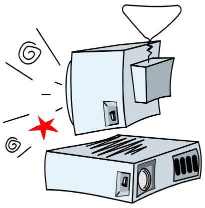
2

Basics

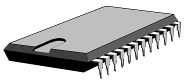
Technical requirements

Requirements on your computer

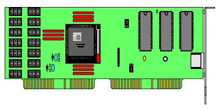
Using WinFACT is possible even on standard PC hardware. Nevertheless comfort and efficiency increase with higher performance hardware components. Minimum requirements for a comfortable work are as following:



A PC with a 486 CPU, Pentium or better

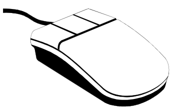


A minimum of 128 MB RAM



A graphic adapter with a resolution of 800x600 pixel (256 colours) or better

Free hard disk space of at least 80 MB



A mouse or a similar pointing device

The operating system Windows 98/NT/2000/XP

Introduction and program philosophy

What is WinFACT?

WinFACT is a modular program system that on one side offers tools for the analysis, synthesis and simulation of conventional closed-loop systems, on the other side WinFACT contains components for the handling of fuzzy systems. The graphical user interface of Windows guarantees an extremely short training period combined with highly comfortable work. A great variety of program interfaces and file formats allows the communication with all kinds of peripheral devices, external processes, user-written programs and standard software products. Therefore WinFACT is of interest for education and training as well as for research and developement.

WinFACT represents a combination of independent but combinable program modules between which data transfer can be achieved via different communication channels. WinFACT in its complete version first contains all components required for the anaysis and synthesis of conventional closed-loop systems. Among these are:

- The identification of linear systems based on measured values of the input and output variable. The identification algorithm used is characterized by its robustness against disturbances like measurement noise and is applicable for any type of input signals.
- The analysis of linear systems by calculation of their step response, Bode plot, Nyquist plot, root locus and pole/zero distribution.
- The synthesis of linear controllers. All types of standard elements are available as controller components.
- The simulation and optimization of closed-loop systems.

Fuzzy logic and fuzzy control

Besides conventional methods a focal point of the program concept is in the area of modern methods like *fuzzy logic* and *fuzzy control*. WinFACT offers tools for all-level access starting with "fuzzy logic experiments" and continuing with an interactive design of rule-based systems up to synthesis and simulation of complex fuzzy control loops. Each level of course allows unrestricted usage of all properties and conecpts that fuzzy logic offers:

- Different types of fuzzy sets (triangular, trapezoid, singleton)
- Various logical operators and inference mechanism
- Multitude of defuzzyfication methods

Design and analysis of hybrid systems - containing convential components and fuzzy components as well as neural networks - is practicable based on a block-oriented simulation. Components for the graphical representation and evaluation of measurements or simulation results based on the Windows MDI standard complete the program system.

WinFACT offers a Windows-typical user interface, beginning with the characteristical menu structure and toolbars. Integrated help features combined with comfortable dialog windows reduce invalid or inconsistent input data to a minimum.

Hardware interfaces

WinFACT offers a great variety of different hardware interfaces which allow a process coupling (e. g. for measurement data aquisition or control) or a transfer of structures (e. g. controllers) designed with WinFACT to a target platform outside the PC:

- By *A/D-D/A-converter cards*. Nearly all customary card types are supported by WinFACT (e. g. Advantech, Meilhaus, National Instruments, Keithley, Sorcus, Wasco, Bitzer, Leybold, Analog Devices, Intelligent Instrumentation, PCMCIA cards etc.). The usage of those cards is especially recommended for training purposes or prototype control systems at real processes.
- By *serial interfaces* in combination with appropriate hardware modules (e. g. ISM-series from Ganter company or ADAM-modules from Advantech). This concept is especially qualified for measurement data aquisition or time-uncritical applications.
- PLC and process control systems
- For a PC-independent, self-sufficient realization the high-efficient *C-code-generator* is recommended. This tool "translates" any system structure into general-purpose ANCI C code which can be implemented on nearly any kind of target platform (e. g. microcontroller boards). This concept allows the real-time-control even of high-dynamical processes.

