Online Bridge Vibration Analysis and Intelligent Fault Detection Approach using FFT-ANN on an embedded system

Jeferson González G.
Escuela de Ingeniería Electrónica
Instituto Tecnológico de Costa Rica
jefg89@gmail.com

Yeiner Arias E.
Escuela de Ingeniería Electrónica
Instituto Tecnológico de Costa Rica
yeinerarias@gmail.com

Johan Carvajal G.
Escuela de Ingeniería Electrónica
Instituto Tecnológico de Costa Rica
johcarvajal@tec.ac.cr

***Abstract*—This paper presents an approach for bridge vibration analysis and fault detection based on the Discrete Fourier Transform (DFT) and Artificial Neural Networks (ANN). The final implementation of the experiment was successfully completed on an Intel Galileo Board, running a customized embedded Linux distribution created with the Yocto Project with a Board Support Package (BSP). The chosen DFT implementation was the FFTW3 C library, ported to the Galileo Board via Yocto recipes. The artificial neural network was implemented and trained with the back-propagation algorithm based on expected outputs for both fault and normal conditions. Experiment results shown its is viable to implement this approach to isolate structural fault elements on a bridge in real time with more than 95% confidence level.**

***Keywords*—** FFT, ANN, Intel Galileo, Yocto Project, vibration, fault isolation, online analysis

I. Introduction

VIBRATION analysis is a technique widely implemented for the study of the dynamic effects on bridges due to live loads, as well as, for the evaluation of potential damage to the structure performance degradation. It is categorized, as a non-destructive test, which is an important aspect, since these structures are expensive, and destructive analysis are not convenient for the structure.

Vibration signal needs to be processed through a digital signal-processing algorithm in order to obtain relevant frequency information, the most common algorithm for this scenario is The Discrete Fourier Transform. The Discrete Fourier Transform (DFT) is a frequency analysis mechanism, by digital means, for non-periodical discrete signals [1].

The information obtained from the DFT, corresponding to the power spectral density for the vibration frequencies, can be analyzed and classified by an intelligent fault diagnosis system: an Artificial Neural Network (ANN). ANN system uses known patterns to establish whether the data collected behave as expected or if, instead, represents a fault behavior.

DFT and ANN algorithms require high computational power, in order to be computed successfully online on the bridge. Therefore, an embedded system is required to perform these operations; for the purposes of the experiment an Intel Galileo board was selected since it has a powerful processor, but with energy efficiency since it is a mobile application. A customize Linux distribution is created through the Yocto Project tool specifically for the Galileo device considering all the software dependencies.

Yocto Project is an open source initiative focused on embedded Linux developers, which allows the creation of customizable Linux images for different hardware platforms, through a Board Support Package (BSP), which is a collection of information about the hardware features in the device and kernel configuration information. Along with any additional required hardware drivers and libraries [2].

This paper will cover Section II, which describes the experiment and summarizes the theoretical aspects of each stage, Section III to presents the result of field application of this approach on an important bridge in Costa Rica, and Section IV to presents some relevant conclusion of this study and comment about further work that will be pursued.

II. Experiment Description

The vibration analysis method used for the experiment was the frequency displacement, wherein the detection of the damage is based on the changes of the natural frequencies of the structure, the foundation of the method is that the frequency at which the bridge vibrates is a function of its stiffness, so that a change in this parameter caused by any damage will be reflected as a decrease in the frequencies [3, 4].

According R. Salgado Estrada [5] the first natural frequency for most of the highway bridges with spans less than 100m is in the range between 1 and 11 Hz and a 19% of stiffness reduction in the structure produces a change in about 10% in the vibration frequencies, considering this as a severe damage.

The experiment developed for this document is divided in three stages: data acquisition, data filtering and processing using DFT, and fault detection through ANN implementation.

A. Data acquisition

Vibration measurement was performed point A (figure 2) by an ADXL335 accelerometer with a electrical sensitivity of 330 mV/g operating on 3,3V power supply and a typical measurement range of +-3,6 g. ADXL335 was read with the Intel Galileo using the built-in AD7298 ADC of the board with a resolution of 12 bits, with a sample frequency of 200 Hz, this process was done by an implemented C routine, running as a single thread in the user space application. Figure 1 shows the connection diagram for Intel Galileo and the ADXL335 accelerometer.

