MATEC Web of Conferences **1**, 03002 (2012)
DOI: 10.1051/matecconf/20120103002
© Owned by the authors, published by EDP Sciences, 2012

Time History Forced Response in Nonlinear Mechanical Systems

K. Ahlin[1], A. Josefsson[1] and M. Magnevall[1]

[1]Blekinge Institute of Technology, Karlskrona, Sweden

Abstract. A formulation of a digital filter method for computing the forced response of a linear MDOF mechanical system is proposed. It is shown how aliasing error effects can be avoided at the expense of a bias error. The bias error is however completely known and it is system independent, as it only depends on the sampling frequency used. The mechanical system is described by its modal parameters, poles and residues. The method is extended to include non-linear elements. A toolbox in MATLAB has been created where nonlinear elements with and without memory can be treated, as well as system described by coupled non-linear equations.

1 Introduction

There exist several methods to obtain the time history response of a mechanical system to an arbitrary excitation. A few of the time domain methods are: the Duhamel integral, the state transition matrix method, and the Runge-Kutta method with variations. The Duhamel integral, also known as the convolution integral, is a well-established method in the literature, its main drawback is the computational burden associated with its non-recursive nature. However, the integral can be used to derive a recursive algorithm, a digital filter, which leads to a faster solution than the naive Duhamel integral method. In the paper, the digital filter method is introduced on linear systems. The method is then extended to nonlinear systems with localized non-linearities.

2 Establishing a Recursive Solution

Working on discrete sampled data necessitates transformation of a model in the continuous time domain into the discrete time domain. The basis for a family of methods is the convolution integral:

$$y(t)=\int_0^t h(t-\tau)\cdot x(\tau)d\tau \tag{1}$$

where x(t) is the input signal, h(t) is the system impulse response and y(t) is the output signal. The convolution integral may be used for any linear, time invariant system. We start by studying the simple analog system:

$$H(s)=\frac{1}{s+a} \tag{2}$$

where s is the Laplace variable. The system in equation (2) has the impulse response:

$$h(t)=\exp(-at) \tag{3}$$

We now calculate y(nT+T), where T is the sampling interval:

$$\begin{aligned}
y(nT+T) &= \int_0^{nT+T}\exp(-a(nT+T-\tau))\cdot x(\tau)d\tau \\
&= \exp(-aT)\cdot\left\{\int_0^{nT+T}\exp(-a(nT-\tau))\cdot x(\tau)d\tau\right\} \\
&= \exp(-aT)\cdot\left\{\begin{aligned}&\int_0^{nT}\exp(-a(nT-\tau))\cdot x(\tau)d\tau+\\&\int_{nT}^{nT+T}\exp(-a(nT-\tau))\cdot x(\tau)d\tau\end{aligned}\right\} \\
&= \exp(-aT)\cdot y(nT) \\
&\quad+\exp(-aT)\cdot\int_0^{T}\exp(au)\cdot x(u+nT)du
\end{aligned} \tag{4}$$

From equation (4) we find that we may calculate y(nT+T) with a recursion formula using only y(nT) and the input signal in the time interval [nT, nT+T]. The system influence shows in the terms exp(-aT) and exp(au) in the last integral in equation (4). The way we use samples of x(t) to evaluate the last integral in equation (4) defines the design method, see figure 1. A recursion formula may be thought of as a digital filter.

As an example, the step invariant method uses the constant value x(nT) in the whole interval, which, using equation (4), gives:

Article available at http://www.matec-conferences.org or http://dx.doi.org/10.1051/matecconf/20120103002

$$
\begin{aligned}
y(nT+T) &= \exp(-aT)\cdot y(nT) \\
&+ x(nT)\cdot\frac{\{1-\exp(-aT)\}}{a}
\end{aligned}
\tag{5}
$$

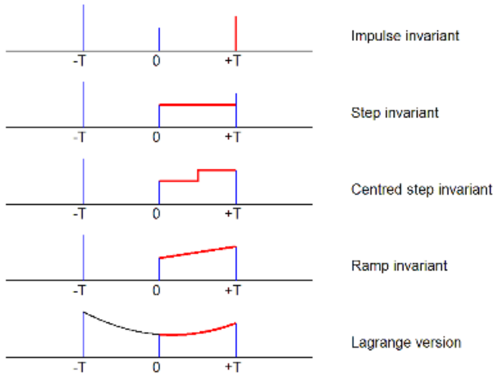

Fig. 1. Different approximations of x(t) in the interval [0, T] from the samples of x(t) define different filter design methods.

