Thomas Westermann

System Analysis

Modelling, Simulation, Visualization

Simulations are performed with Maple

Prof. Dr. Thomas Westermann
Hochschule Karlsruhe - Technik und Wirtschaft
Fakultät EIT
Moltkestr. 30
D 76133 Karlsruhe

http://www.home.hs-karlsruhe.de/~weth0002/

e-mail: thomas.westermann@hs-karlsruhe.de

Additional Material: All Maple-Worksheets which are used and described in this textbook are available on the web-page
http://www.home.hs-karlsruhe.de/~weth0002/Maple

ISBN 978-3-9820650-0-7

ISBN 978-3-9820650-1-4 (eBook)

Einbandsabbildung: Übertragungsverhalten eines Tiefpasses

Einbandsentwurf: Thomas Westermann

Gedruckt auf saurefreiem Papier

Sonderauflage für die Summer School in Chengdu (China) 2019.

Preface

The occasion to compile this material on systems analysis and to publish it in a book was the lecture "Modelling, Simulation and Visualization" which took place at the Summer School in Chengdu from 8th to 19th July 2019. For the participants of the Summer School this special edition was prepared.

The basis is a computer lab in the course of studies Electrical Engineering, which has been taking place for several semesters at the Karlsruhe University of Applied Sciences. In this lecture, topics from mathematics, physics and electronics are combined in order to point out multidisciplinary aspects that play an important role in all of the involved disciplines. The signal and system analysis connects different topics like e.g. the description of oscillations and waves, the Fourier analysis or the description of alternating current circuits.

The contents of the lecture thus form a link between mathematics, in which the mathematical basics in complex numbers, solving differential equations and Fourier analysis, physics, which describes physical effects in the form of laws, and electronics, in which the circuit foundations are laid, and computer science, which provides the basic structures of programming.

In this textbook, we start from a concrete task of signal and system analysis. First we present the mathematical correlations and the mathematical methods in order to work on the problem theoretically. Algorithms are then created to simulate the problem on the computer. The computer framework chosen for programming and visualization is Maple.

Maple distinguishes itself from other simulation and computational tools by the fact that we can implement complex algorithms and present easily the results in the form of graphs, 3D representations and animations. Moreover, the graphic possibilities are outstanding. By knowing a few commands like **plot**, **plot3d**, **animate**, **animate3d** and **Explore** we are not only able to visualize the simulations easily, but we can also display the results very fast and impressive in form of animations.

I would like to thank especially Xiaoqian and Prof. Yu Liu, who enabled me to hold the Summer School, and Chunyang Li for supporting me with her valuable and intensive help for each and every problem I had. Also thanks to my student Mayur Shelke for improving the English version of the course material.

Karlsruhe, March 2019 *Thomas Westermann*

Table of Contents

Chapter 1

Simulating the Transmission Behavior of Electrical Circuits

1 Simulating the Transmission Behavior of Electrical Circuits

Overview: In this chapter we will use the example of RCL electrical filter circuits to determine the transmission behavior of the systems, if the input voltage is sinusoidal or cosine shaped. In this situation the transfer function is determined by the ratio of output voltage to input voltage: $\frac{\widehat{U}_A(t)}{\widehat{U}_0(t)}$, if the input voltage is $\widehat{U}_0(t) = e^{i\,w\,t}$. Because if the input voltage is a complex alternating voltage, for all components an Ohm's law with complex resistors can be assumed. This approach provides a simple introduction to the description of linear systems in the frequency range.

Keywords: Filter circuits, Transfer function, Amplitude ratio, Locus curve, Bode diagram.

We will develop a procedure how to handle the complex transfer function for electrical circuits if the circuit consists of ohmic resistors R, capacitors C and inductors L. Subsequently, special low-pass, high-pass, band-pass and band-stop circuits are discussed. The goal is not only to calculate the transfer function for the given circuit but to design the circuit regarding a given cut-off frequency such that the transfer function $\frac{\widehat{U}_A}{\widehat{U}_0}$ has an appropriate behavior.

Qualitatively, the following slopes of the transfer function $|\frac{\widehat{U}_A}{\widehat{U}_0}|$ are demanded for the different filter circuits:

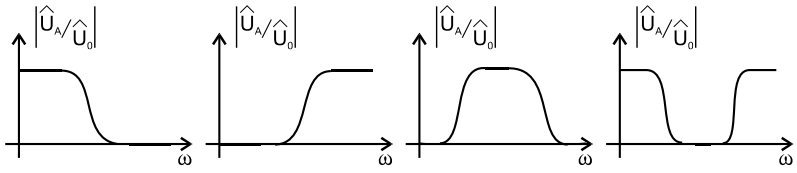


Figure 1.1. Transmission behavior of low-pass, high-pass, band-pass and band-stop

A low-pass filter has the property that low frequencies can pass undisturbed, while high frequencies are blocked. This behavior can be seen in the left picture in Fig. 1.1. Characterizing is the frequency at which the transition from the passing to the blocking area takes place. However for a high-pass filter, the opposite frequency behavior is requested: High frequencies can pass, low frequencies are blocked by the circuit. Obviously, band-pass and band-stop are defined.

For the realization of all filters we will consider only circuits, which consist of identical Π or T elements. Only with this restriction the required transmission characteristics can be achieved.

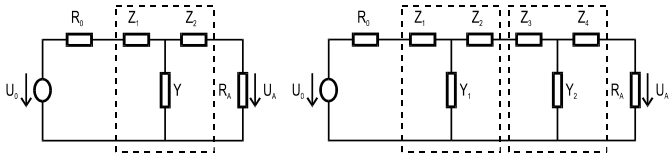


Figure 1.2. Circuits based on T-elements

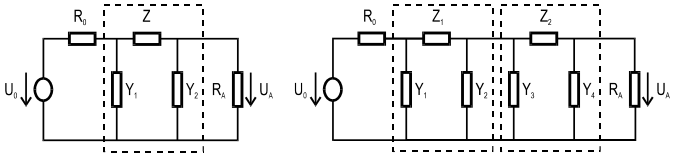


Figure 1.3. Circuits based on Π-elements

The **transfer function** $H(\omega)$ is defined as the ratio of output voltage to input voltage, when the input voltage is assumed to be a complex alternating voltage $\widehat{U}_0(t) = U_0\, e^{i\omega t}$:

$$H(\omega) = \frac{\widehat{U}_A(t)}{\widehat{U}_0(t)}$$

This complex transfer function can be graphically described via the magnitude $|H(\omega)|$ and the phase $\varphi(\omega)$,

$$H(\omega) = |H(\omega)|e^{i\varphi(\omega)}$$

with $|H(\omega)| = \sqrt{H(\omega) \cdot H^*(\omega)}$ and $tan(\varphi(\omega)) = \frac{Im(H(\omega))}{Re(H(\omega))}$.

Note that within the following algorithm, the individual impedance have not yet been specified, Namely the formulas are valid for Π- and for T-elements, as well. The impedance (transverse and longitudinal) must be set according to the subsequent table

R_Ω (ohmic resistance)
$\hat{R}_L = i\omega L$ (impedance of a coil with inductance L)
$\hat{R}_C = \frac{1}{i\omega C}$ (impedance of a capacitor with capacitance C).

With Kirchhoff's laws we obtain for the **complex total resistance (= substitute resistance)** $\hat{R}$ equivalent to two complex resistors $\hat{R}_1$ and $\hat{R}_2$:

(a) Series $\quad \hat{R} = \hat{R}_1 + \hat{R}_2$

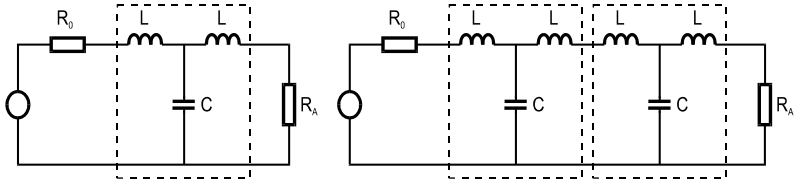
(b) Parallel $\quad \frac{1}{\hat{R}} = \frac{1}{\hat{R}_1} + \frac{1}{\hat{R}_2}$ and $\hat{R} = \frac{\hat{R}_1 \hat{R}_2}{\hat{R}_1 + \hat{R}_2}$

Hence, in the AC circuit the known rules for the equivalent circuit for resistors as used in DC circuits can be applied, if capacitance and inductance become complex resistances:

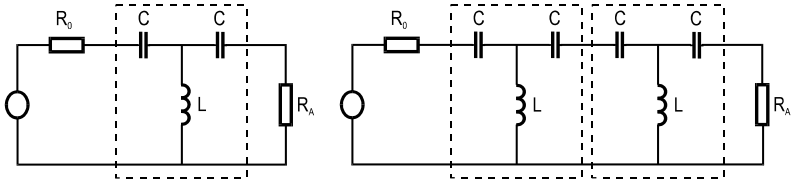
The complex total resistance in series of connected elements is the sum of the complex single resistors. The complex total conductance $\frac{1}{\hat{R}}$ of elements connected in parallel is the sum of the complex individual conductance. The complex total resistance is then the reciprocal of the total conductance.

The following overview shows some typical circuits which will be discussed in this book.

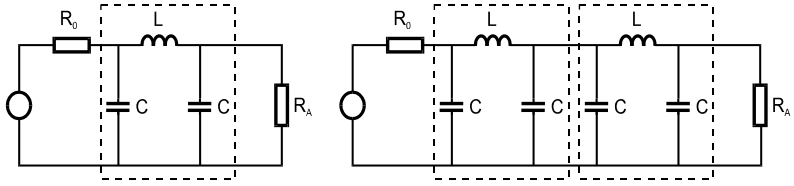
Low-pass filter consisting of T links:



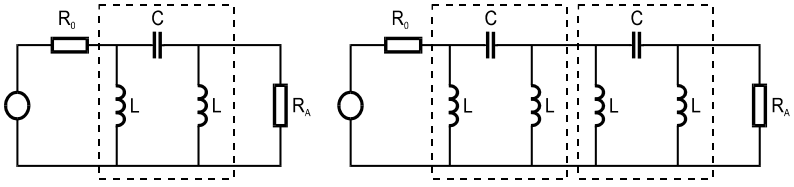
High-pass filter consisting of T links:



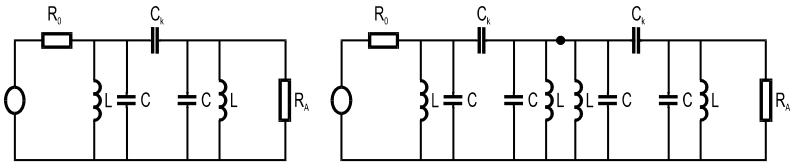
Low-pass filter consisting of Π links:



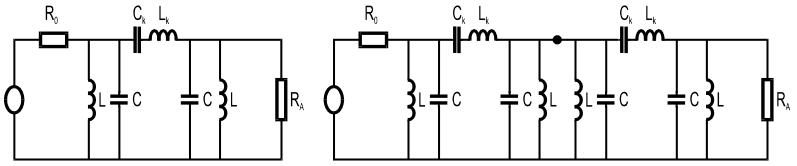
High-pass filter consisting of Π links:



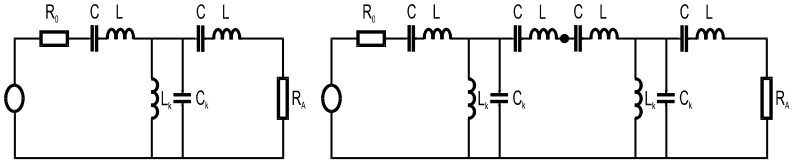
Band-pass filter consisting of Π links:



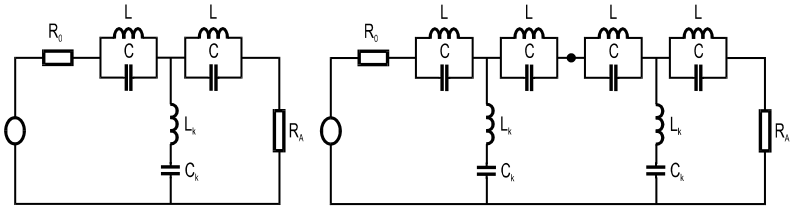
Band-pass filter consisting of Π links:



Band-pass filters consisting of T links:



Band-stop filters consisting of T links:



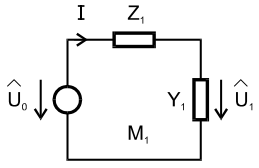
For the naming of the circuits we use the following notation: LP, HP, BP, BS denote the type of circuit: Low-pass, high-pass, band-pass or band-stop; 1Pi, 2Pi, 3Pi or 1T, 2T, 3T for the number of Π- or T-links used. CLC gives the arrangement of the energy storage units in the used Π- or T-links. The circuit is completed with an internal resistor and a load resistor. For example, the name of the second circuit in the third line is: LP2PiCLC.

1.1 Calculation of the Transfer Function $H(\omega)$

We start with the modelling of a circuit with one mesh,and subsequently extend the circuits with more meshes. However, for calculating their transmission behavior, we reduce the number of meshes until we represent the complete circuit only by one mesh. By subsequently applying voltage divider rule we calculate successively all voltages until we finally determine the voltage at the output resistance.

1.1.1 Voltage Divider = circuit with one link

We start with the discussion of a circuit made up of one mesh:



Z_1 = Longitudinal impedance
Y_1 = Cross-impedance

If we apply the mesh rule to the mesh M_1, the input voltage $\widehat{U}_0$ is divided into a voltage drop at Z_1 plus a voltage drop at Y_1:

$$\widehat{U}_0 = Z_1 I + Y_1 I$$

The following applies to the output voltage

$$\widehat{U}_A = Y_1 I$$

so that we can determine the ratio of the two voltages:

$$\frac{\widehat{U}_A}{\widehat{U}_0} = \frac{Y_1 I}{Z_1 I + Y_1 I} = \frac{Y_1}{Z_1 + Y_1}$$

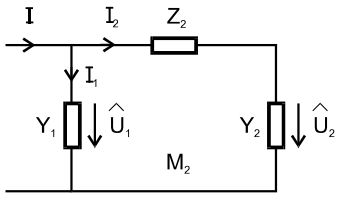
1.1.2 Circuit with two links

Let us now come to a circuit that consists of two meshes. First, we examine the second loop and replace it with a parallel resistor.

$$Y_{1p} = \frac{1}{\frac{1}{Y_1} + \frac{1}{Z_2 + Y_2}} = \frac{Y_1(Z_2 + Y_2)}{Y_1 + Z_2 + Y_2}$$

For the remaining mesh we apply the voltage divider rule

$$\frac{\widehat{U}_1}{\widehat{U}_0} = \frac{Y_{1p}}{Y_{1p} + Z_1}$$



This way we know the voltage drop at Y_{1p} and we can now again apply the voltage divider rule on the second loop

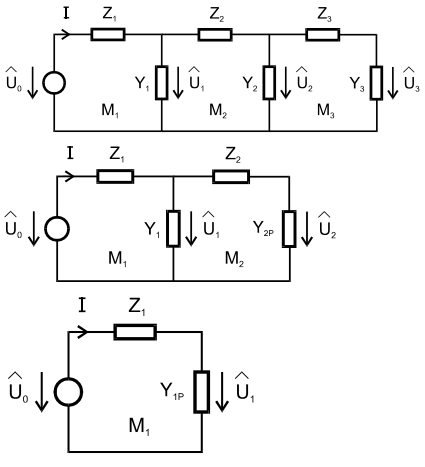
$$\frac{\widehat{U}_2}{\widehat{U}_1} = \frac{Y_2}{Y_2 + Z_2}$$

Algorithm:

Z_1, Z_2 longitudinal impedances
Y_1, Y_2 cross-impedances

$Y_{1p} = Y_1 \cdot (Y_2 + Z_2)/(Y_1 + Y_2 + Z_2)$
$u_1 = u_0 \cdot Y_{1p}/(Z_1 + Y_{1p})$
$u_2 = u_1 \cdot Y_2/(Z_2 + Y_2)$

1.1.3 Circuit with three links



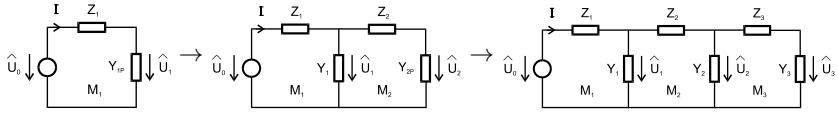
We replace the circuit successively from right to left by parallel resistors, such that at the end only one loop with one parallel resistor remains.

