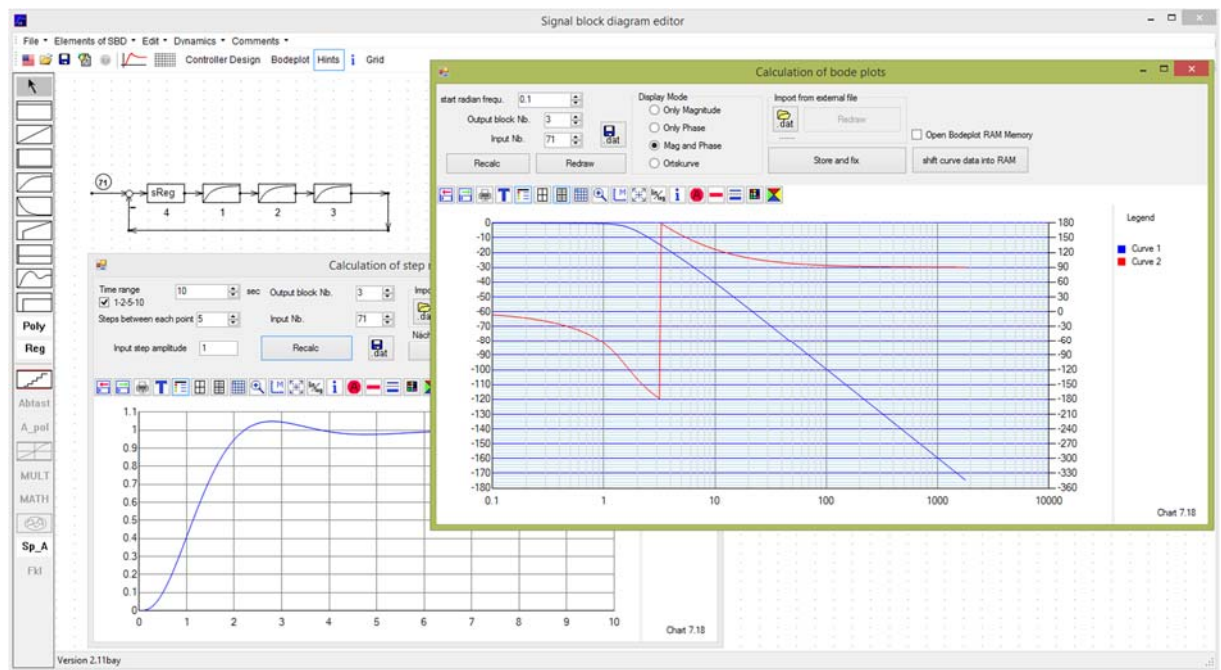


# REGC#

## Control Systems with Windows Fits to Version 3.8a or higher

**Prof. Dr. J. Bayerlein**



## A SHORT DESCRIPTION OF THE FUNCTIONS & TUTORIAL

1<sup>st</sup> English Edition (V1.3)

## Contents list

Preamble: .....	3
System Information and specifications .....	3
2.1 System limitation .....	4
2.2 Restrictions.....	4
The Signal Block Diagram Editor.....	5
4.1 P-Block .....	6
4.2 I-Block, D-Block .....	6
4.3 PT1-Block.....	7
4.4 DT1-Block, PI-Block, PD-Block .....	7
4.5 PT2-Block.....	7
4.6 Delay-Time -Block .....	8
4.7 Polynomial-Block (Transfer function block) .....	8
4.8 Multi Level Block: .....	9
4.9 PIDT1- Controller .....	9
4.9.1 Controller-Type.....	10
4.9.2 PIDT1-Converter .....	10
4.10 Loaded step response block.....	11
4.11 Loaded Bodeplot .....	12
5.1 Step Response Window .....	14
5.2 Bodeplot Window .....	15
5.3 Controller Design.....	17
5.4 Example of Controller design (see Workbook) .....	20
5.5 Report window.....	23

## Preamble:

The control systems lab in the department of electrical engineering and information technology at the FH-Lübeck uses the program REGC# or (speak:) REGCSharp to create the possibility of a fee free use, supporting the control systems lessons and practical courses. That means, you can use the program for free.

This program is developed starting with the former program REGDELPH, which is written by author Prof. Hans-Jürgen Bederke. He has allowed me to use his Pascal code to convert it into modern C# - Code. But the knowledge behind the program is coming from Mr. Bederke. Thanks for his former big part of work and thanks to open his program for all users in the world.

The program author will do his best to eliminate errors and asks the user for a hint if mistakes and or irregularities should be noticed. Furthermore initial ideas and wishes concerning the functionality of the program are very welcome.

Specifications and information contained in this manual could be changed without any further announcement. They are based to program version 2.11. The program was developed with MS Visual Studio 2013 and is Windows- based. It should be noticed, that these programs are under protection of law. No commitments could be derived by exploitation of this document and the program REGC#, the use of program and this document is without any guarantee. No responsibility or liability for any errors or inaccuracies, that may appear in this manual or the program described in it would be assumed.

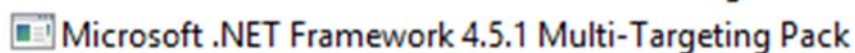
Use of this document and the described software is only permitted according to this agreement.

To this English translation:

Because of my low experience in translation, and a less vocabulary too, it may happen that misunderstandings and use of wrong and/or unusual words occurs in this translation. Therefore the MSOE-ISN-students are kindly asked for useful hints, if such things are noticed. May be we are able to clear up misunderstandings and wrong words for the next edition together.

## System Information and specifications

You can use any windows computer with Windows 7 or higher. It is necessary that Dot Net 4.0 or higher is available, but other MS- programs like Word also use this MS- library. Look into your system, if Dot Net is already installed. Should look like this:



Otherwise go to Microsoft page and download a free Runtime package of latest DotNet called "Redistributable".

REGC# needs only two files: The exe- file **RegCsharp.exe** and in the same path the dll- file **ChartTool.dll**. No further installation is required.

## 2.1 System limitation

**Blocks:** maximal 70, from 1 up to 70; the number of a block is simultaneously his output and respectively the input of the following block. Every block is to be allowed to have 10 inputs at most, which are produced by the summing points in front of it. A Multiplication-block (MU) in front occurs two inputs.

**Segments of Branches:** up to 250

**Summing-points:** up to 30

**Junctions:** up to 30

**Controller:** up to 8.

**External Inputs:** up to 10, numbered from 71 up to 80 (because 1 up to 70 is reserved for blocks). The number of the external inputs are set automatically by program. Before you can do that, a branch as the block-input must be drawn.

The equation system for frequency response calculations is of the same order as the number of blocks. (up to 70<sup>th</sup> order). The coefficients are always complete frequency response vectors, consisting of a set of 256 complex numbers, taken for all inputs, included the external input values. The solution of the equation system is be done by a Bederke developed elimination procedure, which reduces errors and expenditure of calculation.

During the calculation process of the equation system, not more than 150 frequency response vectors may appear at the same time. At what the real requirements depends to the degree of meshes of the control circuit (simple serial circuits will be solved first). Contrary to the enormous expenditure of calculation, this complex calculation is quite fast, comparing with the calculation time for step responses. The screen size is not limited, because even non-visible parts of the signal block diagram, which are outside of the visible area, will be used for the calculation. Normally frequency response calculation will be done over 4.2 decades, in which the user has to choose a suitable radian start frequency for the calculation, at controller design in that way, that the crossover frequency is within the frequency range. Always the radian frequency  $\omega = 2\pi f$  will be represented.

At frequency response calculations we get always only one input and one output. Between this both parts the frequency response will be ascertained, at external inputs including every internal feedback, at internal input signals of the system, the open loop (for example a control loop). But at meshed systems it already could contain feedback loops.

## 2.2 Restrictions

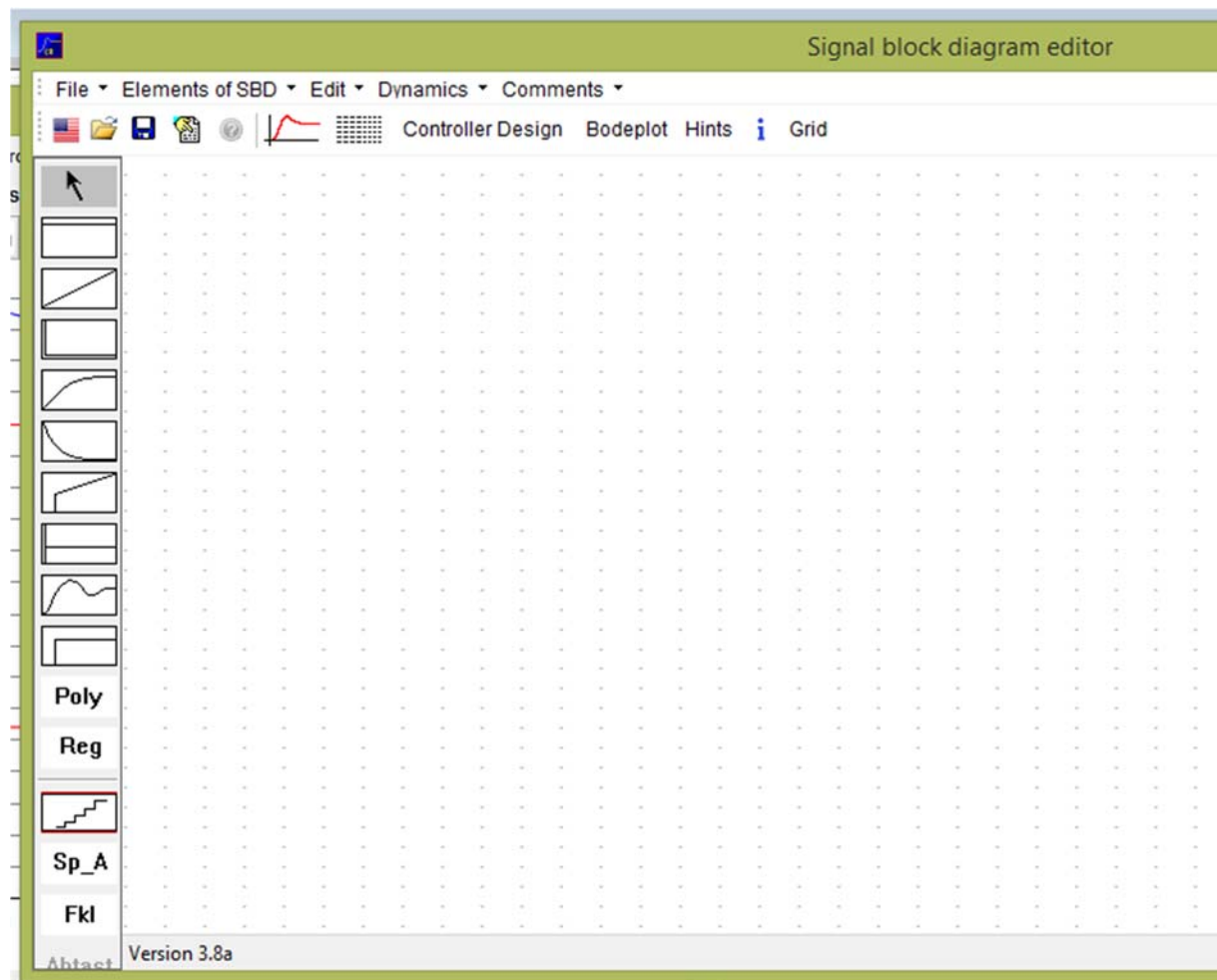
Step responses will be calculated by stepwise integration, what, for D-blocks can cause problems (if a very high calculation time should be avoided.) Therefore a direct feedback across P- or D-blocks is forbidden for D-blocks, PD-blocks and P-blocks too. Because the calculation time affects as delay-time, a stepwise integration will produce unstable systems.