Details may be taken from the corresponding product information.

WinFACT file formats

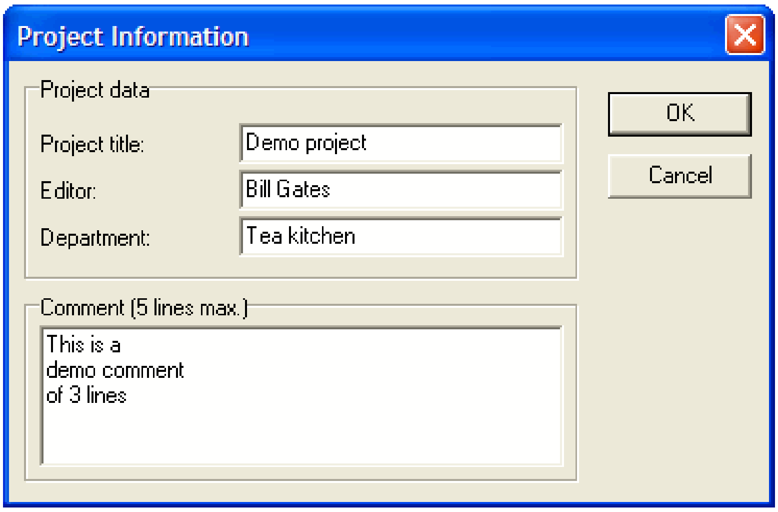
All files used by WinFACT for saving system specific data are ASCII files. So the WinFACT user is able to view (and modify - if necessary) the file content not only out of the corresponding WinFACT programs but also with a standard text editor. Because the files created by WinFACT normally do not extend a length of a few hundred lines, the speed of read and write operations is not decreased by choosing this ASCII format.

The different types of files used by WinFACT are denoted by corresponding file extensions. The format of all relevant file types is described in the following chapters.

Project information

For a detailed specification of the file content all file types can include a *project information* with a maximum length of eight lines that is located at the top of the file above the real data. The corresponding file rows are characterized by a leading exclamation mark. The first five of these rows are reserved for the *project description*, row six for the *project title*, row seven for the *project editor* and row eight for his *department*. If a file containing a project information is opened the project information is displayed automatically if this option was activated by the WFSETUP program (see section *Configuring WinFACT* of the *First steps* chapter); if a file is saved for the first time the project information is requested automatically.

The illustrations below show the project info dialog window and the file header created for the specified project data. All project data are already displayed within the WinFACT standard FILE OPEN dialog before opening the file; thus a fast search for a specific file can be executed using the included project information.



Dialog window for viewing and modification of a project information

```
! This is a
! demo comment
! of 3 lines
!
!
! Demo project
! Bill Gates
! Tea kitchen
```

Corresponding file header created for project information

File types

Transfer functions (UFK-files)

Transfer functions of linear systems with dead time given as

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} \, e^{-Ts}$$

are saved in files with the extension UFK and the following structure:[1]

[1] For more clearness *no* project information is included in the following file listings

```
m
n
b0
.
.
bm
a0
.
.
an
T
```

Simulation results (SIM-files)

A list of pairs of values (t_i, y_i) as it may be created e. g. in form of simulation results is saved by WinFACT one pair under the other and with at least one space character between both values (time and amplitude value). The number of pairs is determined automatically. This file type uses the file extension SIM.

```
t1   y1
t2   y2
t3   y3
.    .
.    .
.    .
```

WinFACT file with pairs of values

General pairs of values (XY-files)

General pairs of values (x_i, y_i) are saved in files with an XY file extension. The file structure is identical to the structure of SIM-files described before.

Multiple pairs of values (MXY-files)

MXY-files are used to save multiple curves (e. g. simulation results) within a single file, e. g. trajectory fields created by the WinFACT module SUSY. Each curve starts with the number of contained pairs of values followed by the pairs of values themselves in the same format used by SIM- resp. XY-files. The end of file is detected automatically, so the number of curves has not to be specified.

