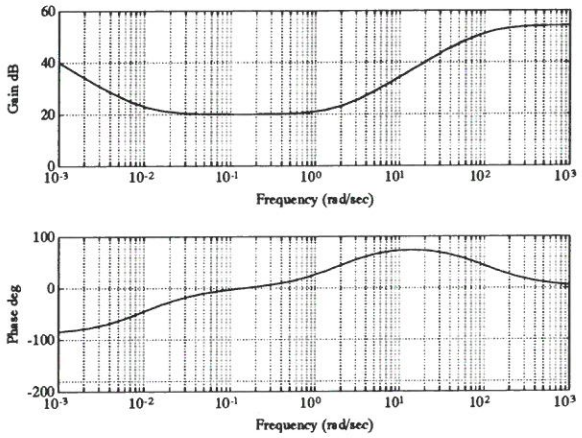


Figure 1.30: *The Bode plot of a PID-controller* ($K_p = 10, T_i = 100, T_d = 0.5, t_c = 0.01$).

1.8 An example of a tuning process: PID-tuning for velocity control of a disk player

A 1DOF-feedback PID-controller is required to do velocity control of the disk player. Figure 1.31 shows the control system to be designed.

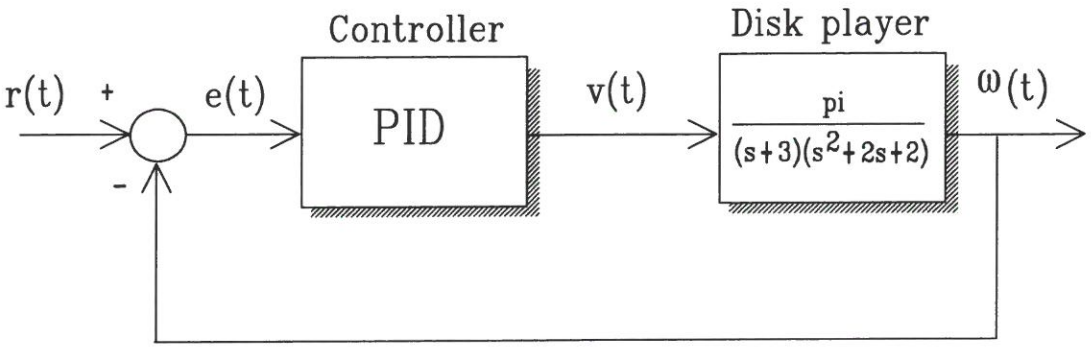


Figure 1.31: *A 1DOF-feedback controller for velocity control of a disk player. The PID-controller calculates the control voltage $v(t)$, such that the disk velocity $\omega(t)$ closely follows a reference speed signal (in most cases modelled by a smooth polynomial), till the desired end velocity is reached. From that moment on, the PID-controller is used for regulating, to maintain the obtained end velocity. To design the controller, the step response and the loop transfer function $L(s)$ are analyzed.*

1.8.1 Step 1: a stabilizing P-controller

Initially, only proportional control is considered. From a root-locus plot (figure 1.32), it can be found that the controlled system is only stable for:

$$-2.22 \leq K_p \leq 10.82$$

To start, a controller gain $K_p = 5$ is chosen: the closed loop system turns out to be stable, but the steady state error is not acceptable at all ($e_{ss} = 0.276$). To decrease e_{ss}, K_p must be increased. However, stability requirements limit K_p to be smaller than 10.82.

1.8.2 Step 2: designing a PI-controller

Since the steady state error can't be reduced to zero, one can conclude that a P-controller can't cope with all our performance wishes. Adding an integrating term ($\frac{K_p}{sT_i} = \frac{I}{s}$) to the controller is the most easy way to guarantee a zero steady state error. To design the PI-controller, different methods can be used.

Method 1: the transient response method of Ziegler-Nichols.

To apply this method, two characteristics must be derived from the step response of the plant: the lag L and the steepest slope or reaction rate $R = \frac{K}{\tau}$. From the step response of the disk player, it is found that $L = 0.4286$ and $R = \frac{\pi/6}{1.7857} = 0.2932$. The controller parameters are then calculated as:

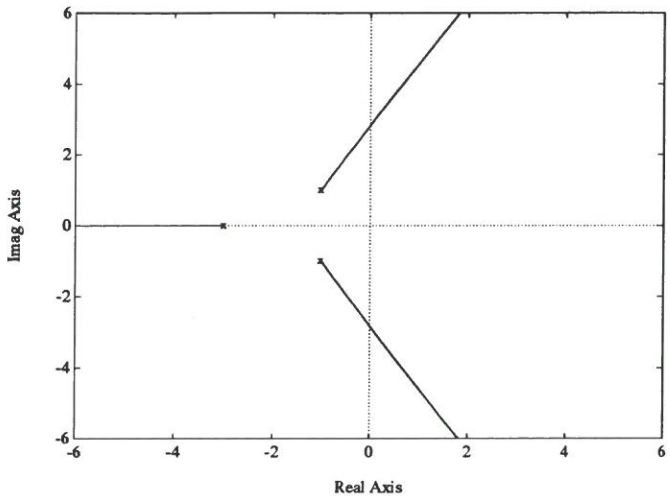


Figure 1.32: *Root locus plot of the P-controlled disk player: the poles of the closed loop transfer function are plotted in the complex plane as a function of the controller gain K_p. For K_p larger than* 10.82, *the controlled system is unstable, since it has closed loop poles in the RHP. It should be noticed that only positive values of K_p are considered here.*

$$K_p = \frac{0.9}{RL} = 7.1614$$

$$T_i = 3.3L = 1.4144$$

which corresponds to:

$$P = 7.1614$$

$$I = 5.0632$$

Method 2: the stability limit method of Ziegler-Nichols.

