Automatic Control Laboratory, ETH Zürich
Profs. M. Morari, J. Lygeros

Manual prepared by: Kimberly Chung
Revision from: September 13, 2013

# IfA Fachpraktikum - Experiment 1.2 :

# MAN IN THE LOOP

---

The Man-in-the-loop [1] [2] exercise is a graphical user interface (GUI) based Matlab simulation that interactively teaches the user some of the basic concepts of feedback control. You will:

- review transfer functions, Bode diagrams, and stability theory.
- use a joystick to follow a moving dot (a reference), and your controller transfer function will be estimated and analyzed.
- control an inverted pendulum and balancing board with the joystick.

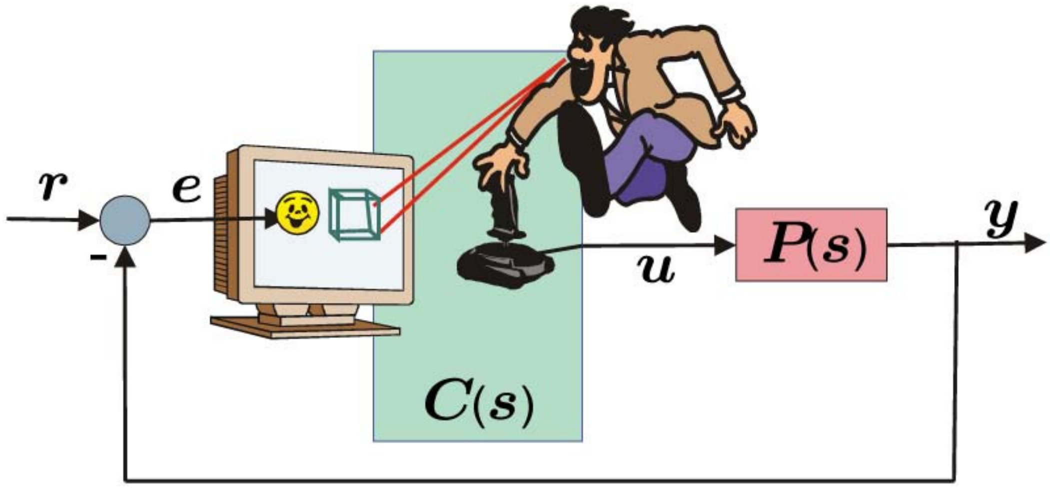



Figure 1: "Man in the loop" simulation exercise framework. Source: [1].

---

[1] Simulation written by René Waldvogel.
[2] Code Maintenance: Kimberly Chung.

# Contents

# Chapter 1

# Simulation Overview

This chapter has an overview of the "Man in the loop" simulation exercise with sample exercises. There are three classes of exercises: control theory review, human controller analysis, and unstable system control.

## 1.1 Control Theory

The user completes exercises on transfer functions, Bode diagrams, and stability theory. Figure 1.1 shows an example.

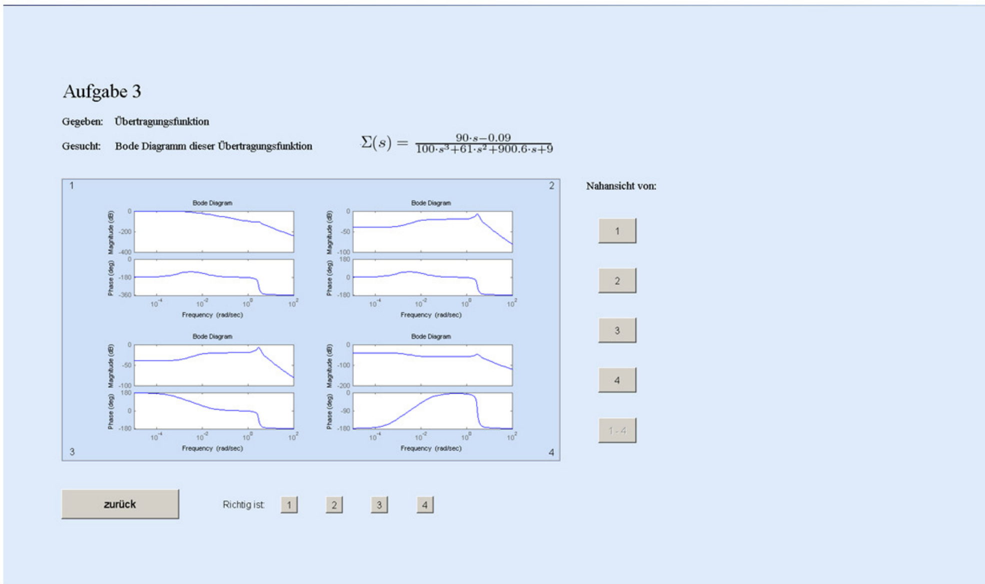



Figure 1.1: Theory review exercise sample.

## 1.2 Human controller

The user follows a moving dot, representing a plant, with a joystick. Then the controller transfer function of the user is estimated and analysis performed. Figure 1.2 shows an example.

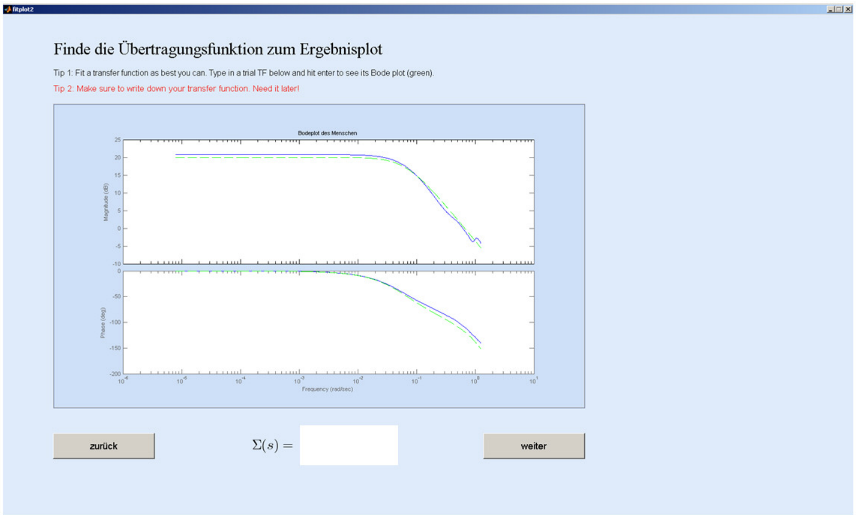



Figure 1.2: Human controller exercise sample.

