



## MADYN 2000 Version 3.7

The following new features were introduced in version 3.7:

1. [Non-Synchronous Characteristics for Tilting Pad Bearings](#)
2. [Parameter Optimization \(for Magnetic Bearing Controllers\)](#)
3. [Custom Block for Modelling](#)
4. [Unbalance Definition by G According to ISO 1940](#)
5. [Fluids: Introduction of the Coefficient  \$m\_q\$  and Speed Dependence](#)
6. [New Commands for Automation \(batch\)](#)

Apart from these feature several bugs were fixed.

Regarding **Radial Fluid Film bearings RFBs** it should be noted, that the grid for the bearing analysis with more than 5 pads has been refined. This may lead slightly different results.

Table of content for detailed description of features:

1. Non-Synchronous Characteristics for Tilting Pad Bearings .....	2
2. Parameter Optimization .....	5
2.1 General Explanations, Methods to Search an Optimum .....	5
2.2 Magnetic Bearing Controller Parameters Optimization .....	7
2.3 Example .....	12
3. Custom Block for Modelling .....	15
3.1 Defining a Black Box Shaft .....	15
3.2 Generalised MADYN Syntax .....	15
3.3 Example .....	16
4. Unbalance Definition by G According to ISO 1940 .....	18
5. Fluids: Introduction of the Coefficient $m_q$ and Speed Dependence.....	19
6. New Commands for Automation (batch) .....	20
7. References .....	22



## 1. Non-Synchronous Characteristics for Tilting Pad Bearings

As described in /1/ tilting pad bearings are frequency dependent, i.e. the damping and stiffness properties do not only depend on speed, but also on the frequency. This means, that at a non-synchronous frequency, the damping and stiffness properties are different from those at synchronous speed. This may be relevant for the assessment of stability by eigenvalue analyses. To consider this effect the bearings must be described as transfer functions TFU. Until now it is common practice to use the synchronous properties only. However, for certain types of tilting pad bearings the frequency dependence is not negligible.

For analysis types DIN table (DinTab) and Constant Adiabatic (c\_ad) the non-synchronous characteristics can be calculated. The analysis is activated by a check box “Calc Non-syn. Coeff.” in the RFB GUI. This check box is present in case of tilting pad bearings and analysis types DinTab and c\_ad (see figure 1.1). The analysis of the non-synchronous characteristics needs the static equilibrium position from the usual ALP3T analysis. In case the check box is activated, the button “CALC” therefore starts the usual analysis, in case results do not yet exist, before starting the additional analysis of the matrices required for the transfer function.

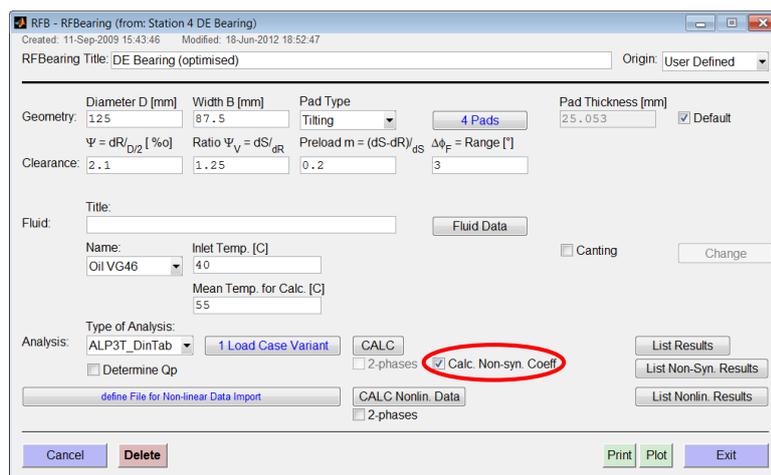


Fig. 1.1: RFB GUI for tilting pad and analysis type DINTab

The matrices to determine the transfer functions are as follows (also see /1/):

- Coefficient matrix C with dim.  $2 \times 2$  for the relation between forces  $F_{2,3}$  and displacements  $x_{2,3}$ .
- Coefficient matrix D with dim.  $2 \times 2$  for the relation between forces  $F_{2,3}$  and velocities  $\dot{x}_{2,3}$ .
- Coefficient matrix E with dim.  $2 \times N_p$  for the relation between forces  $F_{2,3}$  and tilting angles  $\psi_i$ .
- Coefficient matrix H with dim.  $2 \times N_p$  for the relation between forces  $F_{2,3}$  and tilting angular vel.  $\dot{\psi}_i$ .
- Coefficient matrix  $C\sim$  with dim.  $N_p \times 2$  for the relation between tilt. moments  $M_i$  and displacements  $x_{2,3}$ .
- Coefficient matrix  $D\sim$  with dim.  $N_p \times 2$  for the relation between tilt. moments  $M_i$  and velocities  $\dot{x}_{2,3}$ .
- Coefficient matrix  $E\sim$  with dim.  $N_p \times N_p$  for the relation between tilt. moments  $M_i$  and tilt. angles  $\psi_i$ .
- Coefficient matrix  $H\sim$  with dim.  $N_p \times N_p$  for the relation between tilt. moments  $M_i$  and tilt. ang. vel.  $\dot{\psi}_i$ .

$N_p$  hereby corresponds to the number of tilting pads.

The matrices for each speed can be viewed and exported to text files by the button “List Non-Syn. Results”.



From the matrices for the non-synchronous characteristics equivalent frequency dependent stiffness and damping coefficients  $k(f)$  and  $d(f)$  can be calculated (see /1/). They can be plotted for each speed from the window opened by the button “List Non-Syn. Results”.

In figure 2.2 the usual synchronous bearing coefficients of the bearing in figure 2.1 are shown. The pad support has only a small offset. The results at a speed of 100 cps (6'000 rpm) are highlighted by data tips. The frequency dependent damping and stiffness coefficients  $k(f)$  and  $d(f)$  for the same speed of 100 cps are shown in figure 2.3. The frequency dependence can clearly be recognised. The synchronous value is highlighted and agrees well with the synchronous coefficient of figure 2.2.

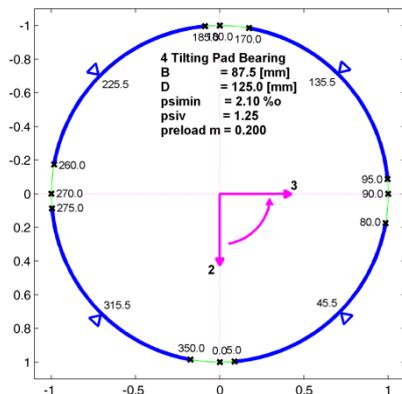


Fig. 2.1: Bearing geometry

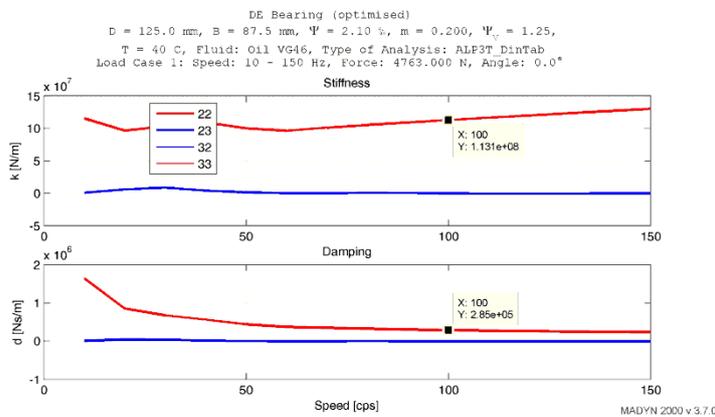


Fig. 2.2: Synchronous stiffness and damping coefficients

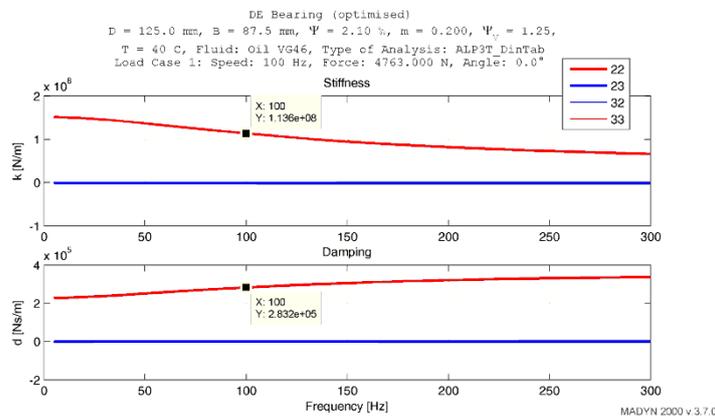


Fig. 2.3: Frequency dependent stiffness and damping coefficients at 100 cps



For the rotor in figure 2.4 the influence of the non-synchronous coefficients on the eigenvalues at the nominal speed of 6'000 rpm (100 Hz) is checked. The results of an eigenvalue analysis with the traditional synchronous coefficients are shown in figure 2.5 and with the nonsynchronous coefficients in figure 2.6. It can be seen, that the damping of the first natural frequency (backward and forward whirling) reduces considerably taking into account the non-synchronous effect. An additional effect of the results with non-synchronous coefficients is the arising of new modes due to the augmented model of the bearing. They are visible in figure 2.6 between 60 and 70 Hz. Since the modes have an extremely high damping ratio of about 90%, they are only visible as lines in the plot with a damping scale up to 0.1.