In your design you should avoid D-blocks if possible. A feedbacked D-block can be replaced by a DT1-block before editing. Mostly D-blocks can be replaced by I-blocks as the branch is inverted (that means changing of the effect-direction) and you get an error free result.

If a controller should be optimized, only one feedback input is allowed. Except the summing point no other summing points in the feedback branch directly in front of the controller are allowed. Otherwise the result for the set value in loop frequency response would be calculated faulty (in case of controller optimization). The set value has to be added with negative sign (not a must, if the negative sign is on another place in the loop, for example at a block-input or another summing point). In some cases, positive feedback value or negative set value, may be, the program is not able to find the right feedback block (number). In this cases you have to type in the fitting number.

## The Signal Block Diagram Editor

After program start the following working sheet will appear on the screen:



The starting program language is German, if you want to have English language you have to click on the “US- flag – button” .

All further screenshots are made with English language.

You can draw and create additionally to the later described blocks the four elements using the hotkeys or menu → “Elements of SBD”.

<b>w</b>	<b>Branches</b>
<b>v</b>	<b>Junctions</b>
<b>i or -</b>	<b>Minus-Sign</b>
<b>s</b>	<b>Summing-Point</b>

Until you press **Esc**, the branch-function **w** is permanently activated, if selected once.

A chosen minus-sign has to be placed by mouse click on any branch.

Junctions could be placed, but they are not necessary for the function.

Summing-Points and junctions can be placed only on begin or end of a branch segment.

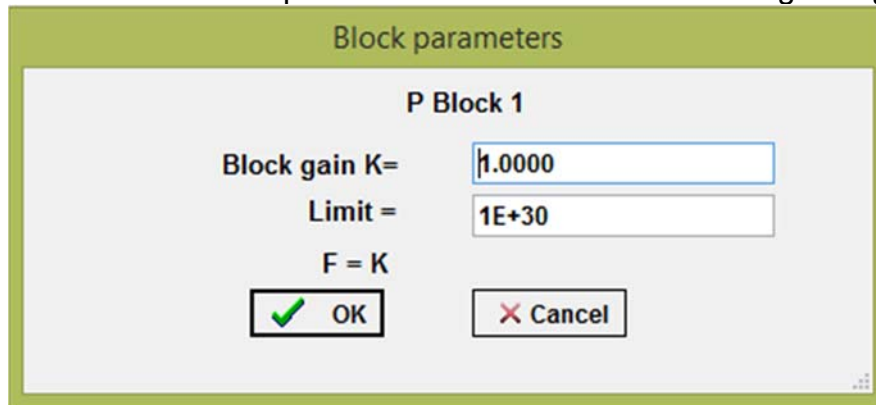
If you want to delete blocks, branches or summing points, just simply select it with mouse click and after selection press the “delete key” on your keyboard. With same method you can shift the elements to other positions.

In opposite of the former Regdelph- program (there you had used drag- drop- method) you can place a systems function block by shortly clicking on the desired block from the left side tool-bar with the left mouse key, and then move mouse (without pressing left mouse key) to the desired position on the Signal block diagram (SBD). Then click the left mouse key a second time to place the block.

In the following text the dialogue-boxes of all available function-blocks will be described.

#### 4.1 P-Block

After selection and placement of a P-block the following dialogue-box appears:



In the blue focused block gain k-area you can type in the value for K.

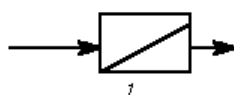
The value in the limit - textbox is default set to 1.0E+30 (infinity). If you want to have a limitation behavior of output signal type your limitation value into this field. Only a symmetrical limitation to + value and – value is implemented.

#### Basics for all Dialogue-Boxes:

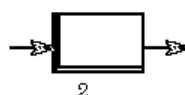
Your input values will be accepted with MouseClickLeft = MCL on the OK-Button. The decimal separation can be a comma (,) or a point / dot (.). Avoid thousand separators.

#### 4.2 I-Block, D-Block

The dialogue-windows of I- and D-block are similar to the P-Block-window.



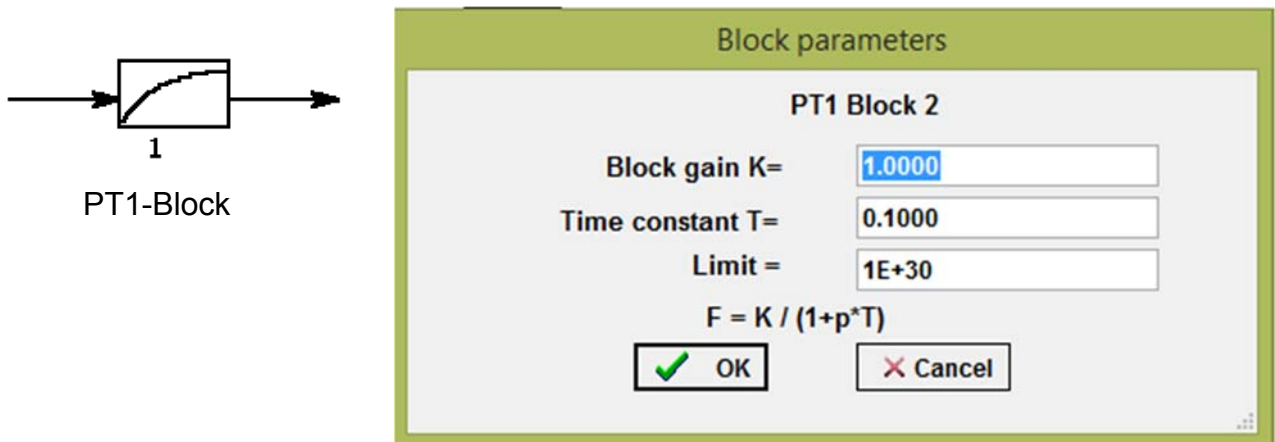
I-Block



D-Block

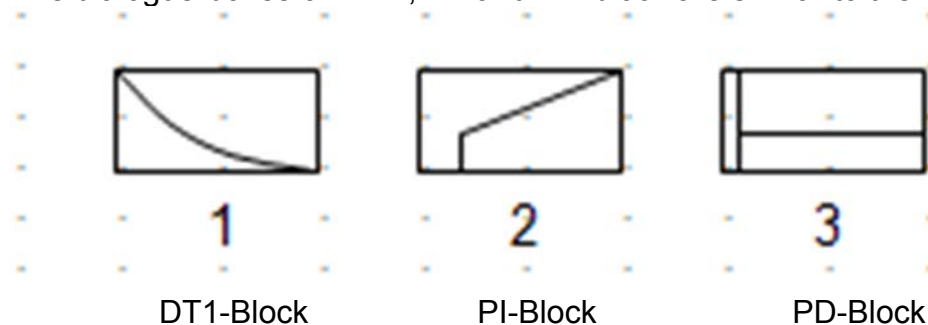
### 4.3 PT1-Block

After selection and placement of a PT1-block the following dialogue-box appears:



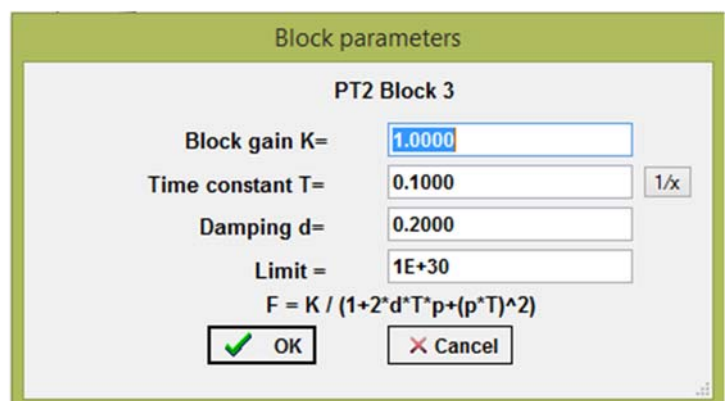
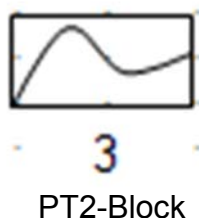
### 4.4 DT1-Block, PI-Block, PD-Block

The dialogue-boxes of DT1-, PI- and PD-block are similar to the PT1-Block-box.