## 1.3 Fun with inverted pendulum and balancing board

The user controls two classic unstable systems with a joystick. The inverted pendulum GUI is shown in Fig. 1.3.

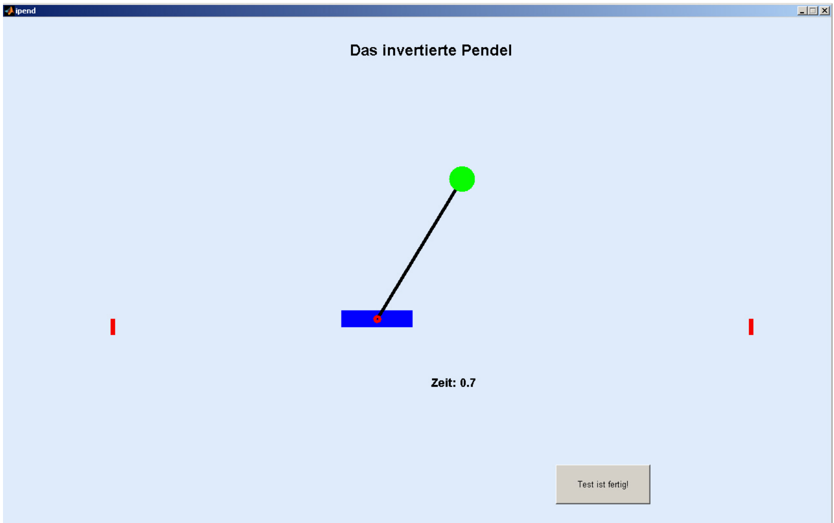



Figure 1.3: Inverted pendulum exercise.

# Chapter 2

# Lab Session Tasks

This chapter describes the instructions for starting the graphical user interface (GUI) simulation in Matlab. It also contains theory that might be helpful in completing some of the exercises.

## 2.1 Running the simulation

### 2.1.1 How to start simulation

1. Double-click the icon named `ifa_1_2.ps` on the Desktop. This will retrieve the necessary files, start Matlab and compile the files which takes about 5 minutes to complete. Please be patient.
2. Type ≪start≫ in the command window (without ≪≫).
3. Follow the screen instructions. Please read Section 2.1.2 as it has a few things the user needs to be aware of about the GUIs. See Section 2.1.3 if Matlab crashes. For assistance with theoretical background, see Section 2.2.

### 2.1.2 Important GUI tips - READ IT!

- Please do not have other programs running - the software is memory intensive!
- **Please be patient with Matlab Graphical User Interfaces (GUIs)**. Please wait until the screen is fully loaded before doing anything. Please do not click around too fast.
- Most exercises with the joystick are limited to 30-40 seconds and will terminate automatically. The inverted pendulum and balancing board simulations at the end are limited to 70 seconds. When joystick simulations are running, on-screen buttons will be disabled even when visible.
- If Matlab freezes, see help in Section 2.1.3 or take a note of the GUI file name at the upper left corner of the window and ask an assistant.

### 2.1.3 When things go wrong

- If the GUI is behaving abnormally, try using the back button "Zurück" and try again.
- If the GUI is not responsive or crashes and must be restarted:
    1. Restart Matlab by double-clicking on the Matlab icon on the Desktop. Note: Do not double-click `ifa_1_2.ps` since this would get a fresh copy of all files and would again do a compilation of these files!
    2. Make sure the current directory is at the root: `C:\Scratch\Man_in_the_loop`

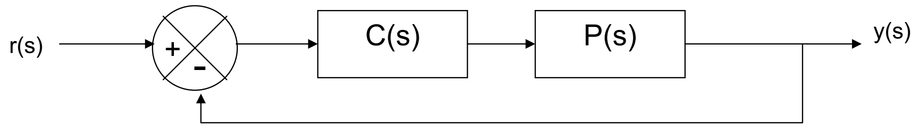



Figure 2.1: Block diagram of a feedback system.

3. Type in ≪restart≫ in the command window (without ≪≫).

This will open the GUI at a state close to where left off. However, it will not always bring it back to exactly where left off. Alternatively, if Matlab has to be restarted, just double-click `ifa_1_2-restart.ps`.

- If the user terminates a joystick exercise using 8 or 9 button, the program unloads. Just press the "test ist fertig" and go to the next screen. There will be an option to try again.

### 2.1.4 After completing the experiment

Please fill out the feedback formular on the registration page under `MyExperiments`. Each Student/Partizipatn has to fill out its own feedback. This will help us to improve the experiment. Thank you for your help.

## 2.2 Theory and formulas

This section provides theory and formulas that may be needed during the exercise. Theory related to closed-loop systems will be based on the block diagram in Fig. 2.1.

### 2.2.1 Nyquist stability criterion

The stability of the *closed-loop* system in Fig. 2.1 can be determined with its *open-loop* properties using the Nyquist stability criterion. The open-loop transfer function of the system in the figure is defined as

$$L(s) = P(s)C(s) \tag{2.1}$$

where $P(s)$ is the plant transfer function and $C(s)$ is the controller transfer function. The Nyquist contour is a plot of the magnitude of $L(s)$ in Eq. (2.1) versus the angle of $L(s)$.

**Notation:**

- $P=$ number of open-loop poles of $L(s)$ in the right-half-plane (RHP) or Rechte-Halb-Ebene (RHE).
- $N=$ number of encirclements of -1 by the Nyquist contour. Counterclockwise is positive and clockwise is negative.
- $Z=$ number of RHP poles (unstable) of the closed-loop system.