To apply this method, two characteristics are required: the ultimate gain K_o for P-control and the oscillation period of the corresponding step response. From P-controller analysis, it is found that a permanent oscillation with a period of 2.2 sec occurs for $K_p = 10.82$, thus $K_o = 10.82$ and $P_o = 2.2$. The controller parameters are then calculated as:

$$K_p = 0.45K_o = 4.86$$

$$T_i = \frac{P_o}{1.2} = 1.83$$

which corresponds to:

$$P = 4.86$$

$$I = 2.65$$

Method 3: solving the Routh Hurwitz conditions.

The Routh Hurwitz criterium gives necessary and sufficient conditions (on the controller parameters P and I) for obtaining a closed loop stable system. To apply them, the loop transfer function $L(s)$ must be calculated first. If $C(s) = P + \frac{I}{s}$ is the model of the PI-controller, $L(s)$ becomes:

$$L(s) = \frac{N_L(s)}{D_L(s)} = \frac{\pi(Ps + I)}{s(s+3)(s^2+2s+2)}$$

and the critical polynomial $l(s)$ becomes:

$$l(s) = D_L(s) + N_L(s) = s^4 + 5s^3 + 8s^2 + (6 + \pi P)s + \pi I$$

The Routh Hurwitz table can easily be computed:

s^4	1	8	πI
s^3	5	$6 + \pi P$	0
s^2	$\frac{34-\pi P}{5}$	πI	0
s^1	$\frac{(\frac{34-\pi P}{5})(6+\pi P)-5\pi I}{\frac{34-\pi P}{5}}$	0	0
s^0	πI	0	0

To have a stable closed loop system, all terms in the second column must have the same sign. Since 1 and 5 are strictly positive, all other terms are also required to be non-negative. After some calculations, one obtains:

$$\frac{34}{\pi} \geq P$$

$$0 \leq I \leq \frac{(34 - \pi P)(6 + \pi P)}{25\pi}$$

These conditions can now be used to calculate all stabilizing I-values, for a given P-value. Assume $P = \frac{34}{2\pi} = 5.41$, the half of the ultimate gain for P-control, then all stabilizing I-values are:

$$0 \leq I \leq 5.194$$

The equivalent condition on T_i is easily calculated:

$$1.041 \leq T_i \leq \infty$$

Finally, a suitable design is derived by varying I, using a trial-and-error method to 'optimize' the closed loop behavior. The chosen controller parameters are:

$$K_p = 5.41$$

$$T_i = 4.56$$

which corresponds to:

$$P = 5.41$$

$$I = 1.186$$

Method 4: tuning using multi-performance optimization techniques

For this method, an approach as described in [16] is used. An objective function $\Phi(P, I)$, equal to:

$$\Phi(P, I) = \lambda_1 \Phi_1(P, I) + \lambda_2 \Phi_2(P, I)$$

is minimized over the (P,I)-space, for different values of λ_1 and λ_2. In fact, $\Phi(P, I)$ is a weighted sum of two performance measures:

- $\Phi_1(P, I)$ is the overshoot *os* of the step response, as a function of P and I.
- $\Phi_2(P, I)$ is equal to the integral of the squared error signal:

$$\Phi_2(P, I) = \int_0^\infty e^2(t, P, I)dt$$

  and is a measure for the static and dynamic tracking abilities of the closed loop system.

By varying the weight vector (λ_1, λ_2), different 'optimal' solutions can be obtained. Depending on the relative sizes of λ_1 and λ_2, the obtained controller makes a trade-off of overshoot and speed. If $\lambda_1 \gg \lambda_2$, the resulting step response will be slow, but with a small overshoot, if $\lambda_1 \ll \lambda_2$ a fast step response, with a much larger overshoot will be obtained.
For the weight vector (λ_1, λ_2) equal to $(0.5, 0.5)$, the controller parameters become:

$$K_p = 2.6119$$

$$T_i = 2.5283$$

which corresponds to:

$$P = 2.6119$$

$$I = 1.0331$$

Otherwise, for the weight vector (λ_1, λ_2) equal to $(0.1, 0.9)$, the controller parameters become:

$$K_p = 3.4742$$

$$T_i = 2.4302$$

which corresponds to:

$$P = 3.4742$$

$$I = 1.4296$$

Comparing the different designs

The following table shows the performance of the different designs. The meaning of the different performance measuresis explained in section 1.4.

Method, Design	1	2	3	4, 1	4, 2
σ	−	+	+	+	+
bw (Hz)		0.266	0.285	0.132	0.201
e_{ss} (rad/s)		0	0	0	0
pgm (dB)		1.9	2.5	8.7	6.2
pm (deg)		7	9	34	22
T_r (sec)		1.1	1.1	1.7	1.4
T_s (sec)		9.9	9.5	8.4	7.6
os (%)		46.5	30.7	3.0	18.9

From this table, it follows that the design of method 4, for $(\lambda_1, \lambda_2) = (0.1, 0.9)$, is the 'best' design (figure 1.33).

1.8.3 Step 3: designing a PID-controller

If the step response of the disk player with the 'best' PI-controller and the open loop step response are compared, it can be seen that no spectacular increase in speed is realized. To do better, D-action is added ($K_p T_d s = Ds$). Similar design methods as for PI-control can be used. To have an implementable design, a low-pass filter $\frac{1}{st_c+1}$ with time constant $t_c = 0.01$ seconds is placed before the differentiator.