### 4.5 PT2-Block

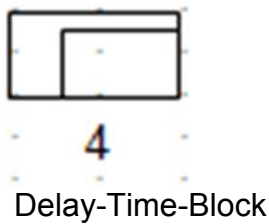
After selection and placement of a PT2-block the following dialogue-box appears:



With the invert- button "1/x" behind the time constant you can invert the number in that text box. That is useful if not the time constant, but the natural radian frequency of PT2 is known, which is  $\omega_0 = 1/T$ .

#### 4.6 Delay-Time -Block

After selection and placement of a Delay-Time-block the following dialogue-box appears:



Block parameters

delay Block 4

Block gain K= 1.0000

Time constant T= 0.1000

Limit = 1E+30

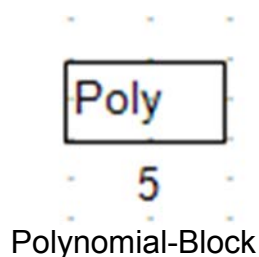
$F = K \cdot \exp(-p \cdot T)$

OK Cancel

#### 4.7 Polynomial-Block (Transfer function block)

If the frequency response of a block is given as a "numerator/denominator polynomial" called complex transfer function, you are able to define or load a polynomial-block up to 5<sup>th</sup> order.

After selection and placement of a Polynomial-block the following dialogue-box appears:



frequency response polynomial

order	denominator	numerator
0	1.0000	1.0000
1	0.1000	xx
2	0.01	xx

Limit 1E+30

OK Cancel

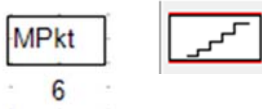
#### Hints to Polynomial-Block:

With this block it's possible to create frequency response equations with a numerator- and denominator-order up to five. Single coefficients may be zero, except the **highest denominator order**. Because, calculating the step response, we have to divide with the highest denominator coefficient, this one should not be unusually small. Set of "xx" instead of a coefficient will break the polynomial progression at this part. At most applications the numerator order is not allowed to be higher than the denominator order. You can load and store ASCII- files with extension .ufk containing degree and coefficients.

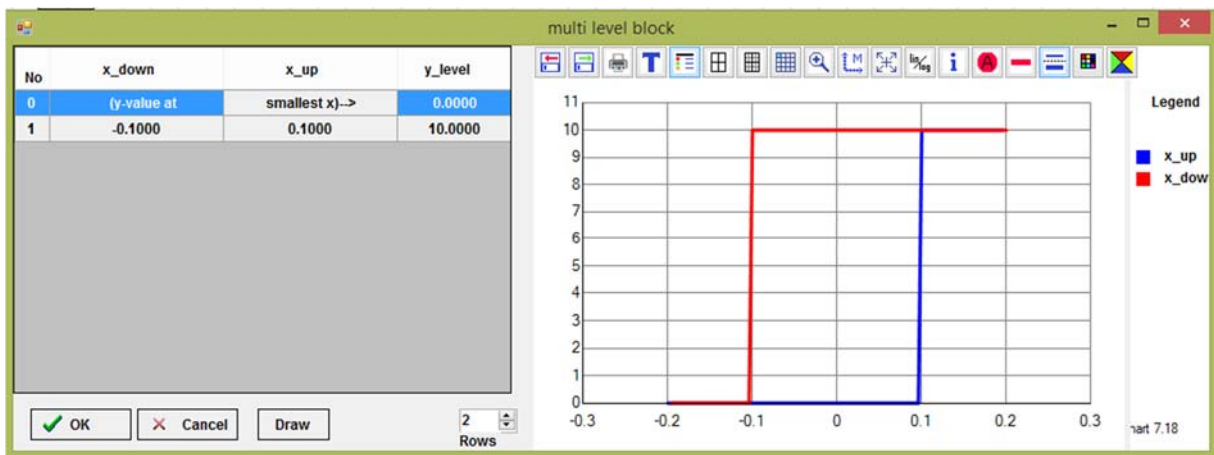


#### 4.8 Multi Level Block:

After selection and placement of a Multi Level Block the following dialogue-box will appear:



Multi Level Block,  
left symbol in SBD,  
right symbol in tool  
bar.



After input of the values, the transfer line could be displayed with “Draw”- button.

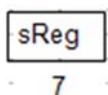
#### **Hints to the Multi Level Block**

Y-level means the y-value at strongly negative x. The transfer line must rise. At hysteresis,  $x_{up}$  has to be more positive than  $x_{down}$ ; otherwise both values will set equal. If no hysteresis is present, you can type in non-numerical signs for  $x_{up}$ , the program will replace the signs by the values of  $x_{down}$ . Non-numerical signs for  $x_{down}$  or y will end the input (xx preset). The maximum number of points is 13. For this block no sort program is available. If frequency responses should be calculated, a replacement for the steady state factor of this block has to be typed in. Handling will be as P-Block.

With this block you can realize two- and three-step-controller. Controller design for these controllers is not supported.

#### 4.9 PIDT1- Controller

Additional to the control system blocks, it is possible to load standard serial controller blocks.



PIDT1- Controller left symbol in SBD, right symbol in tool bar



#### 4.9.1 Controller-Type

PI-, PIDT1-, I-, P- and PDT1-controllers are available. PIDT1-controllers can be converted with the PIDT1- converter. The D-part is defined with its gain = stepdepth =st.

I- and P-controllers will be realized by set Tn to very small or very high values. At the I-controller the integration gain is used as gain.

**Controller, serial structure**

**Controller, block no 7**

proportional gain Kc: 1

PI-time constant Tn: 0.1

D-time constant Tv: 0

D-gain (stepdepth) st: 1

limit: 1E+30 ☒ antiwindup

Transfer function  $F = Kc \cdot (1 + p \cdot Tn) \cdot (1 + p \cdot Tv) / (p \cdot Tn \cdot (1 + p \cdot Tv / st))$

PIDT1- Converter Cancel OK

#### 4.9.2 PIDT1-Converter

After click on this button following windows opens:

**Calculation and conversion of PIDT1-coefficients**

**Block form**  

$$F(p) = K_R \frac{(1 + pT_N)(1 + pT_D)}{pT_N(1 + pT_I)}$$
 KR: 1 TN: 0.1 s Tv: 0 s st: 1 s

**Parallel form**  

$$F(p) = K \frac{1 + 1/pT_I + pT_D}{1 + pT_I}$$
 K: 1 Ti: 0.1 s TD: 0 s T1: 0 s

**Wiki - P + I + DT1- Form**  

$$F(p) = K_a \left( 1 + \frac{1}{pT_a} + \frac{pT_b}{1 + pT_i} \right)$$
 Ka: 1 Ta: 0.1 s Tb: 0 s Ti: 0 s

**Digital controller**  
 Sampling time T0: 0.01 s  
 recursive algorithm:  

$$y(n) = p1 \cdot y(n-1) + p2 \cdot y(n-2) + q0 \cdot x(n) + q1 \cdot x(n-1) + q2 \cdot x(n-2)$$
 p1: 0 q0: 1.05 p2: 1 q1: 0.1 q2: -0.95  
 st effective: 1.05  
 methods see paper  
☐ I. simple PID Approach  
☒ IV Trapezoidal Approach  
☐ Antiwindup

STO PIDT1 Save ZK  
 RCL PIDT1 RCL ZK  
 to report recalculate  
 Copy Close

**Circuit**  
  
 RC-Values:  
 R1: 100 kOhm  
 C2: 1.000 µF  
 R2: 100.000 kOhm  
 C1: 0.000 µF  
 R3: 0.000 kOhm

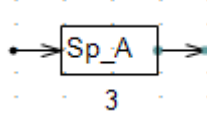
You can convert the different types of PID (block form / parallel form) into the other forms. The same window is available in WindfCsharp. You can exchange PID- values between these program and other modules by using the STO and the RCL – buttons. The data is then stored or restored into / from registry. Also a digital PIDT1 is calculated either in the simple form (design I) ignoring the stepdepth st or with the design IV method described in the CSII- paper. See next table:

PID	$K(1 + pT_D + 1/pT_I) =$	$q_0 = K(1 + T_D/T_0 + T_0/T_i),$ $q_1 = -K(2T_D/T_0 + 1), q_2 = KT_D/T_0, p_1 = 1,$ $st \approx 1 + T_V/T_0$	recursive design I
-----	--------------------------	---	--------------------

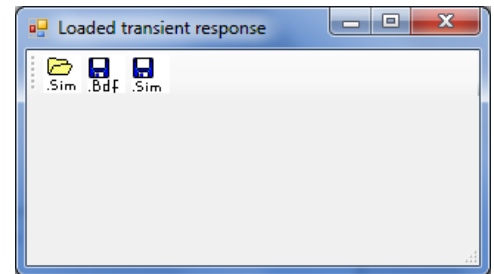
	$\frac{K_R(1+pT_N)(1+pT_V)}{pT_N}$		
PIDT1	$\frac{K(1+pT_D+1/pT_i)/(1+pT_1)}{pT_N(1+pT_1)}$ $\text{st}_{\max} = (T_N+T_V)/T_N+T_V/T_0,$ $\text{st}_{\text{analog}} = T_V/T_1$	$q_0 = b(1+2\frac{T_N}{T_0})(1+2\frac{T_V}{T_0}), \quad b = \frac{K_R T_0}{2T_N(1+2T_1/T_0)}$ $q_1 = b \left( (1-2\frac{T_N}{T_0})(1+2\frac{T_V}{T_0}) + (1+2\frac{T_N}{T_0})(1-2\frac{T_V}{T_0}) \right)$ $q_2 = b(1-2\frac{T_N}{T_0})(1-2\frac{T_V}{T_0}),$ $p_1 = \frac{4T_1/T_0}{(1+2T_1/T_0)} \quad \text{and} \quad p_2 = 1 - p_1$	recursive design IV trapezoidal HF-gains are equal