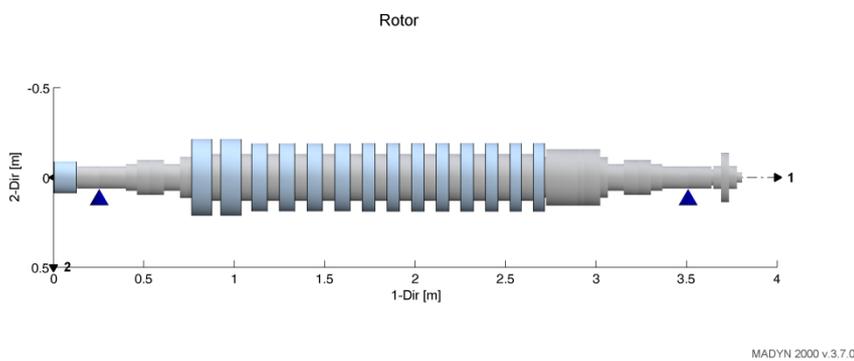


Fig. 2.4: Rotor

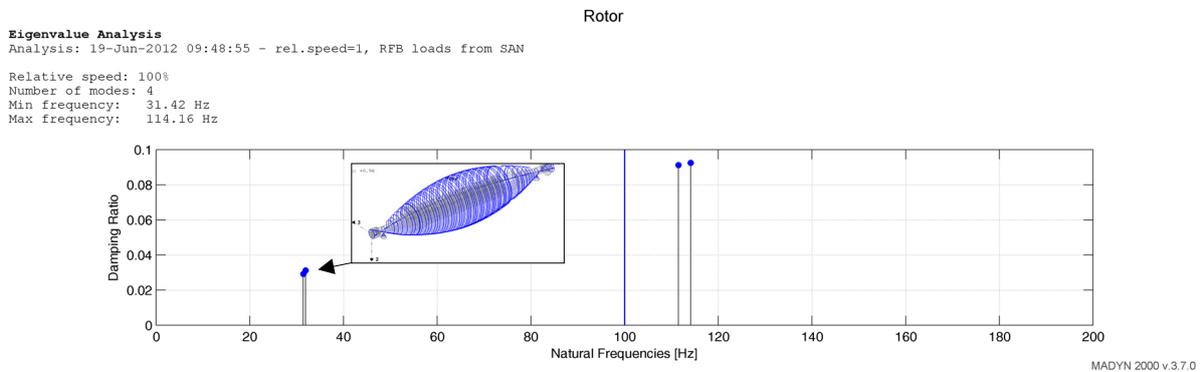


Fig. 2.5: Eigenvalues and 1<sup>st</sup> forward whirling natural mode shape using the synchronous coefficients

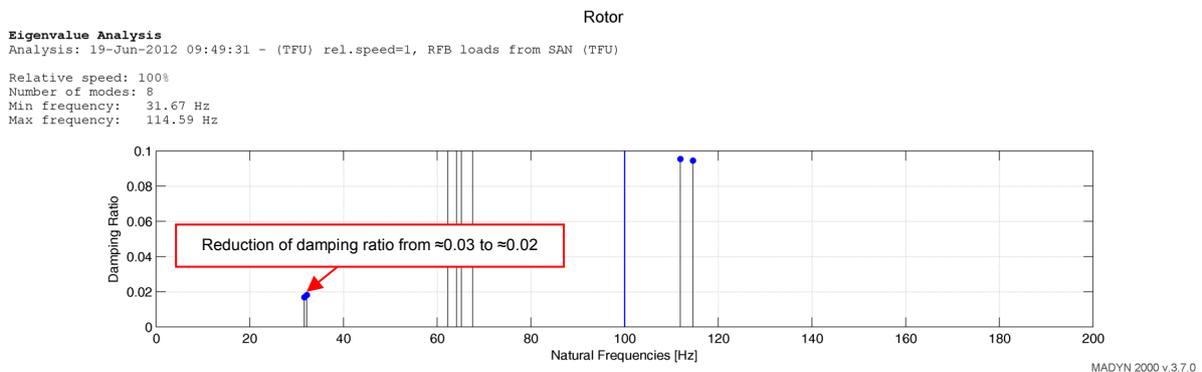


Fig. 2.6: Eigenvalues using the non-synchronous coefficients



## 2. Parameter Optimization

In many cases engineering requires the optimization of model parameters (e.g. the parameters of bearings such as clearances of an RFB or controller parameters for RMBs) or load parameters (e.g. a static displacement load for an optimal alignment).

In version 3.7 the optimization of magnetic bearing controller parameters is introduced, since the controller design is a complex process, which can efficiently be supported by an optimization [2]. Further applications will follow.

For the optimization a new pane in the system explorer is introduced (see figure 2.1)

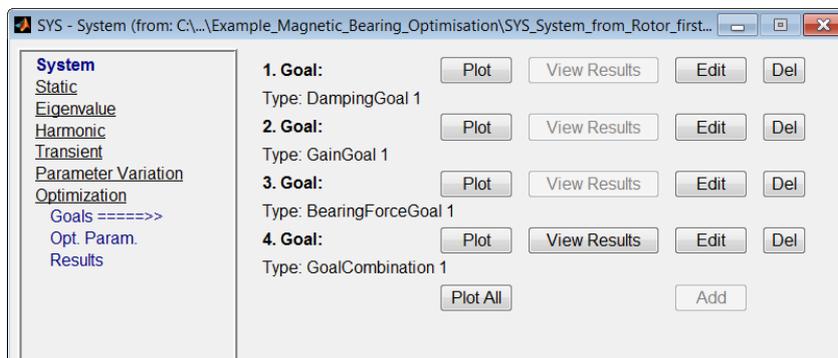


Fig. 2.1: System explorer with optimization pane

### 2.1 General Explanations, Methods to Search an Optimum

The most important part of an optimization process is the goal, which represents the ideal state of the system. MADYN 2000 estimates how close is the system to that ideal state using simple numerical score, computed as a function of the system configuration. Thus the score (value of the goal function) is a numerical representation of the system's quality.

General convention is that the score function yields

- a negative value if the goal is not yet achieved,
- zero if the goal is achieved exactly,
- a positive number if the system behaves better than required.

Score functions compute relative and unit-less measures, therefore a combination of different goals is possible.

To find an optimum the value of a score function is maximised. The definition of goals for an optimization corresponds to the definition of loads for other analysis types.

The calculation of quality score requires analyses such as eigenvalue or harmonic response to be carried out. For example if a damping goal is defined, then an eigenvalue analysis to calculate the damping of natural modes is necessary. The required analyses to calculate the value of the score function must be defined in the system.

The following parameters have to be defined to start an optimization analysis:

1. Parameters of the model to be varied in order to find the maximum quality score, for example parameters of a controller.
2. Limits and starting value for each parameter.
3. Goals among the defined goals must be selected.
4. An algorithm to find the optimum.
5. Maximum number of iterations and a limit for the calculation time.

The corresponding GUI for a magnetic bearing controller optimization is shown in figure 2.2.



Several optimization algorithms (point 4 of the optimization parameters) provided by MATLAB are available. All of them share the same high-level strategy:

- Start with the user-defined solution.
- Calculate the quality of the current solution using a score function as described above.
- Estimate which changes to parameter values are likely to improve the quality.
- Calculate the quality score of proposed solution and select it as the new currently best solution in case of actual improvement.
- Repeat from step c as long as the solution improves.