- **Method 1: the transient response method of Ziegler-Nichols.**

  The value of $\frac{1}{t_c}$ is sufficiently small, compared to the attainable closed loop bandwidth. Therefore, it is assumed that the Ziegler-Nichols rules are still applicable. The controller parameters for $L = 0.4286$ and $R = 0.2934$ are:

$$K_p = \frac{1.2}{RL} = 9.5465$$

$$T_i = 2L = 0.8572$$

$$T_d = \frac{L}{2} = 0.2143$$

- **Method 2: the stability limit method of Ziegler-Nichols.**

  The controller parameters for $K_o = 10.82$ and $P_o = 2.2$ are:

$$K_p = 0.6K_o = 6.48$$

$$T_i = \frac{P_o}{2} = 1.1$$

$$T_d = \frac{P_o}{8} = 0.2750$$

- **Method 3: solving the Routh Hurwitz conditions.**

  Since $\frac{1}{t_c}$ is much smaller than the attainable closed loop bandwidth, it is assumed that the Routh Hurwitz conditions may be derived for an ideal differentiator (since these conditions are easier to compute). For ideal PID-control, the polynomial of interest is:

$$l(s) = D_L(s) + N_L(s) = s^4 + 5s^3 + (8 + \pi D)s^2 + (6 + \pi P)s + \pi I$$

  The Routh Hurwitz table and the resulting conditions can easily be computed. Finally, one obtains:

$$34 + \pi(5D - P) \geq 0$$

$$0 \leq I \leq \frac{(34 + \pi(5D - P))(6 + \pi P)}{25\pi}$$

  These conditions can now be used to calculate all stabilizing D-values, for given P- and I-values. Assume $P = \frac{34}{2\pi} = 5.41$, the half of the ultimate gain for P-control, and $I = 1.186$ as previously designed for PI-control, then all stabilizing D-values are:

$$D \geq -0.835$$

  The equivalent condition on T_d is easily calculated:

$$T_d \geq -0.1543$$

  Using a trial-and-error method, it can be found that $T_d = 1$ gives rise to acceptable performance measures.

- **Method 4: using multi-performance optimization methods.**

  For $(\lambda_1, \lambda_2) = (0.5, 0.5)$, the following design is obtained:

$$K_p = 4.8217$$

$$T_i = 0.9719$$

$$T_d = 1.4459$$

Finally, the 'best' PID-design (figure 1.34) was obtained by doing trial-and-error, with the controller of method 2 as an initial design. The resulting controller is:

$$K_p = 6.4860$$

$$T_i = 1.1$$

$$T_d = 1$$

The following table compares the 'best' PI-controller and the 'best' PID-controller:

	PI	PID
σ	+	+
bw (Hz)	0.201	0.534
e_{ss} (rad/s)	0	0
pgm (dB)	6.2	19.3
pm (deg)	22	42
T_r (sec)	1.4	0.5
T_s (sec)	7.6	1.9
os (%)	18.9	10.6

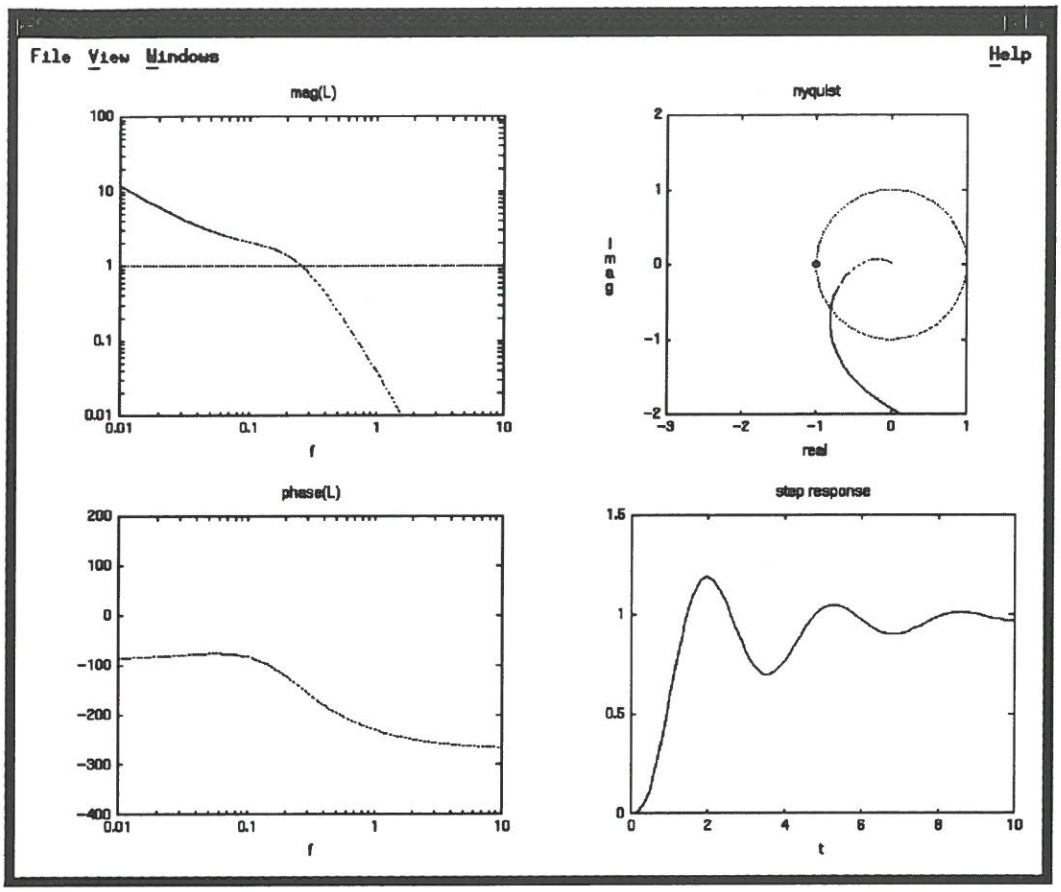


Figure 1.33: *The Bode plot and the Nyquist plot of the loop transfer function $L(s)$, together with the closed loop step response, for the PI-controlled disk player, with $K_p = 3.4742$ and $T_i = 2.4302$.*