**The Nyquist stability criterion**

The Nyquist criterion states

$$Z = P - N \tag{2.2}$$

where the variables are defined in the notation above. In words, Eq. (2.2) states that the number of RHP poles (unstable poles) of the *closed-loop* system is equal to the number of *open-loop* poles in the RHP minus the number of encirclements of -1 by the Nyquist contour. Thus, in order to have a stable closed-loop system, (i.e. $Z = 0$), we want $P = N$. In words, the closed-loop system is stable if the number of unstable open-loop poles is equal to the number of **counterclockwise** encirclements of -1 by the Nyquist contour. See [2] for further details.

### 2.2.2 Crossover frequency and plant limitation

The controllability of a plant is inherently limited by the properties of the plant. We can in fact determine whether a plant can be controlled satisfactorily with any controller *before* a controller is designed. There is no point in designing a controller if the plant can never be controlled to meet performance specifications. One such specification is the crossover frequency, $w_c$, requirement of the open-loop transfer function $L(s)$ defined in Eq. (2.1). A plant $P(s)$ in Fig. 2.1 may have poles in the right-half-plane (RHP), zeros in the RHP, or delay. Any one of these imposes limitations on the achievable cross over frequency of $L(s)$.

**Notation:**

- $\pi^+$ = real RHP (RHE) pole of plant $P(s)$.
- $\zeta^+$ = real RHP (RHE) zero of plant $P(s)$.
- $\tau$ = delay in $P(s)$. [sec]

**Plant limitation on $w_c$**
For a plant with a real RHP pole, a real RHP zero, or delay, the following condition needs to be satisfied for closed loop stability and adequate performance:

$$\max\left\{2\pi^+\right\} \le w_c \le \min\left\{\frac{\zeta^+}{2},\ \frac{1}{\tau}\right\}. \tag{2.3}$$

Equation (2.3) states that $w_c$ must be:

- at least twice as large as the RHP pole of the plant.
- smaller than half of the RHP zero of the plant.
- smaller than the "delay frequency", $\frac{1}{\tau}$.

If the plant has multiple RHP poles or RHP zeros, then the dominant one is the pole with the largest real part or the zero with the smallest real part, respectively. See [4] for further details.

### 2.2.3 Controller gain and system stability using Routh's stability criterion

The stability of a transfer function is determined by the location of its poles. If there are any RHP (unstable) poles, the system is unstable. So how do we determine if there are any unstable poles in the system? Consider the following transfer function expressed as a ratio of polynomials:

$$T(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^n + a_1 s^{n-1} + \cdots + a_n}. \tag{2.4}$$

The poles of the transfer function in Eq. (2.4) are the roots of the denominator. The denominator polynomial is called the *characteristic polynomial.* Thus, determining the stability of the system is equivalent to determining if there are RHP roots to the characteristic equation. One way of doing this is finding all roots of the characteristic equation and determining if any of them are in the RHP (unstable). Another way is to use Routh's stability criterion which tells us the stability of a transfer function without finding the poles directly. This criterion is useful in determining the range of gain allowable for a system with a P-controller to maintain stability. If the criterion is satisfied, the system has no unstable poles and thus stable. The criterion has two parts:

**Condition 2.1** *(Necessary): all the coefficients $a_1, \cdots, a_n$ of the characteristic polynomial in Eq. (2.4) are positive and nonzero.*

**Condition 2.2** *(Necessary and sufficient): all the elements in the first column of the Routh array are positive.*

The Routh array in Condition 2.2 is built using the characteristic polynomial coefficients as follows:

$$\begin{array}{rcccc} s^n: & 1 & a_2 & a_4 & \cdots \\ s^{n-1}: & a_1 & a_3 & a_5 & \cdots \\ s^{n-2}: & b_1 & b_2 & b_3 & \cdots \\ s^{n-3} & c_1 & c_2 & c_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ s^2: & * & * & & \\ s: & * & & & \\ s^0: & * & & & \end{array} \tag{2.5}$$

where the elements in rows $s^{n-2}$ and $s^{n-3}$ (and so forth) are determined using the characteristic polynomial coefficients $a_1, \cdots, a_n$ as follows:

$$\begin{aligned} b_1 &= -\frac{\det\begin{bmatrix} 1 & a_2 \\ a_1 & a_3 \end{bmatrix}}{a_1} \\ b_2 &= -\frac{\det\begin{bmatrix} 1 & a_4 \\ a_1 & a_5 \end{bmatrix}}{a_1} \\ b_3 &= -\frac{\det\begin{bmatrix} 1 & a_6 \\ a_1 & a_7 \end{bmatrix}}{a_1} \\ c_1 &= -\frac{\det\begin{bmatrix} a_1 & a_3 \\ b_1 & b_2 \end{bmatrix}}{b_1} \\ c_2 &= -\frac{\det\begin{bmatrix} a_1 & a_5 \\ b_1 & b_3 \end{bmatrix}}{b1} \\ c_3 &= -\frac{\det\begin{bmatrix} a_1 & a_7 \\ b_1 & b_4 \end{bmatrix}}{b_1} \end{aligned} \tag{2.6}$$

The elements $b_1, b_2, \cdots$ of the Routh array in Eq. (2.5) and Eq. (2.6) are *not* the coefficients of the numerator of the transfer function in Eq. (2.4). To demonstrate how Routh's criterion is used, below is an example using a second order transfer function.

**2nd order system Routh's criterion example**

Consider a second order closed-loop transfer function of the system in Fig. 2.1 as in following

$$T(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{b_0 s + b_1}{s^2 + a_1 s + a_2}. \tag{2.7}$$

For the system in Eq. (2.7) to be stable, its characteristic polynomial has to satisfy the necessary and sufficient conditions of the Routh criterion.