Methods differ mostly in a way of estimating the direction of further search (step c):

- Simplex Direct Search*: Generally speaking, this method operates with one parameter at a time, changing its value as long as the score improves. Once some local optimum is reached, method switches to another parameter and repeats the loop (see /19/).
- Gradient-based methods are *Interior Point*, *Sequential Quadratic Programming*, *Active Set*, *Trust-Region* (see /20/). Such methods compute numerical derivatives of the score function to guess better values for all parameter at once.
- Random search* is not a real optimization algorithm, rather than a set of simple unrelated guesses in the allowed range of parameters.

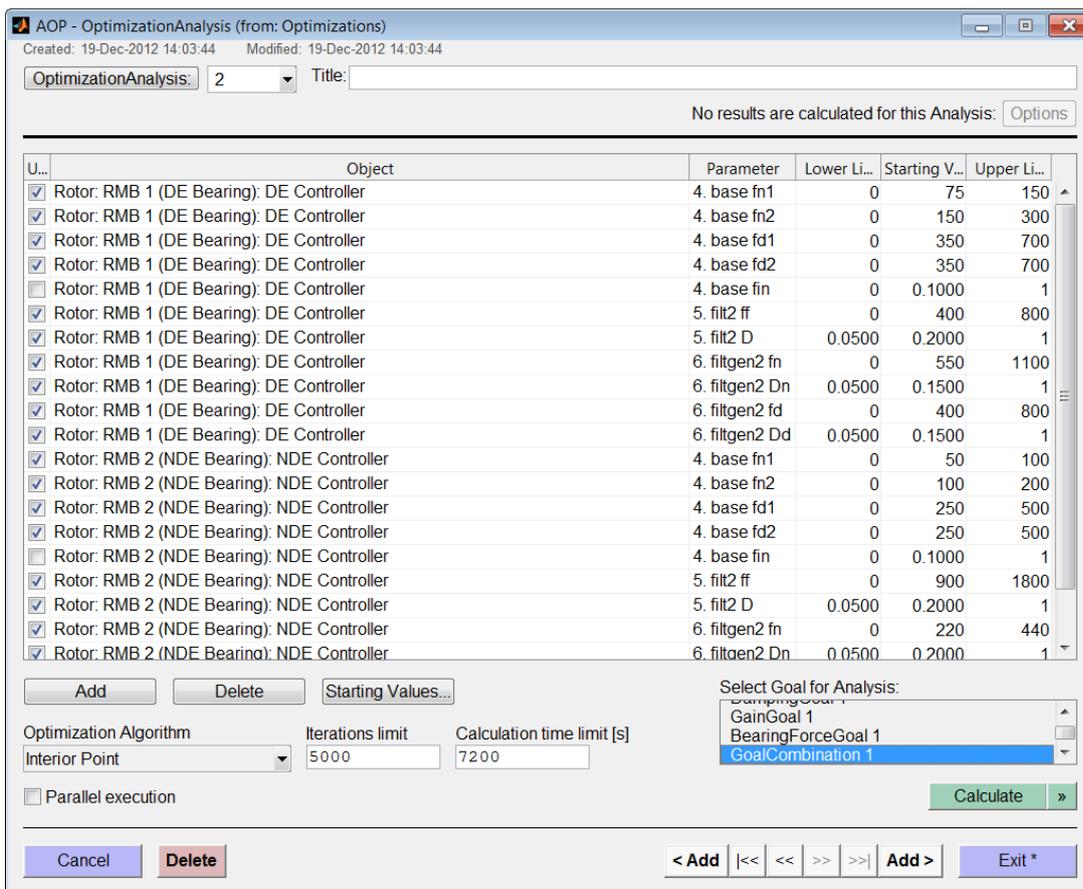


Fig. 2.2: Optimization parameter GUI.



## 2.2 Magnetic Bearing Controller Parameters Optimization

All parameters or a selection of parameters of all RMB controllers in a system consisting of blocks as described in chapter VII of the MADYN 2000 documentation can be varied to find an optimum.

Among the search algorithms described above good experience has been gained with the *Interior Point* algorithm.

The following goals can be defined for the optimization:

- 1.) *Damping goal* of all natural modes in a wide frequency range. The goal can be a function of the frequency, i.e. the target damping of natural modes in the operating speed range for example can be higher than for other modes. The score function for this goal requires eigenvalue results.
- 2.) *Bearing force goal*, i.e. the bearing forces should remain below a certain limit. The score function for this goal requires results of one load case of a harmonic response analysis.
- 3.) *Gain goal*, i.e. the gain of the controller transfer function should remain below a certain limit. This goal was introduced, because a low gain helps to keep the sensitivity low and because high gain can cause saturation of the magnetic bearing amplifiers due to noise.
- 4.) *Sensitivity goal*, i.e. the sensitivity should remain below a certain limit. For the definition of the sensitivity see chapter IV.4.1 of the MADYN 2000 documentation. The score function for this goal requires results of a sensitivity analysis.
- 5.) *Combination* of the goals 1 to 4, where different weights can be set to different goals.

In the following the goal functions are described more in detail.

### Damping Goal

The GUIs to define the damping goal as a function of the frequency can be seen in figure 2.3 and 2.4. The resulting function is shown in figure 2.5.

The damping goal can be defined for four frequency regions: Frequencies below, within and above the operating speed range as well as at higher frequencies. The field "Higher frequency limit" in the GUI of figure 2.4 defines what is considered as higher frequencies.

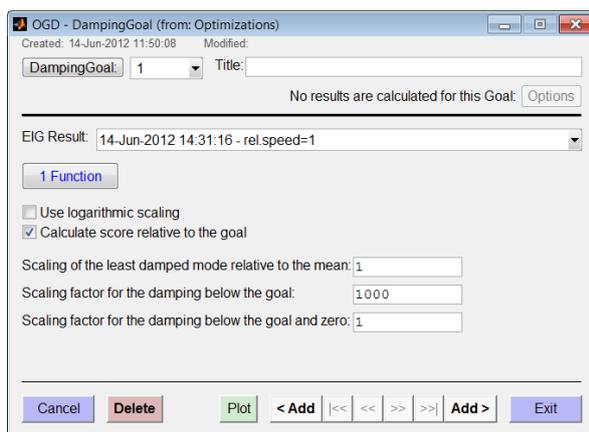


Fig. 2.3: GUI to define damping goal

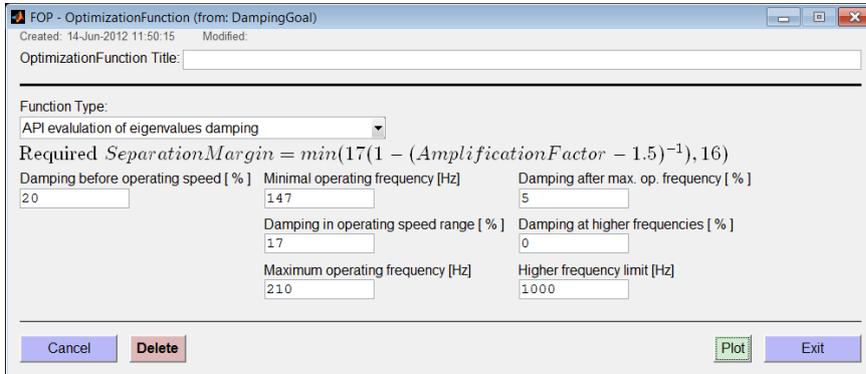


Fig. 2.4: GUI to define the frequency function for the damping goal

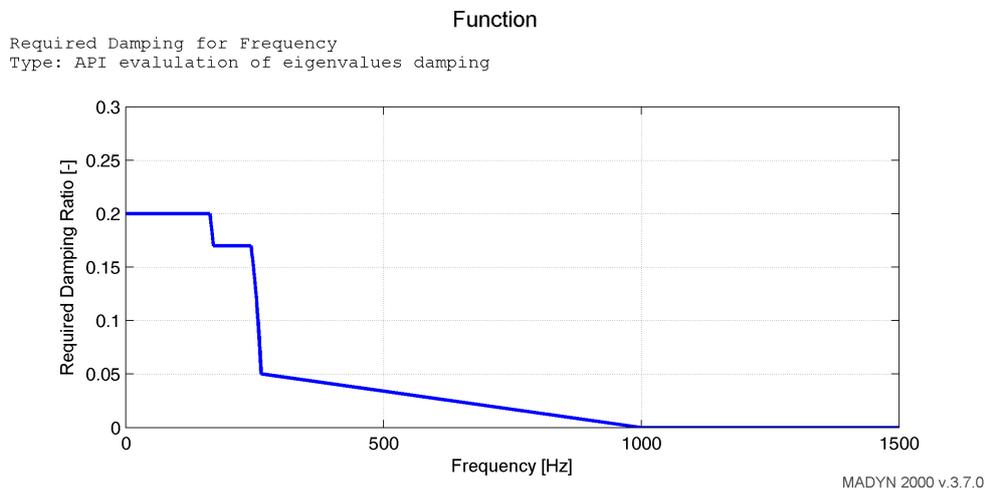


Fig. 2.5: Resulting frequency function for the damping goal

For the transition of the low frequency range to the minimum operating speed and from maximum operating speed to the high frequency range the formulas for the separation margin  $SM$  according to API are applied. They are as follows:

$$SM = 17 \left( 1 - \frac{1}{AF-1.5} \right) \text{ or } 16, \text{ whichever is less,} \quad (2.1)$$

for frequencies below the minimum speed,

$$SM = 10 + 17 \left( 1 - \frac{1}{AF-1.5} \right) \text{ or } 26, \text{ whichever is less,} \quad (2.2)$$

for frequencies above the maximum speed, with

$$AF = \frac{1}{2D\sqrt{1-D^2}} \quad (2.3)$$

and  $D$  as the damping ratio.