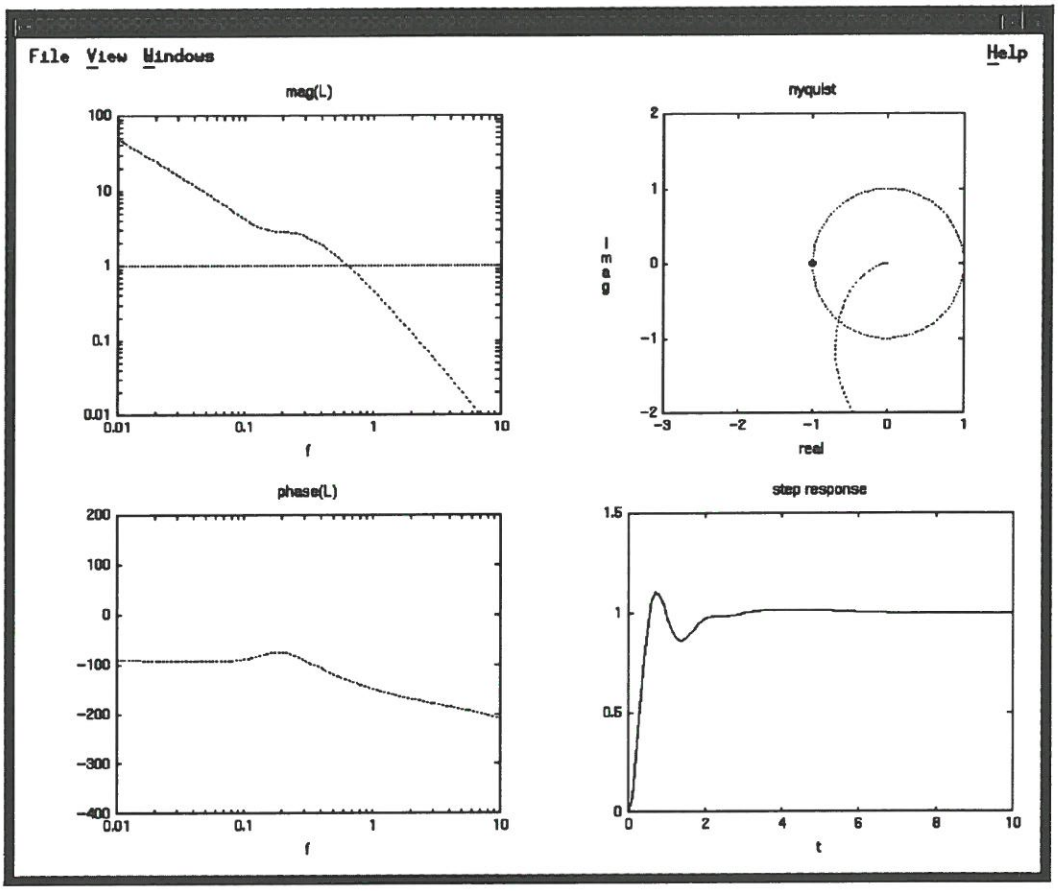


Figure 1.34: *The Bode plot and the Nyquist plot of the loop transfer function $L(s)$, together with the closed loop step response, for the PID-controlled disk player, with $K_p = 6.4840$, $T_i = 1.1$ and $T_d = 1$.*

1.9 Implementation aspects

1.9.1 Analog or digital?

Once a suitable PID-controller is found, it has to be implemented in some way.
In earlier times, all PID-controllers were implemented in hardware, using all kinds of analog technology (electronical, mechanical, pneumatical, hydraulical). Since the controller parameters were determined by geometrical and physical properties, they were mostly fixed for a given controller.
However, due to the enormous success of the digital computer, the actual controllers are mostly implemented in software, where parameters can be changed easily.

The ideal continuous time PID-controller $C(s)$ is given by:

$$C(s) = K_p(1 + \frac{1}{sT_i} + sT_d)$$

$C(s)$ relates the controller output $U(s)$ to the error signal $E(s)$:

$$U(s) = C(s)E(s)$$

If the sampling interval T is small enough, the continuous time behavior of the controlled process can be approximated rather well by stepwise constant signal approximations. In that case, the discrete PID-controller will be implemented by proper numerical approximations of the integration and differentiation operations.

- The proportional part is readily implemented as:

  $$u_p(k) = K_p e(k)$$

  where k refers to the sampling instant kT. In the z-domain, this gives:

  $$U_p(z) = K_p E(z)$$

- In the discrete time domain, the integral action can be approximated by the sum of all previous errors:

  $$u_i(k) = u_i(k-1) + \frac{K_p T}{T_i} e(k)$$

  In the z-domain, this becomes:

  $$U_i(z) = \frac{K_p T z}{T_i(z-1)} E(z)$$

- Finally, differentiation in the discrete domain can be approximated as the first difference:

  $$u_d(k) = K_p T_d \frac{(e(k) - e(k-1))}{T}$$

  and after taking the z-transform:

  $$U_d(z) = K_p T_d \frac{(z-1)}{Tz}$$

Combining all the above results yields the discrete time PID-controller:

$$U_p(z) = K_p(1 + \frac{Tz}{T_i(z-1)} + \frac{T_d(z-1)}{Tz})E(z)$$

1.9.2 Anti-windup

The actuator is the device through which the control input is applied to the process. Since these components have a limited range (e.g. motor driving amplifiers have a minimal and maximal voltage or current output), they introduce a saturating non-linearity into the process (figure 1.35).