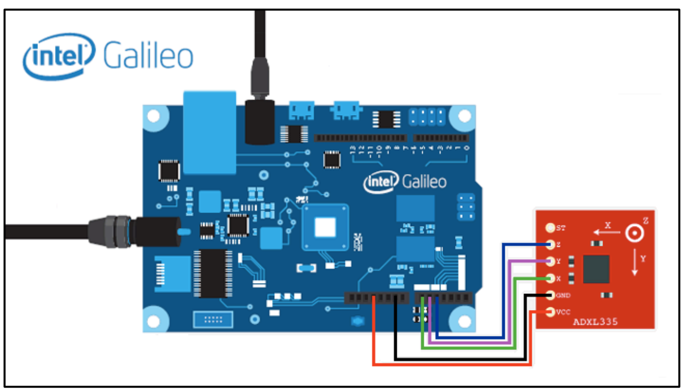

Figure 1. Connection diagram for the embedded system

For the purposes of this paper, only the *x-axis* component of the accelerometer was analyzed, since it is the most significant component of the bridge vibration for the implemented accelerometer installation on the field (see Figure 2).

The accelerometer was connected to Intel Galileo board through the analog input 0, and due to hardware constraints of speed, power and memory required to perform real time DFT and ANN computations that sample frequency was limited to 200 Hz.

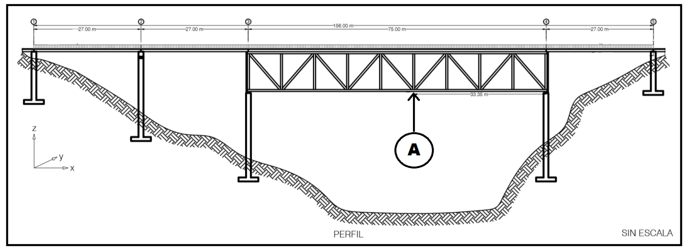

Figure 2. Measurement point in the Virillas's River Bridge

In order to perform the data collection for vibration signal on the bridge, the measurement device was placed at point **A** shown in the Figure. 2. The *x*-axis accelerometer is aligned with the *x*-axis of the bridge shown in the diagram, in which longitudinal vibration modes are measured more conveniently and effectively.

B. Discrete Fourier Transform (DFT)

Once the vibration signal is acquired is passed through the DFT, in which, given a discrete signal $x(n)$, the transformation to its sampled spectrum is [1]:

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} \,, k = 0,1,\ldots,N-1$$

In digital systems, DFT is usually implemented using the Fast Fourier Transform (FFT). Fast Fourier transform scales as $O\ (n \ \log\ n)$ problem, which require high performance processor to be implemented in real time.

Frigo and Johnson [6] developed a C subroutine library for computing *the discrete Fourier transform* in one or more dimensions for both real and complex numbers. It is widely used and very well known as the FFTW. (*Fastest Fourier Transform of the West*).

For any array of real number (case in study), the steps for computing DFT with the FFTW are:

1. Creating a FFTW complex (2D array of double) variable for holding the output data:
   *fftw_complex *out;*

2. Creating and set up a plan, in order to execute the DFT:
   fftw_planplan_forward;
   plan_forward = fftw_plan_dft_r2c_1d (n, in, out, FFTW_ESTIMATE);

3. Execute the plan
   fftw_execute(plan_forward);

The last step will save the FFT complex input data in the output variable defined in 1.

One useful property of DFT for an *n* size array of real data is the *Hermitian symmetry:*

$$Y[n-k] = Y^{*}[k]$$

Using this property, storage space and time can be reduced, by trimming redundant operations within the FFT, by roughly a factor of two [6].

FFTW process for the vibration signal leads to a 2D array of double precision numbers, with the real and imaginary part of the DFT in one dimension each.

In the natural frequency reduction analysis, it is more useful to obtain the magnitude of the DFT, not as function of the sample number k, but of the frequency of each component. Magnitude computing is just the absolute value of the complex number (i.e. square root of the sum of both components to square). Frequency reconstruction, in the other hand, depends directly of the process. Given an input array with N samples, a sample frequency F_S and a sample number k. Frequency value for each sample can be obtained as:

$$f[k] = k * \frac{F_S}{N}$$

Once the frequency array is computed, a graph is constructed, since it is a more intuitive representation of the DFT. An example of this process is shown in the Figure 3.