The main reason for the introduction of these functions between the regions is to get a smooth transition and thus get a better numerical behaviour. It should be noted, that higher values than 24% for the damping goal would destroy the smooth transition and are therefore prohibited.

In the region above the speed range and damping values below the value defined in the field "Damping after max. op. frequency" a linear transition to the value "Damping at higher frequencies" is



assumed. For frequencies above the “Higher frequency limit” a constant damping is assumed according to the value in the field “Damping at higher frequencies”.

The score value for the damping goal is calculated from deviations  $\Delta$  of the actual damping from the required goal damping:

$$\Delta = D_{\text{actual}} - D_{\text{required}} \quad (2.4)$$

As can be seen in the GUI for the damping goal definition (figure 2.3), there are two check boxes for the options “Use logarithmic scaling” and “Calculate score relative to the goal”.

In case of a score calculation relative to the goal the deviation is divided by the required damping:

$$\Delta_{\text{rel}} = \Delta / D_{\text{required}} \quad (2.5)$$

For logarithmic scaling D is transformed as follows:

$$\Delta_{\text{log}} = \log(1 + \max(-1, \Delta)) \quad (2.6)$$

Additionally scaling factors can be applied to the deviations  $\Delta$ ,  $\Delta_{\text{rel}}$  or  $\Delta_{\text{log}}$ , if the deviation is below the goal (since it does not necessarily make sense to further increase damping once it is above the goal), if the deviation is below the goal and below zero (to further emphasised unstable modes in the score) or for the least damped mode relative to the mean (to avoid that single modes with bad damping are negligible in the score value). The additional scaling for deviations below goals and below zero are described in equations (2.7) and (2.8).  $\Delta$  here also represents  $\Delta_{\text{rel}}$  and  $\Delta_{\text{log}}$ .

$$\Delta(\Delta < 0) = \Delta(\Delta < 0) \text{ Scaling}_{\text{below goal}} \quad (2.7)$$

$$\Delta(D_{\text{actual}} < 0 \ \& \ \Delta < 0) = \Delta(D_{\text{actual}} < 0 \ \& \ \Delta < 0) \text{ Scaling}_{\text{below zero}} \quad (2.8)$$

Since exceedingly good damping is not necessary, damping values with a damping ratio of 10% above the goal or three times better than the goal for relative scaling are capped.

The resulting score value is calculated from the deviations  $\Delta$ ,  $\Delta_{\text{rel}}$  or  $\Delta_{\text{log}}$  as follows:

$$\text{Score} = \text{mean}(\Delta) + \min(\Delta) \text{ Scaling}_{\text{least damped Mode}} \quad (2.9)$$

### Gain Goal

The GUI for the gain goal is shown in figure 2.6. For the gain goal a constant gain limit and sampling frequencies at which the gain is checked have to be defined. Moreover a checkbox allows specifying, that in case of goal combinations the gain goal only at natural frequencies shall be used.

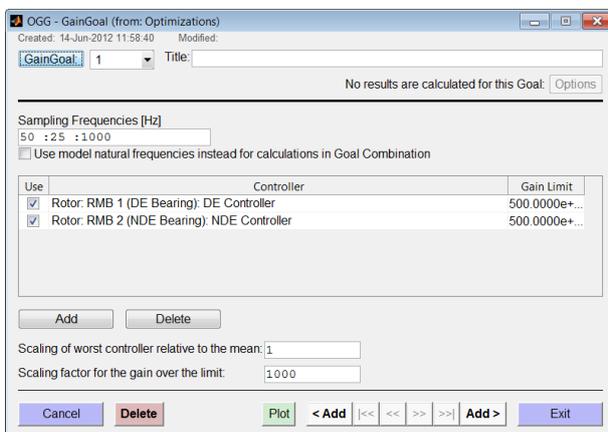


Fig. 2.6: GUI to define gain goal



Deviations from the goal are calculated for each controller by the peak gain value.

$$\Delta = \log(\text{Gain}_{\text{Limit}} / \text{Gain}_{\text{Peak}}) \quad (2.10)$$

A scaling factor can be applied to  $\Delta$  for gains above the limit in a similar manner as for the damping according to equation (2.7). A scaling factor can also be applied for the highest gain peak. The score thus is calculated as follows:

$$\text{Score} = \text{mean}(\Delta) + \min(\Delta) \text{Scaling}_{\text{worst gain}} \quad (2.11)$$

The averaging hereby is done over the controllers.

### Bearing Force Goal

The GUI for the bearing force goal is shown in figure 2.7. For the bearing force goal a harmonic load case with results and a force limit have to be defined. Moreover a checkbox allows specifying, that in case of goal combinations the force only at natural frequencies shall be used. This is justified, since maximum forces normally are in the vicinity of natural frequencies.

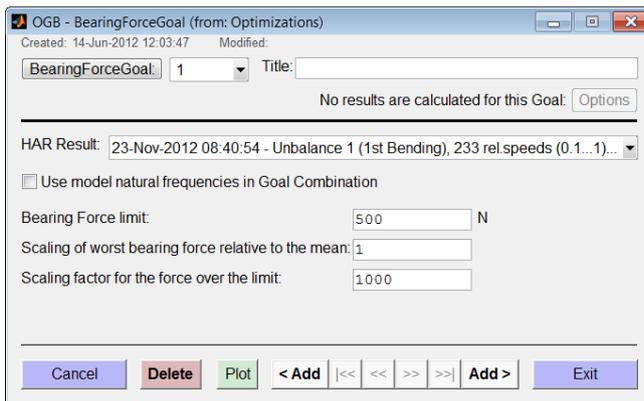


Fig. 2.7: GUI to define bearing force goal

Deviations from the goal are calculated for each bearing by the peak force.

$$\Delta = (\text{Force}_{\text{Limit}} - \text{Force}_{\text{Peak}}) / \text{Force}_{\text{Limit}} \quad (2.12)$$

A scaling factor can be applied to  $\Delta$  for forces above the limit in a similar manner as for the damping according to equation (2.7). A scaling factor can also be applied for the highest force. The score thus is calculated as follows:

$$\text{Score} = \text{mean}(\Delta) + \min(\Delta) \text{Scaling}_{\text{worst force}} \quad (2.13)$$

The averaging hereby is done over the bearings.

### Sensitivity Goal

The GUI for the sensitivity goal is shown in figure 2.8. For the sensitivity goal a sensitivity analysis and a limit have to be defined. Moreover a checkbox allows specifying, that in case of goal combinations the sensitivities only at natural frequencies shall be used. This is justified, since maximum sensitivities are in the vicinity of natural frequencies. The analysis time is considerably reduced using this option, because the sensitivity analysis normally requires small frequency steps due to the small damping at high frequencies.

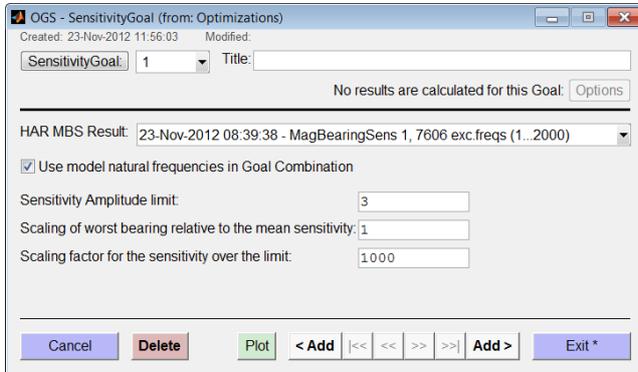


Fig. 2.8: GUI to define sensitivity goal

Deviations from the goal are calculated for each bearing by the peak sensitivity.

$$\Delta = \log(\text{Sensitivity}_{\text{Limit}} / \text{Sensitivity}_{\text{Peak}}) \quad (2.14)$$

A scaling factor can be applied to  $\Delta$  for sensitivities above the limit in a similar manner as for the gain according to equation (2.7). A scaling factor can also be applied for the highest sensitivity. The score thus is calculated as follows:

$$\text{Score} = \text{mean}(\Delta) + \min(\Delta) \text{Scaling}_{\text{worst bearing sensitivity}} \quad (2.15)$$

The averaging hereby is done over the bearings.