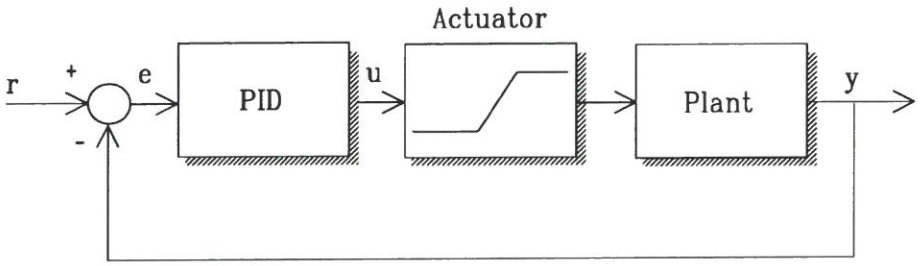


Figure 1.35: *Integrator wind up, due to actuator saturation.*

As an example, consider the PI-control of a certain process. Suppose a setpoint change brings the actuator into saturation. In that case, the error signal will decrease slower then without saturation, so the integrating output term will be higher: as a consequence, the actuator will be saturated during a longer time, since it will not leave the saturation, unless the error has changed sign for a sufficiently long time in order to bring the integral output to a low value. Further, during saturation, the feedback is actually not working: it can be replaced by a fixed-level input, which gives rise to undesirable overshoots for processes with a low stability degree. The previous effect is called *integrator wind up.*
This phenomenon can be avoided by stopping the integration when the actuator saturates. A clever way to do so is to force the output of the controller to stay at the saturation limit u_l as long as the actuator is saturated. This can be done by implementing an extra feedback loop which forces the difference between actuator input and actuator output to be zero at all times. The integral part of the controller can be used for this purpose (figure 1.36).

1.9.3 Incremental PID

For analog PID-control, the output of the regulator is always generated in a direct way. However, digital control allows to bring the algorithms in an incremental form, where at each sampling instance, the regulator output increment is calculated. This is particularly useful when the actuator is of the integrating type (e.g. a stepping motor). The number of pulses to be fed to the stepping motor is then proportional to the regulator increments (the actuator itself reconstructs the required control signal by integrating the increments!). The incremental algoritm is also called the *velocity form* of the PID-algorithm, as opposed to the *position form* of the non-incremental case.
Advantages of this incremental form are:

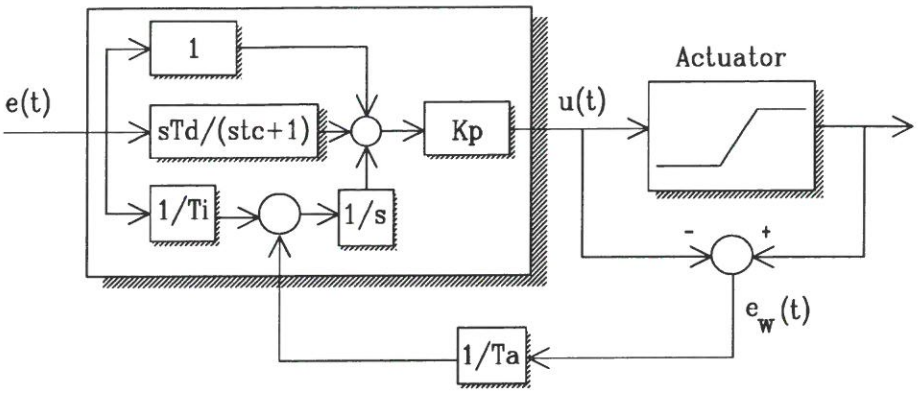


Figure 1.36: *Anti-windup by introducing an extra feedback loop: as long as there is no saturation, $e_w = 0$ and the PID-controller works in the normal way. As soon as $e_w \neq 0$, the extra loop tries to bring e_w back to zero, to a rate which is inversely proportional to T_a. This may drastically reduce the saturation time and improve the step response! Be careful to take T_a not too small, if there is an active D-action ($T_d \neq 0$), as spurious peaks originating from it may drive the actuator into saturation for a short time: too quick resetting during this short peak results in an unnecessary small value for the integrator output after the peak has vanished.*

- Shorter wordlength calculations are needed, only the final summation stage needs higher precision.
- Except for integrating type actuators, the terminal stage is an integrator, which is useful for anti-windup compensation.

1.9.4 Leadlag, a special class of PID-controllers

An alternative for the parallel PID-implementation (summing the three terms) is the series implementation of a PI- and a PD-action. In the Laplace domain, this implementation looks like:

$$C(s) = K_p'(1 + \frac{1}{sT_i'})(1 + sT_d')$$

Its parallel equivalent can easily be calculated, using the following relations:

$$K_p = K_p' \frac{T_i' + T_d'}{T_i'}$$

$$T_i = T_i' + T_d'$$

$$T_d = \frac{T_i' T_d'}{T_i' + T_d'}$$

However, it can be proven that the series equivalent of a given parallel form does not always exist. Consider the controller's transfer function:

$$C(s) = K_p(1 + \frac{1}{sT_i} + sT_d) = \frac{K_p}{sT_i}(s^2 T_d T_i + sT_i + 1)$$

To convert this to the series form, it is required that the polynomial $s^2 T_d T_i + s T_i + 1$ has two real roots, so its discriminant must be non-negative:

$$T_i^2 - 4 T_d T_i \geq 0$$

From this, it follows that the series implementation of a parallel form only exists if:

$$T_i \geq 4 T_d$$

Although the parallel form is more general, the series form is sometimes preferred, while the implementation and the avoidance of windup are very easy.

Chapter 2

PID – tuner, a practical design tool for PID-control

2.1 Introduction

PID – tuner is a graphical tool for interactive design of PID-like controllers, using the functions of *Xmath* and the facilities of the graphical user interface *GUI*.
The tool itself consists of three different windows, each with its own functionality. Detailed help messages on these windows and features are implemented.

- *The main window* is used by the designer who 'turns knobs' in the parameter control panel and 'reads the meters' in the performance measures display area (figure 2.1). It appears automaticaly when the tool is started.
- *The performance plot window* visualizes the characteristics of the controlled system. It can be called by pressing the *Plot* button on any Windows menu (figure 2.2).
- *The ranges window* is used to change the ranges of sliders, bargraphs and plots. It can be called by pushing the *Ranges* button on any *Windows menu* (figure 2.3).

