AMrotorSIM – Toolbox

Vorstellung zur ICAM2017

Johannes Maierhofer

12.04.2017

# Demoanwendung

Zur Demonstration der AMrotorSIM-Toolbox

Inhalt

[Demoanwendung 1](#_Toc479766910)

[Rotor configuration file 2](#_Toc479766911)

[Rotor 2](#_Toc479766912)

[Massescheiben 2](#_Toc479766913)

[Sensors 2](#_Toc479766914)

[Lager 2](#_Toc479766915)

[Loads 2](#_Toc479766916)

[Header 3](#_Toc479766917)

[Import 3](#_Toc479766918)

[Clean up 3](#_Toc479766919)

[Compute Rotor 4](#_Toc479766920)

[Running system analysis 5](#_Toc479766921)

[Running Time Simulation 7](#_Toc479766922)

# Rotor configuration file

## Rotor

cnfg.cnfg\_rotor.name = "Testrotor";  
cnfg.cnfg\_rotor.shear\_def = 1; % 1 = Timoshenko or 0 = Bernoulli beam theoy  
  
cnfg.cnfg\_rotor.rotor\_dimensions = [0,15,0;400,15,0;450,150,0;600,15,0]\*1e-3; %[m] cnfg.cnfg\_rotor.shear\_factor = 0.9;   
  
cnfg.cnfg\_rotor.E\_module = 211e9; %[N/m^2]  
cnfg.cnfg\_rotor.density = 7446; %[kg/m^3]  
cnfg.cnfg\_rotor.poisson = 0.3; %steel 0.27...0.3 [-]  
  
cnfg.cnfg\_rotor.dz=0.01; %set elemet length for meshing

## Massescheiben

cnfg.cnfg\_disc(1).name = 'Zusatzscheibe';  
cnfg.cnfg\_disc(1).position = 300e-3; %disc position [m]  
cnfg.cnfg\_disc(1).radius = 200e-3;  
cnfg.cnfg\_disc(1).m = 5; %disc mass [kg]  
cnfg.cnfg\_disc(1).Ix = 1e-4; %disc mom. of inertia [m^4]  
cnfg.cnfg\_disc(1).Iz = 1e-4; %disc mom. of inertia [m^4]

## Sensors

cnfg.cnfg\_sensor(1).name = 'Sepp';  
cnfg.cnfg\_sensor(1).position=0.050;  
cnfg.cnfg\_sensor(1).type=1;  
  
cnfg.cnfg\_sensor(2).name='Hans';  
cnfg.cnfg\_sensor(2).position=0.550;  
cnfg.cnfg\_sensor(2).type=1;

## Lager

cnfg.cnfg\_lager(1).name = 'Linkes Lager';  
cnfg.cnfg\_lager(1).position=0e-3; %[m]  
cnfg.cnfg\_lager(1).type=1;  
cnfg.cnfg\_lager(1).stiffness=5e6; %[N/m]  
  
cnfg.cnfg\_lager(2).name = 'Rechtes Lager';  
cnfg.cnfg\_lager(2).position=600e-3; %[m]  
cnfg.cnfg\_lager(2).type=1;  
cnfg.cnfg\_lager(2).stiffness=5e6; %[N/m]

## Loads

cnfg.cnfg\_force\_const\_fix=[];  
% Unwuchten  
cnfg.cnfg\_unbalance(1).name = 'Geplante Unwucht';  
cnfg.cnfg\_unbalance(1).position = 300e-3;  
cnfg.cnfg\_unbalance(1).betrag = 5e-1;  
cnfg.cnfg\_unbalance(1).winkellage = 0;

# Header

% Johannes Maierhofer  
% 28.03.2017,29.03.2017,30.03.2017,31.03.2017,03.04.2017,04.04.2017,05.04.2017,06.04.2017,12.04.2017  
%  
% .o. ooo ooooo .  
% .888. `88. .888' .o8  
% .8"888. 888b d'888 oooo d8b .ooooo. .o888oo .ooooo. oooo d8b  
% .8' `888. 8 Y88. .P 888 `888""8P d88' `88b 888 d88' `88b `888""8P  
% .88ooo8888. 8 `888' 888 888 888 888 888 888 888 888  
% .8' `888. 8 Y 888 888 888 888 888 . 888 888 888  
% o88o o8888o o8o o888o d888b `Y8bod8P' "888" `Y8bod8P' d888b