1. Check if $a_1$ and $a_2$ are both positive (Condition 2.1).
2. Check if the elements in the first column of the Routh array are positive (Condition 2.2).

$$\begin{array}{llll} s^2: & 1 & a_2 & 0 \\ s^1: & a_1 & 0 & 0 \\ s^0: & b_1 = -\frac{\det\begin{bmatrix} 1 & a_2 \\ a_1 & 0 \end{bmatrix}}{a_1} = a_2 & b_2 = -\frac{\det\begin{bmatrix} 1 & 0 \\ a_1 & 0 \end{bmatrix}}{a_1} = 0 & 0 \end{array} \tag{2.8}$$

Re-write Eq. (2.8) for clarity

$$\begin{array}{cccc} s^2: & 1 & a_2 & 0 \\ s^1: & a_1 & 0 & 0 \\ s^0: & a_2 & 0 & 0 \end{array} \tag{2.9}$$

For stability, the elements in the first column of the Routh array, $a_1$ and $a_2$, in Eq. (2.9) need to be positive. This is in fact the same as the necessary condition.

The two conditions in Eq. (2.8) and Eq. (2.9) show that for a *second order* characteristic equation, both the necessary and sufficient conditions are satisfied if all the coefficients of the characteristic polynomial are positive. Thus, for a second order system, it is stable iff all the coefficients of the characteristic polynomial are positive; a Routh array does not need to be created to determine stability. If a proportional controller is used so that $C(s) = K$, the coefficients $a_1$ and $a_2$ will be functions of $K$. Then the Routh criterion can be used to determine the range of $K$ that will keep the system stable. See [2] for further details.

# Appendix A

# Library of Transfer Functions

This section contains s-domain transfer functions that are frequently used in control system analysis. For each function, Nyquist, Bode plot, impulse response, step response, and analog/digital realization are provided. The contents in this section were taken directly from [3].