Frequency responses (BD- resp. OK-files)

The frequency response $G(\mathrm{j}\omega)$ of a linear transfer element can be represented in two ways:

- In form of a *Bode plot* consisting of triples of values

$$\left(\omega_i,\ \left|G(\mathrm{j}\omega_i)\right|_{\mathrm{dB}},\ \angle G(\mathrm{j}\omega_i)\right)$$

containing frequency, gain in dB and phase in degree. The corresponding file uses the extension BD and has the following structure:

ω_1	$\left\|G(\mathrm{j}\omega_1)\right\|_{\mathrm{dB}}$	$\angle G(\mathrm{j}\omega_1)$
ω_2	$\left\|G(\mathrm{j}\omega_2)\right\|_{\mathrm{dB}}$	$\angle G(\mathrm{j}\omega_2)$
ω_3	$\left\|G(\mathrm{j}\omega_3)\right\|_{\mathrm{dB}}$	$\angle G(\mathrm{j}\omega_3)$
.	.	.
.	.	.
.	.	.

- In form of a *Nyquist plot* consisting of triples of values

$$\left(\omega_i,\ \mathrm{Re}\left\{G(\mathrm{j}\omega_i)\right\},\ \mathrm{Im}\left\{G(\mathrm{j}\omega_i)\right\}\right)$$

containing frequency, real and imaginary part. The corresponding file uses the extension OK and has the following structure:

ω_1	$\mathrm{Re}\left\{G(\mathrm{j}\omega_1)\right\}$	$\mathrm{Im}\left\{G(\mathrm{j}\omega_1)\right\}$
ω_2	$\mathrm{Re}\left\{G(\mathrm{j}\omega_2)\right\}$	$\mathrm{Im}\left\{G(\mathrm{j}\omega_2)\right\}$
ω_3	$\mathrm{Re}\left\{G(\mathrm{j}\omega_3)\right\}$	$\mathrm{Im}\left\{G(\mathrm{j}\omega_3)\right\}$
.	.	.
.	.	.
.	.	.

Vectors (VEK-files)

Vectors are saved component by component in files with a VEK file extension. The first row (following the project information) must contain the number of vector components. Thus the vector

$$\underline{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

leads to the following file:

```
n
b1
b2
.
.
bn
```

Complex vectors (KVK-files)

Complex vectors (e. g. vectors of eigenvalues)

$$\underline{r} = \begin{pmatrix} a_1 + \mathrm{j}b_1 \\ a_2 + \mathrm{j}b_2 \\ \vdots \\ a_n + \mathrm{j}b_n \end{pmatrix}$$

are saved in files with a KVK extension as following:

```
n
a1  b1
a2  b2
.
.
an  bn
```

WinFACT file containing a complex vector

Matrices (MAT-files)

Matrices are saved row by row, each row contains exactly one matrix element. The first row contains the number of maxtix rows resp. columns separated by at least one space character. Matrix files use the MAT extension. Thus a $m \times n$ -Matrix

$$\underline{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

is saved in a file of following structure:

```
m n
a11
a12
.
.
a1n
a21
a22
.
.
```

State space models (ZRM-files)

State space models written as

$$\underline{\dot{x}} = \underline{A}\underline{x} + \underline{b}u, \quad y = \underline{c}^{\mathrm{T}}\underline{x} + d\,u$$

with the system matrix

$$\underline{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

the control vector

$$\underline{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

and the output vector

$$\underline{c}^{\mathrm{T}} = \left(c_1, c_2, \ldots, c_n\right)$$

are saved in files with a ZRM extension and the following structure:

```
n
a11 a12 ... a1n
a21 a22 ... a2n
 .   .  ... .
 .   .  ... .
an1 an2 ... ann
b1
b2
.
.
bn
c1
c2
.
.
cn
d
```

Function value matrices (FWM-files)

Function value matrices characterize a function of two variables

$$z = f(x, y)$$

in discrete form, i. e. in form of discrete sampling points

$$z_{ij} = f(x_i, y_j), \; i = 1, \ldots, m \quad j = 1, \ldots, n$$

with a range of values $x_{\min} \le x \le x_{\max}$, $y_{\min} \le y \le y_{\max}$. These files may be used to represent the function in form of a 3D-graphic or contour lines, e. g. by the WinFACT module INGO or mathematical programs like *Mathematica*.