A difference equation, such as equation (5), may be regarded as a digital filter. A general expression would be:

$$
y(n) = \sum_{i=0}^{p} b_i \cdot x(n-i) - \sum_{j=1}^{q} a_j \cdot y(n-j)
\tag{6}
$$

3 Application to Linear Mechanical Systems

There are various ways to represent a mechanical system: through a Finite Element Model (FEA) model, a lumped MCK model or an analytical model. All of them can provide quantitative information about the modal parameters, i.e. poles and residues of the system. Here we note that these parameters can as well be obtained by performing an experimental modal analysis on the mechanical system. See figure 2.

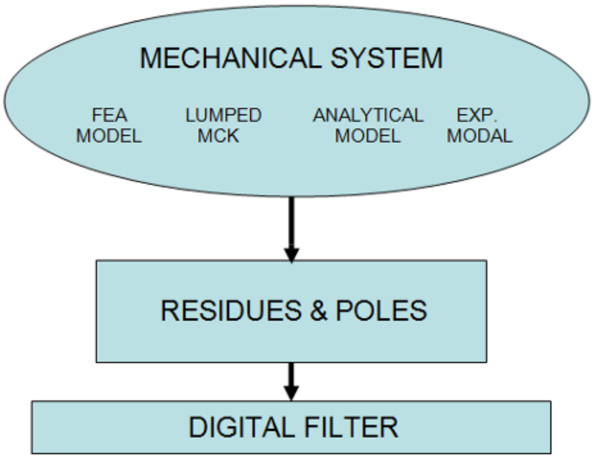

Fig. 2. A linear mechanical system may be characterized by its residues and poles. These may come from different models of the system, or from experiments. The residues and poles then define the corresponding filters.

According to the modal superposition theorem for a MDOF system, consisting of N modes, the frequency response function H(s) connecting one input degree-of-freedom with one output degree-of-freedom can be expressed using partial fraction expansion in terms of the residues R_r and poles s_r as follows):

$$
H(s) = \sum_{r=1}^{N} \frac{R_r}{s-s_r} + \frac{R_r^*}{s-s_r^*}
\tag{7}
$$

For the impulse invariant transform, equation (4) gives:

$$
\begin{aligned}
&y(nT+T) = \exp(-aT)\cdot y(nT) + T\cdot x(nT+T) \\
&\textit{or} \\
&y(n) = \exp(-aT)\cdot y(n-1) + T\cdot x(n) \\
&\textit{or} \\
&Y(z) = X(z)\cdot\frac{T}{1-z^{-1}\cdot\exp(-aT)}
\end{aligned}
\tag{8}
$$

For a complex conjugate pair with residue R_r and pole s_r, as in equation (7), we will get:

$$
\begin{aligned}
&H(z) = \frac{T\cdot R_r}{1-z^{-1}\cdot\exp(s_r T)} + \frac{T\cdot R_r^*}{1-z^{-1}\cdot\exp(s_r^* T)} \\
&\textit{with} \\
&R_r = A + jB \\
&s_r = \sigma + j\omega \\
&\textit{we get} \\
&H(z) = \frac{2A - z^{-1}\cdot\exp(\sigma T)\cdot(2A\cdot\cos(\omega T) - 2B\cdot\sin(\omega T))}{1-2z^{-1}\cdot\exp(\sigma T)\cdot\cos(\omega T) + z^{-2}\cdot\exp(2\sigma T)}
\end{aligned}
\tag{9}
$$

Equation (9) seems as a complicated expression, but may be performed by a few simple steps in MATLAB. In our MATLAB toolbox, the calculation of filter coefficients for one pole pair in the ramp invariant case looks like:

```
N = R/S/S/T*[-1-S*T+exp(S*T),1+(S*T-1)*exp(S*T)];
D = [1 -exp(S*T)];
B = 2*conv(real(N),real(D))+2*conv(imag(N),imag(D));
A = real(conv(D,conj(D)));
```

where

R	complex residue
S	complex pole
B	filter numerator coefficients
A	filter denominator coefficients

To get the complete response, the contributions from all wanted modes are just added.

4 Error Calculations

As in all sampled systems, we will have aliasing. It doesn't help us if our input signal x(t) has aliasing protection, as the system response, h(t), in general has not. As a result, the spectrum of our response will have contributions from the original spectrum Y(f) centred around multiples of the sampling frequency f_s. The aliased spectrum $Y_a(f)$ will be:

$$Y_a(f) = Y(f) + \sum_n Y(f - n \cdot f_s) + Y(f + n \cdot f_s) \tag{10}$$

The error resulting from aliasing is system dependent. If the system frequency response has large amplitude at the sampling frequency, the error in the low frequency region may be considerable, due to the fold back. This error when we use the impulse invariant method is well known and has led to the recommendation not to use impulse invariant technique when handling response to SDOF systems, for instance when calculating shock response spectra. In an international standard, ISO 18431-4, the ramp invariant method is recommended instead.