### Goal Combinations

The GUI to define goal combinations is shown in figure 2.9. All above mentioned goals can be combined with different scaling factors. For each goal different scaling factors can be applied for positive and negative deviation.

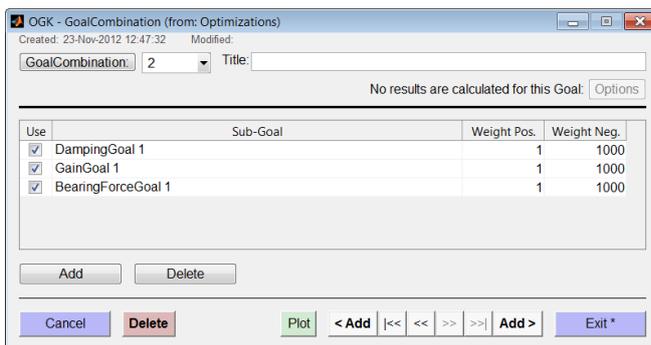


Fig. 2.9: GUI to define goal combination

The score is calculated as follows:

$$\text{Score} = \frac{1}{N} \sum \text{score}_i \cdot \begin{cases} \text{scaling}_+ & \text{if } \text{score}_i < 0 \\ \text{scaling}_- & \text{if } \text{score}_i \geq 0 \end{cases} \quad (2.16)$$

with  $i$  as the index for the different goals and  $N$  as the number of goals.



### 2.3 Example

In figure 2.10 an example of a compressor on magnetic is shown to demonstrate, what can be achieved with the optimization. More details of this example are explained in [2].

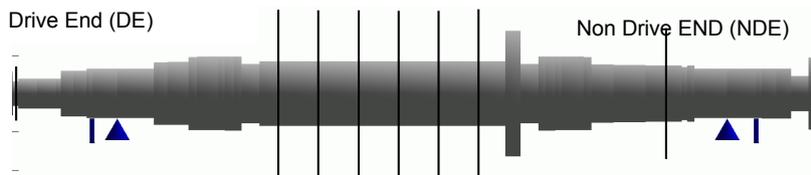


Fig. 2.10: Example rotor, weight 550kg, maximum speed 12'600rpm

Suited controllers for this compressor have been designed. Figure 2.11 and 2.12 summarise the performance of these controllers. They yield a stable system and a good sensitivity. However, the following points are not satisfactory:

- The damping of the first bending mode in the speed range is only 12%, i.e. lower than 20% which corresponds to an amplification factor of 2.5 allowing operation in resonance according to API (figure 2.11).
- The maximum bearing force of the unbalance response exciting the first bending mode is 1000N (figure 2.12). A maximum of only half this value is desirable in order to have sufficient safety margins to unbalances as they can occur in the field due to thermal deformation or fouling.

In order to improve the controller an optimization was run. A damping goal, a bearing force goal and gain goal were applied. The bearing force goal is set to 500 N and the gain goal to 500 kN/mm. A factor of 1/2 was used for the gain goal, i.e. damping and force, were regarded as more important. The sensitivity goal was not applied, since it is usually automatically fulfilled, if the gain is not too high and if the damping goal is fulfilled. This implies that the damping goal function is chosen accordingly, i.e. the damping goal of the modes above the operating speed range has a certain level.

The used damping goal function is shown in figure 2.13. In the upper operating speed range the goal was set to 17%. The reason is that force and damping goal may be contradictory and it was decided to slightly release the damping goal, since the force goal is more important. In the low frequency range the goal is set to 20% in order to have robustness against self-excitation from fluid forces in the seals. Above the speed range the damping goal linearly decreases from a rather high value of 5% to 0% at 1000 Hz. The function in this range is chosen in order to keep the sensitivity low.

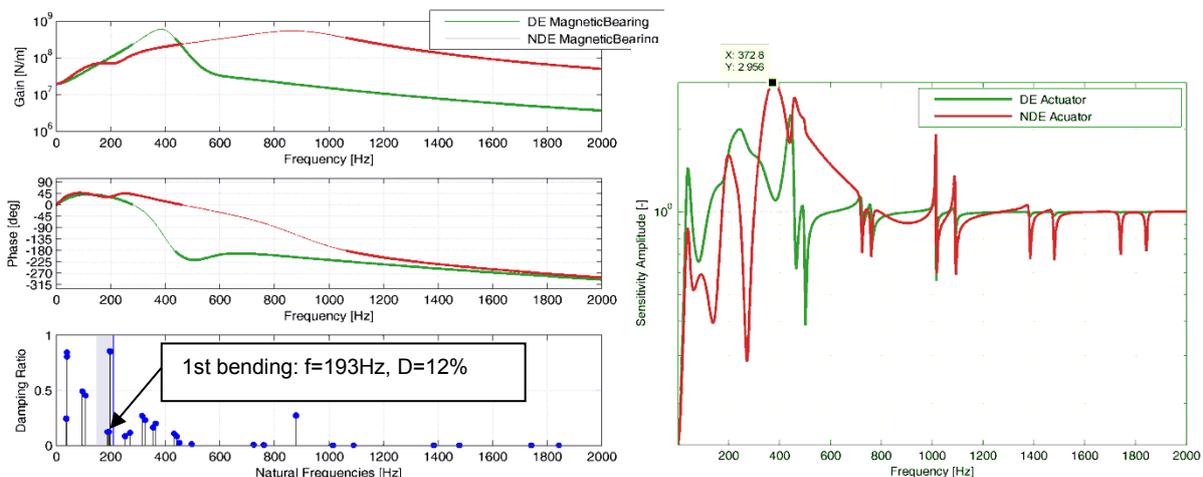


Fig. 2.11: Bearing transfer functions, eigenvalues and sensitivity at maximum speed

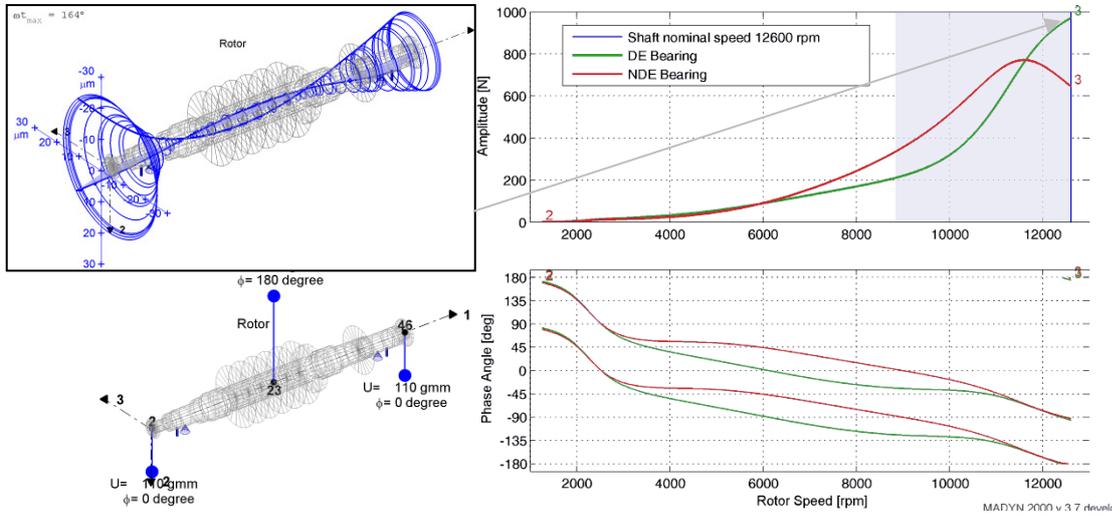


Fig. 2.12: Unbalance load case (magnitude G1) and bearing force response

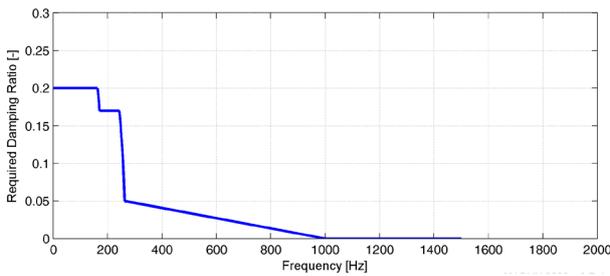


Fig. 2.13: Goal function for the damping ratio

Numerical optimization of totally 20 high-level parameters of both (DE and NDE) controllers was carried out. The performance of the resulting controller can be seen in the figures 2.14 and 2.15. The red dots in the eigenvalue plot indicate damping values below the goal. Values above the goal are green.