First, we replace the last mesh with Y_{2p}

$$Y_{2p} = \frac{Y_2(Y_3 + Z_3)}{Y_2 + Y_3 + Z_3}$$

Next, we once again replace the last mesh with Y_{1p}

$$Y_{1p} = \frac{Y_1(Y_{2p} + Z_2)}{Y_2 + Y_{2p} + Z_2}$$



Subsequently, we apply the voltage divider rule from left to right as it is indicated in the sequence of circuits below:

$$\frac{\widehat{U}_1}{\widehat{U}_0} = \frac{Y_{1p}}{Y_{1p}+Z_1} \quad \rightarrow \quad \frac{\widehat{U}_2}{\widehat{U}_1} = \frac{Y_{2p}}{Y_{2p}+Z_2} \quad \rightarrow \quad \frac{\widehat{U}_3}{\widehat{U}_2} = \frac{Y_3}{Y_3+Z_3}$$

In summary, we obtain the following algorithm. In order to apply this procedure we have to specify the longitudinal impedances as well as the cross-impedances according to the given circuit.

Algorithm:

Z_1, Z_2, Z_3 Longitudinal impedances
Y_1, Y_2, Y_3 Cross-impedances

$$Y_{2p} = Y_2 \cdot (Y_3 + Z_3)/(Y_2 + Y_3 + Z_3)$$
$$Y_{1p} = Y_1 \cdot (Y_{2p} + Z_2)/(Y_1 + Y_{2p} + Z_2)$$

$$U_1 = U_0 \cdot Y_{1p}/(Y_{1p} + Z_1)$$
$$U_2 = U_1 \cdot Y_{2p}/(Y_{2p} + Z_2)$$
$$U_3 = U_2 \cdot Y_3/(Y_3 + Z_3)$$

1.1.4 Simulations

We simulate the complex transfer function for (1) a high-pass, (2) a low-pass, (3) a band-pass, and (4) a band-stop with Maple. The associated Maple worksheets can be directly started by clicking on the links in the table.

(1) HP2TCLC: High-pass consisting of two T links.
(2) LP2PiCLC: Low-pass consisting of two Π links.
(3) BP2PiLCp: Band-pass consisting of two Π links.
(4) BS2TLCp: Band-stop consisting of two T links.

1.1.5 Algorithmic Implementation: The Procedure Chain

We consider a general circuit consisting of n links

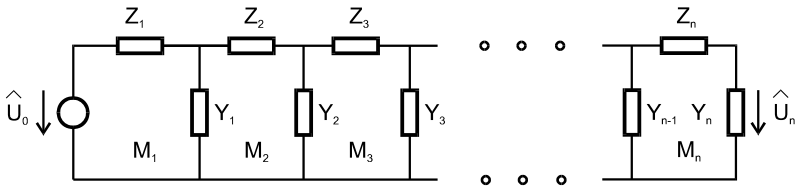


Figure 1.4. General circuit of length n

In generalization of the discussed cases (1) - (3), the algorithm to calculate the transfer function is:

Algorithm:

$Z_1, Z_2, \ldots, Z_n$: Longitudinal impedances
$Y_1, Y_2, \ldots, Y_n$: Cross-impedances

$$Y_{p,n-1} = Y_{n-1} \cdot (Y_n + Z_n)/(Y_{n-1} + Y_n + Z_n)$$
$$Y_{p,i} = Y_i \cdot (Y_{p,i+1} + Z_{i+1})/(Y_i + Y_{p,i+1} + Z_{i+1}) \qquad i = n-2, \ldots, 1$$

U_0 Input voltage

$$U_i = U_{i-1} \cdot Y_{p,i}/(Y_{p,i} + Z_i) \qquad i = 1, \ldots, n-1$$
$$U_n = U_{n-1} \cdot Y_n/(Y_n + Z_n)$$

The Maple procedure **chain** simulates the transfer function for a generalized n-linked chain applying the algorithm above. Ultimately, the procedure chain calculates successively the parallel resistors $(i = n-2, \ldots, 1)$ starting from the outermost mesh to the innermost mesh $(i = 1)$. The second for-loop starts at the innermost mesh and successively applies the voltage divider rule for $(i = 1, \ldots, n-1)$.

When using the procedure, it is important to note that the longitudinal impedances must now be labeled with $Z[1]$, $Z[2]$, $Z[3]$, etc. Similarly, $Y[1]$, $Y[2]$, $Y[3]$, ... apply to the cross-impedances. The global variable determined with the procedure **chain** is the transfer function with the name **H**.

```
> chain := proc(U0,Z,Y,n)
>     # Calculates the transfer function of a chain
>     #  U0: applied voltage
>     #  Z[i]: longitudinal impedances
>     #  Y[i]: cross-impedances
>     #  n: number of links
> local i,Yp,U;
> global H;
>
```

```
>     # Replace the meshes with parallel resistors
>     # starting with mesh n to mesh 1
> Yp[n] := Y[n];
> for i from n-1 by -1 to 1
> do Yp[i] := Y[i]* (Yp[i+1] + Z[i+1]) / (Y[i] + Yp[i+1] + Z[i+1]) od:
>
>     # Reverse resolution of the voltages
> U[0] := U0:
> for i from 1 to n
> do U[i] := U[i-1] * Yp[i] / (Yp[i] + Z[i]) od:
> H:=simplify((U[n]/U0));
> end:
```

With this formalism, the individual elements of the circuit are **not** specified so far. So the algorithm is the same for all Π as well as for all T elements. For impedance we have to specify the resistivity according to the following table

R_Ω	(Ohm's Resistor)
$i\omega L$	(Impedance of a coil with inductance L)
$\frac{1}{i\omega C}$	(Impedance of a capacitor with capacitance C)

The call of the procedure is done with **chain**(U0, Z, Y, n), where U0 is the amplitude of the input voltage, Z[i] the longitudinal and Y[i] the cross-impedances. n specifies the number of links. The result of the procedure is the complex transfer function H. Its magnitude and phase angle is subsequently displayed for visualization purposes.

Example 1.1: Simulation of the high-pass filter HP2TCLC.

We consider a high-pass filter consisting of two T links. The middle capacitor is specified with $\frac{1}{2}C$, because at that place two capacitors are in series.

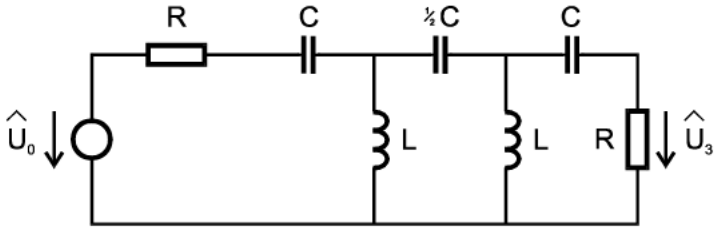


Figure 1.5. High-pass with two T-elements

To determine the transfer function we define the longitudinal impedances

```
> Z[1] := R+1/(I*w*C): Z[2] := 1/(I*w*C/2): Z[3] := 1/(I*w*C):
```

and the cross-impedances

```
> Y[1] := I*w*L: Y[2] := I*w*L: Y[3] := R:
```

As input voltage we select

> U0 := 1:

Since the high-pass consists of 3 links, n=3 and the call of the procedure **chain** is

> chain(U0, Z, Y, 3);

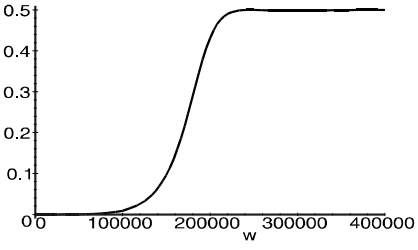
$$\frac{1}{2}\,w^5\,L^2\,C^3\,R\,I\,/(\,w^5\,L^2\,C^3\,R\,I+2\,w^4\,L^2\,C^2-4\,I\,w^3\,L\,C^2\,R \\ -3\,w^2\,L\,C+R^2\,w^4\,C^3\,L-R^2\,w^2\,C^2+2\,I\,R\,w\,C+1)$$

The transfer function is a complex rational function in ω with the highest occurring exponent 5: The circuits contains 5 independent energy storage. This transfer function is assigned the global name H and can then be used for graphical display. Since H is a complex-valued function in w, we must either plot the magnitude $abs(H)$ or the phase $argument(H)$.

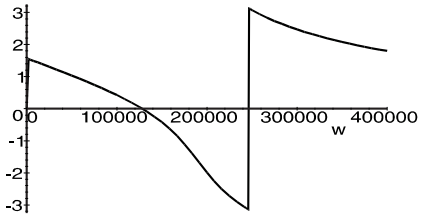
> R:=1000: C:=5.28e-9: L:=3.128e-3:

> plot(abs(H),w=0..400000,thickness=2);

> plot(argument(H),w=0..400000,thickness=2);



Amplitude ratio



Phase diagram

Discussion: It is obvious from the left picture that low frequencies are blocked ($H \approx 0$) but high frequencies can pass ($H \approx \frac{1}{2}$). The cut-off frequency at half maximum amplitude is $\omega_g = 175000\ \frac{1}{s}$. We observe that the maximum value in the transfer behavior is $\frac{1}{2}$. This is because we have an internal resistance R equal to the load resistance. Thus we only get half of the input voltage at the load resistor.

The amplitude ratio can be used to determine exactly the amplitude of the output signal, if the input signal is for example $u_e(t) = 10\sin(300000t)$: Because at frequency $\omega = 300000$ one reads the value $\frac{1}{2}$ in amplitude ratio. I.e. half of the input signal is still available at the output, namely $u_a = 150000$. While with an input signal of $u_e(t) = 10\sin(50000t)$ we receive almost no output signal. From the phase diagram, we read the phase shift between the input and output voltage. □

1.2 Dimensions of High- and Low-Pass Filters

So far we have discussed how to simulate the transfer function of a given circuit and thereby we have identified the cut-off frequency ω_g. Now we move to the more demanding way: We specify the cut-off frequency ω_g and the resistance R. Then, we look for the matching values for L and C.

If we make the following assumptions, we get very good results using Maple simulations.

Prerequisites:

(1) The structure of the circuit is either made up of identical $\Pi-$ or $T-$links.

(2) All Y- and Z- impedances consist of idealized values (only L and C; no R value). This is not feasible, but it can initially be assumed. After adjusted operating parameters are found, the influence of coil resistances can be considered.

(3) The internal resistance R_i is equal to the load resistance R_A (Power adaptation of the circuit or voltage divider rule).

Basic formulas

(1) **Scaling the cut-off frequency:** For the sake of simplicity, we define the cut-off frequency ω_g for a high-pass or low-pass filter as the frequency at which $|H(\omega)|$ decreases to $\frac{1}{\sqrt{2}}$ of the maximum value. It holds

$$\boxed{\omega_g = \frac{k_f}{\sqrt{LC}}} \tag{1.1}$$

if k_f represents a circuit-specific constant.

(2) **Scaling of the resistor:** The relationship between resistance and L, C is

$$\boxed{R = k_R \cdot \sqrt{\frac{L}{C}}} \tag{1.2}$$

if k_R represents a circuit-specific constant.

To find the circuit-specific constants k_f and k_R we choose $L = C = 1$ for a first simulation.

(1.) Within this simulation we vary R such that the transfer function has an appropriate shape. This appropriate R gives the scaling factor k_R.

(2.) From the transfer function we read the scaling factor k_f, which is in case of $L = C = 1$ exactly the **cut-off frequency**.

Appropriate Scaling:

(1) **Simulation:** Set $L = C = 1$ and vary $R_1 = R_A$ until the frequency behavior is as smooth as required $\Rightarrow R_{an}$ (adjusted resistance). If the cut-off frequency ω_g and resistance R is given we are searching for the corresponding value of L and C. The transfer function will now be scaled such that the transmission behavior remains the same.

(2) **Scaling of L and C at given ω_g and R:**

$$(1.1) \cdot (1.2) \rightsquigarrow \omega_g \cdot R = k_R \cdot k_f \cdot \frac{1}{C} \Rightarrow \quad C = \frac{k_f \cdot k_R}{\omega_g \cdot R} \tag{1.3}$$

$$\frac{(1.2)}{(1.1)} \rightsquigarrow \frac{R}{\omega_g} = \frac{k_R}{k_f} \cdot L \Rightarrow \quad L = \frac{k_f \cdot R}{\omega_g \cdot k_R} \tag{1.4}$$

(3) After we have calculated the values for L and C, we again compute the transfer function. Now, the qualitatively behavior is equivalent to the parameters $L = C = 1$, but with cut-off frequency ω_g.

Example 1.2: HP1PiLCL.

We are looking for a high-pass filter consisting of a Π-link, which has a resistance of $R = 500\Omega$ and a cut-off frequency of $\omega_g = 1000s^{-1}$:

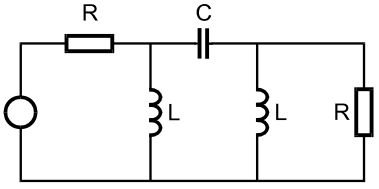


Figure 1.6. High-pass with a Π-link

With

$$Z_1 = R, \quad Z_2 = \frac{1}{i\omega C}, \qquad Y_1 = i\omega L, \quad Y_2 = \frac{1}{\frac{1}{R} + \frac{1}{i\omega L}}$$

it follows

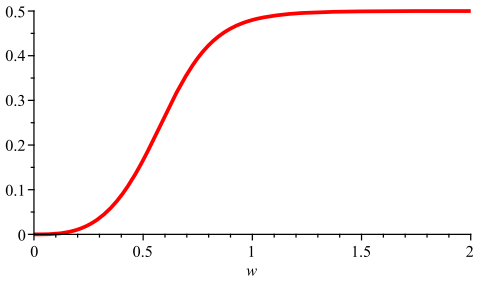
$$Y_{1p} = \frac{Y_1(Y_2 + Z_2)}{Y_1 + Y_2 + Z_2}$$

and

$$u_1 = \frac{Y_{1p}}{Y_{1p} + Z_1}$$

$$u_2 = u_1 \cdot \frac{Y_2}{Y_2 + Z_2}$$

(1) To find the circuit specific constants k_f and k_R we set $L = C = 1$. By varying R in the range $R = 0.5 \ldots 1.5$ we get for $R = 0.8$ a smooth slope of the transfer function.



From the diagram of the transfer function we read the cut-off frequency $\omega_g = 0.57$.

$$\Rightarrow k_R = 0.8 \qquad \text{and} \qquad k_f = 0.57$$

(2) Hence, we get the scaling for L and C:

$$(I) : C = \frac{0.57 \cdot 0.8}{1000 \cdot 500} = 0.912 \cdot 10^{-6} F$$
$$(II) : L = \frac{0.57 \cdot 500}{0.8 \cdot 1000} = 0.355 H$$

Example 1.3: Simulation of the low-pass LP2PiCLC.

Given is the low-pass filter composed of two Π links. We are looking for L and C such that with an internal resistance of $1 M\Omega$ the circuit has a cut-off frequency of $\omega_g = 20000\, s^{-1}$.