#### 4.10 Loaded step response block

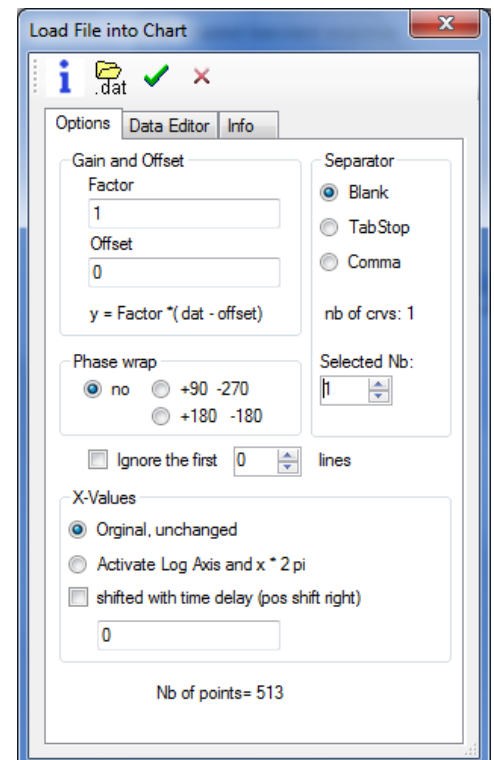
For inserting a step response-file in your signal block diagram editor, you have to use the transient response block, which is displayed in the screenshot on the right hand side.



You get the following screenshot after inserting the block. The first thing you have to do is opening a .sim file by clicking on the “open .sim” button.



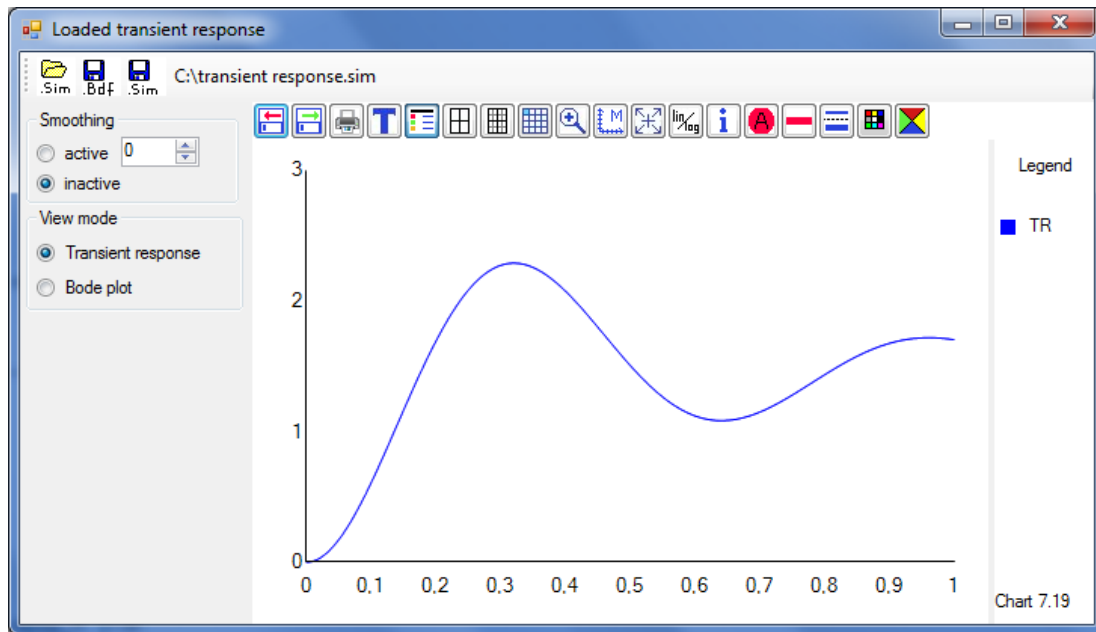
After that, you have to select the .sim file you want to insert. A .sim file contains two columns, the first represents the x-values, the second one represents y-values. The default separator is a space-character. In the next user dialog it is possible to change the settings for the used separator, offset, gain. Note that it is possible to load curves with any point count. When you use default files from Regc# or Windfc#, no changes are needed in this window. You can go on by pressing the green OK button.



The step response will be displayed in the chart. If there is a noise on the signal, you can activate a low pass filter and smooth the curve by changing the value in the numeric up down box within the groupbox “Smoothing”.

Changes of the view mode, step response or bode plot, can be taken on the groupbox “view mode”. For some views you might need to click on the “autoscale” button of chart2D to see the whole curve.

You can save the displayed bode plot by clicking on the “save .bdf file” button. If you want to save the transient response, click on the “save.sim” button.

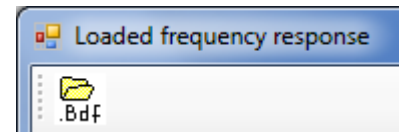


#### 4.11 Loaded Bodeplot

For inserting a frequency response-file in your signal block diagram editor, you have to use the frequency response block, which is displayed in the screenshot on the right hand side.



You get the following screenshot after inserting the block. The first thing you have to do is opening a .bdf file by clicking on the "open .bdf" button.



After that, you have to select the .bdf file you want to insert. A .bdf file contains three columns, the first column contains the values from the x axis, the second column contains magnitude values and the third column contains phase values. The default separator is a space-character. In the next user dialog it is possible to change the settings for the used separator, offset, gain, etc... Regc# uses omega values for calculation, so as default the frequency values are multiplied by  $2 \cdot \pi$ . If you don't want that, you can set it off in the groupbox "X-Values". Note that it is possible to load curves with any point count. When you use default files from Regc# or Windfc#, no changes are needed in this window. You can go on by pressing the green OK button.

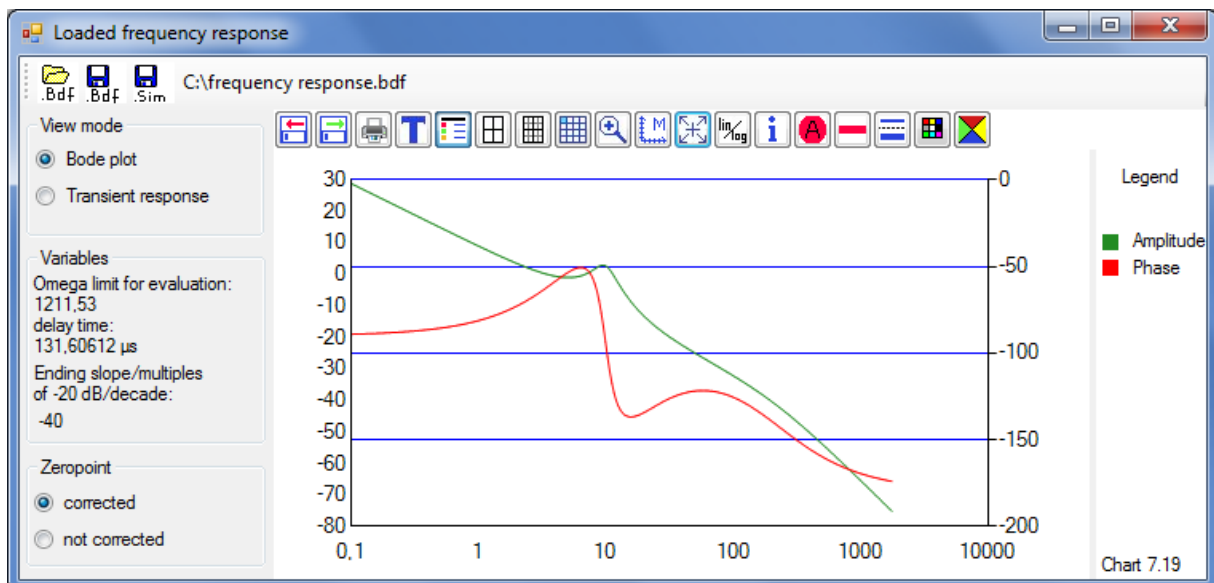
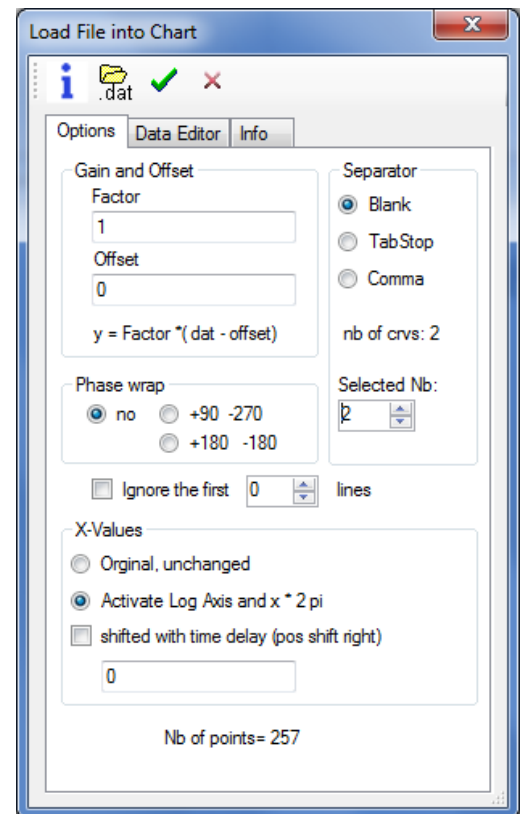
The frequency response will be displayed in the chart. The y-axis on the left-hand side represents the amplitude values, the y-axis on the left hand side represents the phase values.

