

KISSsoft AG - ☎ +41 55 254 20 50
 Uetzikon 4 - ☎ +41 55 254 20 51
 8634 Hombrechtikon - ✉ info@KISSsoft.AG
 Switzerland - 🌐 www.KISSsoft.AG

KISSsys Instructions:

Load Case Table for Standard Load Cases

1 Introduction

Typically, a gearbox is recalculated for several different load cases (e.g. normal use, nominal power, overload case etc.). It is possible to enter the corresponding speed / torques directly as a load, but it is more elegant to create a table of variables with every load case stored in it. The user can then choose the load case he wishes to calculate. The corresponding load data is automatically read from the table and included in the model.

2 Solution

The different load cases (composed by name, rotation speed and torque) are saved in a table using the variables *Loadcase*, *Speed* and *Torque*. These variables are read by dialog windows and they are of type “Variant of Real”, which is similar to an Array.

3 Way to the Solution

3.1 Transfer from the Templates to the Tree Structure

The installation of KISSsoft/KISSsys contains a file *templates.ks* as a standard. This file can be opened by the administrator (click on *Options* and *Administrator*) with a click on *File* and then *Open templates file*. The browser appears and the file *templates.ks* can be opened. A second templates file, *templates-loadcases.ks*, contains the here described extension. This template is available from KISSsoft support and contains only this table. It has to be copied into the installation directory and opened in the same manner as the original templates *templates.ks*. The element *Loadcases* has now to be copied into the tree structure, it is in this example placed under the element *System*.

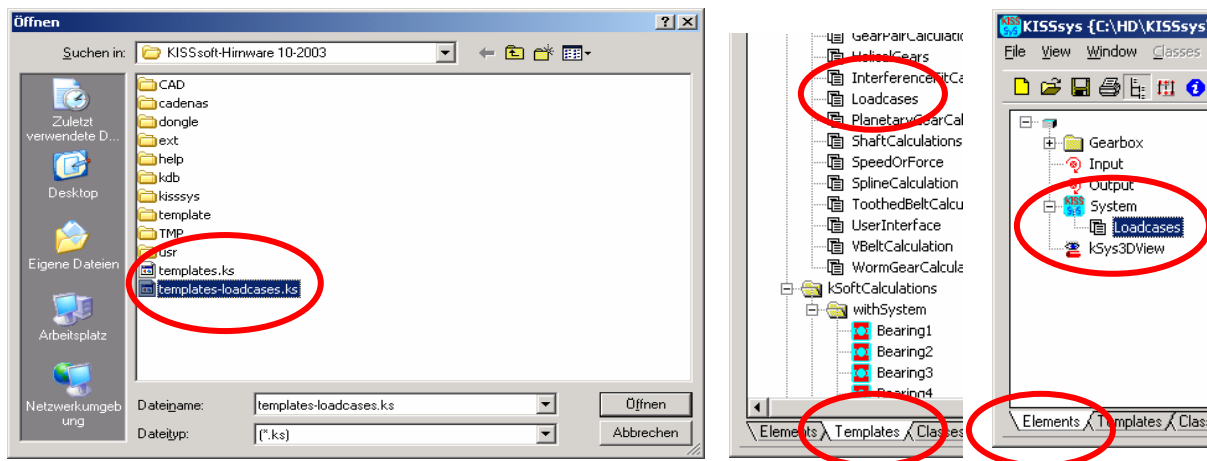


Figure 3-1 Left: Load special templates Right: Copy the loadcases in the tree structure.

3.2 Elements of the Table *Loadcases*

Use right mouse click on *Loadcases*, then *Poperties* and see the variables of the table defined as follows:

Variable	Type	Description
ActiveLoadcase	Real	Number of the active load case (From 0 to „NofLoadcases“)
Dialog	Function	Is not used
Loadcase	Variant of String	Name of the load case
ModifyLoadcase	Function	Changes a defined load case
NewLoadcase	Function	Defines a new load case
NofLoadcases	Real	Number of the different load cases in the table
OnLoadFile	Function	Intern KISSsys function
SelectLoadcase	Function	Chooses a load case
Speed	Variant of Real	Rotation speed
Torque	Variant of Real	Torque
defintion	Array	Definition of the table
kSysPosition	Array	Position of the table on the screen

Display the table using right click on *Loadcases* and *Show*.