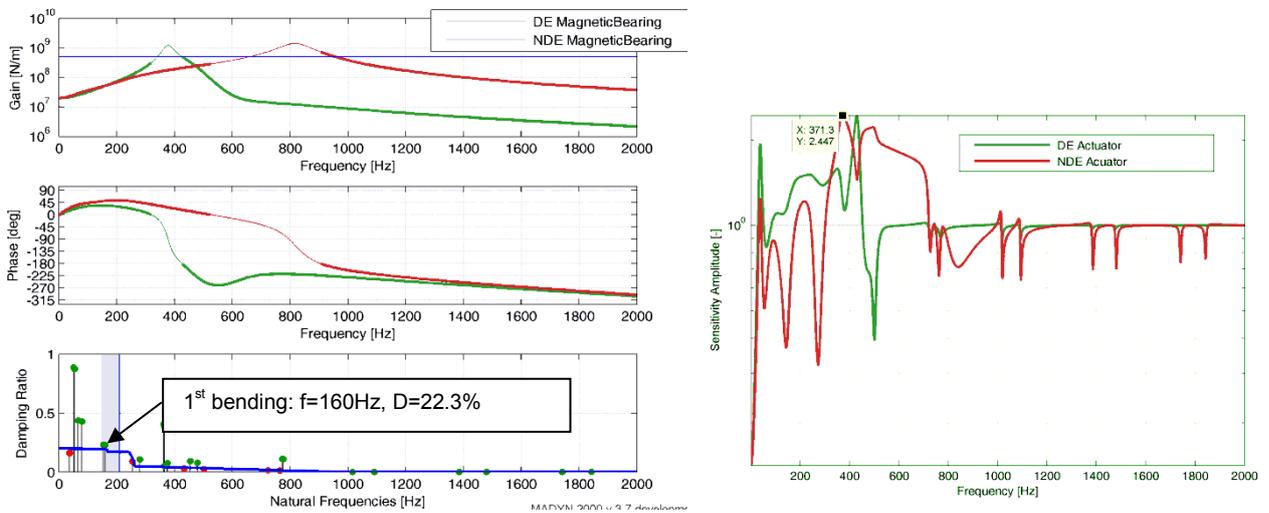


Fig. 2.14: Bearing transfer functions, eigenvalues and sensitivity at max. speed, optimized controller

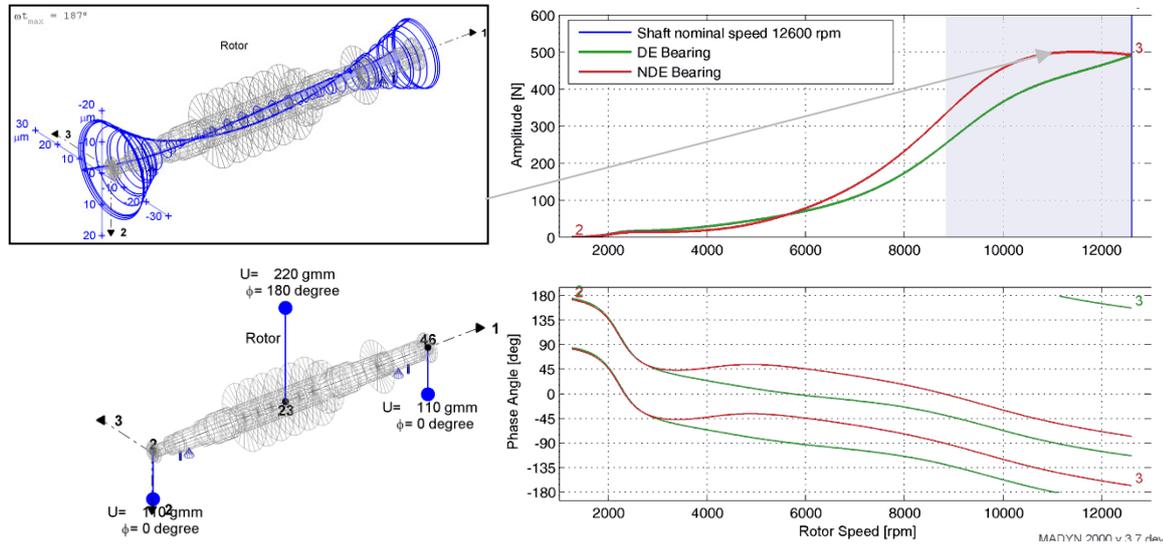


Fig. 2.15: Unbalance load case (mag. G1) and bearing force responses, optimized controller

It can be seen, that the damping ratio of the first bending is above 20%, the sensitivity has slightly further improved and the maximum bearing force to the unbalance response is now 500N only. Thus the controller fulfils the major goals. Such a successful result with an optimization can of course only be achieved if the goals are physically possible.



### 3. Custom Block for Modelling

User can add custom blocks to the model using a generalised MADYN syntax. MADYN syntax is described in Appendix I of the MADYN 2000 documentation. By means of such custom blocks features can be introduced into a model, which are not directly supported by MADYN 2000 model objects.

Using the custom block requires deep knowledge of MADYN. When executing the custom block the resulting MADYN input file (MIF) is checked by MADYN and may result in error messages in the MADYN output file "FILE6". It is the user's responsibility to check and correct these errors, whereas in MADYN 2000 they are widely avoided by input checks in the model GUIs.

#### 3.1 Defining a Black Box Shaft

Custom blocks require additional degrees of freedom, which are introduced by sections of a so called "Black Box Shaft" (BBS). The button "Add Custom Block" in the SYS GUI opens a GUI to define a black box shaft. As for usual shafts degrees of freedom (torsion, lateral ...) can be selected by a TLA case, material and sections can be defined. Moreover the GUI has a text editor, which allows creating the MIF (MADYN Input File) source code in generalised MADYN syntax.

The following rules apply for a Black Box Shaft (BBS):

- MIF code according to the stations of the black box shaft (KOORD in MIF), the degrees of freedom (FESTK in MIF) and a material (WSTOF in MIF) is automatically created. It cannot be seen in the MIF text editor.
- The cross sectional properties are not used at all in the MIF code. They are only used for the geometry plot of the black box shaft, i.e. the plot does not represent the real model anymore.
- Connections of the black box shaft to other shafts are rigid. In MIF STARR is used for this connection<sup>1</sup>.
- The MIF text editor allows creating the custom MIF block using the degrees of freedom of a black box shaft and the shaft material in a general way. Additional nodes and degrees of freedom to those from the black box shaft sections can be introduced if necessary. The custom block must be written in format free generalised MADYN syntax. The syntax is generalised in the sense, that variables and formulas can be used, where fixed numbers are necessary in the actual MADYN input file. The generalised syntax is described in the following chapter.

The complete MIF code including the part of the shaft input and the custom MIF block can be seen in the print of the BBS. In the print variables and formulas are replaced by the actual numbers.

#### 3.2 Generalised MADYN Syntax

In MADYN 2000 the numbers in MIF for nodes, elements and element properties are automatically created. Variable numbers in a BBS are necessary, because the final numbers in MIF are not known and therefore cannot be referred to when defining the custom MIF block.

In the following the generalised syntax to create variable numbers is described:

#STA1, #STA2, #STA3, #STA4 ...: Creates numbers for the stations created by the sections of the black box shaft. #STA1 refers to the 1<sup>st</sup> station #STA2 to the 2<sup>nd</sup> station ... of the shaft. The stations can thus be referenced in the custom block.

---

<sup>1</sup> The user must be aware of this, since this results in a slave node. Using a slave node as further slave node (chains of slaves) is not allowed in MADYN and results in an error in "FILE6".



#: Creates a number, which cannot be further referenced in the custom block. The custom block may contain several such signs. Although they are the same, different numbers are created in MIF.

# followed by a text, e.g. #FD: Creates a number, which can be referenced in the custom block, e.g. for elements such as springs, their properties or for additional nodes.