### A.0.1 Integrator Element

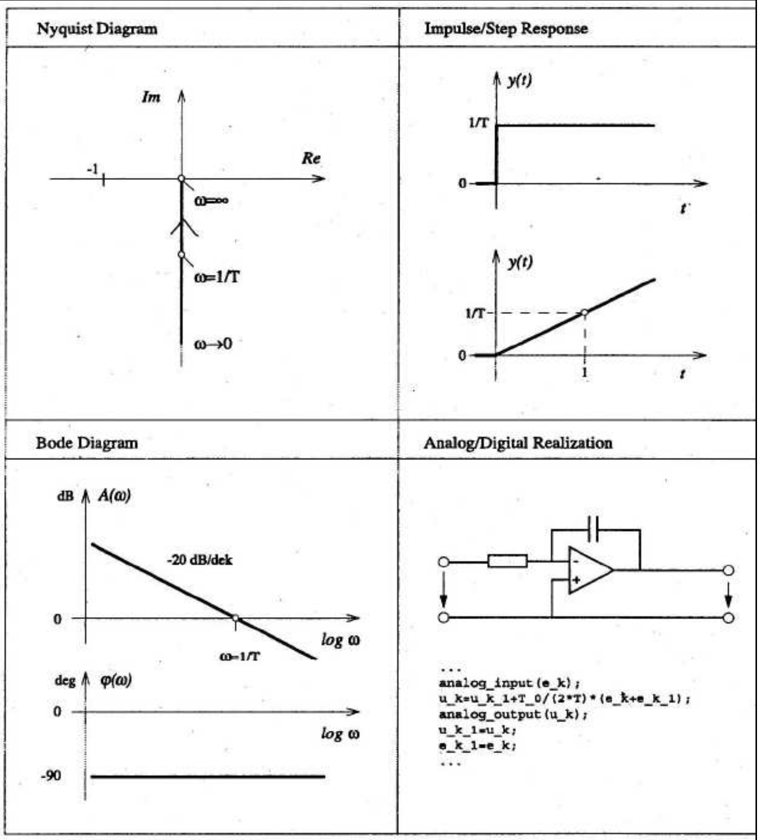
Element Acronym: $\boxed{\mathrm{I}}$

Transfer Function: $\Sigma(s) = \frac{1}{T \cdot s}$

Poles/Zeros: $\pi_1 = 0,\ \zeta_1 = \infty$

Internal Description: $\frac{d}{dt}x(t) = u(t)$

$y(t) = \frac{1}{T} \cdot x(t)$



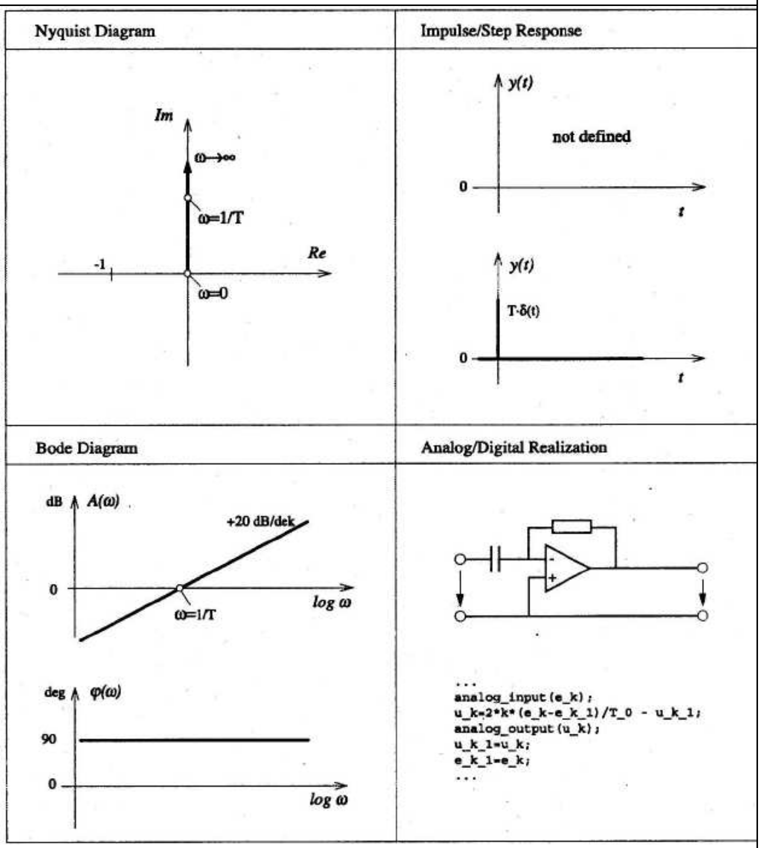

### A.0.2 Differentiator Element

Element Acronym: $\boxed{\mathrm{D}}$

Transfer Function: $\Sigma(s) = T \cdot s$

Poles/Zeros: $\pi_1 = \infty,\ \zeta_1 = 0$

Internal Description: $y(t) = T \cdot \dot{u}(t)$



### A.0.4 Realizable Derivative Element ("Dirty D")

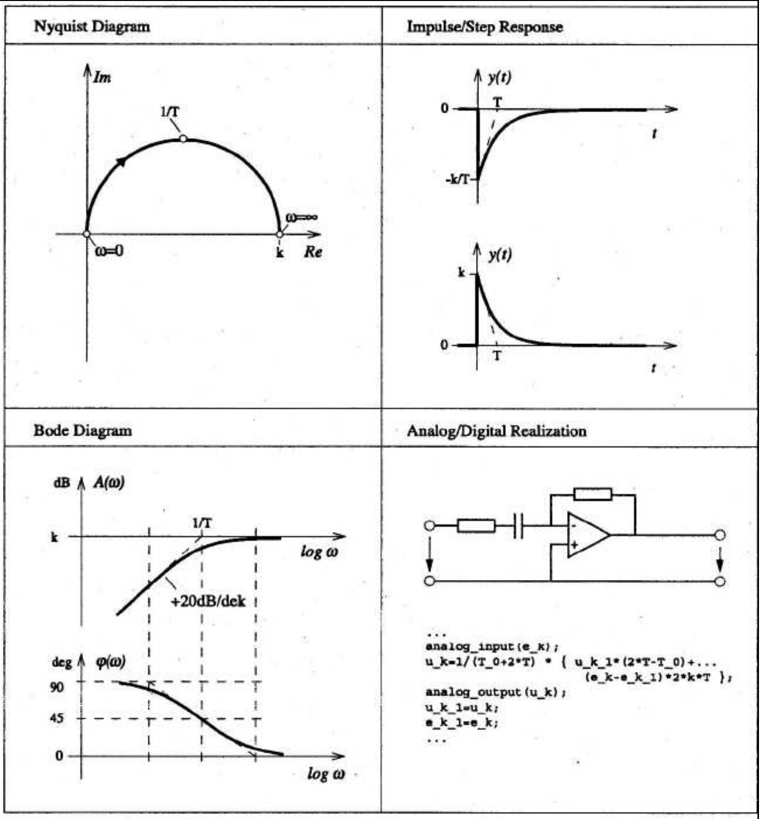
Element Acronym: $\boxed{\text{DT-1}}$

Transfer Function: $\Sigma(s) = k \cdot \frac{T \cdot s}{T \cdot s + 1} = k \cdot \left(1 - \frac{1}{T \cdot s + 1}\right)$

Poles/Zeros: $\pi_1 = -\frac{1}{T}$, $\zeta_1 = 0$

Internal Description: $\frac{d}{dt}x(t) = -\frac{1}{T} \cdot x(t) + \frac{1}{T} \cdot u(t)$

$y(t) = -k \cdot x(t) + k \cdot u(t)$




### A.0.3 First-Order Element

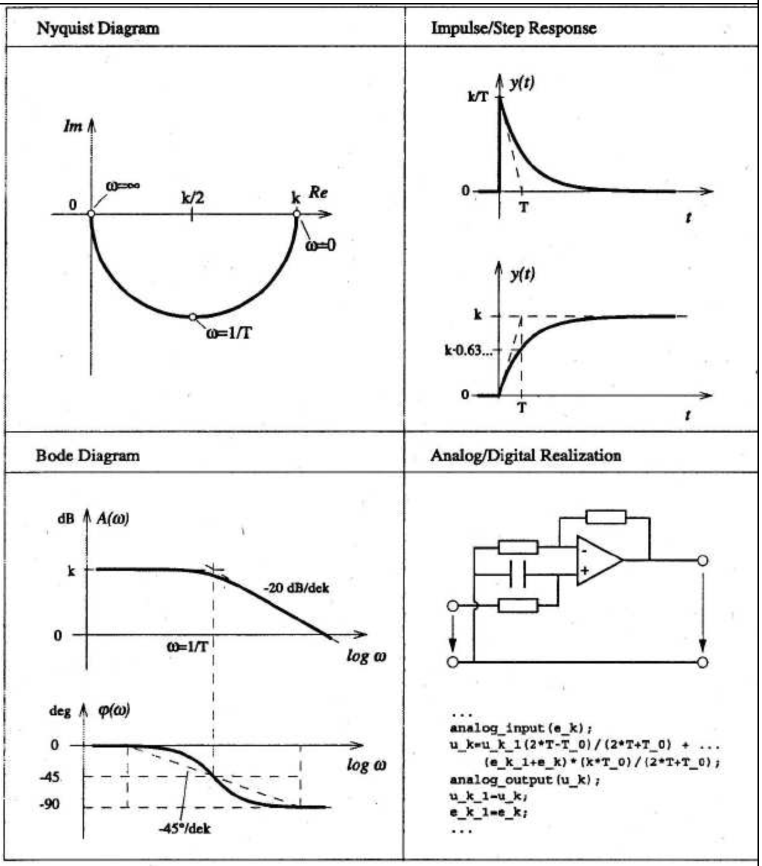
Element Acronym: $\boxed{\text{PT-1}}$

Transfer Function: $\Sigma(s) = \frac{k}{1 + T \cdot s}$

Poles/Zeros: $\pi_1 = -\frac{1}{T}$, $\zeta_1 = \infty$

Internal Description: $\frac{d}{dt}x(t) = -\frac{1}{T} \cdot x(t) + \frac{1}{T} \cdot u(t)$