This window also shows characteristic variables in the groupbox “variables”.

Changes of the view mode, step response or bode plot, can be taken on the groupbox “view mode”. For some views you might need to click on the “autoscale” button of chart2D to see the whole curve.

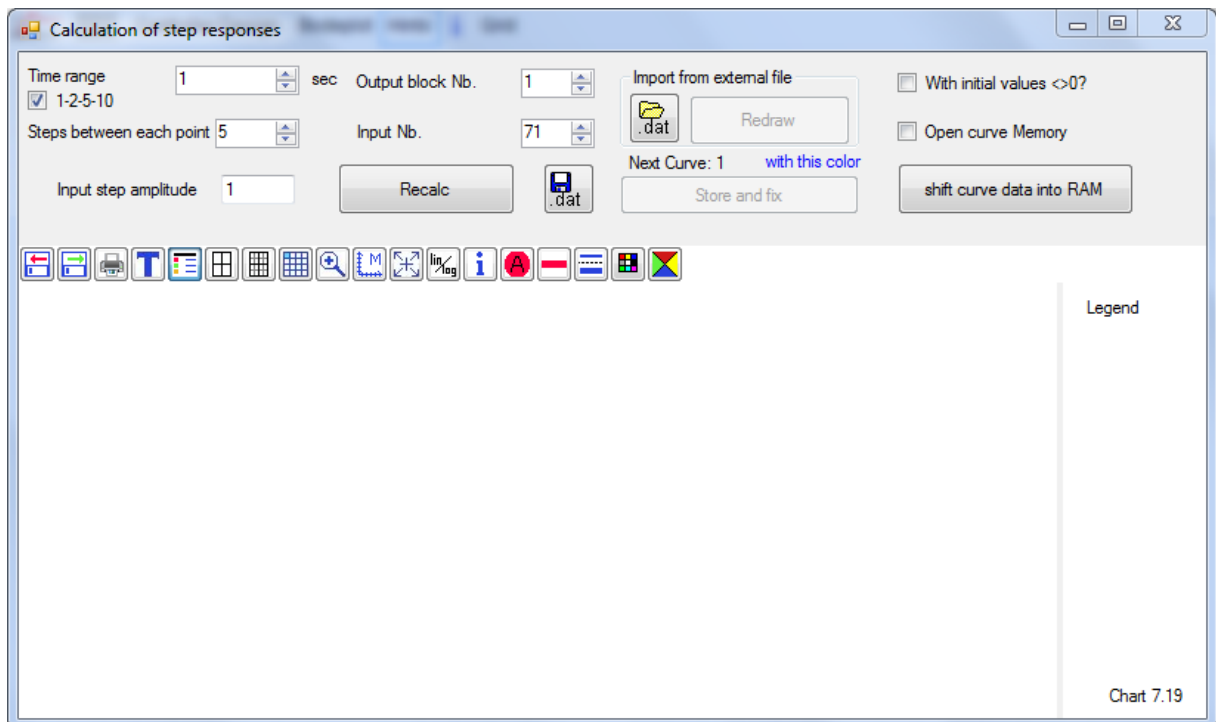
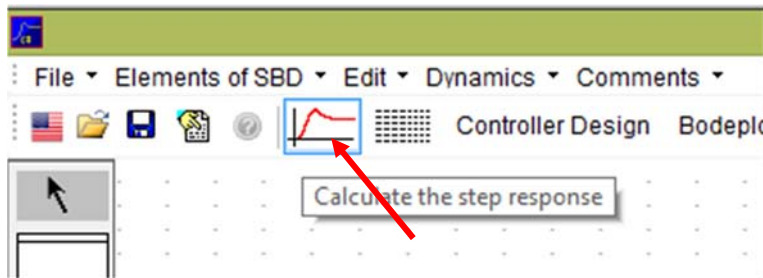
The resulting step response could have an offset. The default settings are that the offset is removed. You can deactivate this by checking the radiobutton “Zeropoint not corrected”.

You can save the displayed bode plot by clicking on the “save .bdf file” button. If you want to save the transient response, click on the “save.sim” button.



## 5.1 Step Response Window

By pressing on the button “calculate the step response”, which is red marked in the screenshot on the right hand side, the “Calculation of step response” window appears.



For simply display the current step response, click on the “Recalc” button.

In this window you can select the output block and input number which are the same numbers as the ones in the signal block diagram window. Regard the following screenshot.

The legend of the curve also displays the block and input number at the right hand side of the graph.

The settings of the time axis of the plot can be modified by changing the value “time range” at the upper left corner. For a better use, the value toggles between 1,2,5 and 10. If you don’t want that, you can switch it off by clicking on the checkbox next to the label. The value of the input step amplitude can be modified in the textbox “Input step amplitude”.

When changing the values in these boxes, the “Recalc” button must be pressed to refresh the curve.

The displayed curve can be stored as a .sim file by clicking on the disc button. The resulting file contains 2 columns, which are separated by a space character. The first column contains the values from the t axis, the second value contains values from y-axis.

Multiple curves can be displayed by clicking on the button “store and fix”. After that, you can add a second curve by pressing “Recalc”. Up to 10 curves can be displayed simultaneously

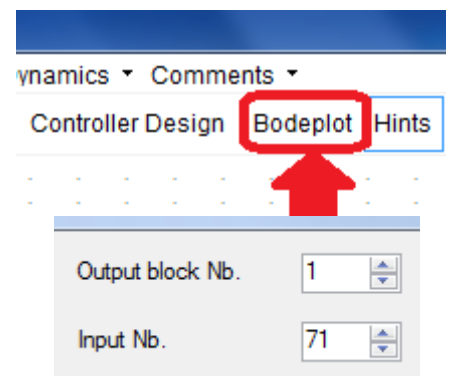
For comparing the displayed curve to others, external curves can be loaded into the diagram by clicking on the button “Import from external file”.

By activating the “initial values unequal to zero” checkbox, you can set initial values for each block in the upcoming table on the right hand side. After inserting the new values, you have to click on the button “Recalc”.

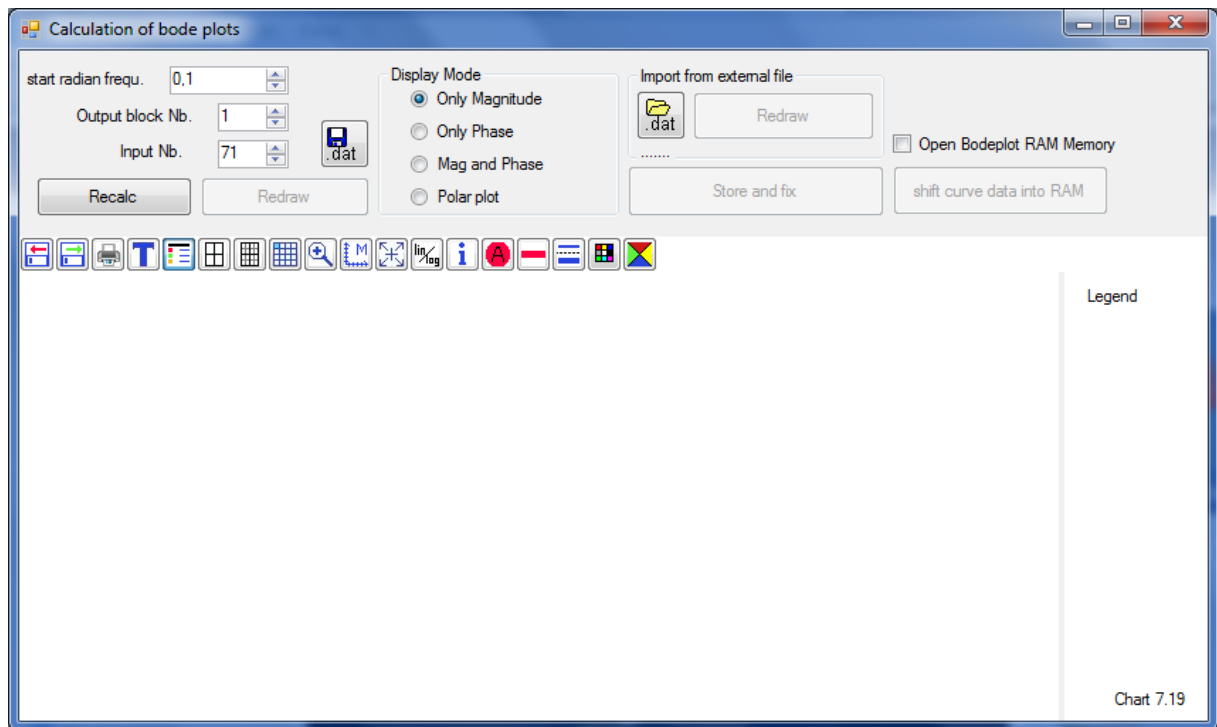
The button “shift curve data into RAM” stores the current curve in RAM. When the checkbox “open curve memory” is checked, all stored curves in RAM are displayed. When you press on the button RCL, the stored curve is displayed on the screen.

## 5.2 Bodeplot Window

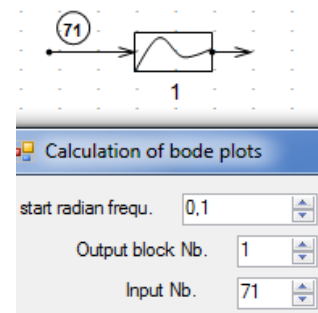
By pressing on the button “Bodeplot”, which is red marked in the screenshot on the right hand side, the “Calculation of bode plots” window appears.







In this window you can select the output block and input number which are the same numbers as the ones in the signal block diagram window. See example in the right screenshot.