Example: The function

$$z = x^2 + y^2, \quad 0 \le x \le 5, \; 0 \le y \le 10$$

with $m = 6$ sampling points for x and $n = 6$ sampling points for y results in the following function value matrix:

$$\underline{A} = \begin{pmatrix} 0 & 4 & 16 & 36 & 64 & 100 \\ 1 & 5 & 17 & 37 & 65 & 101 \\ 4 & 8 & 20 & 40 & 68 & 104 \\ 9 & 13 & 25 & 45 & 73 & 109 \\ 16 & 20 & 32 & 52 & 80 & 116 \\ 25 & 29 & 41 & 61 & 89 & 125 \end{pmatrix}$$

The file format corresponds to the MAT-file format; the file has the following structure:

```
m n
a11
a12
.
.
a1n
a21
a22
.
.
amn
[xmin xmax]
[ymin ymax]
```

The specification of the range of values (last two rows of the file) is optional and can be omitted; in this case INGO represents the characteristic map over a free-selectable range.

Block lists (BL-files)

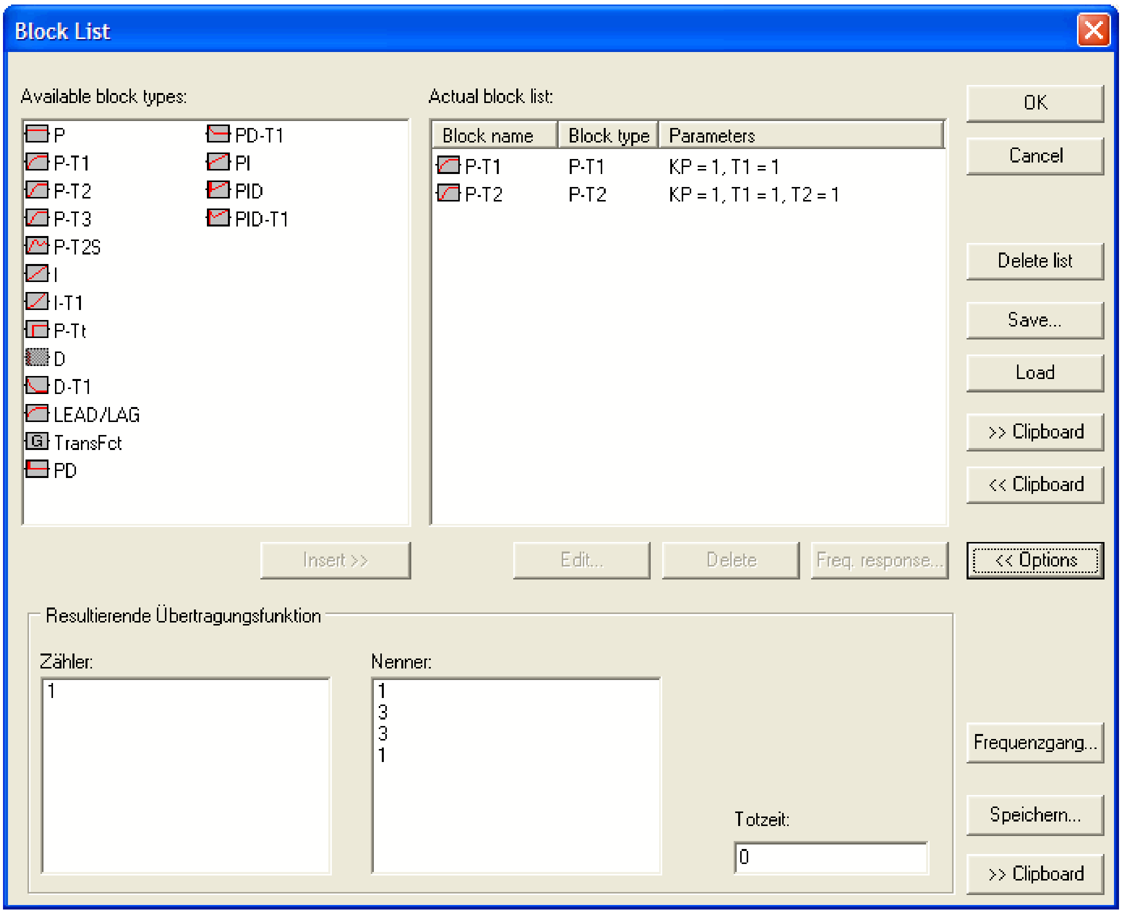
Block lists are supported e. g. by the programs LISA, RESY and BORIS. They contain a list of (normally several) linear transfer elements like P-T1, P-T2 etc. and are used e. g. for the specification of composed controllers or plants. The table below lists all available block types.