$y(t) = k \cdot x(t)$



### A.0.5 Second-Order Element

Element Acronym: $\boxed{-}$

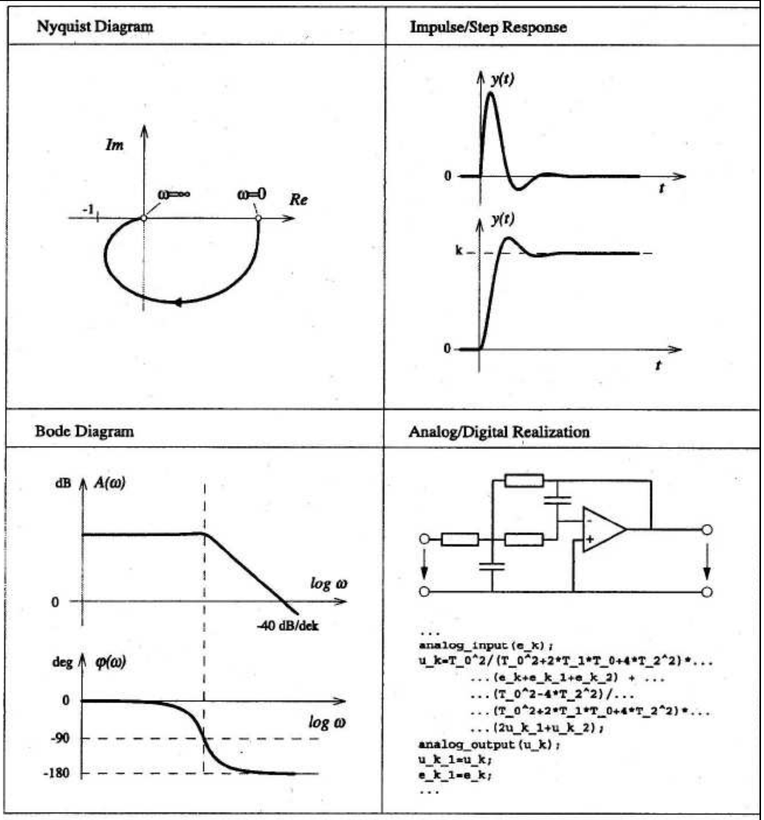
Transfer Function: $\Sigma(s) = k \cdot \frac{\omega_0^2}{s^2+2\cdot\delta\cdot\omega_0\cdot s+\omega_0^2}$

Poles/Zeros: $\pi_{1,2} = -w_0 \cdot \delta \pm w_0\sqrt{\delta^2-1}$, $\zeta_{1,2} = \infty$

Internal Description: $\dot{x}_1(t) = x_2(t)$,

$$\dot{x}_2(t) = -\omega_0^2 \cdot x_1(t) - 2\cdot\delta\cdot\omega_0\cdot x_2(t) + u(t)$$

$$y(t) = k \cdot \omega_0^2 \cdot x(t)$$




### A.0.6 Lag Element

Element Acronym: $\boxed{\text{LG-1}}$

Transfer Function: $\Sigma(s) = k \cdot \frac{1+T\cdot s}{1+\alpha\cdot T\cdot s} \qquad 1 < \alpha$

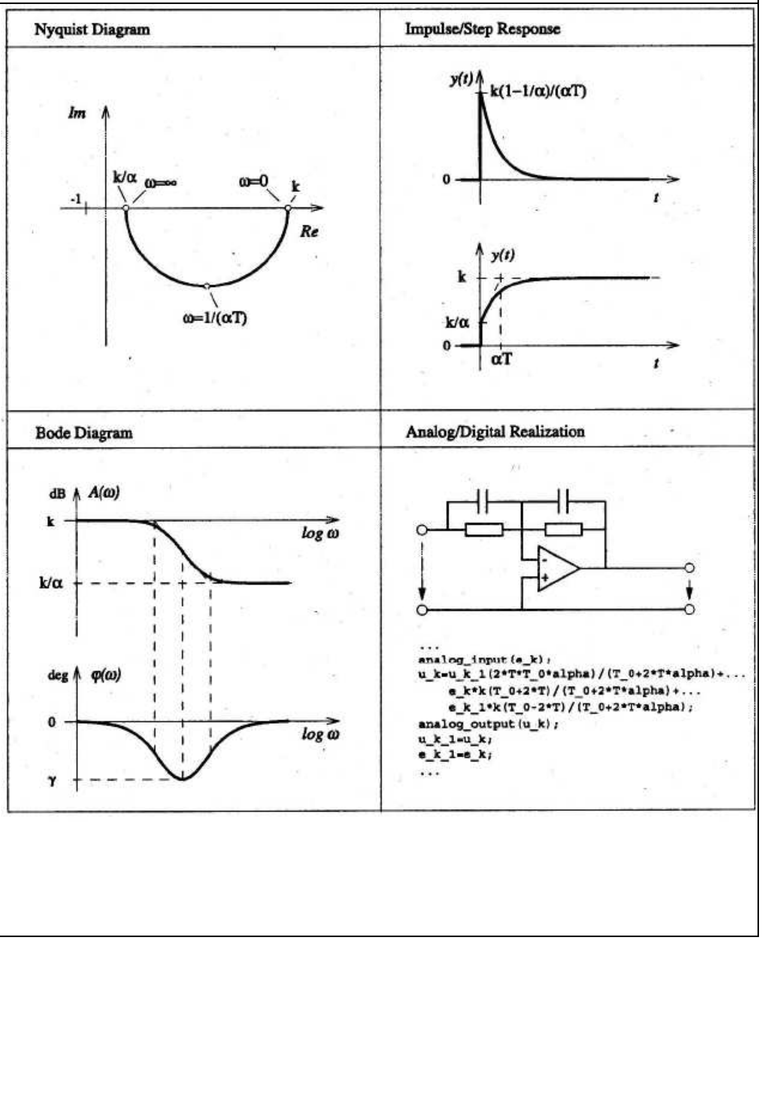
Poles/Zeros: $\pi_1 = -\frac{1}{\alpha\cdot T}$, $\zeta_1 = -\frac{1}{T}$

Internal Description: $\frac{d}{dt}x(t) = -\frac{1}{\alpha\cdot T}\cdot x(t) + u(t)$

$$y(t) = \frac{k\cdot(\alpha-1)}{\alpha^2\cdot T}\cdot x(t) + \frac{k}{\alpha}\cdot u(t)$$