It is not so well known, however, why other methods lead to better result. The handling of the input x(t) in the last integral in equation (4) and figure 1 may be seen as a convolution between the input samples and a convolution kernel. In the case of ramp invariant, the convolution kernel w(τ) is:

$$w(\tau) = \begin{cases} 0 & \tau < -T \\ \frac{1}{T} \cdot (\frac{\tau}{T} + 1) & -T \le \tau \le 0 \\ \frac{1}{T} \cdot (-\frac{\tau}{T} + 1) & 0 \le \tau \le T \\ 0 & \tau > T \end{cases} \tag{11}$$

which is a triangle with base from –T to +T. As we have a convolution in the time domain, this corresponds to a multiplication in the frequency domain. The multiplier is the Fourier transform of the convolution kernel. For the ramp invariant the amplitude of the Fourier transform is:

$$|F(w(t))| = \left(\frac{\sin(\frac{\pi \cdot f}{f_s})}{\frac{\pi \cdot f}{f_s}} \right)^2 \tag{12}$$

This function is zero at the sampling frequency and quite flat in the frequency region around, which helps to reduce the aliasing error considerably. The price to pay is a bias error in the low frequency domain, but it is known and starts at zero for low frequencies. The aliasing error is dependent on the system and the chosen sampling frequency, while the bias error (for a certain frequency) is only depending on the sampling frequency. This makes the ramp invariant version the preferred one in many applications, such as shock response spectrum calculations. Another benefit is that (as opposed to step invariant) there is no phase error for the ramp invariant. This is very important when we study nonlinear systems.

5 Forced Response in Nonlinear Systems

As a simple case, we consider a non-linear MDOF system for which both the applied force and the non-linearity are at the same degree of freedom as shown in Figure 3. More specifically, the single non-linear component is located between the input force DOF and ground.

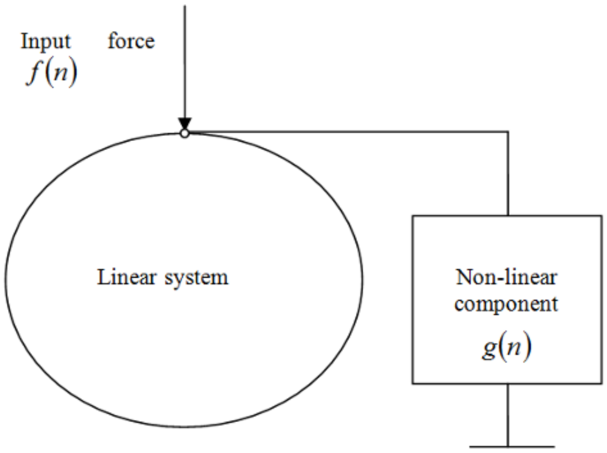

Fig. 3. A non-linear component is introduced to a linear system. The input force and the non-linearity are applied at the same degree-of-freedom.

The equation of motion for the system of interest in discrete time domain is governed by the following equation:

$$M\ddot{x}(n) + C\dot{x}(n) + Kx(n) + g(n) = f(n) \tag{13}$$

with the usual notations and where g(n) represents a single non-linear component which is a function of x. Equation (13) in the frequency domain (z-plane) is conveniently written as:

$$B(z) \cdot X(z) + G(z) = F(z) \tag{14}$$

where B(z) is the system impedance matrix and from which equation the system response X(z) can be obtained as:

$$X(z) = H(z) \cdot [F(z) - G(z)] \tag{14}$$

With H(z) as the system Frequency Response Matrix. With equation (6) written in the z domain we may write for the r:th mode:

$$H_r(z) = \frac{b_0^r + b_1^r \cdot z^{-1} + b_2^r \cdot z^{-2}}{1 + a_1^r \cdot z^{-1} + a_2^r \cdot z^{-2}} \tag{15}$$

where we use the superscript r to mark the mode number. Re-writing the resulting relationship in the time domain gives the system response at time n for the r^{th} mode as:

$$\begin{aligned} x_r(n) &= b_0^r\, f(n) - b_0^r\, g(n) \\ &+ b_1^r\, f(n-1) - b_1^r\, g(n-1) + b_2^r\, f(n-2) \\ &- b_2^r\, g(n-2) - a_1^r\, x_r(n-1) - a_2^r\, x_r(n-2) \end{aligned} \tag{16}$$