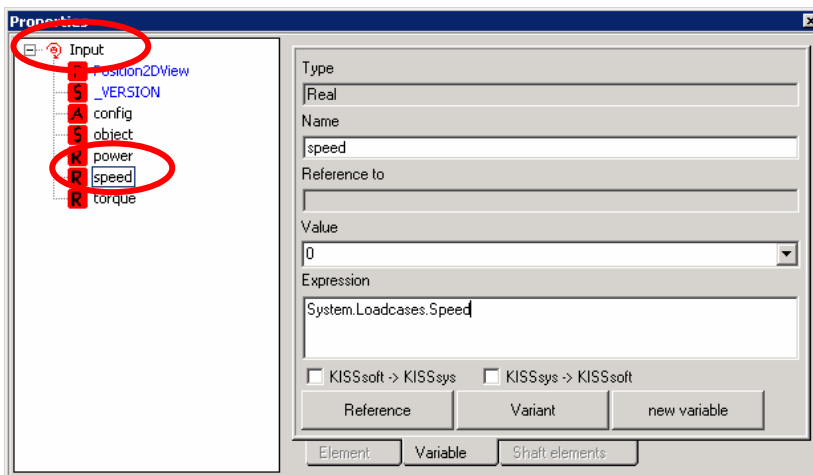
	Loadcase	Speed	Torque
0	Case 1	5	5
1	New Loadcase	1200	-34
2	Case 3	99	99
3	Case 4	0	0
4	Case 5	12	12
5	Overload 1	34	56
6	New Loadcase	1200	-34

Figure 3-2 Load Case Table with the Number of the Load Case on the Left

Is the value of the variable *ActiveLoadcase* equal to 3, the variable *Loadcase* has then the value *Case 4*.

3.3 Link of the Rotation Speed / Torque

A rotation speed and a torque are saved in the load case table. Depending on the application this rotation speed / torque are applied to different or the same load input elements. In this example the KISSsys model contains only two load input elements (*kSysSpeedOrForce*), one for the power input and one for the power output. The rotation speed on the input and the torque on the output should be defined in this example. The element *Input* and the variable *speed* should contain the following expression:

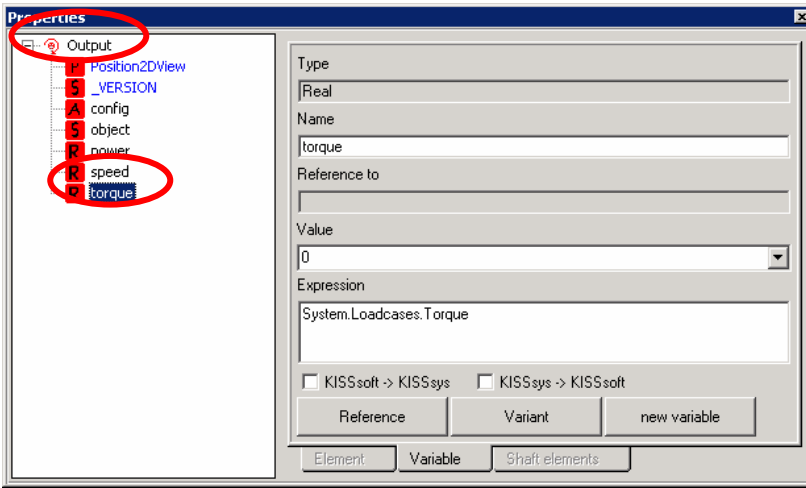


The rotation speed for a load case is saved in the variable *System.Loadcases.Speed*.

The rotation speed on the *Input* receives the value of this variable.

Figure 3-3 Link Input Speed with Load Case Table

The same procedure is used to define the output, here is the torque defined.



The torque for a defined load case is defined in the variable *System.Loadcases.Torque*.

The torque on the *Output* receives the value of this variable.

Figure 3-4 Link of the Output Torque with the Load Case Table

3.4 Integration in a User Interface

The table Loadcases contains three functions which should be called by the user interface. These functions are:

- *SelectLoadcase*
- *ModifyLoadcase*
- *New Loadcase*

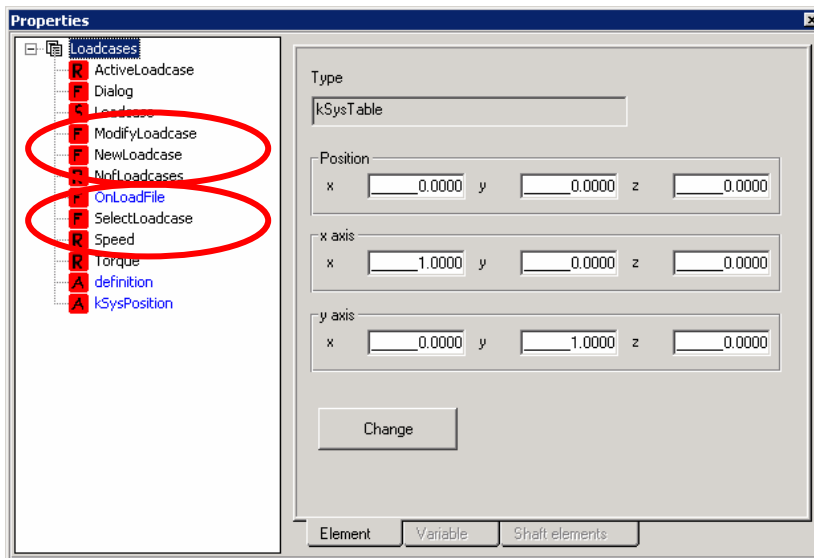


Figure 3-5 Function for the Selection, Creation and Modification of a Load Case

These three functions should be integrated in the user interface. An additional function *Show Loadcases* may be introduced in the user interface, which shows the *Load Case Table*.

Function	Input	Speed	Torque
Select Loadcase			
New Loadcase	Input	34	2.6452
Modify Loadcase	Output	2.6452	56
Show Loadcases			
		Gear 1	Gear 2

Figure 3-6 User Interface with the 4 Function Calls

The single functions are defined as follows:

The dialog box is titled "Change Variable" and has a close button (X) in the top right corner. It contains the following fields and controls:

- Type: Function
- Name: Select Loadcase
- Reference to: (empty)
- Value: (empty)
- Expression: System.Loadcases.SelectLoadcase(); System.calcKinematic(); kSys_Refresh();
- Checkboxes: KISSsoft -> KISSsys, KISSsys -> KISSsoft
- Buttons: Reference, Variant, Ok

Function call for the choice of a new load case.

Special attention has to be paid to the calculation of the kinematics, which should be done right after the choice of the new load case, but not the KISSsoft calculations. The command kSys_Refresh() makes a refresh on the whole model.

The dialog box is titled "Change Variable" and has a close button (X) in the top right corner. It contains the following fields and controls:

- Type: Function
- Name: New Loadcase
- Reference to: (empty)
- Value: (empty)
- Expression: System.Loadcases.NewLoadcase();
- Checkboxes: KISSsoft -> KISSsys, KISSsys -> KISSsoft
- Buttons: Reference, Variant, Ok

Function call for the creation of a new load case.

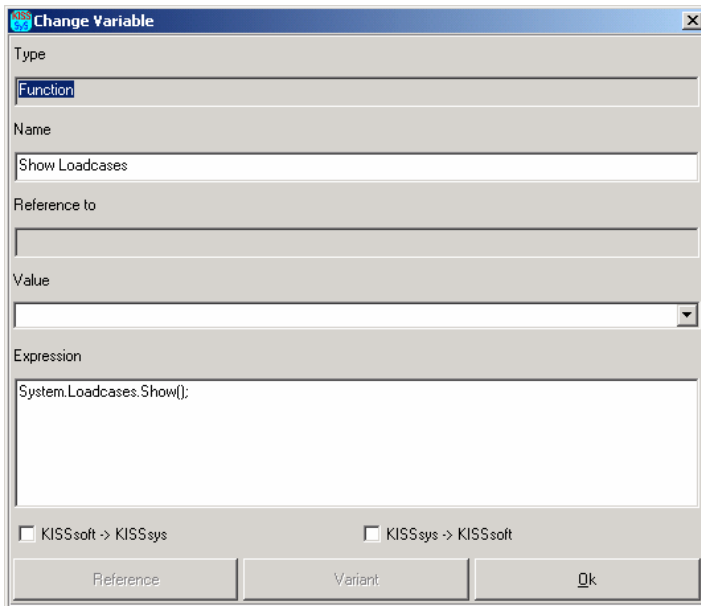
Only the function itself is called.

The dialog box is titled "Change Variable" and has a close button (X) in the top right corner. It contains the following fields and controls:

- Type: Function
- Name: Modify Loadcase
- Reference to: (empty)
- Value: (empty)
- Expression: System.Loadcases.ModifyLoadcase(); System.calcKinematic(); kSys_Refresh();
- Checkboxes: KISSsoft -> KISSsys, KISSsys -> KISSsoft
- Buttons: Reference, Variant, Ok

Function call for the modification of a already existent load case.

Special attention has to be paid to the calculation of the kinematics, which should be done right after the choice of the new load case, but not the KISSsoft calculations. Mainly because the changed load case could be the active load case.