Phase minimum: $\gamma = \arctan(1/\sqrt{\alpha}) - \arctan(\sqrt{\alpha})$

at frequency: $\hat{\omega} = [T\cdot\sqrt{\alpha}]^{-1}$



### A.0.7 Lead Element

Element Acronym: $\boxed{\text{LD-1}}$

Transfer Function: $\Sigma(s) = k \cdot \frac{1+T\cdot s}{1+\alpha\cdot T\cdot s} \quad 0 < \alpha < 1$

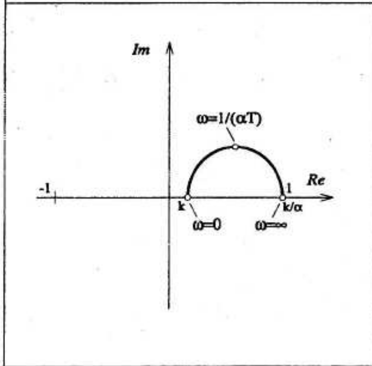
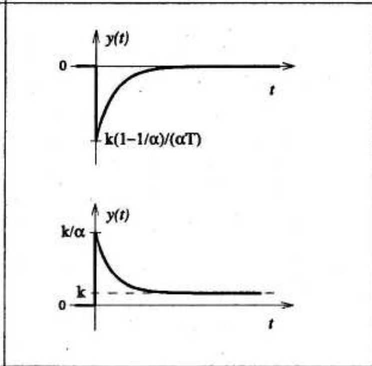
Poles/Zeros: $\pi_1 = -\frac{1}{\alpha\cdot T}$, $\zeta_1 = -\frac{1}{T}$

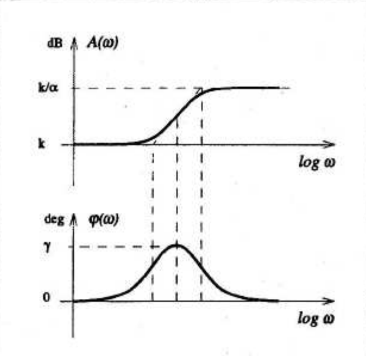
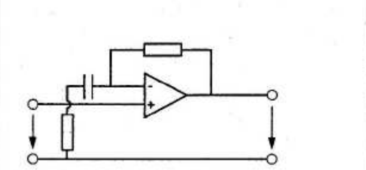
Internal Description: $\frac{d}{dt}x(t) = -\frac{1}{\alpha\cdot T}\cdot x(t) + u(t)$

$y(t) = \frac{k\cdot(\alpha-1)}{\alpha^2\cdot T}\cdot x(t) + \frac{k}{\alpha}\cdot u(t)$

Phase maximum: $\gamma = \arctan(1/\sqrt{\alpha}) - \arctan(\sqrt{\alpha})$

at frequency: $\hat{\omega} = [T\cdot\sqrt{\alpha}]^{-1}$

| Nyquist Diagram | Impulse/Step Response |
|---|---|
|  |  |

| Bode Diagram | Analog/Digital Realization |
|---|---|
|  |  |

```
...
analog_input(e_k);
u_k=u_k_1(2*T*T_0*alpha)/(T_0+2*T*alpha)+...
     e_k*k(T_0+2*T)/(T_0+2*T*alpha)+...
     e_k_1*k(T_0-2*T)/(T_0+2*T*alpha);
analog_output(u_k);
u_k_1=u_k;
e_k_1=e_k;
...
```

### A.0.8 PID Element

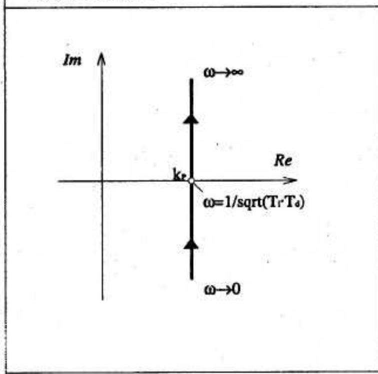
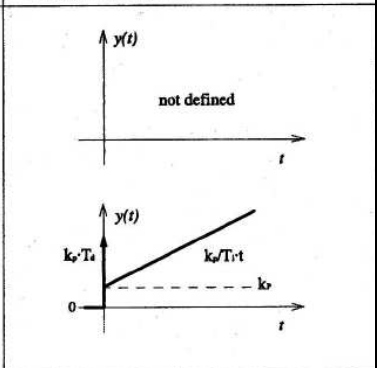
Element Acronym: $\boxed{\text{PID}}$

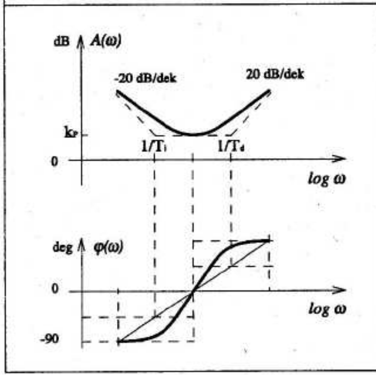
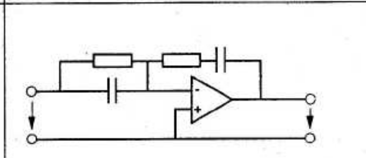
Transfer Function: $\Sigma(s) = k_p \cdot (1 + \frac{1}{T_i\cdot s} + T_d\cdot s)$

Poles/Zeros: $\pi_1 = 0$, $\pi_1 = \infty$, $\zeta_{1,2} = -\frac{1}{2T_d} \pm \sqrt{\frac{1}{4T_d^2} - \frac{1}{T_i T_d}}$

