Software Lab WT 2006/07

contact person: Dipl.-Ing. M. Buchschmid, Dipl.-Ing. M. Egger

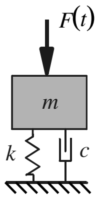
bau mechanik tu münchen

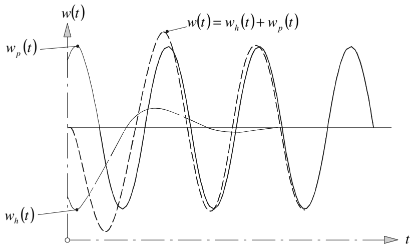
Visualisation of vibrating SDOF-Systems

Overview

A software tool has to be developed, which can be used in “Structural Dynamics”–lectures in order to visualize the vibration of SDOF-systems for different types of loads to convey general terms in dynamics like natural frequency, damping, resonance or dynamic amplification to the students.

System

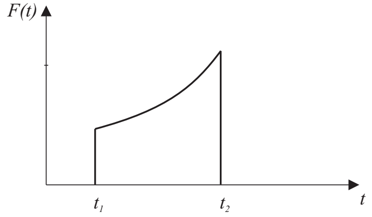




types of vibration

1. Free vibration for given initial conditions $w(t=0), \dot{w}(t=0)$
2. Forced vibration for
   a. Harmonic loads $F(t)=F_0 \cdot \sin(\Omega \cdot t)$
   (Ω, m, k, c are given!)
   b. Periodic loads $F(t)=C_0+\sum_{n=1}^{n_{\max}} C_n \cdot \sin(n\frac{2\pi}{T}t+\alpha_n)$
   ($n_{\max}, C_n, \alpha_n, m, k, c$ are given!)
   c. Arbitrary non periodic loads

$$F(t)=A_n(t-t_1)^n+A_{n-1}(t-t_1)^{n-1}+\ldots+A_1(t-t_1)+A_0$$



($t_1, t_2 n, A_n, m, k, c$ are given!)

3. Root Point excitation
   $w(t), \dot{w}(t), or\ \ddot{w}(t)$ are given as
   a. Harmonic
   b. Periodic
   c. Arbitrary non-periodic
   Function (like shown for forced vibration)

Usability

Different menus should be provided, containing the following features:

1. System
   a. Definition of m, k, c
   b. Output of natural (circular) frequency in order to check the input

2. Loads
   a. Definition of different load-cases (Superposition for Output)
   b. Graphical output of the load in order to check the input

3. Output (choose boxes for)
   a. Animated graphics of the vibration system
   b. Displacement $w(t)$
   c. Velocity $\dot{w}(t)$
   d. Acceleration $\ddot{w}(t)$
   e. (Amplification function $V(\eta)$)

4. Setup
   a. Language for output: English or German
   (Language for input is German)
   b. Colours and fonts

Organisation and support

1. General solution of the mechanical problems, which were mentioned above using a computer algebra system with a symbolic processor (e.g. maple) with the aim of understanding the problems, finding enclosed solutions and verifying these solutions.
   (This part is supported by Lehrstuhl für Baumechanik)

2. Structuring the results of 1.) and developing a concept to build the required software out of the given data structure and choosing a language (e.g. Java)
   (This part is supported by Lehrstuhl für Baumechanik and Lehrstuhl für Bauinformatik)

3. Programming the software
   (This part is supported by Lehrstuhl für Bauinformatik)