The legend of the curve also displays the block and input number at the right hand side of the graph.

The settings of the frequency axis of the plot can be modified by changing the value "start radian frequ." at the upper left corner. The next five frequency decades from this value will set the frequency axis of the plot.

By clicking on "Recalc", the bodeplot will be calculated and displayed in the chart.

The legend of the curve on the right hand side displays the block and input of the plot.

The selected radiobutton in the groupbox "Display Mode" determines the view of the plot. You can change between magnitude, phase, magnitude + phase and polar plot. After selecting a radiobutton, you have to click on the button "Redraw", and the new curve appears on the window.

The displayed curve can be stored as a .bdf file by clicking on the disc button. The resulting file contains 3 columns, which are separated by a space character. The first column contains the values from the x axis, the second column contains magnitude values and the third column contains phase values.

Multiple curves can be displayed by clicking on the button "store and fix". After that, you can add a new curve by pressing "Recalc". Up to 10 curves can be displayed simultaneously.

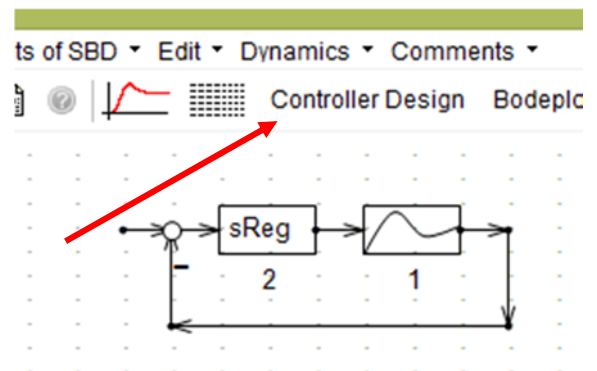
For comparing the displayed curve to others, external curves can be loaded into the diagram by clicking on the button "Import from external file".

The button "shift curve data into RAM" stores the current curve in RAM. When the checkbox "open curve memory" is checked, all stored curves in RAM are displayed. When you press on the button RCL, the stored curve is displayed on the screen.

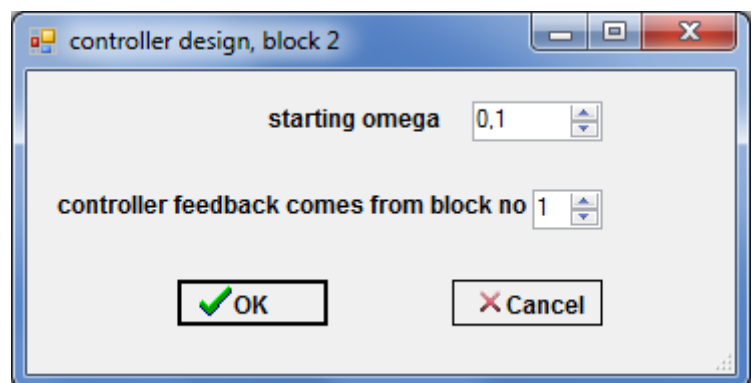


### 5.3 Controller Design

A controller can be designed by Regc# by clicking on “Controller Design”. Make sure, that you have placed the “sReg” block in a control loop. The screenshot on the right hand side shows an example for a correct control loop.

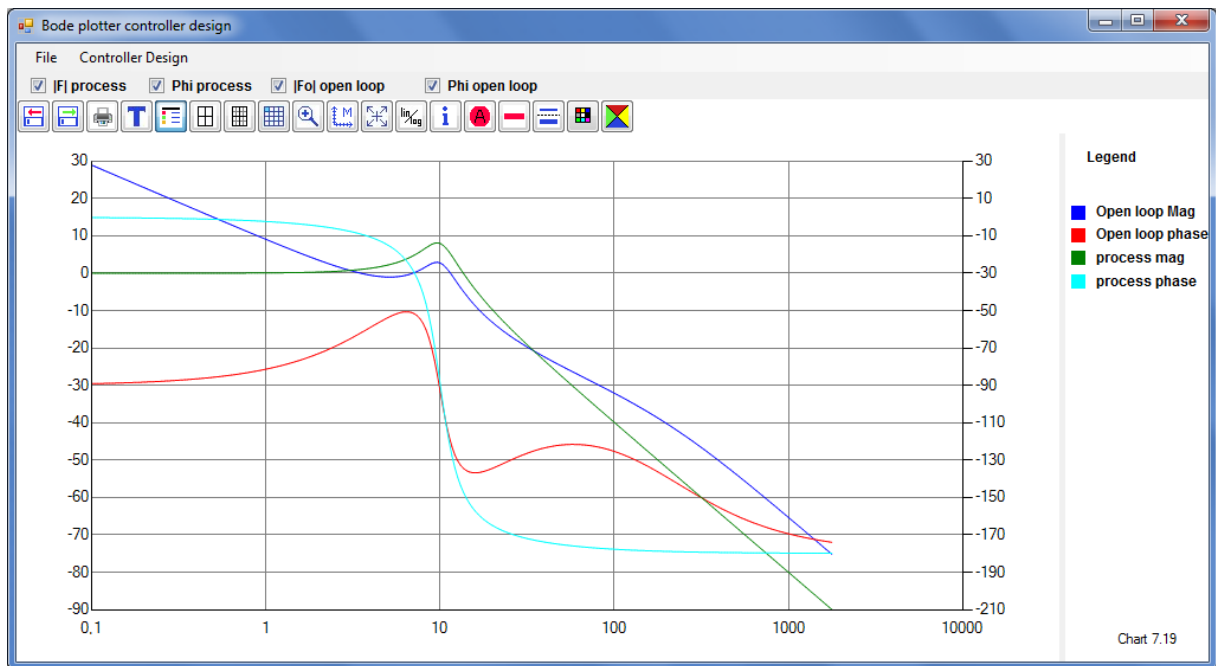


After clicking on “Controller Design”, the following window appears. The software automatically detects the block number of the controller in the headline. Here settings for the starting omega and the controller feedback can be changed.



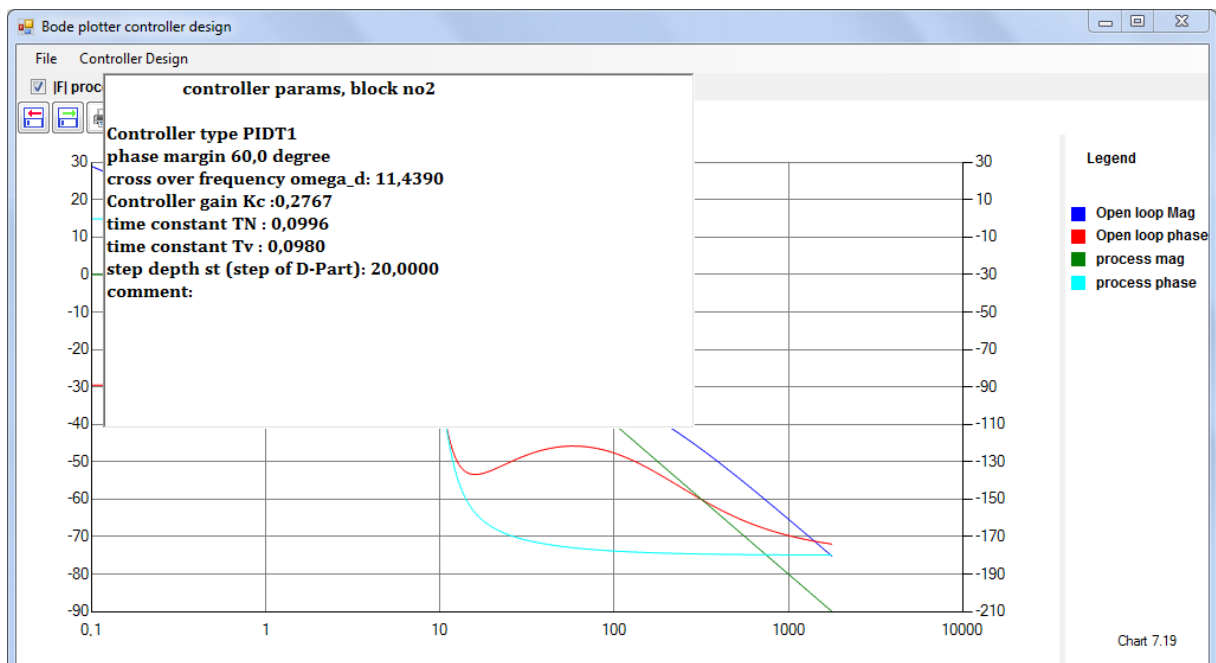
The next step is to choose a value for the D-part. This depends on the process characteristics. For this example, a PT2 bloc is used without I-part. When you want to design a controller with D-Part, select the checkbox. After that, a groupbox appears where you get a recommendation for the D-time constant value and stepdepth. When you want to use the proposed value, make sure to click on the downward arrow. After that, activate the next window with a click on OK.

In the upcoming window you can set the TN time constant, the desired phase margin, the use of symmetrical optimum and the choice of the controller type. Make also sure to press on the downward arrow if you want to use the proposal TN with -12 dB-Method. In this example a PIDT1 controller will be designed, so after clicking on "PIDT1" a chart with the result will be displayed in a new window.




In this chart you can see 4 curves of open loop magnitude and phase and of the process magnitude and phase. By using the checkboxes, you can make the curves visible or invisible. If you want to save the curves you can click on the menu item "store as ASCII file", which is in the dropdown menu of the menu item file. The controller parameter can be changed by using the menuentries of the menuitem "controller design".

