Acoustic emission source localisation in thin plates through a dispersion compensation approach

Krzysztof GRABOWSKI *, Mateusz GAWROŃSKI *, Wiesław Jerzy STASZEWSKI *, Tadeusz UHL *, Ireneusz BARAN **, Wojciech SPYCHALSKI ***, Paweł PAĆKO *

* AGH University of Science and Technology, Krakow, Poland
** Cracow University of Technology, Krakow, Poland
*** Warsaw University of Technology, Warsaw, Poland

Prague, 06.10.2014
1. Introduction
2. TDDT algorithm
3. Numerical Simulation
4. Experimental Verification
5. Conclusions
Introduction

While identifying acoustic emission events it is important to remove the dispersion phenomena.

- Time reversal process
- Linear mapping
- Wrapped frequency transform
- TDDM
- Etc.

Very often TDDM (time-distance domain mapping) is used, however this technique deforms the signal (after removing the dispersion).

Therefore, there have been developed technique for active sensing by Cai et al. 2013 called - TDDT (time-distance domain transform), which not only removes the dispersion but gives undeformed signal.
Dispersion removal through the TDDT

Issues in current localisation:
- Arbitrary assumption of wave velocity in the material
- Broadband signals
- Dispersion phenomena

Given:
- Position of sensors
- Geometry and properties of the structure

Unknown:
- Localisation of AE event

$\text{inv}(f-k)$
Considering PZT sensors, Lamb wave, single wave packet in frequency domain we can describe as:

\[ V(\omega) = E_a(\omega)E_s(\omega)G(r_0, \omega)V_a(\omega) \]  \hspace{1cm} (1)

Where \( V_a(\omega) \), \( V(\omega) \), \( E_a(\omega) \) and \( E_s(\omega) \) are excitation signal, signal received by the sensors, and electro-mechanical efficiency of the PZT respectively. Considering transfer function from the source signal to the sensor we may further denote it as:

\[ V(\omega) = E_a(\omega)E_s(\omega)V_a(\omega)A(r_0, \omega)e^{-iK(\omega)r_0} \] \hspace{1cm} (2)

\[ = V_a(\omega)H(\omega), \]

Where \( H(\omega) = H[K(\omega)] = e^{-ikr_0}|_{k=K(\omega)} \) is transfer function of the whole procedure considering Lamb waves, its’ propagation and receive.
TDDT Algorithm

Considering following equations we may note that the phase of the received signal is reduced by $K(\omega)r_0$ comparing to the source signal. Considering dispersion curve $K(\omega)$ is nonlinear function dependent on the frequency. Therefore, different frequencies of the signal will have different time delays.

TDDT transforms original signal in the time domain for the signal within the spatial domain considering wave number with correlation to the dispersion phenomena. Concluding we may simply note the equation for the signal received in spatial domain as:

$$v(r) = v_a(r) * h(r)$$ \hspace{1cm} (3)

In broadband TDDT, distance is received by interpolation transfer function of the plate through the dispersion curve (sampling is done by the wave numbers).

$$H(k) = H[\Omega(k)]$$

$$\Omega(k) = K^{-1}(\omega),$$ \hspace{1cm} (4)
We may proceed with the same way with source signal:

\[ V_a(k) = V_a[\Omega_{\text{non}}(k)] \]
\[ \Omega_{\text{non}}(k) = c_g k, \]

therefore our final signal in the spatial domain looks as following:

\[ \nu(r) = \text{IFFT}[V_a[\Omega_{\text{non}}(k)]] \ast \text{IFFT}[H[\Omega(k)]] \]

Moreover, for the narrowband signals we may introduce correction factor:

\[ \nu(r) = \text{IFFT}[V_a[\Omega_{\text{non}}(k)]H[\Omega(k)]] \]
\[ = \text{IFFT}[(V_a[\Omega(k)]H[\Omega(k)]) \times (V_a[\Omega_{\text{non}}(k)]/V_a[\Omega(k)])] \]
\[ = \text{IFFT}[V[\Omega(k)]C(k)], \]
Dispersion removal through the TDDT

TDDT algorithm – step by step.
The following governing equation is considered:

\[
\frac{\partial}{\partial x_l} \left( S_{klmn} \frac{\partial w^m}{\partial x_n} \right) + f_k = \rho \frac{\partial^2 w^k}{\partial t^2} + R_k \frac{\partial w^k}{\partial t}, \quad k, l, m, n = 1, 2, 3
\]

\[
AW_{xx} + BW_{yy} + CW_{xy} + f = \rho W_{tt} + R W_t
\]

For the numerical simulations – Local Interaction Simulation Approach, which is widely used by our department have been chosen.