Function call to show the load case table.

3.5 Manipulation

3.5.1 Choice of the Load Case

With a double click on the function *Select Loadcase* in the user interface appears following dialog, in which the desired load case may be selected, press *Ok* to confirm the choice.

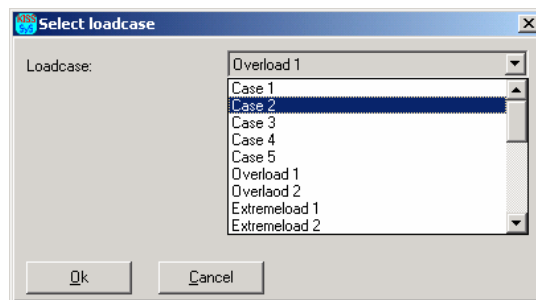


Figure 3-7 Choice of a Load Case

After this manipulation the new load case is used, the values for *speed* and *torque* on the *In-* and *Output* are changed and the calculation of the kinematics is repeated (note that the KISSsoft calculations are not executed yet).

Select Loadcase		Speed	Torque
New Loadcase	Input	5	0.00664
Modify Loadcase	Output	0.00004	5
Show Loadcases			
Active Loadcase			
Case 1	Gear 1		Gear 2
	SF	28.543	29.547
	SH	3.0011	3.3

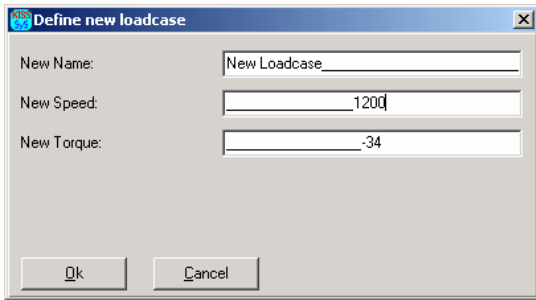
Select Loadcase		Speed	Torque
New Loadcase	Input	22	0.00665
Modify Loadcase	Output	0.0072	2
Show Loadcases			
Active Loadcase			
Case 2	Gear 1		Gear 2
	SF	28.543	29.547
	SH	3.0011	3.3

Figure 3-8 Changed Load Values after the Choice of a new Load Case

If the dialog is canceled, nothing is changed.

3.5.2 Create a New Load Case

Double click on the function *New Loadcase* in the user interface starts the following dialog, where the new load case can be defined.



Enter the name for the new load case. The default value of that field is a string *No Name*.

Enter the torque and the speed.

Attention: use + or – to define the sense of rotation or the direction of the torque vector.

The entry is confirmed and the load case table extended:

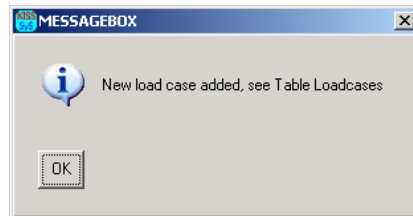


Figure 3-9 Confirmation of the Entries

	Loadcase	Speed	Torque
0	Case 1	5	5
1	New Loadcase	1200	-34
2	Case 3	99	99
3	Case 4	0	0
4	Case 5	12	12
5	Overload 1	34	56

Figure 3-10 Load Case Table Before and After the Creation of a New Load Case

3.5.3 Change the Load Case

A *double click* on the function *Modify Loadcase* shows the following window for the choice of the load case which has to be changed:

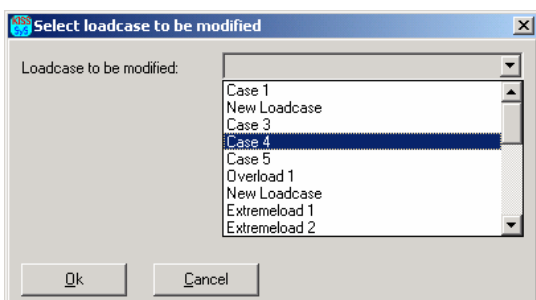


Figure 3-11 Choice of the Load Case to Change

Afterwards appears the following window in which the new data can be entered:

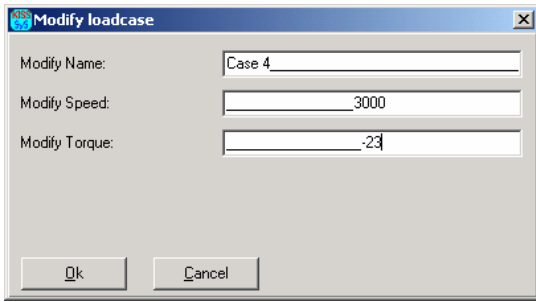


Figure 3-12 Load Case Modification

	Loadcase	Speed	Torque
0	Case 1	5	5
1	New Loadcase	1200	-34
2	Case 2	99	99
3	Case 4	0	0
4	Case 5	12	12
5	Overload 1	34	56
6	New Loadcase	1200	-34

Figure 3-13 Load Case Table Before and After the Change of Case 4

3.5.4 Display the Load Case Table

With *double click* on the function *Show Loadcases* in the user interface or with *right click* on the element *Loadcases* and *Show* in the tree structure, the load cases table is shown.

4 Description of the Functions

4.1 Select Load Case

<pre> VAR array,res; array = kSys_GetVariantArray("System.Loadcases.Loadcase"); res = CADH_VarDialog(["Select loadcase",400,200,0.4], [C:VDLG_StrCom,"Loadcase:",array,array[System.Loadcases.ActiveLoadcase],1]); IF res[0] THEN System.Loadcases.ActiveLoadcase= res[1]; ENDIF; kSys_Refresh(); </pre>	<p>The names of the load cases which stand in the variable Loadcase are read in an array. Then follows a dialog in which the array is shown as a list. If res[0] TRUE (if the dialog is confirmed with Ok) then the actual load case is equal to the selected load case. The position of the selected element is saved in the variable res[1] for that.</p>
--	---

4.2 New Load Case

<pre> VAR res,Oldactiveloadcase,OldLoadcase,Oldspeed,Oldtorque; Oldactiveloadcase=System.Loadcases.ActiveLoadcase; System.Loadcases.NofLoadcases=System.Loadcases.NofLoadcases+1; System.Loadcases.ActiveLoadcase=System.Loadcases.NofLoadcases-1; kSys_Recalc(); res = CADH_VarDialog(["Define new loadcase",400,200,0.4], [C:VDLG_Str,"New Name:", "New Loadcase"], [C:VDLG_Int,"New Speed:"], [C:VDLG_Int,"New Torque:"]); </pre>	<p>If a new load case is created, the number of the load cases increases by one and the actual load case is the new one.</p> <p>It appears a dialog where the name, the speed and the torque can be entered.</p>
---	--

<pre> IF res[0] THEN IF res[1]=" " THEN res[1]="No Name"; System.Loadcases.Loadcase=res[1]; ELSE; System.Loadcases.Loadcase=res[1]; ENDIF; System.Loadcases.Speed=res[2]; System.Loadcases.Torque=res[3]; ELSE System.Loadcases.NofLoadcases=System.Loadcases.NofLoadcases-1; System.Loadcases.ActiveLoadcase=Oldactiveloadcase; System.Loadcases.Loadcase=OldLoadcase; System.Loadcases.speed=Oldspeed; System.Loadcases.torque=Oldtorque; ENDIF; CADH_Message("New load case added, see Table Loadcases"); kSys_Refresh(); </pre>	<p>If no name is given for the new load case, it receives the name <i>No Name</i>.</p> <p>If the dialog is <i>Cancelled</i>, the new load case is deleted and the old load case is reactivated.</p> <p>The entry is confirmed by a message box.</p>
--	---

4.3 Modify Load Case

<pre> VAR array,res,resres,Oldspeed,Oldtorque; array = kSys_GetVariantArray("System.Loadcases.Loadcase"); res = CADH_VarDialog(["Select loadcase to be modified",400,200,0.4], [C:VDLG_StrCom,"Loadcase to be modified.",array,System.Loadcases.Loadcase,1]); IF res[0] THEN System.Loadcases.ActiveLoadcase=res[1]; resres = CADH_VarDialog(["Modify loadcase",400,200,0.4], [C:VDLG_Str,"Modify Name:",System.Loadcases.Loadcase], [C:VDLG_Int,"Modify Speed:",0], [C:VDLG_Int,"Modify Torque:",0]); kSys_Recalc(); IF resres[0] THEN System.Loadcases.Loadcase=resres[1]; System.Loadcases.Speed=resres[2]; System.Loadcases.Torque=resres[3]; ENDIF; ENDIF; kSys_Refresh(); </pre>	<p>At first the active load case is chosen by choice of the list.</p> <p>If the dialog is confirmed with <i>Ok</i>, a second dialog appears, in which values for the name, speed and torque can be changed.</p> <p>If this second dialog is confirmed with <i>Ok</i>, the values are saved in the corresponding variables.</p>
---	--