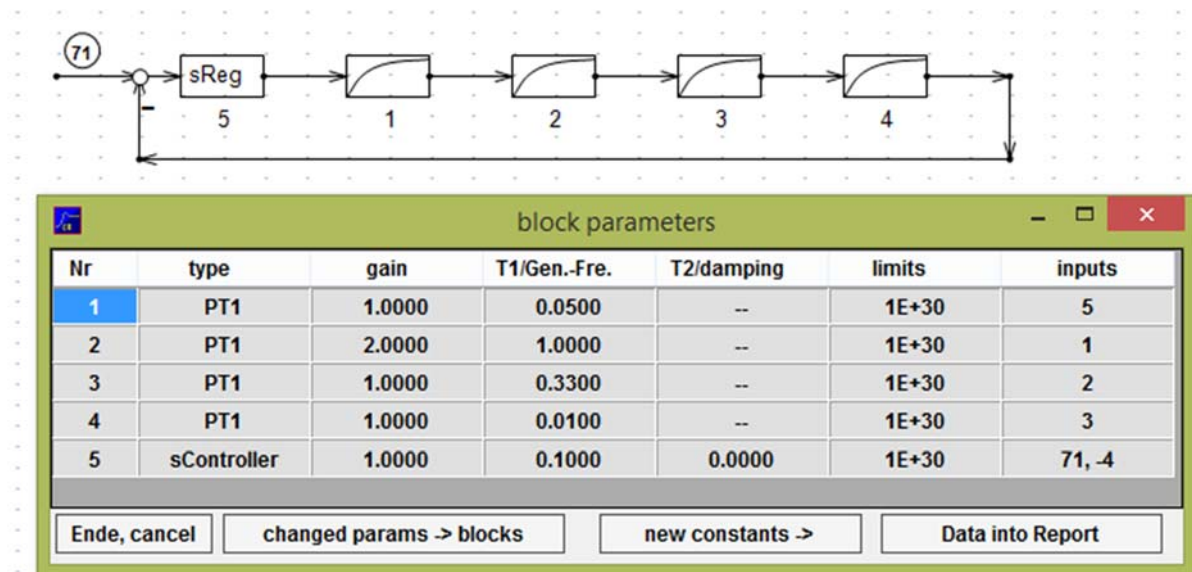
When you close the form, all controller parameters will be displayed on top of the chart, like it is in the following screenshot.



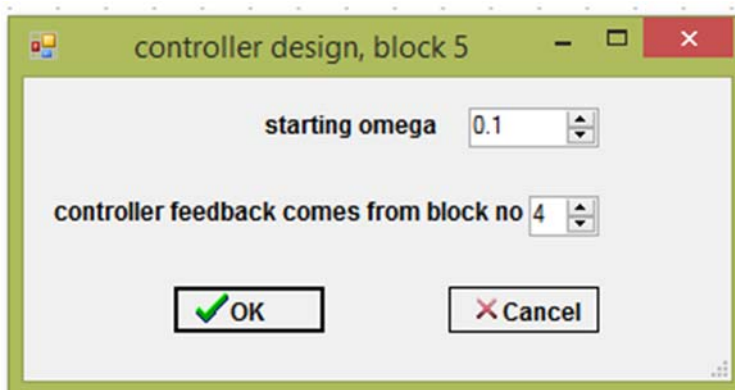
After closing this window the controller is designed and you can calculate the reference step response or other things.

### 5.4 Example of Controller design (see Workbook)

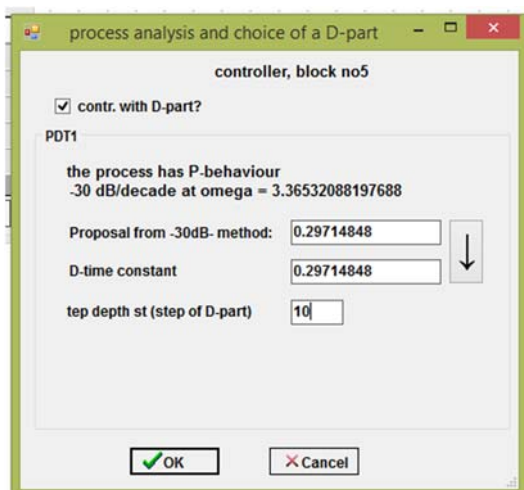
The 4 PT1- Example of the lecture is now used as an example. With the button  the parameter settings could be displayed in a table.



Now a PIDT1- controller should be designed. Phase margin is  $60^\circ$ , the proposed values of the controller time constants calculated with the -12dB and the -30 dB- methods should be used. Stepdepth should be 10. Then you get the following screenshots:



Click OK.




Click, checkbox "with D- part", accept the proposed value with the arrow- button and change stepdepth to 10. Then OK.

controller type and phase margin'

**controller, block no 5**

gain gradient -12 dB/dec at omega = 1.1600  
this omega gives a good approximation for the compensation of the largest time constant

Proposal TN with -12dB-Method  

PI-time constant

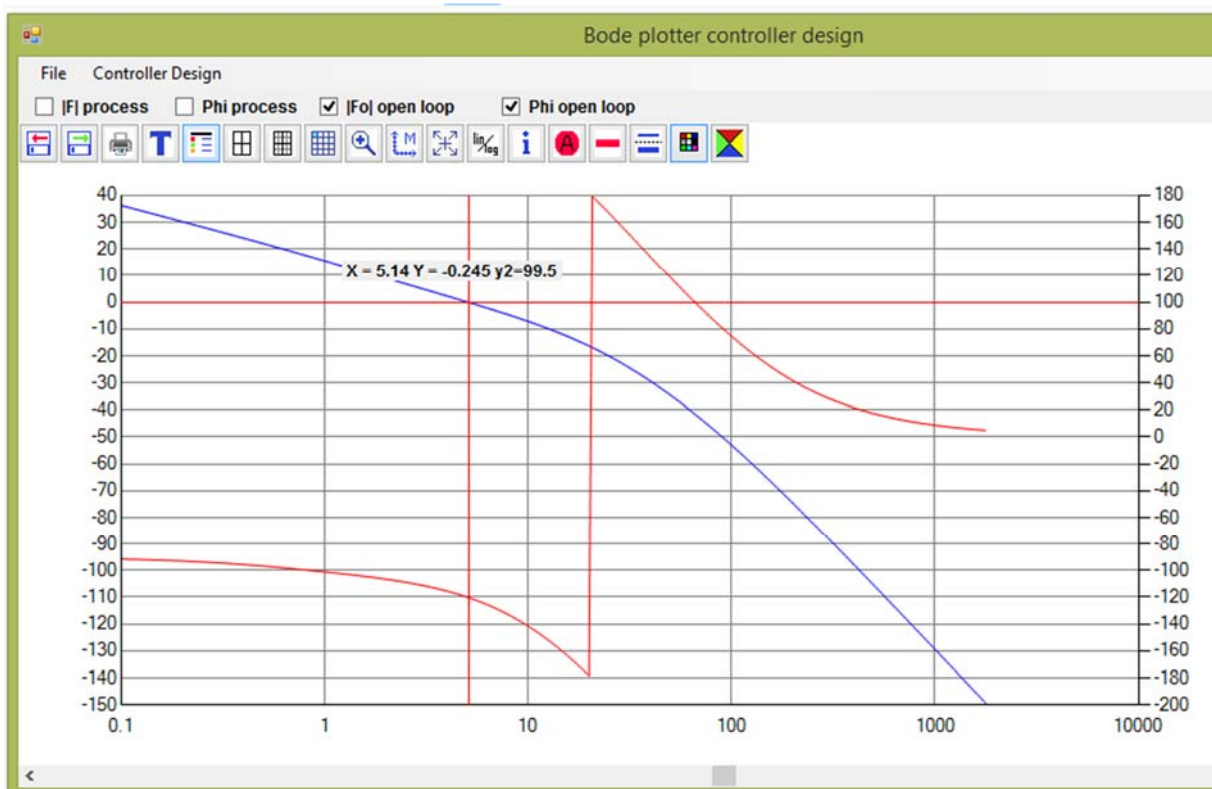
☐ symmetrical optimum?

phase margin (degree)

**controller type**

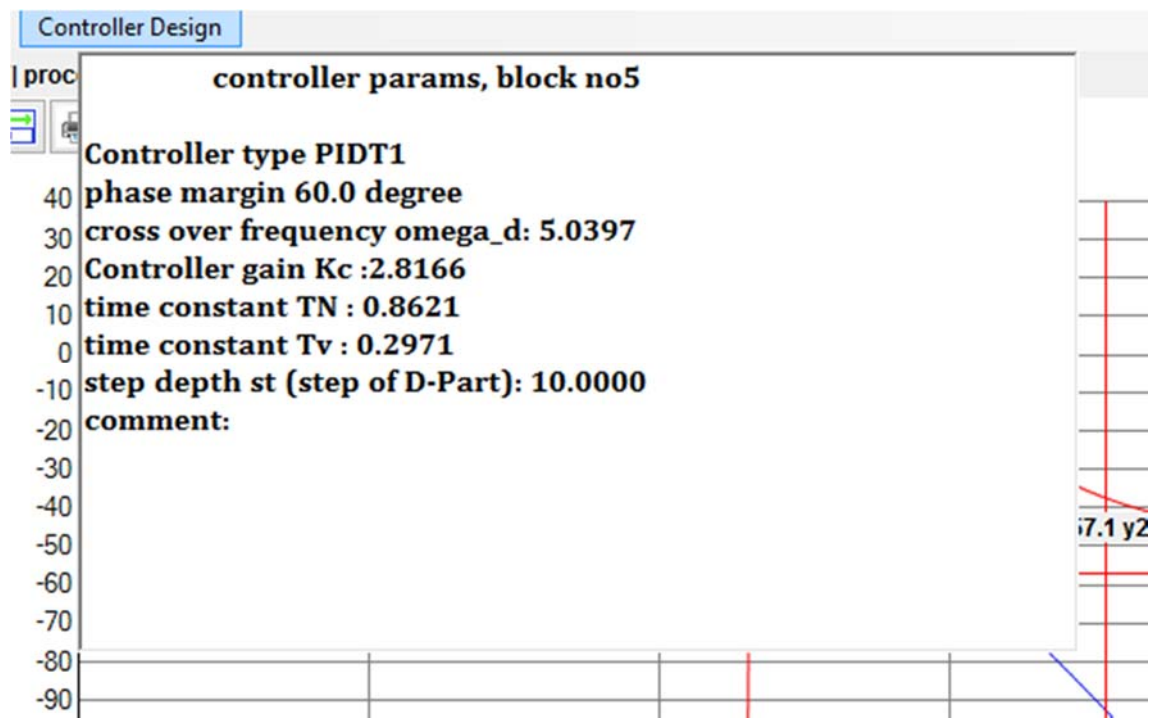
Accept proposed TN- value and change if necessary phase margin and click on PIDT1- button.