Arithmetic expressions as in MATLAB can be used between curly brackets, e.g.:

{-1/474.286 \* 2}

Arithmetic expressions can also be used in combination with node numbers, e.g. in case of KIKOPs, where a coordinate is referred to as node number times 10 plus a number between 1 and 6 referring to the six degrees of freedom of a node (1 axial displacement, 2 & 3 radial displacements, 4 torsional angle, 5 & 6 angles for bending):

{#STA2\*10+4}

### 3.3 Example

The following simple example demonstrates the modelling of a hinge as connection between two shafts. This is not directly possible in MADYN 2000, where two shafts can either be rigidly connected or via a flexible coupling, which consists of two hinges and a spacer.

The GUI to model the hinge by a BBS is shown in figure 3.1.

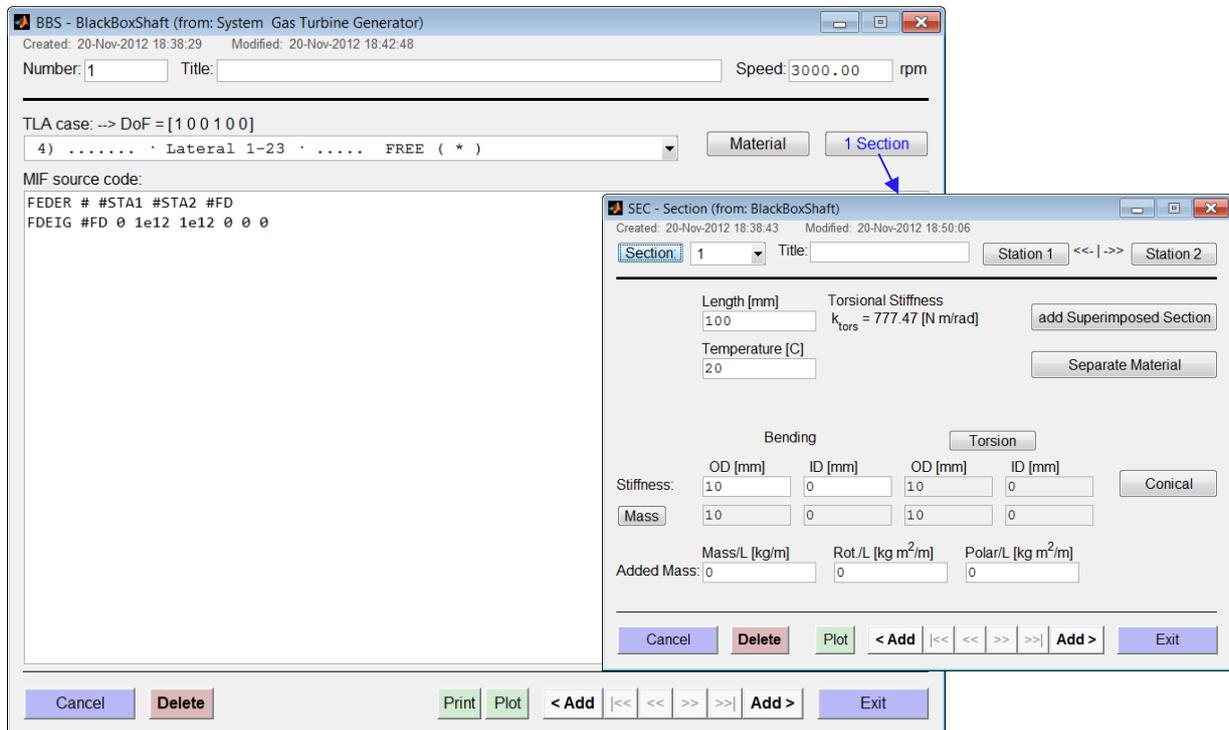


Fig. 3.1: GUI to define hinge by custom block with section GUI

The section of the BBS introduces two nodes to the system. In the MIF source code they are referenced as #STA1 and #STA2. The degrees of freedom are for lateral analyses, i.e. coordinate 2,3,5,6 are active. The BBS can be connected to two other shafts, one on each side, by the usual connection in the SYS GUI. These connections are rigid i.e. #STA1 can be considered as part of one





#### 4. Unbalance Definition by G According to ISO 1940

The magnitude of an unbalance can be defined by the balancing quality G according to the ISO standard 1940. For this a reference mass and reference speed are necessary (G value x yields eccentricity e via  $e\Omega = x$  [mm/s] with  $\Omega$  as the speed in [rad/s]  $\rightarrow U = m_{\text{Reference}} e$ ). The shaft nominal speed is taken as the reference speed and the rotor mass or the mass of selected sectors as reference mass.

In figure 4.1 the new GUI for the unbalance definition is shown with the highlighted controls for the definition in G. To define the reference mass from shaft sectors a separate GUI can be opened as shown in the figure. The definition of several sectors is possible. The reference mass is the sum of all sectors.

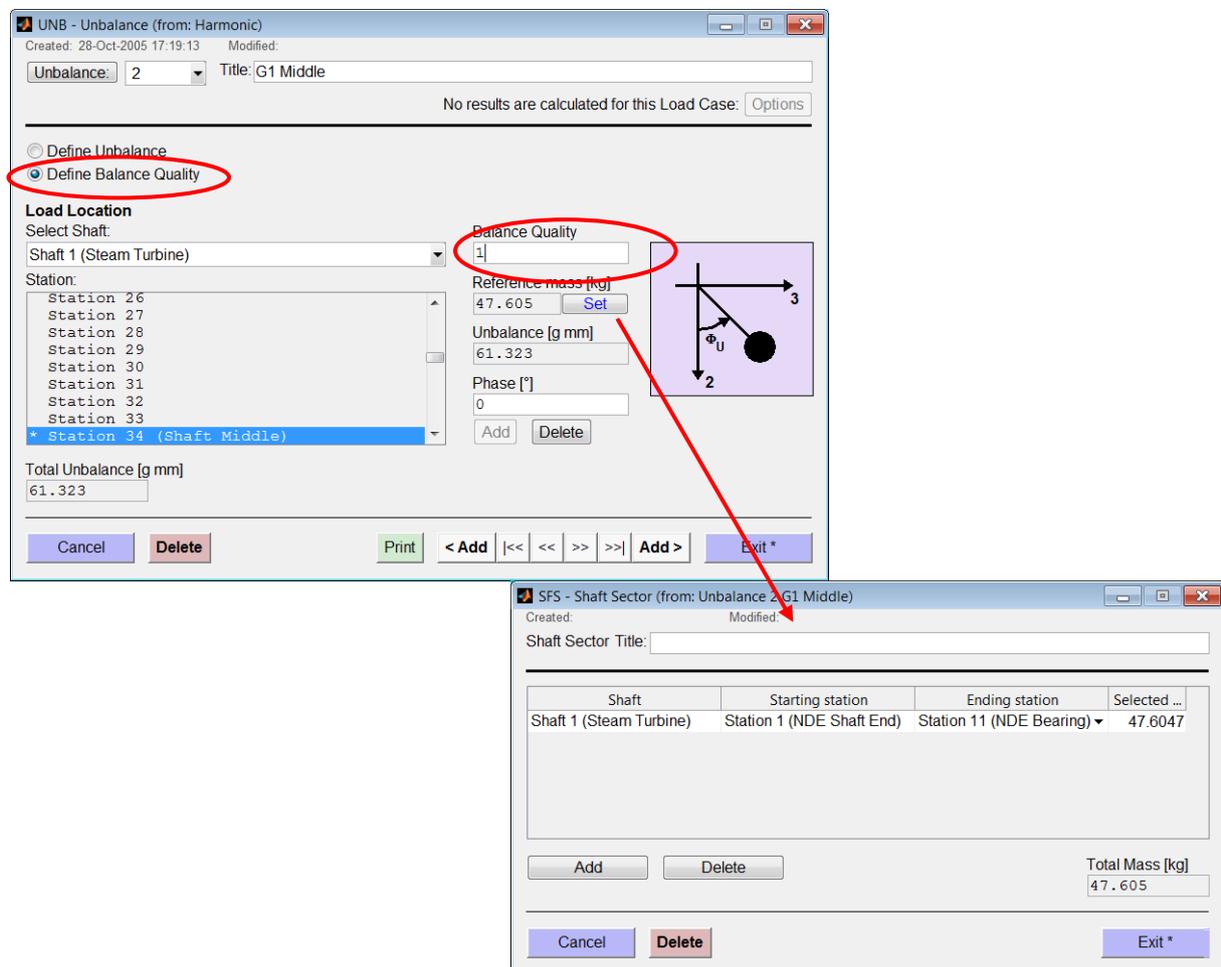


Fig. 4.1: GUI for the definition of an unbalance by G according to ISO 1940

Sectors can be defined in a menu, which is opened by the “set” button under the title “Reference mass”. The complete reference mass is shown in a field. It corresponds to the sum of the masses of all sectors.