We first determine the transfer function setting $L = 1$ and $C = 1$ and vary R until we get an appropriate shape of the transfer function. Here we can identify κ_r to be equal to the optimal resistance and the cut-off frequency κ_f from the transmission diagram can be read. These are the two circuit-specific constants, which are then included in the calculation of L and C.

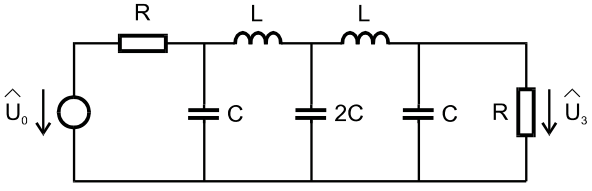


Figure 1.7. Low-pass filter

To calculate the transfer function of the low-pass, we use the procedure **chain**. For this we define the longitudinal impedances

```
> Z[1] := R: Z[2] := I*w*L: Z[3] := I*w*L:
```

and the cross-impedances

```
> Y[1] := 1/(I*w*C): Y[2] := 1/(I*w*2*C): Y[3]:= 1/(I*w*C +1/R ):
```

As input voltage we select

```
> U0 := 1:
```

As the low-pass consists of three meshes, we select $n = 3$ and the call reads

```
> chain(U0, Z, Y, 3);
```

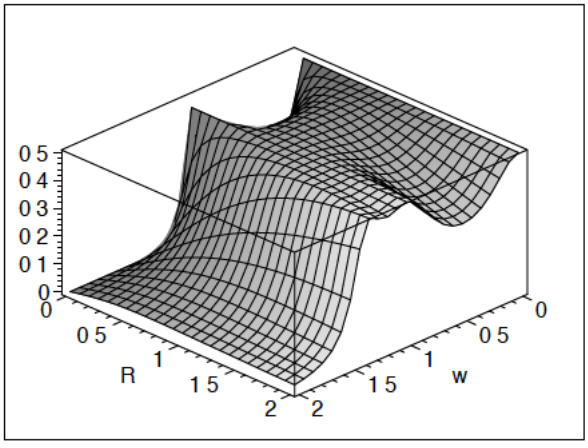
$$\frac{-1}{2}\, I\, R/(-I\, R + 4\, I\, w^2\, L\, R\, C + w\, L - 2\, I\, R\, w^4\, L^2\, C^2 - w^3\, L^2\, C + 2\, R^2\, w\, C - 3\, w^3\, L\, L\, C^2\, R^2 + w^5\, L^2\, C^3\, R^2)$$

The transfer function is a complex rational function in ω with the highest occurring exponent 5: The chain contains 5 independent energy devices. For dimensioning L and C, we first set

```
> C:=1: L:=1:
```

and draw the transfer function 3-dimensional

```
> plot3d(abs(H), w=0..2, R=0..2,axes=boxed);
```



The transfer function has a smooth transition at $R \approx 0.9$. We therefore draw $H(w)$ for $0.5 \leq R \leq 1.5$ in more detail, by using the Explore-command to indicate single values:

```
> Explore(plot(abs(H),w=0..2),
>         parameters=[R=0.5 .. 1.5], initialvalues=[R=1.]);
```

At the optimal resistance.

> $k_r := 0.8$:

the cut-off frequency is taken from the diagram.

> $k_f := 1.4$:

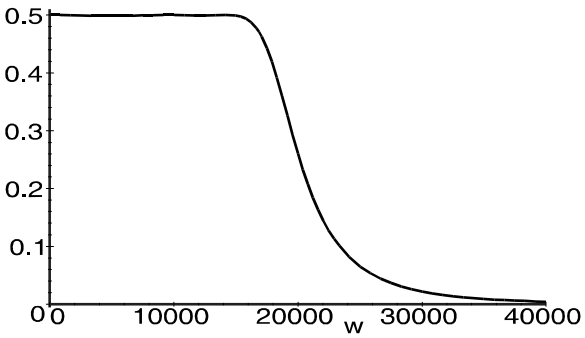
With these values of the constants k_r and k_f we calculate L and C for given quantities $R = 500$ and $\omega_g = 20000$:

> $R := 1000$: $w_g := 20000$:
> $C := k_f * k_r/(w_g * R)$;
> $L := k_f/k_r * R/w_g$;

$$C- = .56000000000\,10^{-7} \quad L- = .08750000000$$

The transfer function for these adapted quantities is

> plot(abs(H), w=0..40000, thickness=2);



We recognize very well that now the cut-off frequency lies approximately at $\omega_g = 20000$ and the qualitative behavior of the transmission remains. □

Example 1.4 : Maple simulations.

In the following two worksheets a high-pass and a low-pass filter are scaled. The associated Maple worksheets can be started by clicking on the hyperlink.

(1.) Low-pass: Given is a low-pass consisting of two Π links. We are looking for L and C, such that at $R = 1000\Omega$ the cut-off frequency is $\omega = 20000\frac{1}{s}$.

(2.) High-pass: Given is a high-pass consisting of a Π link. We are looking for L and C, such that the cut-off frequency is $\omega = 1000\frac{1}{s}$ at $R = 500\Omega$. □

1.3 Locus Curves

An alternative to the graphic representation of the transfer function with respect to magnitude and phase is the **locus curve**. In this representation of $H(\omega)$, the the magnitude $|H(\omega)|$ and the phase angle $\varphi(\omega)$ are drawn simultaneously into a diagram with polar coordinates. The magnitude corresponds to the radius and the phase angle to the angle. Thereby $|H(\omega)|$ and $\varphi(\omega)$ vary simultaneously as a function of ω.

Example 1.5 : Simulation of the locus curve.

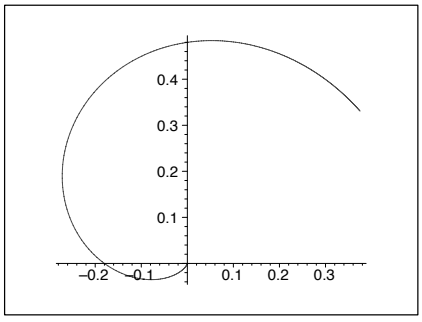
In Maple, the locus curve is represented either by the command
> **plot**([**Re**(H(ω)), **Im**(H(ω)), $\omega = 0 \ldots 2$]);

where $Re(H(\omega))$ and $Im(H(\omega))$ are the real part and imaginary part of the complex function $H(\omega)$, respectively. Or
> with(plots):
> **polarplot**([**abs**(H(ω)), **argument**(H(ω)), $\omega = 0..2$]);

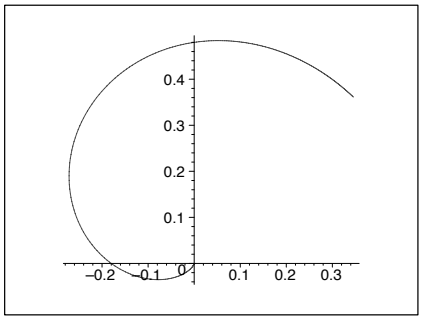
where $abs(H(\omega))$ means the magnitude of the transfer function and $argument(H(\omega))$ the phase of $H(\omega)$. □

Example 1.6 : Visualization of the locus curve of HP1TCLC.

In case of Example 1.2, we visualize the locus curve for the parameters $L = C = 1$. Then we vary the frequency ω from $0 \ldots 2$ (see Figure (a)). Subsequently, we take the same high-pass with one Π link but with parameters $L = 0.4H, C = 0.9 \cdot 10^{-6}F$. Here, we vary ω from $0 \ldots 3000$ (see Figure (b)).



(a) $L = C = 1$. ω varies from $0 \ldots 2$.



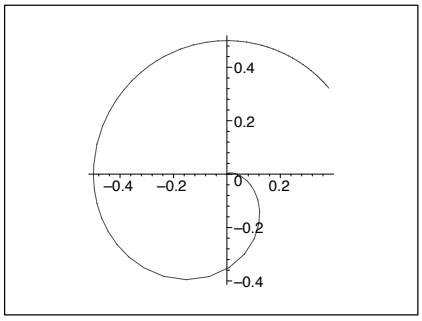
(b) $L = 0.4H, C = 0.9 \cdot 10^{-6}F$. ω varies from $0 \ldots 3000$.

Figure 1.8. Locus Curves

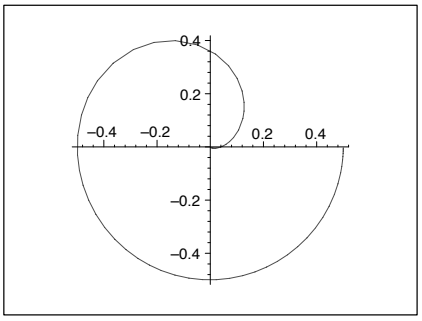
The two locus curves for the corresponding parameter ranges in ω are identical. □

Example 1.7 : Locus curves.

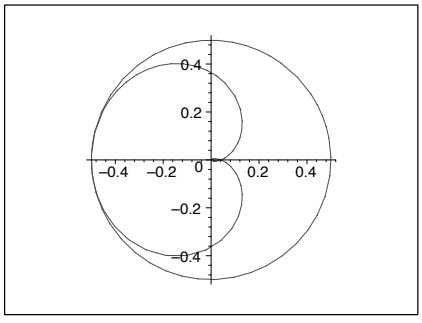
For the examples given in 1.1.4 of the (1) high-pass, (2) low-pass, (3) band-pass and (4) band-stop, the following results applies for the locus curves:



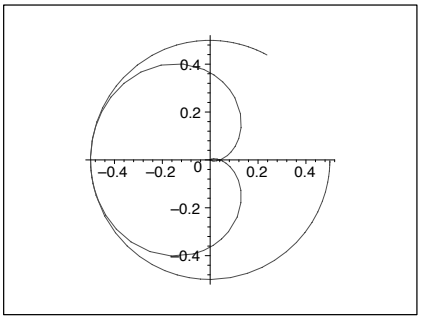
(a) High-Pass HP2TLCL



(b) Low-Pass LP2PiCLC



(c) Band-pass BP2PiLCr



(d) Band-stop BS2TLCp

Figure 1.9. Locus Curves

Watching the phase angle: When varying the frequency from 0 to ∞ the phase angle moves over $N \cdot 90°$. Where N is the number of energy stores. □

1.4 Bode Diagram

As we have already seen, the transfer behavior $H(\omega)$ is graphically represented separately by the magnitude of the transfer function, $|H(\omega)|$, or the phase, $argument(H(\omega))$, as function in ω.

In addition to this representation, the 10th logarithm of the magnitude in logarithmic scaling of the ω axis is established. This is the so-called **Bode Diagram**. Exactly said

$$20\,dB\ log_{10}(|H(\omega)|)$$

with logarithmic scaling of the ω axis is graphically displayed. In Maple, the Bode diagram is realized by the command

```
> with(plots);
> semilogplot(20*log[10](abs(H(ω)), ω = 0.1...ω_max);
```

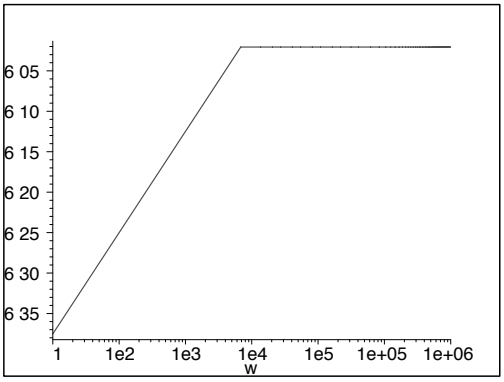


Figure 1.10. Bode Diagram

Note that ω must not start at 0 because a logarithmic scaling of the ω axis is performed and $log(0)$ is not defined!

Summary of the chapter: In this chapter we have described electrical circuits in their frequency behavior. The transfer function $H(\omega)$ was introduced for this purpose as the ratio of output voltage to input voltage when the input voltage is $\widehat{U}_0(t) = e^{i\,w\,t}$.

Hence, the output signal can be calculated for any frequency. As beautiful and as simple this description is, prerequisite is that the input signal is cosine or sinusoidal. As soon as the circuit receives a square wave signal, triangle signal or similar signals, the transfer function cannot be used to make statements about the output signal. For this task we have to formulate the description in a more general way. This leads us to the next chapter: Modelling the circuits or more general linear systems in the time domain.

Chapter 2

Simulating the Time Behavior of Electrical Circuits

2 Simulating the Time Behavior of Electrical Circuits

Overview: In the previous chapter the frequency behavior of RCL filter circuits was investigated. For this purpose we have performed simulations with an AC input voltage $\widehat{U}_0(t) = e^{i\,w\,t}$.

However, if the input signal is a square-wave voltage, a ramp signal or another non sinusoidal voltage signal, that method can **no longer** be directly used to compute the transfer function or to simulate the corresponding output signal. For treating these signals, we reformulate the model equations. This leads to the modelling of the systems by means of differential equations which then describe the time behavior.

Keywords: Electrical circuits, Euler's method, Differential equations, Impulse response, Step response.

In this chapter first order differential equations for electrical circuits consisting of RCL elements are established. Then they are calculated by means of the Euler method: The differential equations are solved numerically. The goal is to simulate the output signal for **any** input signal.

First of all, we provide a general relationship between current and voltage for the considered components. Since we no longer restrict ourselves on AC voltages, we no longer can use ohmic laws as introduced in Chapter 1. Now, we model the circuits by systematically applying mesh and node rules. This description leads to a set of coupled, linear differential equations of the first order.

Having more than three coupled differential equations, it is nearly impossible to analytically solve the problem. Therefore we are dependent on approximation procedures. In order to simulate the model equations (=differential equations) numerically, we introduce Euler's method.

With applying the Euler method to each individual differential equation we solve the model equations for any input signal. Three special input signals are of importance: alternating voltages, pulse signal and step signal, which then returns the sinusoidal response, the impulse response, and the step response.

2.1 Physical Laws of Construction Elements

Resistance: 

$$\boxed{U(t) = R \cdot I(t)}$$

The current through an ohmic resistor R is proportional to the voltage $U(t) = R\,I(t)$, when $U(t)$ is the voltage drop at resistor R.

Coil with inductance L: 

$$\boxed{U(t) = L\frac{dI(t)}{dt}}$$

If a current I flows through a coil L, then the voltage drop at the coil U is proportional to the current change $\frac{dI}{dt}$. The constant of proportionality is called inductance L: $U(t) = L\frac{dI(t)}{dt}$.

Capacitor with capacity C: 

$$\boxed{I(t) = C\frac{dU(t)}{dt}}$$

If the voltage at the capacitor is denoted with $U(t)$, then the capacitor charge is $Q(t) = C \cdot U(t)$, if C is the capacitance of the capacitor. Because of $I = \frac{dQ}{dt}$ the current through the capacitor is $I(t) = C \cdot \frac{dU(t)}{dt}$.

2.2 Setting up the DEq of a Circuit

(1) According to the physical laws, each energy store is assigned a state variable: **Each capacitor C_i will be assigned its voltage U_i, each coil L_i is assigned its current I_i as state variable.** No state variable is assigned to the resistors, since resistors are not energy storage devices but only energy consumers.

(2) The mesh and the node rules are applied to the circuit. The goal is to obtain a 1st order DEq for each state variable. **If coil and capacitor are connected in series, formally another node introduced.**

(3) Usually, the procedure leads to one DEq for each state variable. Other than the derivative of a state variables does not occur in the system of DEq.

⚠ **Note:** Sometimes there are more than one 1st order derivative in a single DEq. Then, first a linear system of equations must be solved for the derivatives in order to obtain the desired structure.