2.2 Getting started/stopped

- Start *Xmath* by typing **xmath**. The command window of *Xmath* will occur.
- Change your directory by typing:
  set directory = "/users/sista/vanvuche/bdm" in the command edit space.
- Start the tool by typing **pidcontrol** in the command edit space. The main window of the tool will arise. Move the mouse to the option *Windows* and select the windows you want to open.
- Remark: don't decrease the size of any of the windows you opened, because this causes *Xmath* to stop. Only a process kill will enable you to work on your computer again!
- To stop the tool, go to the option *File* in the main menu, and press *Exit*. To stop *Xmath*, type **quit** in the command edit space.

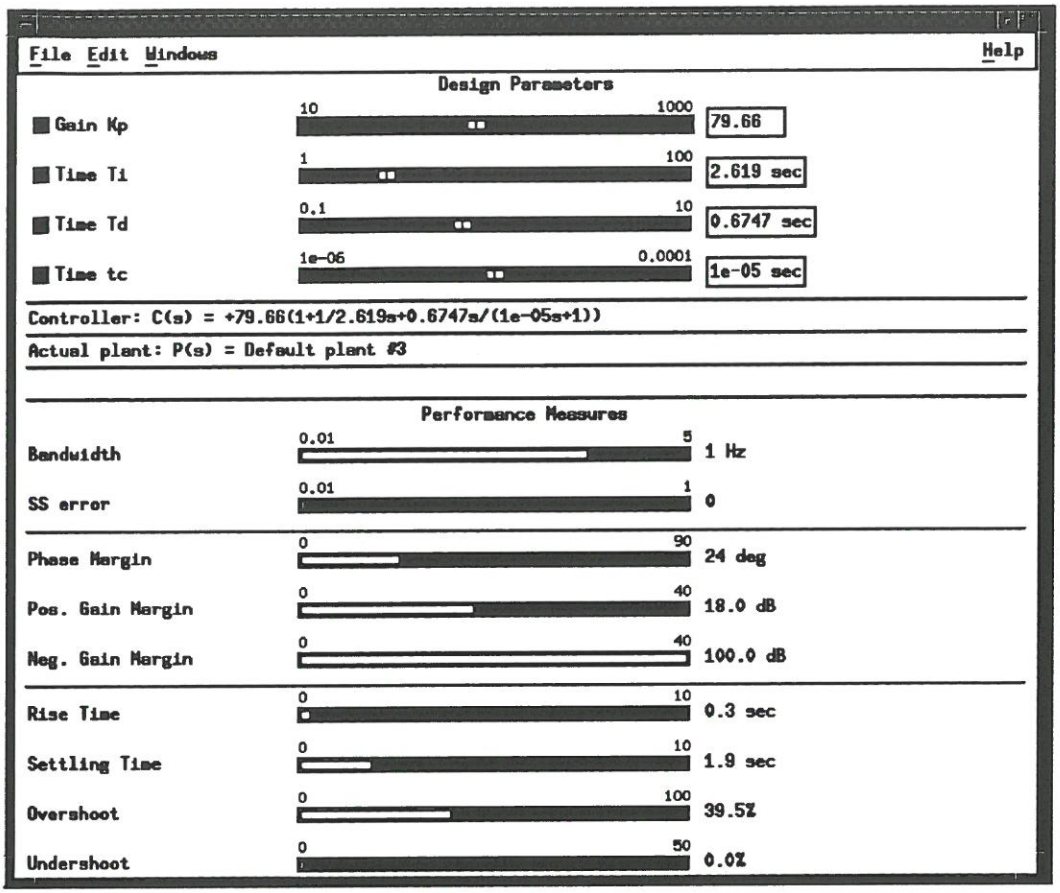


Figure 2.1: *From top to bottom, the main window consists of a control panel that is used to change the design parameters and form of the controller, a display that gives the current controller transfer function (the "current design"), a display that gives the name of the plant, a display area that is used for warning messages and a display area that is used to report various performance measures for the current controller.*

2.3 Selecting a plant

Two possibilities are available to select the plant to be controlled:

- Select a plant from the default plant list: go to the option *File* in the main menu and push the button *Default Plant List* .... A list of 17 different plants appears, each with its own specific properties. A plant can be selected by clicking on the knob in front of its model, followed by pressing *OK*.

- If you want to enter your own system, enter the plant via the *Xmath* command edit space, by typing

  plant=system(makepoly([numcoef]),makepoly([dencoef]))

  The coefficients must be ordered with decreasing powers of s and separated by commas. Then go to the option *File* in the main menu and push the button *Get Plant From Xmath*. A prompt dialog appears, asking for a plant name. Type in **plant** and select *OK*.