Equation (16) is identified as a recursive equation which can be written down for each mode r (i.e. for each digital filter). Using the superposition theorem then, the total response is the sum of each of the individual digital filters. Taking a step further, Equation (16) can be rewritten in a compact fashion for each mode r and at time n as follows:

$$x_r(n) + b_0^r\, g(n) = Q_r \tag{17}$$

where it can be seen that the term Q_r embodies the terms in the second line of equation (16) and also the term $b_0^r\, f(n)$. In this respect we note that Q_r is completely determined by past parameter values together with the current applied force f(n). Now, equation (17) can be recognized as being a non-linear equation in x with g being some non-linear function of x. Equation (17) is computed for each mode r and then summed up for all the modes; after which the resulting equation is solved to yield the desired total response at time step n. The procedure is summarized in figure 4.

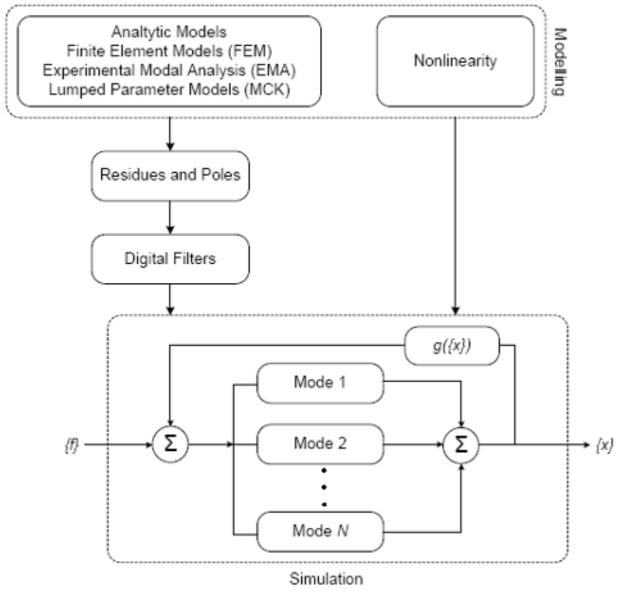

Fig. 4. Illustration of the forced response simulation method. The underlying linear system is described by its residues and poles and the nonlinearities are modeled as additional external forces.

6 Extensions

The idea described here has been extended in many ways, and a toolbox with simulation functions has been created in MATLAB.

The functions have been extended to incorporate many non-linearities, coupled between different degrees of freedom for the linear system. This means that we have to solve a non-linear system of equations for each time step.

Very efficient routines to accomplish that have been implemented into the scripts. In order to make the execution fast, the non-linearities have to be defined within the script.

As an example, the function *nonlindxmck* has the following syntax:
x = nonlindxmck(F, fs, M, C, K, FD, nlD, modes, resps)

The input is the force vector F with a sampling frequency of fs. The linear system is in this case defined by the matrices M, C and K. The force is applied to degree of freedom FD, and the non-linearities are applied to degrees of freedom, DOFs, defined by nlD. Modes is a vector containing the modes that we want in the calculation. Finally, resps is a vector defining in which degrees of freedom we want the response(s).

As an example, let the system be a simple four degree of freedom linear system. We apply one cubic nonlinearity from one DOF to ground and we have another cubic nonlinearity between two other DOFs. We then apply a random force to the remaining DOF and calculate all four responses. See figure 5. To calculate all the responses in 100,000 samples takes 11 seconds on a laptop computer.

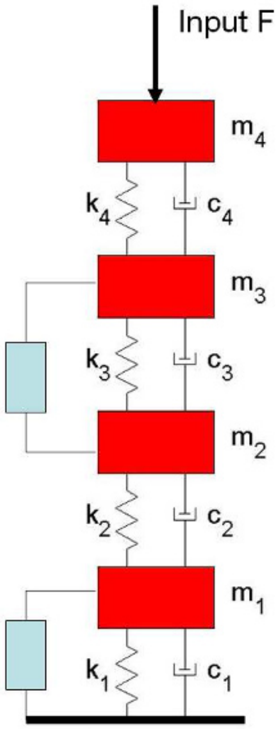

Fig. 5. A simple four degree-of-freedom linear system with two attached cubic non-linearities.

7 Summary

In this paper, a formulation of a digital filter method for computing the forced response of a linear MDOF mechanical system has been proposed. It is shown how aliasing (and phase) error effects can be avoided at the expense of a bias error. The bias error is however completely known and it is system independent, as it only depends on the sampling frequency used. The mechanical system is described by its modal parameters, poles and residues.

The method is extended to introduce non-linear elements. A toolbox in MATLAB has been created where nonlinear elements with and without memory can be treated, as well as system described by coupled non-linear equations. The functions are very effective and the computation time is minimized.