2.3 Solving the DEq with Euler's Method

Given is the differential equation with initial condition

$$y'(t) = f(t, y(t)) \tag{2.1}$$
$$y(0) = y_0 \tag{2.2}$$

We will solve this differential equation with initial condition numerically for times $t_0 = 0 \leq t \leq T$. To do this we divide the interval $[t_0, T]$ into N sub-intervals of the length

$$h = dt = \frac{T - t_0}{N}$$

The sizes h and dt are called **step size** and **time step**, respectively. We get as intermediate times

$$t_j = t_0 + j \cdot dt \qquad j = 0, \ldots, N$$

and we will calculate the solution only at these discrete points in time t_0, t_1, $t_2, \ldots, t_N$: Beginning from the initial value y_0 we determine approximations $y_1, y_2, \ldots, y_N$ for the function values $y(t_1)$, $y(t_2)$, …, $y(t_N)$ of the solution of (2.1). This procedure is called the **discretization** of the problem.

For the initial value (t_0, y_0) we know the exact gradient $\tan\alpha$ of the solution according to equation (2.1).

$$y'(t_0) = \tan\alpha = f(t_0, y_0)$$

For a small increment h the function y is approximated within the interval $[t_0, t_0 + h]$ by its tangent line (see Fig. 2.1a). For the function value $y(t_1)$ we subsequently apply

$$y(t_1) = y(t_0 + h) \approx y(t_0) + y'(t_0) \cdot h$$

We set

$$\boxed{y_1 := y_0 + f(t_0, y_0)\, h}$$

Thus, the function value $y(t_1)$ has been approximated at point in the time t_1 by y_1. Continuing from this erroneous value y_1, the next value is calculated with equation (2.1). We use y_1 and y_1' for the calculation of an approximate value for the subsequent time $t_2 = t_1 + h$:

$$y(t_2) = y(t_1 + h) \approx y(t_1) + y'(t_1)\, h \qquad \text{(Linearization)}$$

Here, we define

$$\boxed{y_2 := y_1 + f(t_1, y_1)\, h}$$

y_2 is an approximation for the exact value of the solution $y(t_2)$ at time t_2.

We repeat the described procedure for the next step $P_2 = (t_2, y_2)$ and so on. Generally we can describe the approximation method schematically by:

new value = old value + change of solution

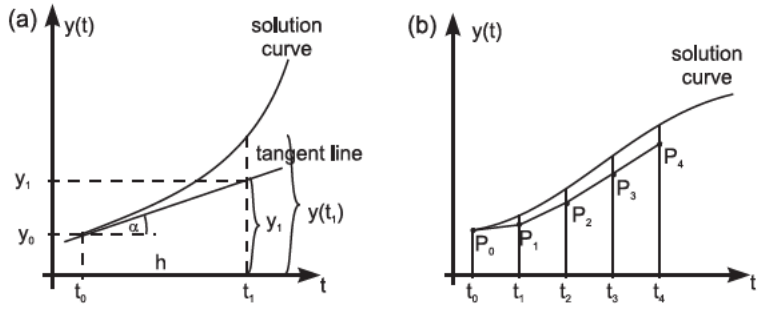


Figure 2.1. Euler's method

or

$$y_{new} = y_{old} + f\,(t_{new},\, y_{old}) \cdot h.$$

Starting from point $P_0 = (t_0,\, y_0)$ we successively evaluate the values

$$y_{i+1} = y_i + h\, f\,(t_i,\, y_i) \qquad i = 0,\, 1,\, 2, \ldots,\, N-1.$$

The solution curve is composed of straight lines, such that the approximation is a line section (see Fig. 2.1b). This method is called according to its geometric meaning, the **Polygon course procedure**, or according to its inventor also **Euler's method**. This simple procedure provides for sufficiently small step h sufficiently good approximations $y_1,\ y_2,\ \ldots,\ y_N$ for the searched values $y\,(t_1),\ y\,(t_2),\ \ldots,\ y\,(t_N)$.

The Euler procedure starts with the initial condition $y(0) = y_0$ and iterates using the following algorithm:

```
y = y0
for t from tmin by dt to tmax do
y = y + f(t,y) * dt
end do
```

To be able to draw the numerical solution, we store the point in time t as well as the function value y in a list called $data$. The algorithm then reads as

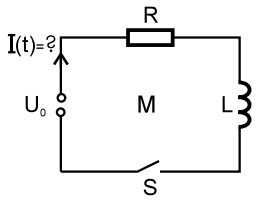
```
y=y0
data[0] = [tmin, y]
i=1

for t from tmin by dt to tmax do
y = y + f(t,y) * dt
data[i] = [t+dt, y]
i = i + 1
end do
```

2.4 Examples with L, C, and R

Example 2.1: Series circuit with coil L and resistor R.

The circuit in this example consists only of an energy storage: an inductance L. We assign this inductance its state variable: the current through the coil $I(t)$. By applying the mesh rule we get a 1st order differential equation for the current $I(t)$. Since we close the switch S at time $t = 0$ the additional initial condition is $I(t = 0) = 0$.



Energy storage: L
State variable: I

Mesh Set: $U_0 = U_R + U_L = R \cdot I + L\frac{dI}{dt}$

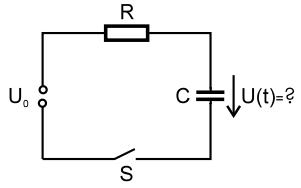
$\Rightarrow \boxed{\frac{dI}{dt} = (U_0 - RI)/L}$ $\rightarrow$ DEq for I

Algorithm:

$U_0 = A \cdot sin(2\pi ft)$
$i = 1$
$I = 0$
for t from t_{min} by dt to t_{max}
do
    $I = I + (U_0 - RI)/L \cdot dt$
    $data[i] = [t + dt, I]$
    $i = i$+1
end do

Example 2.2: Series circuit of capacitance C and resistance R.

This series circuit also consists of only one energy storage device: the capacity C. We assign this capacity its state variable: the voltage drop across the capacitor $U(t)$. By applying the mesh rule, we obtain an 1st order differential equation for the voltage $U(t)$. Since we close the switch S at the time $t = 0$, the additional initial condition is that the voltage at time $t = 0$ equals zero.



Energy storage: C
State variable: U

Mesh rule: $U_0 = U_R + U_L = R \cdot I + L\frac{dI}{dt}$

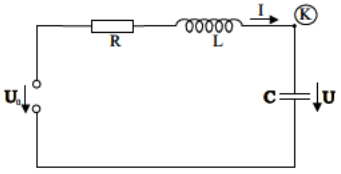
$\Rightarrow \boxed{\frac{dI}{dt} = (U_0 - RI)/L}$ $\rightarrow$ DEq for I

Algorithm:

$U_0 = A \cdot sin(2\pi ft)$
$i = 1$
$I = 0$
for t from t_{min} by dt to t_{max}
do
    $I = I + (U_0 - RI)/L \cdot dt$
    $data[i] = [t + dt, I]$
    $i = i + 1$
end do

Example 2.3: Series circuit of R, C and L.

This circuit consists of **two** energy storage devices: an inductance L and a capacitance C. The inductance is assigned its state variable current $I(t)$ and the capacity the voltage $U(t)$. By applying the mesh rule we only get *one* 1st order differential equation for the current $I(t)$. According to the statement described in 2.2.1: **If coil and capacitor are in series, another node is formally introduced.** We introduce an additional node between coil and capacitance. We thus receive two first order differential equations for $I(t)$ and $U(t)$.



Two energy stores: L, C
Two state variables: I, U

Mesh rule: $U_0 = U_R + U_L + U = R \cdot I + L\frac{dI}{dt} + U$

Node rule K: $I = C \cdot \frac{dU}{dt}$

$$\Rightarrow \quad \boxed{\begin{array}{ll} \frac{dI}{dt} = (U_0 - U - RI)/L & \rightarrow \text{DEq for } I \\ \frac{dU}{dt} = I/C & \rightarrow \text{DEq for } U \end{array}}$$

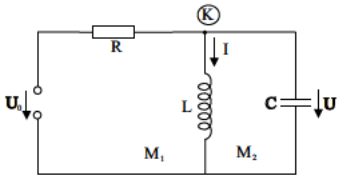
Since we close the switch S at the time $t = 0$, the initial conditions are that current and voltage in point in time $t = 0$ equal zero.

Algorithm:

$U_0 = A \cdot sin(2\pi ft)$
$U = 0$ and $I = 0$
...
for t from t_{min} by dt to t_{max}
do
    $I = I + (U_0 - U - RI)/L \cdot dt$
    $U = U + I/C \cdot dt$
    ...
end do

Example 2.4 : Parallel resonant circuit.

This parallel circuit consists of **two** energy stores: an inductance L and a capacitor C. We assign the inductance its state variable $I(t)$ and the capacitor $U(t)$. By applying the mesh rule and the node rule we obtain two first order differential equations for the current $I(t)$ and the voltage $U(t)$.



Two energy stores: L, C
Two state variables: I, U

Mesh rule for M_1: $U_0 = U_R + U = R \cdot I_0 + U$

Node rule for K: $I_0 = I + C\frac{dU}{dt}$

Mesh rule for M_2: $L\frac{dI}{dt} = U$

⚠ **Note:** Note that a value I_0 occurs in the equations not assigned to any energy storage device. Therefore, we have to eliminate it from the equations. At the end, we get two DEq for the two energy storage devices:

$$U_0 = R(I + C\tfrac{dU}{dt}) + U \quad \text{and} \quad L\tfrac{dI}{dt} = U$$

or

$$\frac{dU}{dt} = ((U_0 - U)/R - I)/C$$
$$\frac{dI}{dt} = U/L$$

Algorithm:

$U_0 = A \cdot sin(2\pi f t)$
$I = 0$ and $U = 0$

for t from t_{min} by dt to t_{max}
do
$U = U + ((U_0 - U)/R - I)/C \cdot dt$
$I = I + U/L \cdot dt$
...
end do

Typical application of this circuit: Regarding the voltage U, the circuit acts as a band-pass: High frequencies can not passed because of C, low frequencies do **not** pass due to L. Only a frequency band around the medium frequency $\omega = \frac{1}{\sqrt{LC}}$ gets through.

2.5 Modelling and Simulation of Filter Circuits

2.5.1 Setting up the DEq for the low-pass filter (LP2PiCLC)

Given is the low-pass filter, which consists of two Π-links LP2PiCLC:

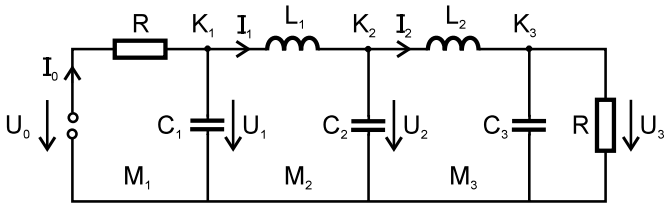


Figure 2.2. Low-pass filter

The low-pass consists of 5 energy stores C_1, C_2, C_3, L_1, L_2; these energy stores are assigned 5 state variables U_1, U_2, U_3, I_1, I_2. Systematically applying the mesh rule and node rule yields the following equations when we consider the physical laws $U_\Omega = R \cdot I$ (Ohm's Law), $I_C = C \cdot \dot{U}$ (current at capacitor), $U_L = L \cdot \dot{I}$ (voltage drop at the coil):

M1: $U_0 = RI_0 + U_1$
K1: $I_0 = C_1\dot{U_1} + I_1$
M2: $U_1 = L_1\dot{I_1} + U_2$
K2: $I_1 = C_2\dot{U_2} + I_2$
M3: $U_2 = L_2\dot{I_2} + U_3$
K3: $I_2 = C_3\dot{U_3} + U_3/R$

⚠ **Note:** I_0 must be replaced as it is **not** assigned to any energy store!

Initially, these are 6 equations, where equation M_1 does **not** represent a differential equation. However, in the system to be solved only the 5 state variables $U_1, U_2, U_3; I_1, I_2$ may occur as variables. Therefore, we eliminate variable I_0 from the equations M_1 and K_1, since it is not assigned to any energy store. We insert equation K_1 into equation M_1. Thus, we get five 1st order differential equations:

$$\begin{aligned}
\dot{U_1} &= ((U_0 - U_1)/R - I_1)/C_1 \\
\dot{I_1} &= (U_1 - U_2)/L_1 \\
\dot{U_2} &= (I_1 - I_2)/C_2 \\
\dot{I_2} &= (U_2 - U_3)/L_2 \\
\dot{U_3} &= (I_2 - U_3/R)/C_3
\end{aligned}$$

Note that only one derivative per differential equation is present and the Euler method provides us with the following algorithm for solving the system:

Algorithm:

$U_0 = \ldots$ given (see below)
$U_1 = 0$, $I_1 = 0$, ... (Initialize all unknown variables)

for t from t_{min} by dt to t_{max}
do
$U_1 = U_1 + ((U_0 - U_1)/R - I_1)/C_1 \cdot dt$
$I_1 = I_1 + (U_1 - U_2)/L_1 \cdot dt$
$U_2 = U_2 + (I_1 - I_2)/C_2 \cdot dt$
$I_2 = I_2 + (U_2 - U_3)/L_2 \cdot dt$
$U_3 = U_3 + (I_2 - U_3/R)/C_3 \cdot dt$
...
end do

Comments:

(1) Note that the DEqs should be solved in order of their occurrence in the physical modelling, from the input voltage towards the output voltage.

(2) By not naming the variables U_1^{new} and U_1^{old}, a renaming of these variables in the algorithm is avoided and the actual (i.e. recalculated) data are always taken into account.

We select $L_1 = L_2 = 1$; $C_1 = C_3 = 1$; $C_2 = 2$; $R = 0.8$ ($k_f = 1.314$) (see chapter 1 on the transfer function). For the numerical simulation we select $\Delta t = 0.005, t_{max} = 90$.

(1.) For the input voltage we choose

$$U_0(t) = \sin(\omega t) \text{ with } \omega = 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0$$

and obtain for the amplitude of the output voltage

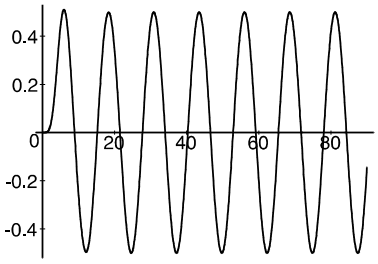


Figure 2.3. Output signal for the input signal $u_0 = 1 \cdot sin(\omega t)$ with $\omega = 1$

ω	0.5	0.75	1.0	1.25	1.5	1.75	2.0
U_A	0.5	0.5	0.5	0.42	0.17	0.065	0.029

We can see that the output amplitude drops drastically at frequencies between 1.25 and 1.5.

(2.) To simulate a turn-on process, we set the input voltage to $U_0(t) = S(t)$. Where $S(t)$ is the step function. For negative times the function has the value 0 and jumps to the value 1 at $t = 0$. In Maple this function is called **Heaviside(t)**. We have to specify the value at the position $t = 0$ explicitly. The response signal belonging to the step signal is shown below. Since this is the reaction of the system to the step function, the response signal is subsequently called the *step response*.

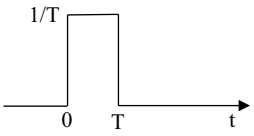


Figure 2.4. Impulse signal

(3.) Next, we select as input signal (input voltage) a pulse with width T and height $1/T$. Thus, we excite the system impulsively. We define the impulse in Maple with $\frac{1}{T}(S(t)-S(t-T))$ or with Maple syntax

```
> U0(t) := 1/T*(Heaviside(t)-Heaviside(t-T));
> Heaviside(0):=0;
```

The associated output voltage is called the *impulse response*.

2.5.2 Simulations with Maple

We set-up the differential equations for (1) a high-pass, (2) a low-pass, (3) a band-pass, and (4) a band-stop filter on and solve them with Maple.

HP2TLCL: High-pass filter consisting of two T links.
LP2PiCLC: Low-pass filter consisting of two Π links.
BP2PiLCp: Band-pass filter consisting of two Π links.
BS2TLCp: Band-stop filter consisting of two T links.