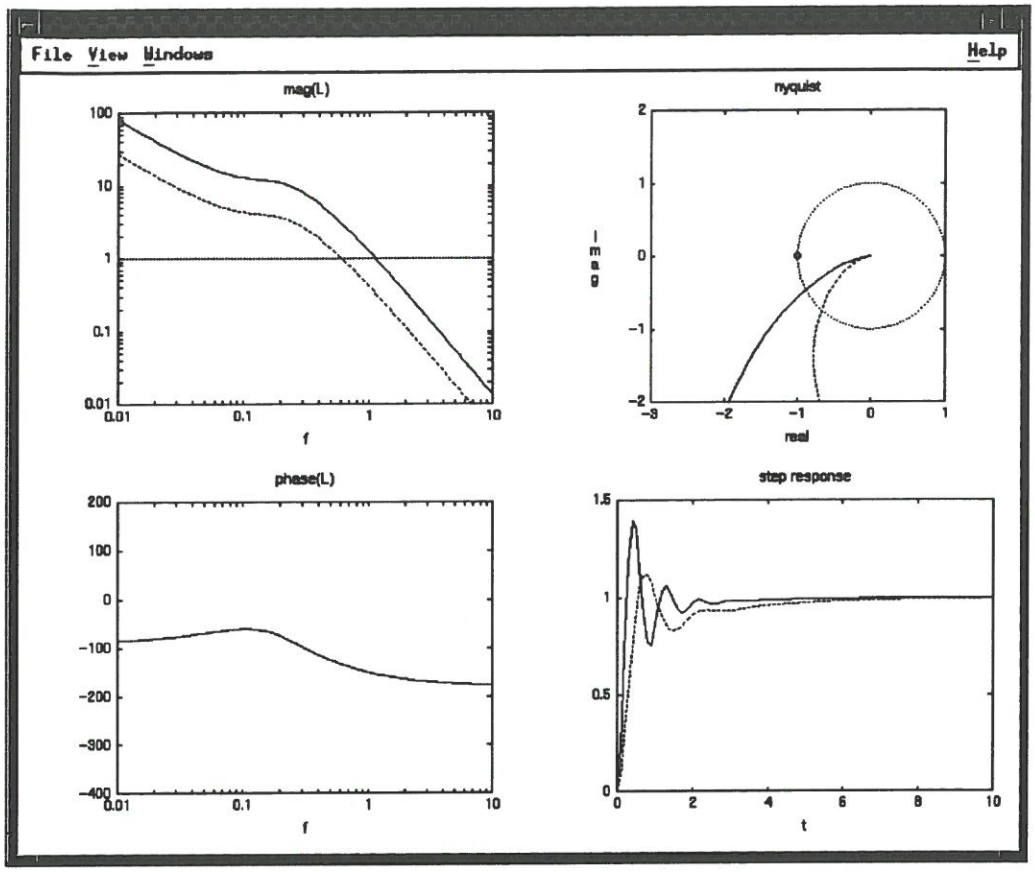


Figure 2.2: *In the performance plot window, the Bode plot and the Nyquist plot of the loop transfer $L(s)$ are shown, together with the closed loop stepresponse, for the actual design (full line) and the previous design (dotted line).*

The actual plant is shown in the main window, either by a number (default plants) or by a name (own defined plants).

2.4 Selecting the controller values

In the main window of the tool, a control panel is used to change the type of controller and its design parameters.
Pushbuttons can be used to (de)activate the different controller terms (choose the control law), while sliders and edit-boxes can be used to set the values of the control parameters. To set the value of a parameter to a value higher then the slider ranges, the edit-box must be used.
The transfer function of the actual controller is also displayed in the main window.

2.5 Analyzing the performance

Two possibilities are available to analyze the performance of the actual design:

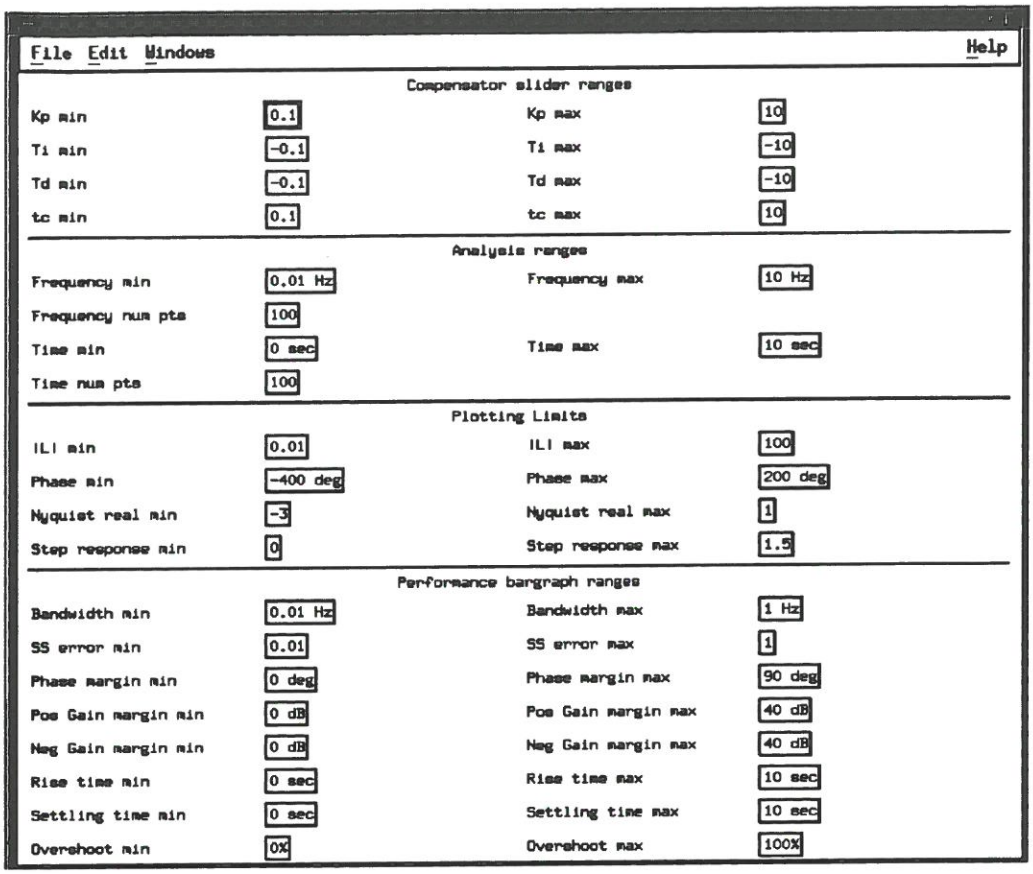


Figure 2.3: *The range window.*

- Bargraphs and read-boxes (in the main window) indicate the value of the most important quality measures, if the closed loop system is stable. A warning message occurs if the closed loop system is unstable.
- In the performance plot window, the Bode plot and the Nyquist plot of the loop transfer $L(s)$ are shown, together with the closed loop step response. If the closed loop system is unstable, nothing will be plotted!

2.6 Getting experienced in PID-control

To allow the user to get experienced in PID-control, $PID - tuner$ provides a list of well chosen plants, each of them having properties that introduce a specific control difficulty. In the following list, the plants and their properties are described shortly.