## Import

import AMrotorSIM.\*

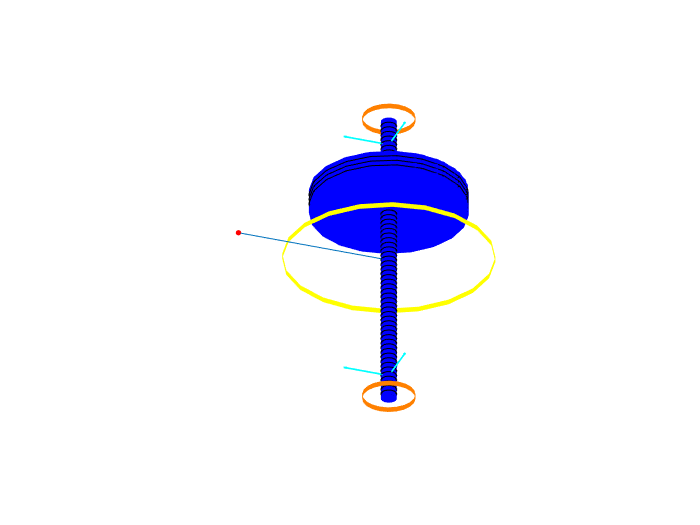
## Clean up

close all  
clear all  
clc

# Compute Rotor

Config\_Sim  
  
r=Rotorsystem(cnfg,'System');  
r.show;  
  
r.rotor.mesh()  
  
g=Graphs.Visu\_Rotorsystem(r);  
g.show();  
  
r.compute\_matrices();  
r.compute\_loads();  
%r.reduce\_modal(10);

--------------- Rotorsystem --------------  
System  
Testrotor  
Zusatzscheibe  
Linkes Lager  
Rechtes Lager  
----------------------------------------------  
--------------- Sensors ------------------------  
Sepp  
Hans  
----------------------------------------------  
--------------- Loads ------------------------  
Geplante Unwucht  
----------------------------------------------  
Mesh ....  
Rotor Gesamtsystem  
Berechne Lagersteifigkeit  
Berechne Lagersteifigkeit



# Running system analysis

m=Experiments.Modalanalyse(r);  
  
% m.calculate\_rotor\_only(4,0:100:1000);  
% esf = Graphs.Eigenschwingformen(m);  
% esf.plot();  
  
m.calculate\_rotorsystem(4,0:100:1000);  
esf2= Graphs.Eigenschwingformen(m);  
esf2.plot();  
  
m.calculate\_rotorsystem(3,0:100:3000);  
cmp = Graphs.Campbell(m);  
cmp.plot();

Berechne Modalanalyse Rotorsystem  
Eigenkreisfrequenzen  
18.8487 Hz  
94.1186 Hz  
158.7975 Hz  
451.9409 Hz  
Berechne Modalanalyse Rotorsystem

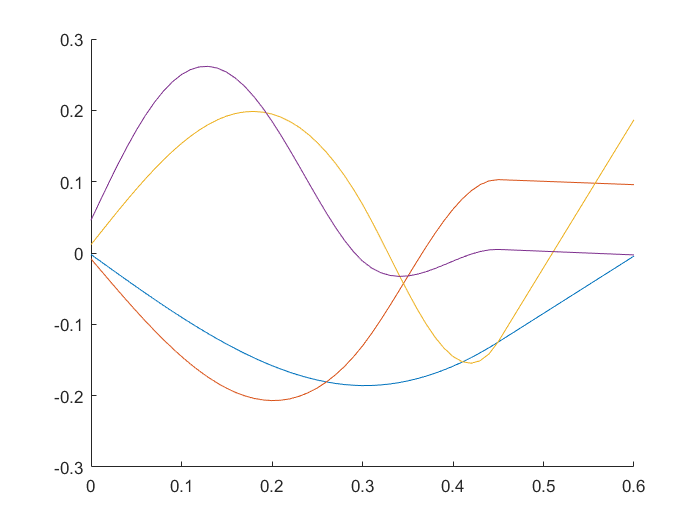


Abbildung : Eigenformen des Rotorsystems

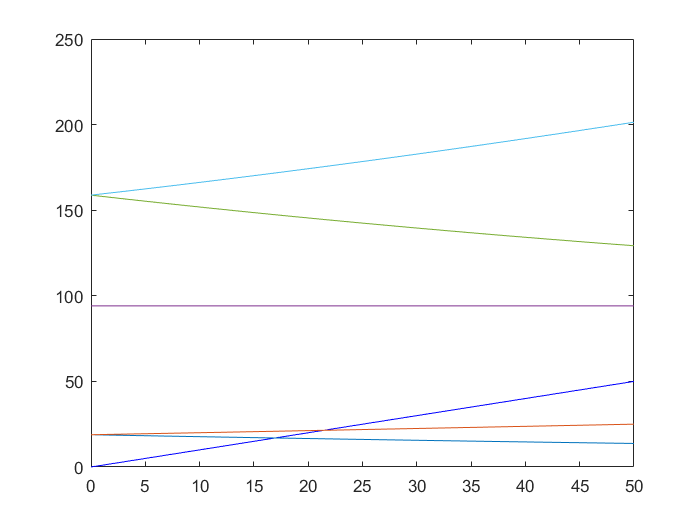


Abbildung : Campbelldiagramm des Rotorsystems

# Running Time Simulation

St\_Lsg = Experiments.Stationaere\_Lsg(r,1000,[0 2]);  
 St\_Lsg.show()  
 St\_Lsg.compute()  
  
 w = Graphs.Wegorbit(r);  
 w.plot(r.sensors);

Stationäre Lösung  
Compute.... ode15 ....  
 --- Plot Wegorbit ---