2.6 Simulation of the Low-Pass LP2PiCLC

We simulate the low-pass filter LP2PiCLC from section 2.5.1. Therefore, we describe the Maple commands in detail, like they have to be executed.

1. Setting up the differential equations: We transfer the differential equations from the previous section directly to Maple. For $\frac{dU_1}{dt}$, $\frac{dU_2}{dt}$, $\frac{dU_3}{dt}$, $\frac{dI_1}{dt}$ and $\frac{dI_2}{dt}$ we choose in Maple shorthand dU1, dU2, dU3, dI1, dI2:

```
> eq1:= R*(C1*dU1 + I1) = Ue-U1;
> eq2:= U1 = L1*dI1 + U2;
> eq3:= I1 = C2*dU2 +I2;
> eq4:= U2 = L2* dI2 + U3;
> eq5:= I2 = C3* dU3 +U3/R;
```

$$eq1 := R\,(C1\,dU1 + I1) = Ue - U1$$

$$eq2 := U1 = L1\,dI1 + U2$$

$$eq3 := I1 = C2\,dU2 + I2$$

$$eq4 := U2 = L2\,dI2 + U3$$

$$eq5 := I2 = C3\,dU3 + \frac{U3}{R}$$

2. To ensure that we can apply Euler's method to each of the differential equations, we isolate the derivative of the state variables we are looking for with the **solve** command. In order to be able to use the result afterwards the **assign** command is used to assign the formulas to the variables dU1, dU2, dU3, dI1, dI2.

```
> sol:=solve({eq1,eq2,eq3,eq4,eq5}, {dU1,dU2,dU3,dI1,dI2});
> assign(sol);
```

$$sol := \{dU2 = -\frac{-I1 + I2}{C2}, dI1 = -\frac{-U1 + U2}{L1}, dI2 = -\frac{-U2 + U3}{L2}, dU3 = -\frac{-I2\,R + U3}{C3\,R}, dU1 = -\frac{R\,I1 - Ue + U1}{R\,C1}\}$$

Setting the parameters and all initial conditions

```
> R:=0.8: C1:=1: C2:=C1*2: C3:=C1: L1:=1:L2:=L1:
> U1:=0: U2:=0: U3:=0: I1:=0: I2:=0:
```

Before we solve the differential equations numerically with Maple, the input voltage must be defined. The worksheet treats three different input voltages:

AC voltage in the passband: All we have to do is to define

```
> Ue:=sin(w*t): w:=1.:
```

and run different simulations with modified w.

Step function: The step function is activated in Maple with **Heaviside**(t). However, this function is not explicitly defined at $t = 0$. I.e. we have to define it:

```
> Heaviside(0):= 0:
> Ue:=Heaviside(t):
```

Pulse function: The pulse function with pulse width T is calculated as the difference between two step functions:

```
> T:=0.5: Ue:=Heaviside(t)-Heaviside(t-T):
```

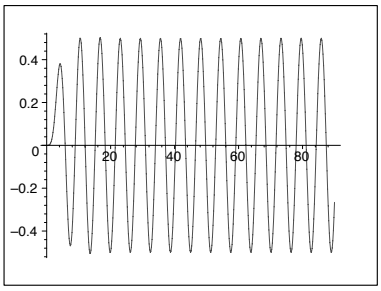
3. Solving each DEq with Euler's method

```
> tmax:=90:
> dt:=tmax/500:
> i:=0:
> for t from 0 by dt to tmax
> do
>       i:=i+1:
>       U1:=U1 + dt*dU1:
>       I1 := I1 + dt* dI1:
>       U2:= U2 + dt * dU2:
>       I2:= I2 + dt * dI2:
>       U3:= U3 + dt * dU3:
>       data1[i]:=[t,U3]:
>       data2[i]:=[t,Ue]:
> end do:
```

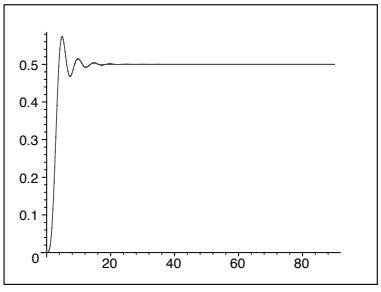
4. Displaying the solution for the different input signals

```
> plot([seq(data1[n],n=1..i)], thickness=2);
```

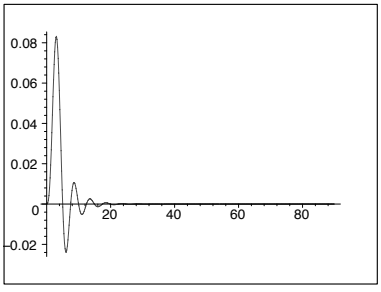
System's response for an AC voltage as input signal at w=1 (in the passband):



System's response for a step function as input signal: Step response.



System's response for a pulse function as input signal: Impulse response.



Summary of the chapter: By modelling the circuits via differential equations, we are able to provide for any input signal the corresponding output signal at least approximately by the Euler method. A deeper relationship specifically to the frequency behavior of the circuits, however, we cannot yet see. In order to deeper understand correlations in the description of systems we still need more theory, which we will explain in the chapter 9 on the description of LTI systems.

Chapter 9

Analysis of Linear Systems

Analysis of Linear Systems

9

Overview: In this chapter we transfer the analysis of signals to the analysis of systems. For linear systems there exist a special function describing the system completely: the **impulse response**. Through the knowledge of the impulse response we are able to predict the output signal for any input signal. In this chapter we also discuss the relationship between the impulse response and the transfer function.

Keywords: Impulse response, transfer function, system function, LTI systems, step response, Fourier transform.

In order to determine the transmission behavior of systems, we investigated the relationship between the input signal $f(t)$ and the associated output signal $g(t)$.

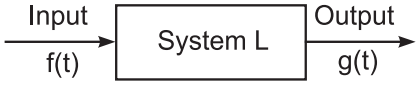


Figure 9.1.

We will characterize *linear* systems by means of the impulse response. It turns out that a linear system is completely described by the *impulse response* (reaction of the system on the impulse excitation): By knowing the impulse response, we are able to compute the reaction of the system to any input signal $f(t)$. In many cases, however, it is easier to determine the Fourier transform of the impulse response ($=$ *system function*). The aim of this chapter is to show this relationship between system function and impulse response and to highlight its importance.

9.1 LTI Systems

An **analogue system** L is a transformation that assigns each input signal $f(t)$ a corresponding output signal $g(t)$. An analog system is therefore a transformation L, which assigns an output function g (*Output*) to each input function f (*Input*):

$$g(t) = L[f(t)]$$

Since we only look at analog systems, we subsequently call L always just by the term system. A system L is called **linear** if the superposition principle is valid:

$$\text{(L)} \qquad L\left[k_1\, f_1(t) + k_2\, f_2(t)\right] = k_1\, L\left[f_1(t)\right] + k_2\, L\left[f_2(t)\right]$$

for any input functions $f_1(t)$, $f_2(t)$ and constants $k_1, k_2 \in \mathbb{R}$. The superposition principle states that the response of a linear system to a superposition of input functions results in the same superposition of response functions. Important special cases of linear systems are systems that can be described by linear differential equations.

A system L is called **time invariant**, if the following condition is satisfied:

$$\text{(TI)} \qquad g(t) = L\left[f(t)\right] \;\Rightarrow\; g(t - t_0) = L\left[f(t - t_0)\right].$$

For example, all networks that are built up from constant components of resistors R, capacitors C, and inductors L are time-invariant systems.

A system is called **causal**, if the reaction of the system $g(t)$ does not start until the cause $f(t)$ is effective:

$$\text{(C)} \qquad f(t) = 0 \text{ for } t < t_0 \;\Rightarrow\; g(t) = L\left[f(t)\right] = 0 \text{ for } t < t_0.$$

In non-causal systems, the reaction may already start when the cause is not yet present ($\rightarrow$ ideal low pass). Note that only causal systems are physical systems.

In the following we limit ourselves to the description of linear, time-invariant, causal systems (LTI systems).

Example 9.1 : Impulse response of a linear system.

As a first example let's have a look on an RC circuit. This circuit can be modeled by the differential equation

$$g'(t) + \alpha\, g(t) = f(t) \qquad \text{with} \qquad g(0) = 0.$$

This system is an LTI system. $f(t)$ is the cause, $g(t)$ is the system's response to $f(t)$. If the input signal is $\delta_\varepsilon(t)$, the corresponding output signal is $h_\varepsilon(t)$. The input signal is a rectangle with width ε and height $1/\varepsilon$ (see Figure 9.2).

In the range $0 \le t \le \varepsilon$ the voltage at the capacitor increases according to

$$\frac{1}{\alpha\,\varepsilon}\left(1 - e^{-\alpha\, t}\right)$$

(*switch on*) and in the $t > \varepsilon$ range the voltage decreases like

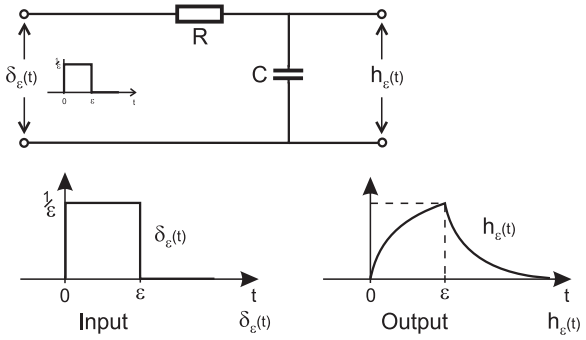


Figure 9.2. Input and output signal of an RC circuit

$$\frac{1}{\alpha\,\varepsilon}\,\left(e^{\alpha\,\varepsilon}-1\right)\,e^{-\alpha\,t}$$

(*switching off*). This time behavior can be confirmed by inserting the corresponding expressions into the differential equation. Hence, the system response to $\delta_\varepsilon\,(t)=\frac{1}{\varepsilon}\,\left(S\,(t)-S\,(t-\varepsilon)\right)$ is given by

$$h_\varepsilon\,(t)=\begin{cases}\dfrac{1}{\alpha\,\varepsilon}\,\left(1-e^{-\alpha\,t}\right) & 0\le t\le\varepsilon\\[2ex] \dfrac{1}{\alpha\,\varepsilon}\,\left(e^{\alpha\,\varepsilon}-1\right)\,e^{-\alpha\,t} & t\ge\varepsilon\end{cases}$$

We now consider the case $\varepsilon\to 0$, i.e. the excitation of the system is done by the δ-pulse $\left(\delta\,(t)=\lim\limits_{\varepsilon\to 0}\,\delta_\varepsilon\,(t)\right)$. Since we are interested in the behavior of the response signal for times $t>0$, we take the values of $h_\varepsilon\,(t)$ for $t\ge\varepsilon$ and determine the limit $\varepsilon\to 0$. For $t>0$ we obtain with the rule of l'Hospital

$$h\,(t)=\lim_{\varepsilon\to 0}\,h_\varepsilon\,(t)=\lim_{\varepsilon\to 0}\,\frac{e^{\alpha\,\varepsilon}-1}{\alpha\,\varepsilon}\,e^{-\alpha\,t}\overset{\frac{0}{0}}{=}\lim_{\varepsilon\to 0}\,\frac{\alpha\,e^{\alpha\,\varepsilon}}{\alpha}\,e^{-\alpha\,t}=e^{-\alpha\,t}$$

$$\Rightarrow\quad\boxed{h\,(t)=e^{-\alpha\,t}\,S\,(t)}$$

$h\,(t)$ is called *impulse response*, because it's the response to the impulse function $\delta\,(t)$ □

9.2 Impulse Response

The procedure that we have chosen in the previous example is generalized for any linear system: Be L be an LTI system and $\delta_\varepsilon\,(t)$ the family of rectangular functions. For each $\delta_\varepsilon\,(t)$ we calculate the response signal $h_\varepsilon\,(t)=L\,[\delta_\varepsilon\,(t)]$. Since $\delta_\varepsilon\,(t)\to\delta\,(t)$ for $\varepsilon\to 0$ and L is a linear system, it follows that

$$h\,(t)=\lim_{\varepsilon\to 0}\,h_\varepsilon\,(t)=\lim_{\varepsilon\to 0}\,L\,[\delta_\varepsilon\,(t)]=L\,\left[\lim_{\varepsilon\to 0}\,\delta_\varepsilon\,(t)\right]=L\,[\delta\,(t)]$$

$h\,(t)$ is the response of the system to the impulse function (= delta function) $\delta\,(t)$ and is therefore called the **impulse response**.

The meaning of the impulse response is highlighted by the following theorem, which states that the system response $g(t)$ to **any** input signal $f(t)$ can be calculated if the impulse response of the system is known:

Convolution Theorem: Let L be a linear, causal, time-invariant system. $h(t)$ is the impulse response and $f(t)$ any input signal. Then the response of the system $g(t) = L[f(t)]$ is given by

$$g(t) = (f * h)(t) = \int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, d\tau$$

The system reaction $g(t) = L[f(t)]$ on any input signal f is calculated by the convolution integral of the impulse response h with the input signal f. This central theorem of system theory is explained below:

Proof of the convolution theorem:
Due to the hide property of the δ-function, $\int_{-\infty}^{\infty} f(\tau)\, \delta(t-\tau)\, d\tau = f(t)$, and the definition of the δ function as the limit of the function family $\delta_\varepsilon(t)$, $\delta(t) = \lim_{\varepsilon\to 0} \delta_\varepsilon(t)$, it holds

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\, \delta(t-\tau)\, d\tau = \lim_{\varepsilon\to 0} \int_{-\infty}^{\infty} f(\tau)\, \delta_\varepsilon(t-\tau)\, d\tau$$

According to the algebraic definition of the integral we replace the integral by a sum over rectangles $\Delta\tau_j \cdot f(\tau_j)\, \delta_\varepsilon(t-\tau_j)$:

$$f(t) = \lim_{\varepsilon\to 0} \lim_{N\to\infty} \sum_{j=0}^{N} f(\tau_j)\, \delta_\varepsilon(t-\tau_j)\, \Delta\tau_j$$

The response of the system to the input signal $f(t)$ is then given by

$$g(t) = L[f(t)] = L\left[\lim_{\varepsilon\to 0} \lim_{N\to\infty} \sum_{j=0}^{N} f(\tau_j)\, \delta_\varepsilon(t-\tau_j)\, \Delta\tau_j\right]$$

$$= \lim_{\varepsilon\to 0} \lim_{N\to\infty} L\left[\sum_{j=0}^{N} f(\tau_j)\, \delta_\varepsilon(t-\tau_j)\, \Delta\tau_j\right]$$

Since L is a linear system, the superposition principle can be applied: The reaction of the system to a sum of input signals is given by the sum of the response functions

$$g(t) = \lim_{\varepsilon\to 0} \lim_{N\to\infty} \sum_{j=0}^{N} f(\tau_j)\, L[\delta_\varepsilon(t-\tau_j)]\, \Delta\tau_j$$

Due to the time invariance, $L[\delta_\varepsilon(t-\tau j)] = h_\varepsilon(t-\tau_j)$, we finally conclude

$$g(t) = \lim_{\varepsilon\to 0} \lim_{N\to\infty} \sum_{j=0}^{N} f(\tau_j)\, h_\varepsilon(t-\tau_j)\, \Delta\tau_j$$

$$= \lim_{N\to\infty} \sum_{j=0}^{N} f(\tau_j)\, h(t-\tau_j)\, \Delta\tau_j = \int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, d\tau \quad \square$$

Example 9.2 : Calculation of the impulse response.