Type	Transfer function
P	K_P
P-T1	$\frac{K_P}{1+T_1 s}$

P-T2	$\dfrac{K_\mathrm{P}}{(1+T_1 s)(1+T_2 s)}$
P-T3	$\dfrac{K_\mathrm{P}}{(1+T_1 s)(1+T_2 s)(1+T_3 s)}$
P-T2S	$\dfrac{K_\mathrm{P}}{\left(\dfrac{s}{\omega_0}\right)^2+\dfrac{2D}{\omega_0}s+1}$
I	$\dfrac{K_\mathrm{I}}{s}$
I-T1	$\dfrac{K_\mathrm{I}}{s(1+T_1 s)}$
P-Tt	$K_\mathrm{P}\ e^{-T_t s}$
D-T1	$\dfrac{T_\mathrm{D} s}{1+T_1 s}$
LEAD/LAG	$K_\mathrm{P}\dfrac{1+T_1 s}{1+T_2 s}$
TransFct	$\dfrac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} e^{-T_t s}$
PD	$K_\mathrm{P}(1+T_\mathrm{V} s)$
PDT1	$K_\mathrm{P}\left(1+\dfrac{T_\mathrm{V} s}{1+T_1 s}\right)$
PI	$K_\mathrm{P}\left(1+\dfrac{1}{T_\mathrm{N} s}\right)$
PID	$K_\mathrm{P}\left(1+\dfrac{1}{T_\mathrm{N} s}+T_\mathrm{V} s\right)$

PIDT1	$K_P\left(1+\frac{1}{T_N s}+\frac{T_V s}{1+T_1 s}\right)$

The screenshot below shows the user dialog for the configuration of block lists.



Dialog window for the configuration of block lists

Within the left area of the dialog all available block types are listed; a block can be inserted into the actual block list by a double click or by selecting the block first and pressing the *Insert>>* button afterwards. The right area of the dialog contains all blocks of the actual block list with their name, block type and block parameters. By a double click on one of these blocks the corresponding block can be modified. The *Freq. response* button allows the calculation and graphical representation of the frequency response of the block in form of a Bode plot. The buttons located at the right dialog margin allow among other things the saving and loading of block lists to resp. from a block list file or the Windows clipboard. The *Options* >> button in the lower window area enlarges the dialog and displays the transfer function of the actual block list; this function can be saved in a UFK-file or copied to the clipboard via the corresponding dialog buttons.

A detailed knowledge of the internal file format is not necessary for working with block lists.

Closed-loop systems (SCL-files)

Standard control loops designed with the WinFACT module RESY are saved in files with the SCL extension (*Standard Control Loop*). These contain controller and plant, each represented by a block list. A detailed knowledge of the internal file format is not necessary for working with this file type.