Internal Description: $\dot{x}_1(t) = \frac{1}{T_i}\cdot u(t)$

$y(t) = k_p\cdot(u(t) + x_1(t) + T_d\cdot\dot{u}(t))$

| Nyquist Diagram | Impulse/Step Response |
|---|---|
|  |  |

| Bode Diagram | Analog/Digital Realization |
|---|---|
|  |  |

```
...
analog_input(e_k);
u_k=e_k*kp(1+T_0/(2*T_i)+2*T_d/T_0)+...
     e_k_1*kp(T_0/T_i-4*T_d/T_0)+...
     e_k_2*kp(-1+T_0/(2*T_i)+2*T_d*/T_0)+...
     u_k_2;
analog_output(u_k);
u_k_1=u_k;
e_k_1=e_k;
...
```

### A.0.9 First-Order All-Pass Element

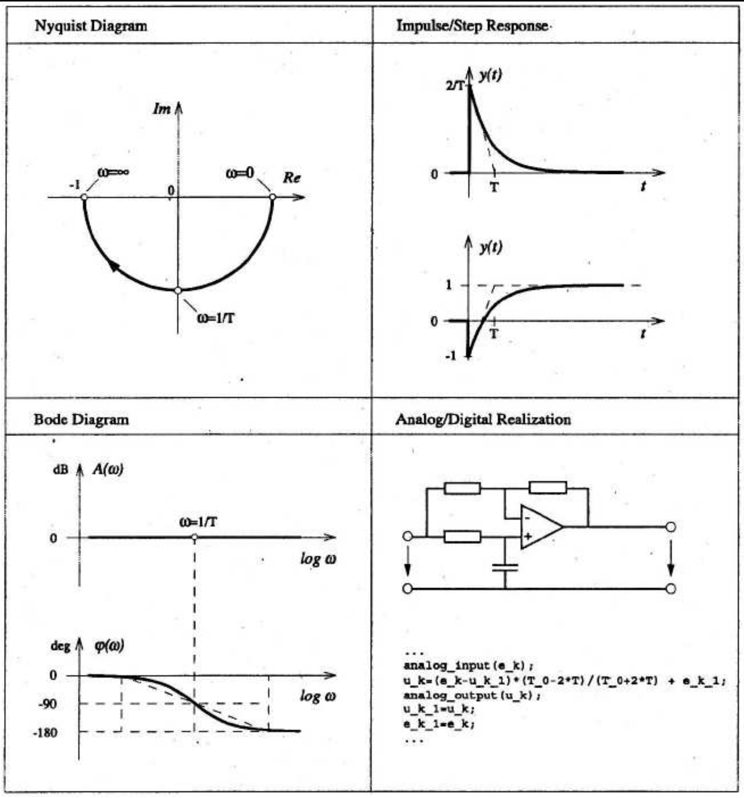
Element Acronym: $\boxed{\text{AP-1}}$

Transfer Function: $\Sigma(s) = \frac{1-T\cdot s}{1+T\cdot s}$

Poles/Zeros: $\pi_1 = -\frac{1}{T}$, $\zeta_1 = \frac{1}{T}$

Internal Description: $\dot{x}(t) = -\frac{1}{T}\cdot x(t) + u(t)$

$y(t) = \frac{2}{T}\cdot x(t) - u(t)$



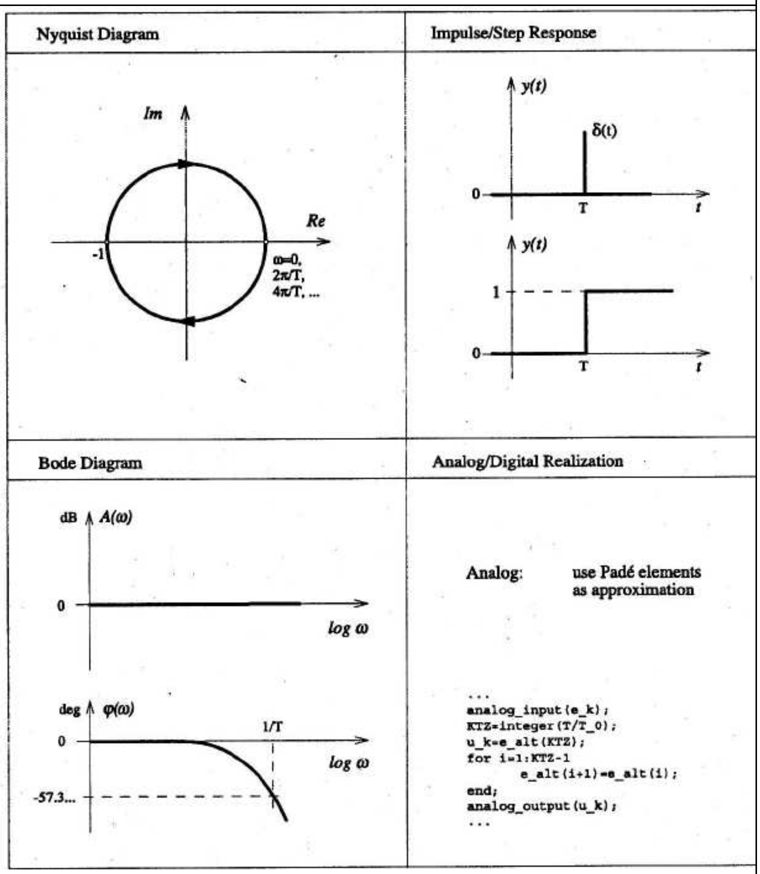

### A.0.10 Delay Element

Element Acronym: $\boxed{-}$

Transfer Function: $\Sigma(s) = e^{-s\cdot T}$

Poles/Zeros: not a real-rational element

Internal Description: $y(t) = u(t-T)$



# Bibliography


[1] Block diagram of "man in the loop" [online image]. Retrieved May 17, 2009, from www.imrt.ethz.ch. http://www.imrt.ethz.ch/education/lectures/control_laboratory/experiments#man.

[2] Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems*, 5th edition ed. Prentice Hall, 2006.

[3] Guzzella, L. *Analysis and Synthesis of Single-Input-Single-Output Control Systems*. Vdf Hochschulverlag ETH Zurich, 2007.

[4] Skogestad, S., and Postlethwaite, I. *Multivariable Feedback Control*, 2nd edition ed. John Wiley and Sons, 2005.