We are looking for the impulse response of the linear system represented by the following differential equation:

$$g'(t) + \alpha\, g(t) = f(t)$$

The impulse response $h(t)$ is the system's response to the input signal $f(t) = \delta(t)$:

$$h'(t) + \alpha\, h(t) = \delta(t)$$

We apply to this differential equation the Fourier transform and use the derivation rule (F_7): $\mathcal{F}(h'(t)) = i\,\omega\,\mathcal{F}(h(t))$:

$$\mathcal{F}(h'(t)) + \alpha\,\mathcal{F}(h(t)) = \mathcal{F}(\delta(t))$$

$$i\,\omega\,\mathcal{F}(h(t)) + \alpha\,\mathcal{F}(h(t)) = 1$$

$$\Rightarrow \mathcal{F}(h(t))(\omega) = \frac{1}{\alpha + i\,\omega}$$

The time function associated with $\frac{1}{\alpha+i\,\omega}$ is according to Example 6.2

$$h(t) = e^{-\alpha\, t}\, S(t)$$

□

By the knowledge of the impulse response we are able to determine the reaction of a LTI system to any input signal by using the convolution theorem. Thus, the important question is how to determine the impulse response. For this basically two different procedures are used:

(1) On the one hand, in some cases it is possible to first determine the Fourier transform of the impulse response when -as in our example- the linear system is described by a differential equation. The impulse response is then calculated by the inverse Fourier transform. For electrical networks, it is shown that the Fourier transform of the impulse response can be determined directly via the complex impedances of the R, C and L components. This leads to the concept of **transfer function** or **system function** which is described in Section 9.3.

(2) If the arrangement of the linear system is not known in detail, i.e. the system is only available as a 'black box' system, then the impulse response can be determined experimentally by exciting the system with the impulse signal. The associated system reaction is then the impulse response. The step function (= *jump function*) is easier to realize experimentally than the pulse function. The impulse response can also be calculated via the *step response* (= reaction of the system to the step function $S(t)$). The relationship between step and impulse response is presented in Section 9.4.

9.3 The System Function (Transfer Function)

The concept of the system function provides a calculus to compute the Fourier transform of the impulse response.

9.3.1 1st Method:

We select as special input signal for a LTI system L the complex exponential

$$x(t) = e^{i\omega t}$$

If $h(t)$ is the impulse response of the system, then the system response $y(t)$ is determined via the convolution integral of $x(t)$ with $h(t)$:

$$y(t) = (h * x)(t) = \int_{-\infty}^{\infty} e^{i\omega(t-\tau)} h(\tau)\, d\tau = e^{i\omega t} \int_{-\infty}^{\infty} e^{-i\omega\tau} h(\tau)\, d\tau$$

The response of the system is again an exponential signal multiplied by the complex amplitude

$$H(\omega) := \int_{-\infty}^{\infty} e^{-i\omega\tau} h(\tau)\, d\tau \tag{1}$$

$H(\omega)$ is called **system function** or **transfer function**. **The system function is the Fourier transform of the impulse response!** The reaction of the LTI system to the input signal $x(t) = e^{i\omega t}$ is a function with time behavior $e^{i\omega t}$ and a complex amplitude $H(\omega)$. So if we select as input signal an harmonic with frequency ω and amplitude 1, then the system reaction again is a harmonic signal with the **same** frequency ω but with different amplitude $H(\omega)$. Since $H(\omega)$ is a complex amplitude, it contains information about both the magnitude of the output $|H(\omega)|$ and the phase shift between input and output signal $\tan\varphi(\omega) = \frac{\operatorname{Im} H(\omega)}{\operatorname{Re} H(\omega)}$.

9.3.2 2nd Method:

Since $y(t) = H(\omega)\, e^{i\omega t}$ is the system reaction to the input signal $x(t) = e^{i\omega t}$,

$$y(t) = L[x(t)]$$

$$\Rightarrow H(\omega)\, e^{i\omega t} = L\left[e^{i\omega t}\right] \hookrightarrow H(\omega) = \frac{L\left[e^{i\omega t}\right]}{e^{i\omega t}}$$

The system function is therefore given by

$$H(\omega) = \left.\frac{y(t)}{x(t)}\right|_{x(t)=e^{i\omega t}} \tag{2}$$

We receive the transfer function $H(\omega)$ to a certain frequency ω, if we select as the input singnal $x(t) = e^{i\omega t}$ and compute the ratio of the corresponding output signal to the input signal in the time domain.

Example 9.3 : System function.

In Example 9.2 the Fourier transform of $h(t)$ was determined by selecting $\delta(t)$ as input. Now we calculate again the system function

$$g'(t) + \alpha\, g(t) = f(t) \qquad (*)$$

by selecting $f(t) = e^{i\,\omega\,t}$ as input signal. Then the associated reaction is $g(t) = H(\omega)\; e^{i\,\omega\,t}$. Now we substitute f and g into the differential equation $(*)$, which provides

$$i\,\omega\, H(\omega)\; e^{i\,\omega\,t} + \alpha\, H(\omega)\; e^{i\,\omega\,t} = e^{i\,\omega\,t}$$

$$\Rightarrow \quad \boxed{H(\omega) = \frac{1}{\alpha + i\,\omega}}$$

This is exactly the same result as in Example 9.2 which we have obtained as the Fourier transform of the impulse response. □

Important: Impulse response and system function are equivalent descriptions of a linear system. Both are related by means of the Fourier transform. Systems with the same impulse response react to the same input signals with identical output signals.

9.3.3 3rd Method:

In addition to equation (1) and (2), we specify a third method to determine the system function. For this we take an arbitrary input signal $f(t)$. Be $g(t)$ the corresponding response signal, then with the help of the convolution theorem we get

$$g(t) = (f * h)(t)$$

We apply to this equation the Fourier transform and use the convolution theorem (F_7):

$$\mathcal{F}(g) = \mathcal{F}(f * h) = \mathcal{F}(f) \cdot \mathcal{F}(h) \qquad (3)$$

Denotes $F(\omega)$ the Fourier transform of the input signal $f(t)$, $G(\omega)$ the Fourier transform of the output signal $g(t)$ and $H(\omega)$ the Fourier transform of the impulse response $h(t)$, so equation (3) is rewritten:

$$G(\omega) = F(\omega) \cdot H(\omega)$$

$$\Rightarrow \quad \boxed{H(\omega) = \frac{G(\omega)}{F(\omega)}} \qquad (4)$$

Equation (4) states that the transfer function $H(\omega)$ can be determined by specifying for an arbitrary signal $f(t)$ the spectrum of the corresponding output signal $G(\omega)$. Subsequently, we have to divide the spectrum of the output signal $G(\omega)$ by the spectrum of the input signal $F(\omega)$.

This third alternative for calculating the transfer function is the most general and contains alternatives (1) and (2) as special cases.

Summary: (System function). There are three equivalent ways for the calculation of the **system function** $H(\omega)$ of a LTI system:

(1) $H(\omega)$ is the Fourier transform of the impulse response:

$$H(\omega) = \int_{-\infty}^{\infty} h(t)\, e^{-i\,\omega\,t}\, dt$$

(2) Choosing the special input function $x(t) = e^{i\,\omega\,t}$ then $H(\omega)$ is the complex amplitude of the response function $y(t) = H(\omega)\, e^{i\,\omega\,t}$:

$$H(\omega) = \left.\frac{y(t)}{x(t)}\right|_{x(t)=e^{i\,\omega\,t}}$$

(3) Is $F(\omega)$ the spectrum of the input and $G(\omega)$ the spectrum of the corresponding output signal, then

$$H(\omega) = \frac{G(\omega)}{F(\omega)}$$

Example 9.4 : Summary example.

Given is a system that is represented by the differential equation

$$y'''(t) + a_1\, y'(t) + a_0\, y(t) = f(t)$$

We determine the system function $H(\omega)$ with all the methods introduced above.

(1) $H(\omega)$ is the Fourier transform of the impulse response $h(t) = L[\delta(t)]$. So we define $f(t) = \delta(t)$, then $y(t) = h(t)$

$$h'''(t) + a_1\, h'(t) + a_0\, h(t) = \delta(t)$$

By applying the Fourier transform to the differential equation, we conclude

$$\mathcal{F}(h''') + a_1\, \mathcal{F}(h') + a_0\, \mathcal{F}(h) = \mathcal{F}(\delta)$$

Because of $H(\omega) = \mathcal{F}(h)$ (the transfer function = Fourier transform of the impulse response) it follows with the derivation rule (F_8)

$$(i\,\omega)^3\, H(\omega) + a_1\, i\,\omega\, H(\omega) + a_0\, H(\omega) = 1 \qquad (*_1)$$

$$\Rightarrow \quad H(\omega) = \frac{1}{(i\,\omega)^3 + a_1\, i\,\omega + a_0} \qquad (*_2)$$

(2) If we set as special input signal $x(t) = e^{i\omega t}$ $(= f(t))$, then the system response is $y(t) = e^{i\omega t} H(\omega)$. Inserting into the differential equation, it follows

$$\left(e^{i\omega t} H(\omega)\right)''' + a_1 \left(e^{i\omega t} H(\omega)\right)' + a_0 \left(e^{i\omega t} H(\omega)\right) = e^{i\omega t}$$

$$\hookrightarrow (i\omega)^3 e^{i\omega t} H(\omega) + a_1 (i\omega) e^{i\omega t} H(\omega) + a_0 e^{i\omega t} H(\omega) = e^{i\omega t}$$

Dividing this equation by $e^{i\omega t}$ returns $(*_1)$ and thus also $(*_2)$ as the transfer function.

(3) If $f(t)$ is the input signal and $g(t)$ is the associated output signal, the relationship between f and g is given by the differential equation

$$g'''(t) + a_1 g'(t) + a_0 g(t) = f(t)$$

Applying the Fourier transform

$$\mathcal{F}\left(g'''\right) + a_1 \mathcal{F}\left(g'\right) + a_0 \mathcal{F}(g) = \mathcal{F}(f)$$

and derivation rule (F_6) together it results in

$$(i\omega)^3 \mathcal{F}(g) + a_1 (i\omega) \mathcal{F}(g) + a_0 \mathcal{F}(g) = \mathcal{F}(f)$$

With $F(\omega) = \mathcal{F}(f)$ and $G(\omega()) = \mathcal{F}(g)$ applies

$$\left((i\omega)^3 + a_1 (i\omega) + a_0\right) G(\omega) = F(\omega)$$

$$\Rightarrow H(\omega) = \frac{G(\omega)}{F(\omega)} = \frac{1}{(i\omega)^3 + a_1 i\omega + a_0}$$

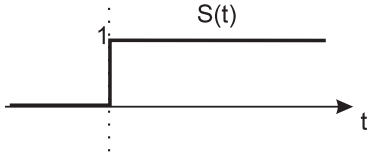
This is the same result for $H(\omega)$ as in (1) and (2). □

In many cases the system function can be determined more easily than the impulse response. If it is known, then the impulse response is given by the *inverse Fourier transform*. Which of the three alternatives for calculating the system function is chosen depending on the problem.

9.4 Relation between Step and Delta Function

We define as the step function (Heaviside function) the function

$$S(t) = \begin{cases} 0 & \text{for } t < 0 \\ 1 & \text{for } t > 1 \end{cases}$$



The **step function** approaches the power-on process for systems that are turned off for all $t < 0$ and on for all $t > 0$. With the help of $S(t)$ a simple determination of the impulse response is possible. Since $S(t)$ is not continuous in point $t = 0$, this function cannot be easily differentiated in $t = 0$. It turns out, however, that the derivative of $S(t)$ can still be formed if the delta function is allowed as a result! For clarification, we consider the sequence of images in Fig. 9.3 (a) - (d).

First, in Fig. (a) the family $S_\varepsilon(t)$ converges for $\varepsilon \to 0$ to the step function $S(t)$ (b):

$$S(t) = \lim_{\varepsilon \to 0} S_\varepsilon(t)$$

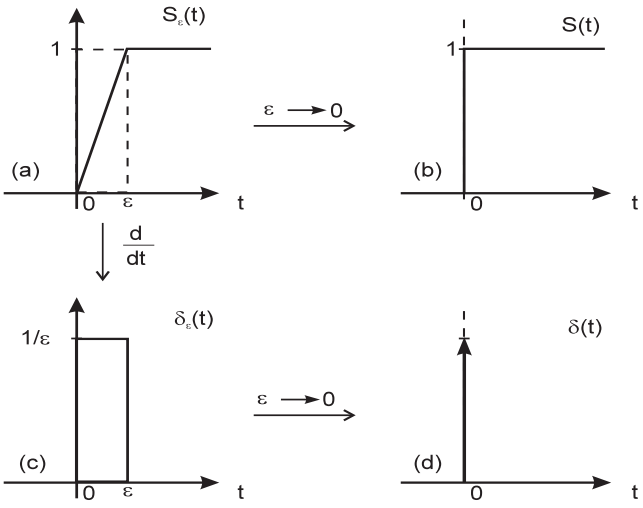


Figure 9.3. From step function to delta function

In (c) we form the derivative $S'_\varepsilon(t) = \delta_\varepsilon(t)$. For $\varepsilon \to 0$ it applies:

$$\lim_{\varepsilon \to 0} \delta_\varepsilon(t) = \delta(t)$$

As the result, we win the relationship

$$\frac{d}{dt} S(t) = \delta(t)$$

This relationship can also be used to differentiate other functions that have a jump.

Relation between impulse and step response. We define the **step response** $SA(t)$ as the reaction of a LTI system to the jump function $S(t)$:

$$SA(t) = L[S(t)] \qquad \text{(step response)}$$

Because of the relationship $\delta(t) = \frac{d}{dt} S(t)$ it applies to the derivation of the step response

$$\frac{d}{dt} SA(t) = \frac{d}{dt} L[S(t)] = L\left[\frac{d}{dt} S(t)\right] = L[\delta(t)]$$

$L[\delta(t)]$ is the reaction of the system to the impulse signal, hence the impulse response $h(t)$.

$$\Rightarrow \quad h(t) = \frac{d}{dt} SA(t)$$

The derivative of the step response is the impulse response.

Example 9.5 : Impulse response.

Given is the step response $SA(t)$ of the RC circuit.

$$SA\,(t) = S\,(t)\,\left(1 - e^{-\frac{1}{RC}\,t}\right)$$

By the relationship $h\,(t) = \frac{d}{dt}\,SA\,(t)$ the impulse response is

$$h\,(t) = SA'\,(t) = \delta\,(t)\,\left(1 - e^{-\frac{1}{RC}\,t}\right) + S\,(t)\,\frac{1}{RC}\,e^{-\frac{1}{RC}\,t}$$

Because of the property of the δ function $\delta\,(t)\;f\,(t) = \delta\,(t)\;f\,(0)$ it applies subsequently

$$h\,(t) = \delta\,(t)\cdot 0 + S\,(t)\,\frac{1}{RC}\,e^{-\frac{1}{RC}\,t} = S\,(t)\,\frac{1}{RC}\,e^{-\frac{1}{RC}\,t}$$

□

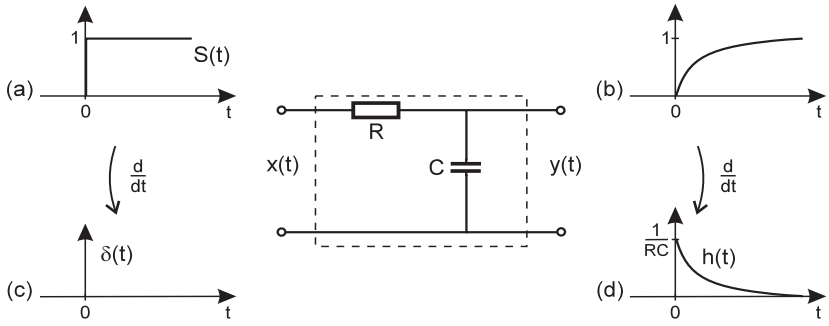


Figure 9.4. Relation between step response and impulse response

Summary: (Impulse response). A linear, time-invariant, causal system L (LTI system) is completely represented by the **impulse response** $h\,(t)$. Because according to the *convolution theorem*, for any input signal $f\,(t)$ the output signal $g\,(t) = L\,[f\,(t)]$ is calculated by

$$g\,(t) = (f * h)\,(t) = \int_{-\infty}^{\infty} f\,(\tau)\;h\,(t-\tau)\;d\tau$$

There are three alternative ways to compute the impulse response $h\,(t)$ of a system:

(1) The impulse response is the reaction of the system to the input signal $\delta\,(t)$:

$$h\,(t) = L\,[\delta\,(t)]$$

(2) The impulse response is the derivative of the step response $SA\,(t) = L\,[S\,(t)]$:

$$h\,(t) = \frac{d}{dt}\,SA\,(t)$$

(3) If $H\,(\omega)$ is the system function (transfer function), then $h\,(t)$ is the inverse Fourier transform of $H\,(\omega)$:

$$h\,(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H\,(\omega)\;e^{i\,\omega\,t}\,d\omega$$

9.5 Simulation of the Transmission Behavior

9.5.1 Frequency analysis of the double pendulum system

Given are the two pendulums of length l shown in Fig. 9.5. At their end masses m_1 and m_2 are located. They are coupled by a spring with spring constants D (see Fig. 9.5). The two masses are deflected by the angle φ_1 and φ_2.