## 5. Fluids: Introduction of the Coefficient $m_q$ and Speed Dependence

For fluids skew symmetric mass matrices can now be considered. In previous versions this was not possible due to traditional restrictions. In the previous MADYN eigenvalue solver unsymmetrical mass matrices could not be considered. In MADYN 2000 this restriction does not exist.

Moreover it is possible to enter speed dependent coefficients.

The new GUI for fluids is shown in figure 5.1. The edit field for  $m_q$  (cross coupling mass) is now active and a new edit field for speeds exists, which is activated by the check box "Speed-dependent fluid interpolation".

In general it is more convenient to import tables for fluid coefficients. These tables may now have columns for speeds (as shown in the table after figure 5.1) and columns for cross coupling masses denoted by " $m_q$ ".

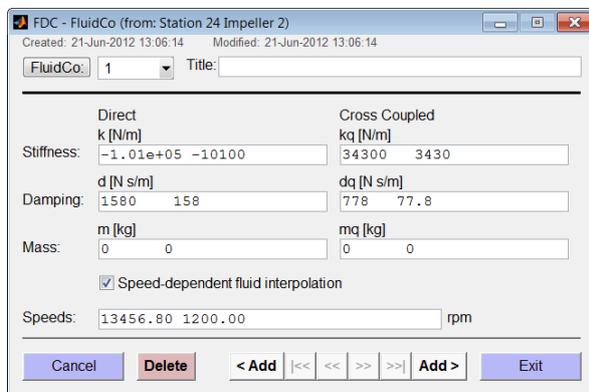


Fig. 5.1: Fluid GUI

\* This is an example with denotation for the stations and speed dependence

Station	Speed	$k$	$k_q$	$d$	$d_q$
'Impeller 1'	20.00	-8.65E+04	1.33E+05	1.41E+02	2.17E+02
'Impeller 2'	20.00	-1.01E+04	3.43E+03	1.58E+02	7.78E+01
'Impeller 3'	20.00	-1.53E+04	8.74E+03	2.06E+02	1.10E+02
'Impeller 4'	20.00	-1.70E+04	1.01E+04	2.28E+02	1.23E+02
'Impeller 5'	20.00	-1.86E+04	1.22E+04	2.53E+02	1.35E+02
'Impeller 6'	20.00	-1.73E+05	2.54E+05	2.77E+02	4.38E+02
'Impeller 7'	20.00	-1.89E+05	2.76E+05	3.00E+02	4.75E+02
'Impeller 1'	224.28	-8.65E+05	1.33E+06	1.41E+03	2.17E+03
'Impeller 2'	224.28	-1.01E+05	3.43E+04	1.58E+03	7.78E+02
'Impeller 3'	224.28	-1.53E+05	8.74E+04	2.06E+03	1.10E+03
'Impeller 4'	224.28	-1.70E+05	1.01E+05	2.28E+03	1.23E+03
'Impeller 5'	224.28	-1.86E+05	1.22E+05	2.53E+03	1.35E+03
'Impeller 6'	224.28	-1.73E+06	2.54E+06	2.77E+03	4.38E+03
'Impeller 7'	224.28	-1.89E+06	2.76E+06	3.00E+03	4.75E+03



## 6. New Commands for Automation (batch)

Available commands are listed in the following table, in order of precedence. The commands “Get” and “Set” are new in version 3.7. They allow getting and modifying values of any object in a MADYN 2000 systems.

Command	Description	Possible Parameters
<b>Execute</b>	Reads and executes commands from a text file. Commands should be entered	<b>File name</b> , .txt extension is recommended
<b>Import</b>	Reads MADYN2000 batch file and imports it to opened system.	<b>File name</b> , *.mbp
<b>Set</b>	Modify the value of existing property in the model, loads, variations or analysis parameters	<b>Object Address</b> and new value, separated by '='
<b>Clear</b>	Discard currently stored results and calculated system matrices	'all' – discard all results 'RMB' – the same as 'all', but runs faster for RMB-only changes by preserving non-RMB related information
<b>Calculate</b>	Calculate data for empty result placeholders that are present in the system or were imported.	'all' – finer control is planned for future versions
<b>Get</b>	Retrieve current value of some numerical property in results or model	<b>Object Address</b>
<b>Check</b>	Check currently open model for possible errors, like missing objects	'optional' – any problems will be reported as warnings  'required' – model will not be saved if check fails
<b>Quit</b>	Save the system and close MADYN2000 after performing all operations	Output <b>File name</b> , *.md3

In the following an example for the use of object addresses is described. The object addresses can be found with the help of the context menu opened by a right mouse click in any object GUI<sup>2</sup>. The item “Properties” will open a window, which contains among others the “Batch Address”.

The folder “My documents\MADYN 2000\Examples\Batch processing\Open Interface” contains a MATLAB script “djsBatchTestVariation.m”, which gets a parameter of a system (either speed or the section length) changes it and recalculates the system. For these purposes the “Get” and “Set” commands are used. The eigenvalues as a function of the parameter are then plotted. In the following the parameter “Object Address” of the “Get” and “Set” command is described.

In general, the address of an object is a path in MADYN 2000 data structure, where individual steps are separated with dots. For example, the address of the third station of the first shaft is:

```
System.Shaft(1).Section(3)
```

Therefore, to change the length of that section, one can add this command to a batch command line:

```
Set System.Shaft(1).Section(3).Length=0.1
```

Values are always represented in default SI units, i.e. for lengths values are in “m”.

To get data from Eigenvalue results, one should send two Get commands:

```
Get Processing:Eigenvalue.Result(1).ResCalcRef.Parameter.Frequency
Get Processing:Eigenvalue.Result(1).ResCalcRef.Parameter.Damping
```

<sup>2</sup> This feature will be available in version 3.7.1, not yet in version 3.7.0.



- 21 -

In the following is an example to execute MD3 program with several parameters. Line breaks are added for clarity:

```
md3 C:\...\batch\SYS_Steam_Turbine.md3
Set System.Shaft(1).Section(3).Length=0.1
Clear all
Calculate all
Get Processing:Eigenvalue.Result(1).ResCalcRef.Parameter.Frequency
Get Processing:Eigenvalue.Result(1).ResCalcRef.Parameter.Damping
Quit -
```

The printed result of the commands will be:

```
Loaded: C:\DeltaJS\Documents\User_Meeting_2012\batch\SYS_Steam_Turbine.md3
ObjInfo: Created: 22.11.2012 17:50:57 Ver=3.7.0-16.10.2012 ObjVer=432.5798
MIF=4.1.19.02.2012
```

```
Set System.Shaft(1).Section(3).Length=0.1
Clear all
Calculate ALL
```

```
Index of the model: 211231016.
Model created on 11-23-12 at 10:16 AM .
Model: Steam Turbine
Get Processing:Eigenvalue.Result(1).ResCalcRef.Parameter.Frequency
[0.0409148137807399 0.250028945463774 0.500045745915878 ... 265.91349468272]
Get Processing:Eigenvalue.Result(1).ResCalcRef.Parameter.Damping
[4.67507295248687e-05 0.000226868434722369 0.00011343734639332 ...
4.47207584029606e-07]
Quit MADYN2000
```

For each pair of command-line arguments MADYN 2000 prints back the requested command and the response. Dot is always used as decimal point in numbers. Vector values are printed in Matlab format.



## 7. References

- /1/ J. Schmied, A. Fedorov, B. Grigoriev: Nonsynchronous tilting pad bearing characteristics. Proceedings of the 8<sup>th</sup> IFToMM International Conference on Rotordynamics, September 12-15, Seoul Korea.  
[http://www.delta-js.ch/cms/upload/pdf/english/technical-papers/Non-Synchronous\\_Tilting\\_Pad\\_Bearing\\_Characteristics\\_IFTOMM\\_2010.pdf](http://www.delta-js.ch/cms/upload/pdf/english/technical-papers/Non-Synchronous_Tilting_Pad_Bearing_Characteristics_IFTOMM_2010.pdf)
- /2/ J.Schmied, A. Kosenkov: Practical Controller Design for Rotors on Magnetic Bearings by Means of an Efficient Simulation Tool. 10. Internationale Tagung Schwingungen in rotierenden Maschinen (SIRM), Berlin, February 2013.