1. $\frac{1}{(s+3)}$: a stable first order plant; easy to control.
2. $\frac{1}{(s^2+2s+2)}$: a stable second order plant with complex conjugated poles; easy to control.
3. $\frac{1}{(s+3)(s^2+2s+2)}$: a stable third order plant; easy to control.
4. $\frac{(1-s)}{(s+3)(s^2+2s+2)}$: a stable, non-minimum phase plant; the undershoot is expected to make the closed loop system slower.

5. $\frac{(3-s)}{(s+3)(s^2+2s+2)}$: a non-minimum phase plant, but such that it is a series combination of an all pass filter (which approximates a delay of 0.66 seconds) and a stable second order system.
6. $\frac{1}{s(s+3)}$: adding a single integrator to a first order system (e.g. add a stepping motor as actuator).
7. $\frac{1}{s^2(s+3)}$: adding two integrators to a first order system.
8. $\frac{(3-s)}{s^2(s+3)}$: a series combination of an all pass filter and two integrators.
9. $\frac{2}{(s+1)(s+2)(s+4)}$: a third order system with only real poles.
10. $\frac{2(s+5)}{(s+1)(s+2)(s+4)}$: adding a minimum phase zero to the previous plant.
11. $\frac{2(s+5)}{(s+100)(s+1)(s+2)(s+4)}$: adding a fast pole to the previous plant (e.g. a simplified model of the actuator).
12. $\frac{2(s+5)}{s(s+1)(s+2)(s+4)}$: adding a slow pole instead of a fast pole.
13. $\frac{2}{(s-1)(s+2)(s+4)}$: an unstable system; stabilization is the first job to do!
14. $\frac{2(s+5)}{(s-1)(s+2)(s+4)}$: an unstable system with a minimum phase zero.
15. $\frac{2(s-5)}{(s-1)(s+2)(s+4)}$: an unstable system with a non-minimum phase zero.
16. $\frac{2}{s(s-1)(s+2)(s+4)}$: adding a single integrator to an unstable system.
17. $\frac{(-0.8864s^2+23.67s-111.0046)}{s(s^2+27.388s+12.6244)}$: an identified model of the ball and beam practicum.

For all those plants, a wide variety of control problems can be solved using different design techniques/rules.

- *Classical tuning:* if the methods of Ziegler-Nichols are applicable, controller values can be derived in the case of P-, PI- and PID-control.
- *Stabilization of the plant:* techniques to stabilize the plant were discussed in chapter 1.
- *Performance-based tuning:* the problems to solve can be formulated in many ways:
  - All quality measures are required to have desired values or 'better' ones (e.g. 'realize an overshoot of 2% or less and a bandwidth of 10 Hz or more').
  - A single quality measure must be optimized, with or without limiting one or more other quality measures (e.g. 'minimize overshoot' versus 'minimize overshoot, but guarantee settling in 2 seconds' versus 'minimize overshoot, but guarantee settling in 2 seconds and a bandwidth of at least 10 Hz').
  - Multiple quality measures must be optimized simultaneously, with or without limiting one or more other quality measures (e.g. 'minimize overshoot and rise time' versus 'minimize overshoot and rise time, but guarantee settling in 2 seconds' versus 'minimize overshoot and rise time, but guarantee settling in 2 seconds and a bandwidth of at least 10 Hz').

Bibliography

[1] Arora J.S. *Introduction to optimum design.* Mc Graw-Hill, New York, 1989.

[2] Astrom K.J., Wittenmark B. *Computer Controlled Systems.* Englewood Cliifs, Prentice Hall, New York, 1984.

[3] Boyd S.P., Baratt C. *Linear Controller Design, Limits of performance.* Prentice Hall Inc., 1990.

[4] Cool J.C., Schijff F.J., Viersma T.J. *Regeltechniek.* Agon Elsevier, Amsterdam, 1972.

[5] De Schutter J., Van Brussel H. *Lecture notes oft the short course on Computer Controlled Motion.* K.U.L., Heverlee, Belgium, 1992.

[6] Dorf R.C. *Modern Control Systems.* Addison-Wesley, New York, 1989.

[7] Eppinger S.D., Seering W.P. *Modelling robot flexibility for end point force control.* IEEE International Conference on Robotics and Automation, 1988.

[8] Fleming P.J., Pashkevich A.P. *Computer-Aided Control System Design Using a Multi-Objective Optimization Approach.* Control '85 Conference, Cambridge, U.K., 174-179, 1985.

[9] Francis B.A. *A Course in H_∞ Control.* Springer-Verlag, 1988.

[10] Franklin G.F., Powell J.D. *Digital Control of Dynamic Systems.* Addison-Wesley, New York, 1980.

[11] Grace A.C.W. *Computer-Aided Control System Design Using Optimization Techniques.* PhD. thesis, University of Wales, Bangor, Gwynedd, UK, 1989.

[12] Nye W.T., Tits L.T. *An application-oriented, optimization-based methodology for interactive design of engineering systems.* Int. J. Control, Vol. 43, No. 6, 1693-1721, 1986.

[13] Olsson G., Piani G. *Computer Systems for Automation and Control.* Prentice Hall, New York, 1992.

[14] Scholten H. *Regelen met Fuzzy Logic en PID.* Elektuur B.V., Beek (NL), 1992.

[15] Toivonen H.T. *A multiobjective linear quadratic Gaussian control problem.* IEEE Trans. on Automatic Control, AC-29, 279-280, 1984.

[16] Vanvuchelen P, De Moor B. *A Multi-Objective Optimization Approach for Parameter Setting in System and Control Design.* ESAT-SISTA report, 1992.

[17] Zakian V., Al-Naib U. *Design of dynamical and control systems by the method of inequalities.* Proc. IEEE, Vol. 120, No. 11, November 1973.

[18] Ziegler J., Nichols N. *Optimum Settings for Automatic Controllers.* Transactions on the A.S.M.E., 1942.