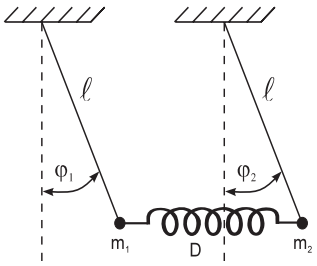


Figure 9.5. Coupled pendulum

We take into account that during the movement on the masses a friction force acts that is proportional to the speed

$$F_R = -\gamma\, l\, \dot{\varphi}_i\,(t) \qquad \text{friction force.}$$

The equations of motion for small deflections φ_1 and φ_2 are:

$$\begin{aligned} \ddot{\varphi}_1\,(t) &= -\frac{g}{l}\,\varphi_1\,(t) - \frac{\gamma}{m_1}\,\dot{\varphi}_1\,(t) + \frac{D}{m_1}\,(\varphi_2\,(t) - \varphi_1\,(t)) \\ \ddot{\varphi}_2\,(t) &= -\frac{g}{l}\,\varphi_2\,(t) - \frac{\gamma}{m_2}\,\dot{\varphi}_2\,(t) + \frac{D}{m_2}\,(\varphi_1\,(t) - \varphi_2\,(t)) \end{aligned} \qquad (*)$$

if $\varphi_1\,(t)$ and $\varphi_2\,(t)$ are the deflections of the masses m_1 and m_2 at time t.

We apply Fourier analysis to determine the eigenfrequencies of the double pendulum system. For this we solve the system of differential equations for the angular deflections $\varphi_1\,(t)$ and $\varphi_2\,(t)$, if the pendulum (1) is affected by an impulse impact $\delta\,(t)$. Thus the differential equations for the pendulum system are without friction ($\gamma = 0$) and with two equal masses ($m = m_1 = m_2$):

$$\ddot{\varphi}_1\,(t) = -\frac{g}{l}\,\varphi_1\,(t) + \frac{d}{m}\,(\varphi_2\,(t) - \varphi_1\,(t)) + \delta\,(t)$$

$$\ddot{\varphi}_2\,(t) = -\frac{g}{l}\,\varphi_2\,(t) + \frac{d}{m}\,(\varphi_1\,(t) - \varphi_2\,(t))$$

These differential equations are translated directly into Maple:

```
> deq1 := diff(phi1(t), t$2) = -g/l * phi1(t) + d/m * (phi2(t) - phi1(t)) + Dirac(t);
> deq2 := diff(phi2(t), t$2) = -g/l * phi2(t) + d/m * (phi1(t)-phi2(t));
```

$$deq1 := \frac{\partial^2}{\partial t^2}\,\phi1\,(t) = -\frac{g\,\phi1\,(t)}{l} + \frac{d\,(\phi2\,(t) - \phi1\,(t))}{m} + Dirac\,(t)$$

$$deq2 := \frac{\partial^2}{\partial t^2}\,\phi2\,(t) = -\frac{g\,\phi2\,(t)}{l} + \frac{d\,(\phi1\,(t) - \phi2\,(t))}{m}$$

We transfer the system of differential equation into the frequency domain by applying the Fourier transform. With the **fourier** command we get two algebraic equations for $\Phi_1\,(\omega)$ and $\Phi_2\,(\omega)$, the Fourier transformed angular deflections $\varphi_1\,(t)$ and $\varphi_2\,(t)$:

$$\Phi_1\,(\omega) = \mathcal{F}\,(\varphi_1)\,, \qquad \Phi_2\,(\omega) = \mathcal{F}\,(\varphi_2)\,.$$

```
> with(inttrans):
> eq1 := fourier(deq1, t, w);
> eq2 := fourier(deq2, t, w);
```

$$eq1 := -w^2\,\mathrm{fourier}\,(\phi1\,(t)\,,\,t,\,w) =$$

$$-\frac{g\,\mathrm{fourier}\,(\phi1\,(t)\,,\,t,\,w)}{l} + \frac{d\,(\mathrm{fourier}\,(\phi2\,(t)\,,\,t,\,w) - \mathrm{fourier}\,(\phi1\,(t)\,,\,t,\,w))}{m} + 1$$

$$eq2 := -w^2\,\mathrm{fourier}\,(\phi2\,(t)\,,\,t,\,w) =$$

$$-\frac{g\,\mathrm{fourier}\,(\phi2\,(t)\,,\,t,\,w)}{l} + \frac{d\,(\mathrm{fourier}\,(\phi1\,(t)\,,\,t,\,w) - \mathrm{fourier}\,(\phi2\,(t)\,,\,t,\,w))}{m}$$

For a clearer presentation we shorten the expressions of the form $fourier\,(\phi\,(t)\,,\,t,\,w)$ through $\Phi\,(w)$ with the **alias** command.

```
> alias(Phi1(w) = fourier(phi1(t), t, w):
> alias(Phi2(w) = fourier(phi2(t), t, w):
```

The two linear equations for $\Phi_1\,(\omega)$ and $\Phi_2\,(\omega)$ are

```
> eq1;   eq2;
```

$$-w^2\,\Phi1\,(w) = -\frac{g\,\Phi1\,(w)}{l} + \frac{d\,(\Phi2\,(w) - \Phi1\,(w))}{m} + 1$$

$$-w^2\,\Phi2\,(w) = -\frac{g\,\Phi2\,(w)}{l} + \frac{d\,(\Phi1\,(w) - \Phi2\,(w))}{m}$$

By solving the system of linear equations with the **solve** command we get the transformed of the angular deflections.

```
> sol := solve({eq1, eq2}, {Phi1(w), Phi2(w)});
> assign(sol);
```

$$sol := \left\{\Phi1\,(w) = \frac{l\,\left(-w^2\,l\,m + g\,m + d\,l\right)}{-2\,m\,w^2\,l\,g + m\,g^2 + m\,w^4\,l^2 + 2\,g\,d\,l - 2\,w^2\,l^2\,d}\right.,$$

$$\left.\Phi2\,(w) = \frac{l^2\,d}{-2\,m\,w^2\,l\,g + m\,g^2 + m\,w^4\,l^2 + 2\,g\,d\,l - 2\,w^2\,l^2\,d}\right\}$$

$\Phi_1(\omega)$ and $\Phi_2(\omega)$ are the transfer functions for pendulums (1) and (2) when we excite pendulum (1). We display both transfer functions in Fig. 9.6 and Fig. 9.7 graphically:

```
> parameter := {g=10, m=0.1, l=1, d=1}:
> plot(subs(parameter, Phi1(w)), w = -10..10);
> plot(subs(parameter, Phi2(w)), w = -10..10);
```

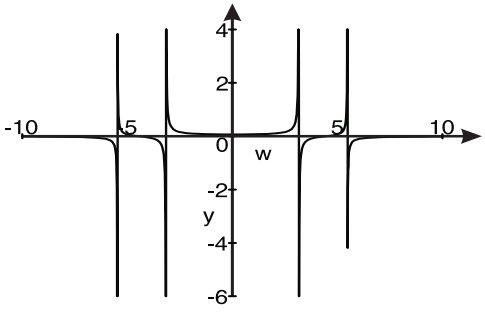


Figure 9.6. Frequency spectrum of the double pendulum system (Pendulum 1)

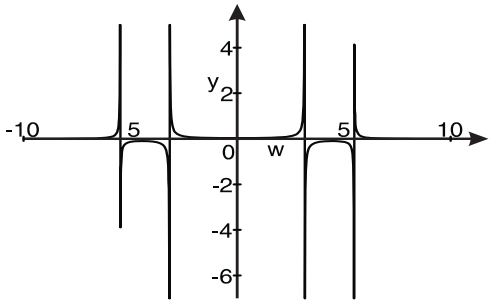


Figure 9.7. Frequency spectrum of the double pendulum system (Pendulum 2)

The resonance frequencies are identified from the two graphs: $\omega_1 = 3.1$ and $\omega_2 = 5.48$. This means that the double pendulum system has two natural frequencies. Since no frictional forces are taken into account when simulating, the natural frequencies correspond exactly to the pole positions of the transfer function: If the system is excited with these frequencies, a resonance catastrophe occurs. We get the pole positions via the zeroes of the denominator of $\Phi_1(\omega)$:

```
> sol := solve(denom(Phi1(w)) = 0, w);
```

$$sol := -\frac{\sqrt{g\,l}}{l}, \frac{\sqrt{g\,l}}{l}, -\frac{\sqrt{m\,l\,(g\,m+2\,d\,l)}}{m\,l}, \frac{\sqrt{m\,l\,(g\,m+2\,d\,l)}}{m\,l}$$

These are the formulas for the eigenfrequencies

$$\omega_1 = \sqrt{\frac{g}{l}} \quad \text{and} \quad \omega_2 = \sqrt{\frac{g}{l} + 2\,\frac{d}{m}}$$

Summary: With the Fourier analysis of the impulse excitation of a linear system, we determine the characteristic frequencies of the system. The resonance points are the natural frequencies of the system. □

9.5.2 Frequency analysis of a high-pass filter

Given is the HP2TCLC high-pass filter, which is composed of two T links (see Example 1.1 from Chapter 1). To determine the system function (=transmission behavior), we select as input voltage the delta function (=pulse function). Then the output signal is the impulse response. The Fourier transform of the impulse response is again the system function!

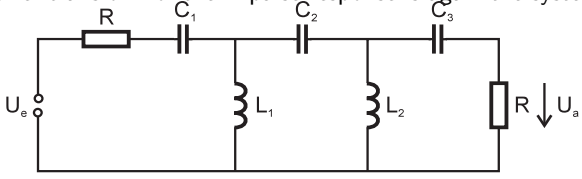


Figure 9.8. High-pass filter with two T-links

The differential equations according to Example 1.1 are as follows

```
> deq1 := R*C1*diff(U1(t), t) + L1*diff(I1(t), t) = Dirac(t) - U1(t):
> deq2 := C1*diff(U1(t), t) - C2*diff(U2(t), t) = I1(t):
> deq3 := L1*diff(I1(t), t) - L2*diff(I2(t), t) = U2(t):
> deq4 := C2*diff(U2(t), t) - C3*diff(U3(t), t) = I2(t):
> deq5 := L2*diff(I2(t), t) - R*C3*diff(U3(t), t) = U3(t):
```

We have selected as input signal $U_e(t) = \delta(t)$ the impulse function. Now we apply the Fourier transform to get the impulse response in the frequency domain. By applying the **fourier** command to the differential equations, a system of linear equations follows for the Fourier transforms of the variables $U_1(t)$, $U_2(t)$, $U_3(t)$, $I_1(t)$, $I_2(t)$.

```
> with(inttrans):
> eq1 := fourier(deq1, t, w):
> eq2 := fourier(deq2, t, w):
> eq3 := fourier(deq3, t, w):
> eq4 := fourier(deq4, t, w):
> eq5 := fourier(deq5, t, w);
```

$$eq5 := I\,L2\,w\,\text{fourier}\left(I2\left(t\right),\,t,\,w\right) - I\,R\,C3\,w\,\text{fourier}\left(U3\left(t\right),\,t,\,w\right) = \text{fourier}\left(U3\left(t\right),\,t,\,w\right)$$

We replace $fourier\left(f\left(t\right),\,t,\,w\right)$ with $F\left(\omega\right)$ using the **alias** command.

```
> alias(U1(w) = fourier(U1(t), t, w)):
> alias(U2(w) = fourier(U2(t), t, w)):
> alias(U2(w) = fourier(U2(t), t, w)):
> alias(I1(w) = fourier(I1(t), t, w)):
> alias(I2(w) = fourier(I2(t), t, w)):
```

Subsequently, we resolve the linear equations for the variables $U_1(\omega)$, $U_2(\omega)$, $U_3(\omega)$; $I_1(\omega)$, $I_2(\omega)$ with the **solve** command.

```
> sol := solve({eq1, eq2, eq3, eq4, eq5},
```

```
>                {U1(w), U2(w), U3(w), I1(w), I2(w)});
> assign(sol);
```

For the transmission behavior, however, not the quantities calculated above are of interest, but the voltage at the ohmic resistor.

$$U_a(t) = R \cdot C_3 \cdot U_3'(t)$$

$U_a(t)$ is the system response to the input signal $\delta(t)$, so the impulse response. The system function (transfer function) is therefore the Fourier transform of $U_a(t)$

$$H(\omega) = \mathcal{F}\left(R \cdot C_3 \cdot U_3'(t)\right)(\omega) = RC_3\,\mathcal{F}\left(U_3'(t)\right)(\omega)$$

$$= R \cdot C_3 \cdot i\omega \cdot U_3(\omega)$$

For the parameters $R = 1000\,\Omega$, $C_1 = C_3 = 5.28 \cdot 10^{-9}\,F$, $C_2 = \frac{1}{2}C_1$, $L_1 = L_2 = 3.128 \cdot 10^{-3}\,H$ we display the transfer function in Fig. 9.9 graphically.

```
> R:=1000: C1:=5.28e-9: C2:=C1/2: C3:=C1: L1:=3.128e-3: L2:=L1:
> H(w) := R*C3*I*w*U3(w):
> plot(abs(H(w)), w = 0..500000);
```

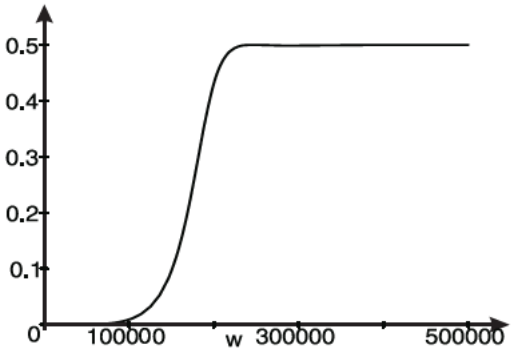


Figure 9.9. Frequency spectrum of the high-pass filter

We can clearly see the high-pass characteristics of the transfer function: Low frequencies are blocked ($H(\omega) \approx 0$) but high frequencies can pass ($H(\omega) \approx \frac{1}{2}$). The cut-off frequency at half of the amplitude is $\omega_g = 175000\,\frac{1}{s}$. $H(\omega)$ fits exactly to the transfer function that we received in Example 1.1 with a complex calculation. □

Summary of the chapter: In this chapter, we have introduced the description of LTI systems using the concept of the impulse response. We also explained the relation between impulse response and system function. Both are related via the Fourier transform. With the Fourier analysis of the impulse excitation of a linear system, we determine the characteristic frequencies of the system or the frequency behavior of the transfer function.