Result (with right mouse button activated coord- display at wd):

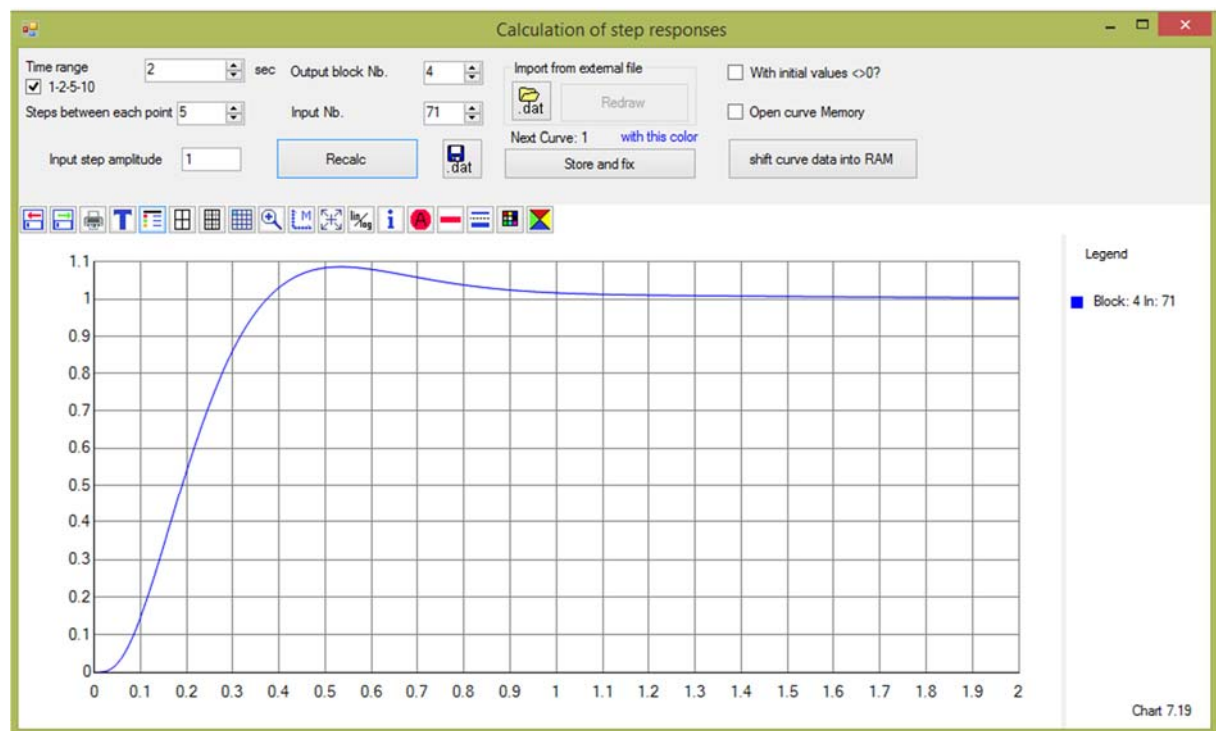


wd is at 5.14, there the open loop phase curve (red) crosses the -120°- value which results in the desired phase margin value of 60°.

After clicking on "close window button"  you will get the parameters:

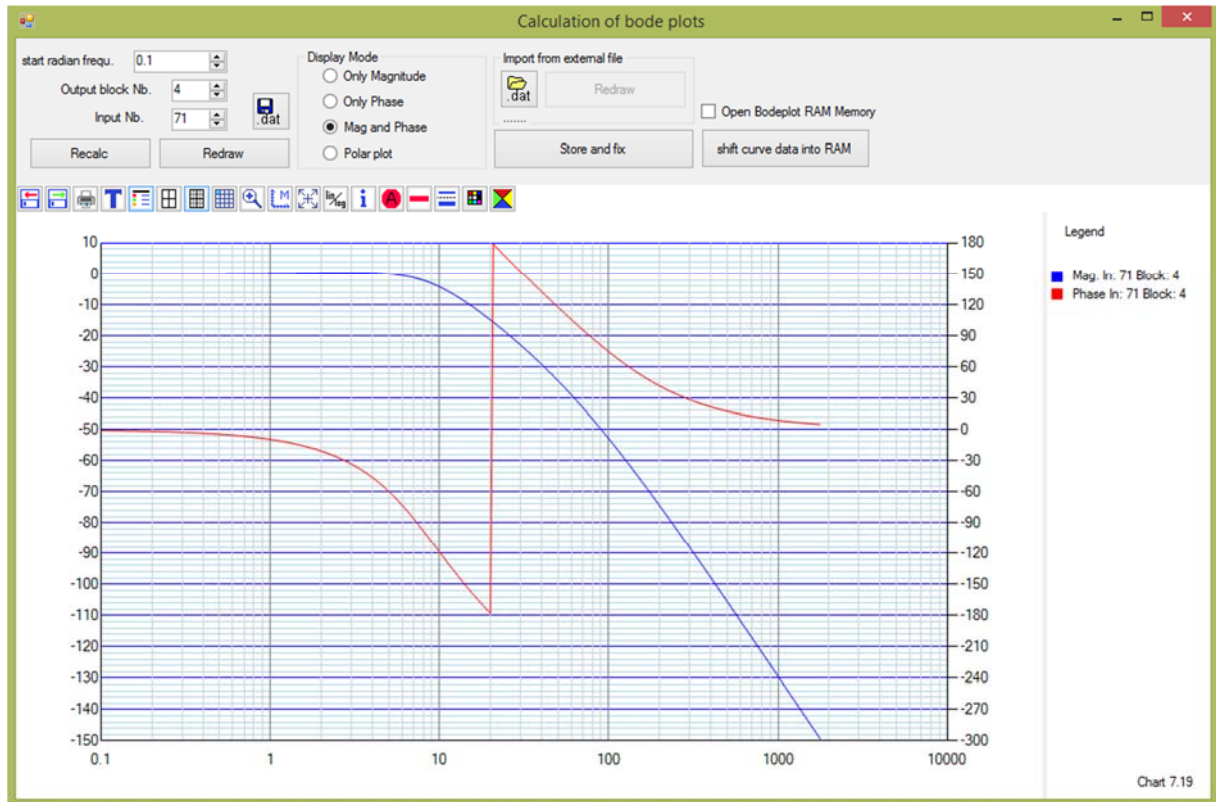


These values are automatically stored into the report page (see Chap. 5.5).  
The reference step response looks like this:




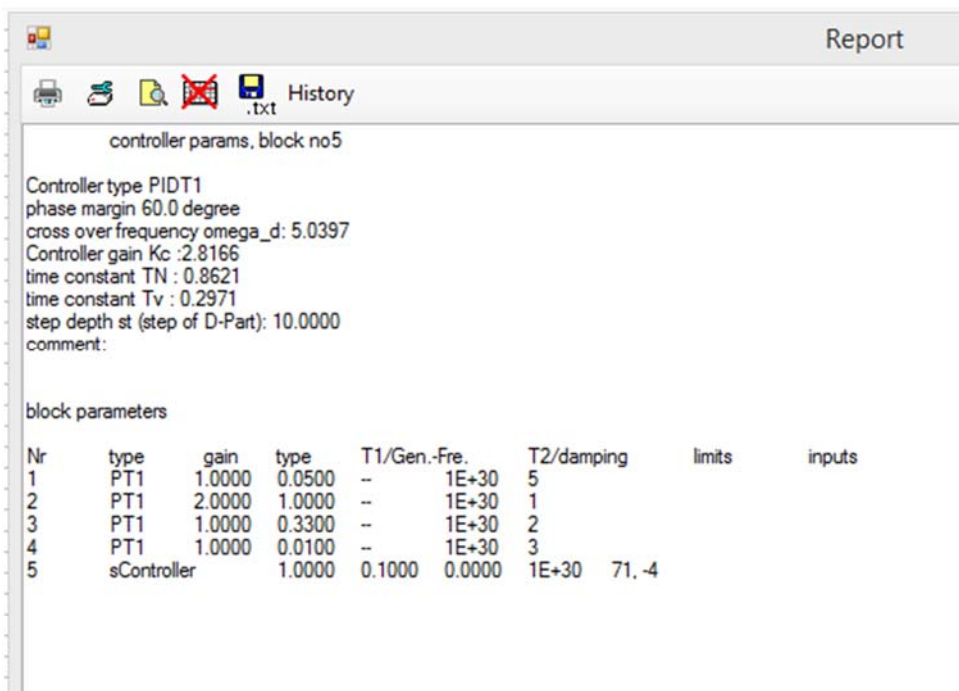
The closed loop bode plot looks like this:





## 5.5 Report window

With the report button  in the toolbar in main window you can open the report page. There results are stored as text information. Some other pages have report buttons to store calculated results, the controller results are stored automatically. After clicking on the button "Data into report" on the block parameter box you can see this screen:



Finally you can print, store and clear this text. With click on "history" you can see the actual history file of RegC#.