Specific help functions

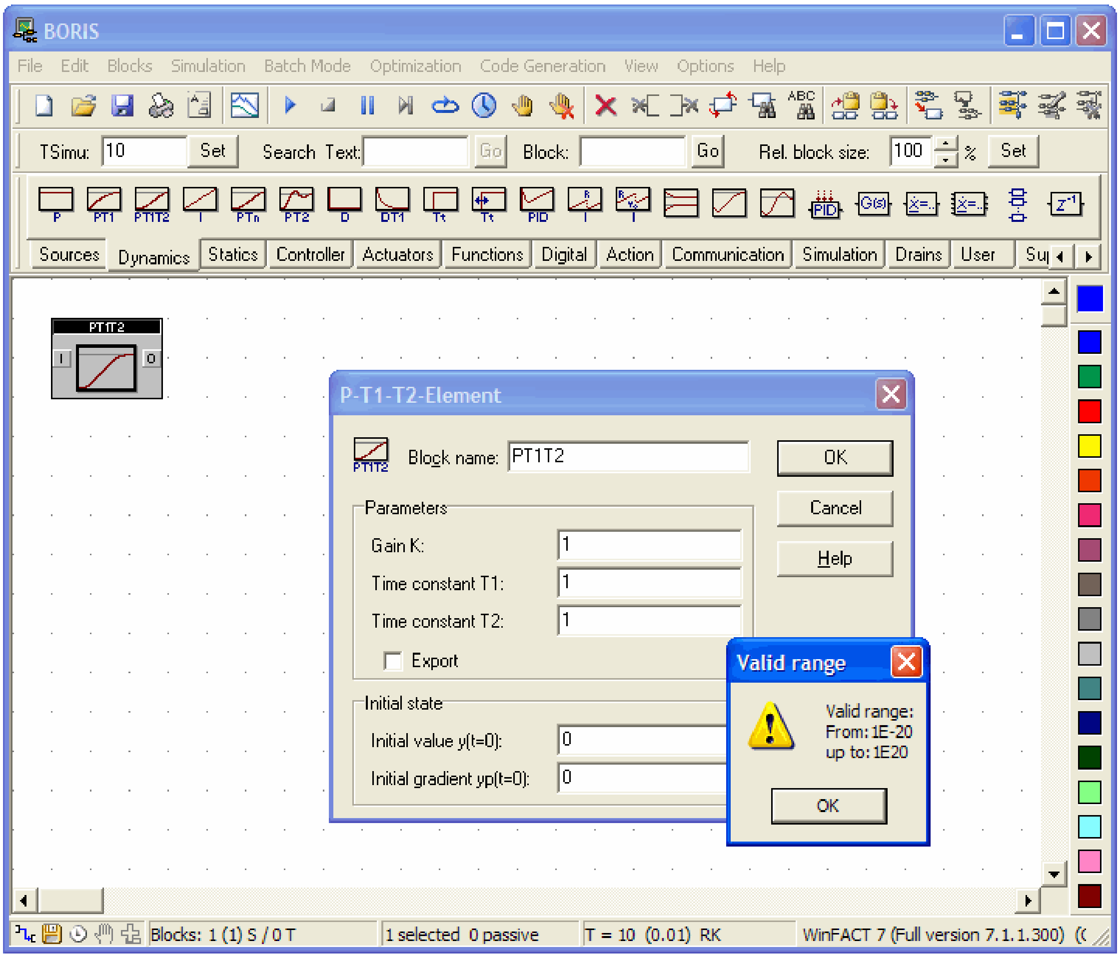
Toolbar help

An important facilitation especially for beginners is represented by the toolbar help function which can be activated by using the WFSETUP program described earlier. If this help modus is activated a Windows typical hint window appears if the mouse cursor is located over a toolbar button for more than a second. Thus the operations executed by different toolbar buttons can be learned in a very easy way.

Toolbar help function

Range of numerical parameter values

If numerical values have to be specified in most cases it is desirable to know the valid range of the parameter values in advance to avoid troublesome error messages or warnings. For this purpose WinFACT offers an automatical range check. Within all numerical input fields of parameter dialogs a right mouse click opens a popup menu that allows to check the valid parameter range. The screenshot below illustrates this feature.



Displaying the valid range of values for numerical parameters