Appendix A

Introduction to the Maple User Interface

Introduction to the Maple User Interface

A

Overview: Maple is a computer program with which we can do mathematics on a computer in a way that was originally only possible with pen and paper. Not only the calculation with real or complex numbers is carried out exactly, but basic mathematical problems of signal analysis such as solving equations or differential equations, Fourier transform and much more is calculated exactly in Maple.

This appendix describes Maple's user interface to provide an orientation for working with the program and provides examples of the most important commands.

A.1 Basics about the User Interface of Maple

A.1.1 Basic settings, first start

After starting Maple, we first have to set the mode of operation: *Worksheet-Mode* or *Document-Mode*. We select the **Worksheet-Mode** for all further simulations, because we combine several commands, activate or deactivate command options, use programming elements or create procedures. Worksheet mode and document mode are identical in their functionality.

All simulations for signal and system analysis and all simulations described in this book are based on this worksheet mode. If the default setting is Document Mode, you can switch to Worksheet Mode at any time:

Tools ⟶ Options ⟶ Interface ⟶ Document ↱ Worksheet ⟶ Apply Globally

After selecting the mode, one is forwarded to the user interface of Maple, which is shown in Fig. A.1:

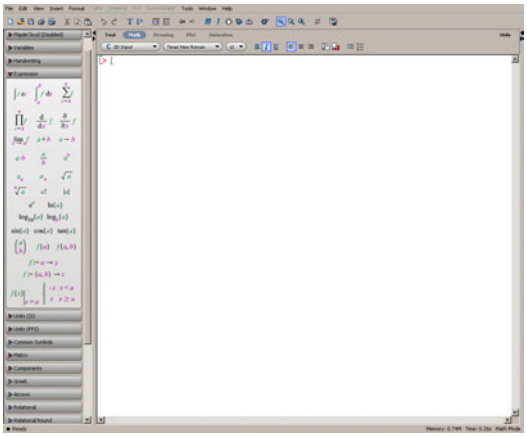


Figure A.1. User interface of Maple (Worksheet-Mode)

At the top there are toolbars for file management, formatting and execution of the worksheet and links to the Maple Help. On the left side there are the palettes, which make the input easier, especially if you don't have any Maple knowledge yet. With the Expression palette the commands can be created very easily, symbol-oriented. At the top of the worksheet additional formatting and selection elements are available.

The Maple input line is then given by a slash [>/ or vertical input prompt [>|. Maple is now expecting an input. There are also two different modes for the *representation* of the input. On the one hand the command oriented Text mode or the symbol oriented Math mode. The only difference between the two modes is Maple's input display. We use the command oriented Text mode.

Although the symbol-oriented input display seems to be more convenient for entering Maple, but the command-oriented input is not only version-independent, but clearer and less error-prone. By default, Maple is in Math mode. **We therefore switch to command mode once:**

Tools → Options → Display → Input display Maple Notation → Apply Globally.

A.1.2 Maple Input

In text mode, an input is made according to the Maple syntax. The input must be preceded by a **;** or **:** and finalized by pressing the **Return** button. An example:

```
> 5*4;
```

$$20$$

The output appears in blue color, one line lower, and centered. Then a prompt appears again. All commands used in this book are listed in this Text mode. For example, if the function $x^2 \sin(x)$ has to be *integrated*, the **int** command is specified:

```
> int(x^2*sin(x), x);
```

$$-x^2 \cos(x) + 2 \cos(x) + 2x \sin(x)$$

The Maple input is described in the following chapters of the book as specified in the text mode. With the **diff** command the *derivative* of $x^2 \sin(x)$ is determined.

```
> diff(x^2*sin(x), x);
```

$$2x \sin(x) + x^2 \cos(x)$$

We can enter an expression

```
> y^2 * sin(x);
```

and instead of pressing Return we right-click this expression. Then Maple opens a context menu (see Figure A.2) from which we can select operations to be applied to the expression.

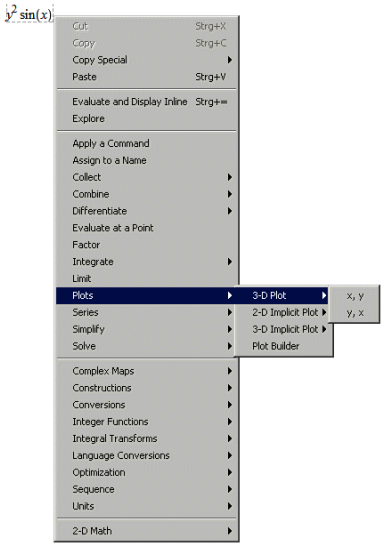


Figure A.2. Interactive Manipulation

A.1.3 Maple Output

As an alternative to right-clicking on the Maple input, the Maple output can be further edited. For clarification, let us discuss an integral task $\int x^2 \sin(x)\, dx$. In order to assign the result of the calculation to a variable *expr*, you use the variable assignment with := *in front of* the Maple command:

```
> expr := int(x^2*sin(x), x);
```

$$expr := -x^2 \cos(x) + 2 \cos(x) + 2x \sin(x)$$

Alternatively, the **%** operator (ditto operator) is available. **%** is used to refer to the result of the last Maple computation. Then, the variable assignment takes place *after* the **int**-command by

```
> expr := %;
```

$$expr := -x^2 \cos(x) + 2 \cos(x) + 2x \sin(x)$$

Afterwards, on *expr* formula manipulations can be performed:

If you mark the result of the Maple calculation (Maple output) and press the right mouse button, possible arithmetic operations can be applied to the result. E.g. *Differentiate* $\longrightarrow$ *x* differentiates the result.

If, instead of differentiating with the right mouse button, you select e.g. *Plots* $\longrightarrow$ *2D-Plot*, the function is drawn in a *Smartplot*. The scaling of the x axis is always from -10 to 10.

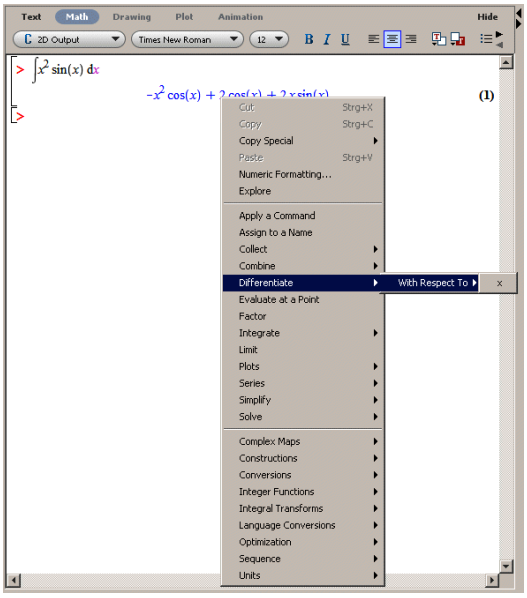


Figure A.3. Manipulation of output

A.2 More Support on Maple Input

A.2.1 Expression Palette

Standard mathematical tasks are already preset as commands in the palettes. You can easily recognize the tasks by the given symbols: $\frac{d}{dx}f$ symbolizes the derivation of f with respect to the variable x or $\int_a^b f\,dx$ stands for the definite integral. Like through clicking on the symbol

$$\frac{d}{dx}f$$

for the derivative appears in the input line
> diff(f, x);

In this input line you have to enter the color coded specified symbols **f** and **x**. Through pressing the Return key, the command is executed.

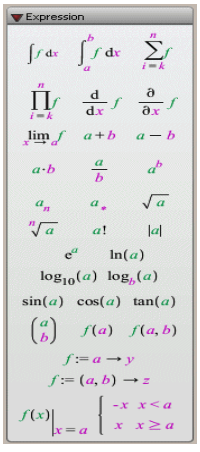


Figure A.4. Expressions

A.2.2 Interactive PlotBuilder

The interactive **PlotBuilder** is very extensive. To use it, define the function to be drawn, e.g. with y:=sin(x); click with the right mouse button on the Maple output and follow the menu guidance

Plots ⟶ *Plot Builder* ⟶ *Options* ⟶ ... ⟶ *Plot*

Through the PlotBuilder, whose surface is shown on the left column in Fig. A.5 is displayed, the type of display (e.g.: 2-D plot) is selected. Further options of the plot command can be selected via the Options menu (see right column). If the Plot button is pressed, the image appears in the worksheet; if Command is pressed, the Maple command with all specified options is displayed.

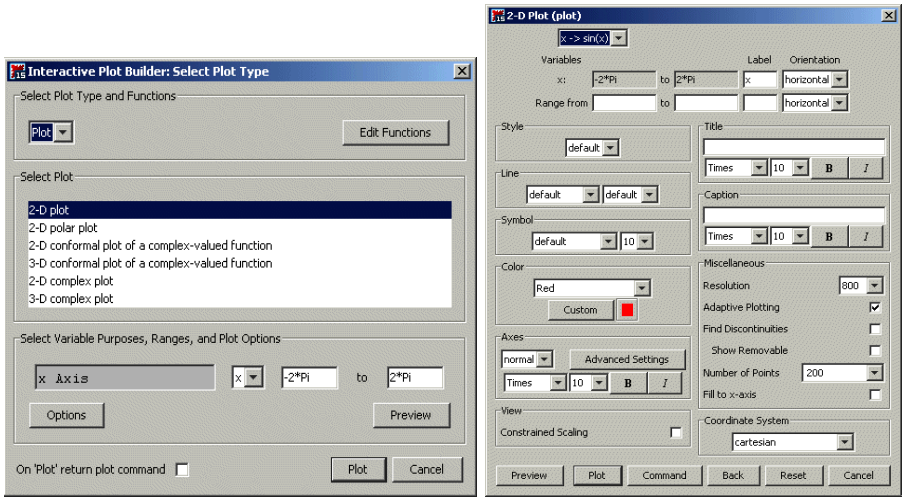


Figure A.5. PlotBuilder

A.2.3 Interactive DE Solver

The interactive **DE Solver** is very extensive. To use it, define the differential equation to be solved, right-click on the Maple output and follow the context menu

Solve DE Interactively

In this menu, initial conditions or parameters of the differential equation can be specified. You decide whether the differential equation is to be solved numerically or analytically and receive a further menu according to your choice, where you can specify options for the solution. You decide whether the Maple commands should be displayed and which output you want to have in the worksheet (Plot/ Solution/ Maple Command) or (Plot/ NumericProcedure/ Maple Command) in the case of the numeric variant.

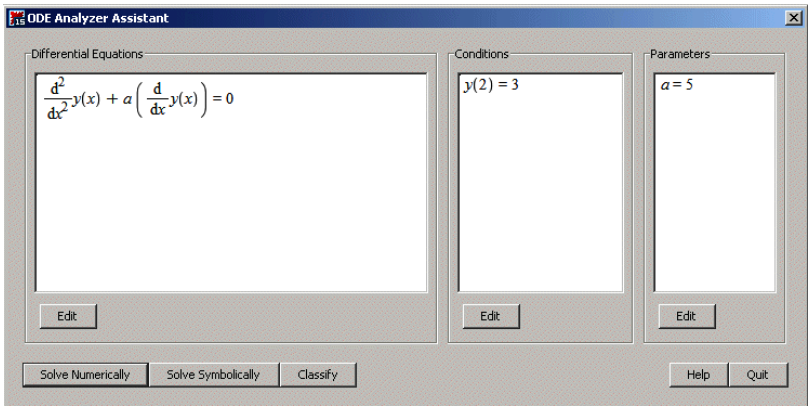
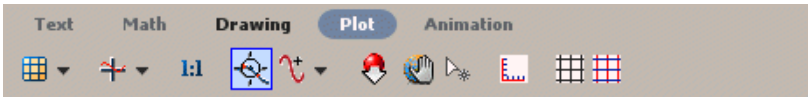


Figure A.6. Interaktiver DE Solver

A.2.4 Maple Graphic

By clicking on a graphic created in Maple (e.g. created by the command plot(x^2, x=0..2);) it appears a new toolbar at the top taskbar, with which you can change the graphic interactively.



Now the **Plot** mode is active. You can, for example, change the axes label, insert grid lines, move the graph, zoom or change properties of the graph such as line thickness, color and much more. The **Drawing** mode is also available. With this option you can create a new graphic in the selected graphic. Insert further graphic elements, which can be found under the corresponding icons. Alternatively, the right mouse button is again available. Thus, there is a comfortable way to create legends, to integrate them into the graphics and to export the graphics into one of the formats <eps, gif, jpg, bmp, wmf>.

In particular, if an animation is created by **animate** or **display**, the image must be clicked. The symbol for the **Animation** mode appears. If you press the start button, the animation starts to run. For animations the **Plot**- and **Drawing** modes can also be selected.

A.3 Maple Structures

Like any programming language, Maple has symbols for operators that match the standard operators in the case of basic arithmetic operations, assignment operators, and much more. Special conventions apply to the brackets for Maple, since each has its own meaning. For completeness, this section also describes the programming structures such as for loops or if queries.

Operators

+	Addition	<	smaller
-	Subtraction	<=	less equal
*	Multiplication	>	larger
/	Division	>=	larger equal
**	Power	=	same
^	Power	<>	uneven
.	Matrix multiplication		

Zero operators

:=	Assignement
;	Command end for execution and output of result
:	Command end for execution *without* output of result
%	last calculated expression (ditto operator)
\"	Quotes for texts in Maple commands

Brackets

(...)	Brackets in a mathematical formula
[., .,..., .]	Generating a list
< ., .,..., .>	Generating a column vector
< .| .|...| .>	Generating a row vector
{ ., .,..., .}	Generating a set

Programming Structures

for-Loop
for <*index*> from <*start*> by <*step*> to <*end*>
do <*instruction*> end do;

while-Loop
while <*condition*>
do <*instruction*> end do;

if-Condition
if <*condition*> then <*instruction*> endif;

if/else
if <*condition*> then <*instruction*>
else <*instruction*>
endif;

Procedures
p:= proc(<*parameter*>)
local <*variables*>;
<*instructions*>
end;

A.4 Maple's Help

Maple offers both the beginner as well as the advanced users a very extensive help, which can be accessed via the Menu item

Search ⟶ Maple Help

A separate window then opens (see Fig. A.7), in which you then select the desired assistance for the command entered. Particularly valuable are the examples that can directly copied to the current worksheet. So you have already a syntactically correct example.

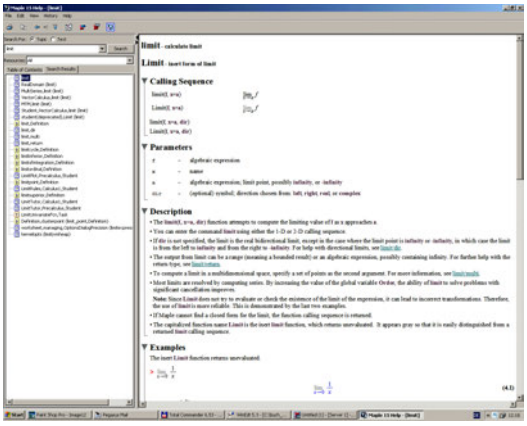


Figure A.7. Separate Help Window

All possible commands starting with the specified characters are displayed. If you are not sure about the command name, then you can pretend like this: If we are looking for the help page for the command with which the limit values is calculated, it is best to first enter *li* (for Limes). Then you enter *limit* or click directly on the Maple proposal **limit**.

Alternatively, you can specify in the worksheet

> ?limit

then the help menu will open, but now directly to the **limit** command. Or you click in Math mode with the right mouse button on the symbol and select *Help on Command*.

With

> ???limit

the advanced user will immediately receive the examples from the Maple help; the remaining sections are then hidden.

A quick overview of the functionality, Hotkeys, often used icons, or user interfaces provides the quick reference: ? ⟶ Quick Reference.

Index