Material is homogenous, orthotropic (for each cell)

\[
W^{n+1} = 2W - W^{n-1} + f(W_i, W)
\]
Numerical simulation - CUDA-based LISA solver

Viewer

OpenGL-based tool for large models visualization (36m in DOF)

Arbitrary geometry import from FE meshes

CAD geometry  →  FE mesh  →  Grid generation
Numerical simulation - Coupled LISA-FE piezo-transducers modelling

Coupled FE piezo-transducers modelling

FE transducer model → Data conversion module → Wave propagation simulation (CUDA)

Solution time [min]

FE

FE+LISA

Pure FE solution

FE-LISA solution

FE – 24h
FE+LISA – 15+4 min = 19 min

500x500x2mm, aluminium
Numerical simulation - Developed system

PRE/POST-PROCESSOR

MATLAB®

FINITE ELEMENT METHOD

COUPLING

SIMULATION SETTINGS

RESULTS

SIMULATION OF WAVE PROPAGATION PHENOMENA

PARALLEL COMPUTING

EXPERIMENTAL DATA

SIMULATION-BASED SETTINGS

EXPERIMENT
Numerical simulation – input parameters

Creating model requires few essential information about object:

- Time step: \( dt \)
- Number of cells per unit: \( dc \)
- Object dimensions: \( x, y, z \) [mm] (for Cartesian)
- Width of damping layer: \( ndw \)
- Maximum value of damping: \( dvm \)

Excitation signal parameters for actuators such as:

- Frequency: \( freq \) [Hz]
- Number of periods: \( nT \)

or input data file with amplitude values for every iteration step.

Generated signal example
Numerical simulation – How does it work?

For specified **dimensions** and **dc**, density of cubes is calculated:
Example: \(x = 2 \text{ [mm]}, \ y = 2 \text{ [mm]}, \ z = 1 \text{ [mm]}\) \(dc = 0.5\) results in 16 cubes.

**Every element** has its state calculated individually based on the state of the neighbours and its own for specified parameters.
Using MatLab scripts or modifying data files, user defines positions of sensors and actuators in simulation as well as the definition of extorting signals, material parameters and cube positions in model.

Properly defining **time step** for our simulation is crucial. **Too small** value results in wave propagation error. **Maximum** wave speed is calculated from **dc** and **dt values**. With \(dc = 1\) and \(dt = 0.1\), we acquire maximum speed of 10000 m/s (Nie jestem pewien!! Trzeba się upewnić u Pawła).
First the method was verified using numerical simulations. For this purpose, the Graphical Processing Units (GPU) based Local Interaction Simulation Approach (LISA) has been used. A 3D model of a plate has been built and excited using a model source. The time signal representing the source was obtained through an inverse analysis of experimental test (Worden et. al. 2011). Virtual sensors were placed at the plate’s surface to collect time responses. Then, the TDDT was applied and damage localisation performed.
Numerical simulation – test case 2
Numerical simulation – test case 1

HSU – 200 mm time domain

HSU – 200 mm distance domain

HSU - 400 mm time domain

HSU - 400 mm distance domain
Steel material properties were assumed as:

\[ E = 180 \, 000 \, [\text{MPa}] \quad \rho = 7500 \, \left[ \frac{\text{kg}}{\text{m}^3} \right] \quad \text{poisson ratio} = 0.33 \, [-] \]

For the magnets there were assumed:

\[ E = 230 \, 000 \, [\text{MPa}] \quad \rho = 7500 \, \left[ \frac{\text{kg}}{\text{m}^3} \right] \quad \text{poisson ratio} = 0.24 \, [-] \]

Responses were acquired as time signals of out-of-plane displacement component of a single grid point.

Excitation was applied in several positions to verify the proposed approach. The plate was excited by two previously identified time signals modelling a HSU-like source. The responses were recorded by four sensors. The transient dynamic analysis consisted of 20000 time steps.
Numerical simulation – test case 2

Results obtained after importing sensor data to MATLAB and running proposed localization method are summarized below.

**Red stars** show localization results, **black circles** are points, where actuators were placed. **Blue circles** are sensors positions

It can be seen that the accuracy of the algorithm is approximately 75% when it comes to events detection. For very close and very far excitation points, the method was not found accurate due to edge reflections and internal reflection from the magnets.
Numerical simulation – pros and cons

**Advantages** of LISA approach:

- Flexible environment, allowing for creation of complicated objects
- Ability to adjust simulation parameters for the needs of the research
- Clear visualization of final results
- Fully compatible with MatLab environment
- Easy way to import external data to files (.txt format)

**Disadvantages** of LISA approach:

- Object creation through code only which requires more advanced programming skills of the user
Experimental Setup

Set of experiments were performed to verify numerical simulation. Measurement hardware used was Vallen GMBH units with four channels and National Instruments PXI-1093 with FlexRIO card.

Four sensors of type VS150-M with preamplifiers AEP4H were used with Vallen and pinducersr – broadband type of PZT sensors were used with NI PXI.

**Measurement details:**
- Aluminum plates
- Different configuration of sensors
- Localization
- Source identification (HSU)
- Dispersion removals
Experimental setup 1

An aluminium plate of dimensions 500x500x2 [mm], was equipped with 3 sensors.

The setup was analogous to numerical model. Measurement hardware used was Vallen GMBH unit with four channels.
Experimental setup 1
Experimental setup 2

Second test case – alliminium thin (2 mm) plate with the notch
Experimental setup 2

[Graphs showing data points and distribution]

25
Conclusions:

- Proposed algorithm has shown promising results when applied to localization of artificial sources in metallic plates.
- As the method is still in early development stage, there is great potential for creating an efficient tool for AE source localisation.
- The ultimate goal of the work is to avoid providing tuneable parameters for damage localisation. Instead, only physical parameters, such as elastic constants and plate thickness are used.
Thank you for your attention.

Questions?