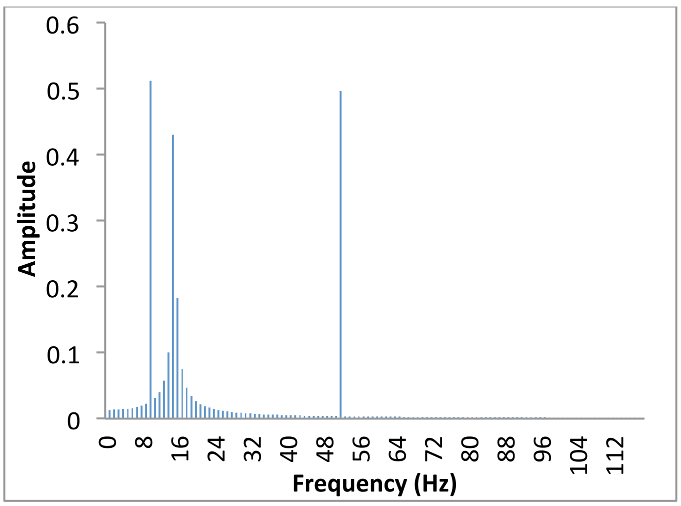

Figure 3. DFT for a composed random signal, using FFTW

C. *Artificial Neural Network (ANN)*

An artificial neural network (ANN) is an simplified model of the human neural system [7]. A single unit of this network is called *neuron*. Figure 4 represents a model for a neuron, which will be used in this approach for fault isolation due to vibration frequency shifts.

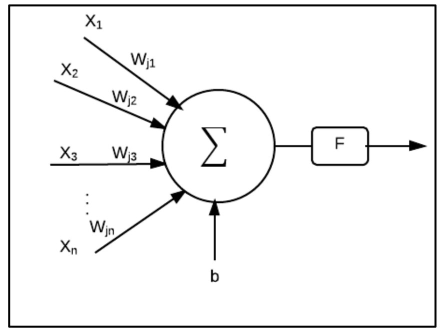

Figure 4. Representation of an artificial neuron

The output for a single neuron, like the one above, can be calculated as a function of the sum the X*i* input multiplied by its weight factor (W*ji)* plus a bias or offset value. For a *linear* neuron *F,* also called the activation function, the output value is 1, and the can be expressed as:

$$Fout = \sum_{i=1}^{N} X_i W_{ji} + b$$

Assuming this configuration for each neuron, an artificial neural network consists on certain number of neurons distributed in three or more layers: the *input layer,* (one or more) *hidden layers* and the *output layer*.

The architecture of the ANN chosen for this experiment is composed by one input layer, which contains a single input derived from the FFT spectrum, a hidden layer with ten neurons and an output layer with a single neuron. Figure 5 represents the artificial neural network (ANN) topology implemented in this experiment for structural fault isolation process.

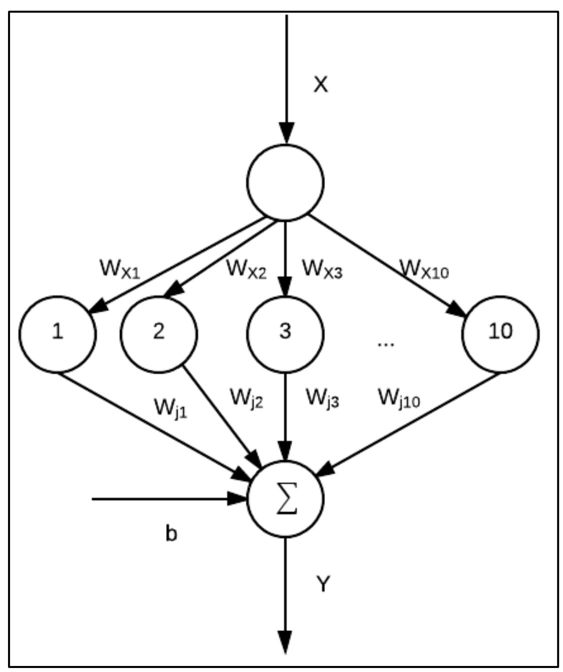

Figure 5. ANN architecture chosen for the experiment

The activation function for the hidden neurons was the widely known *sigmoidal function,* which gives the neurons a nonlinear behavior:

$$f(out) = \frac{1}{1+e^{-out}}$$

D. ANN Training

The artificial neural network, described above, was implemented from scratch using standard C-language as a routine planned to run on an individual OS thread. But before its final implementation, as usual, it was necessary to train the network with real data from Virilla´s river bridge vibrations, in order to "teach" (supervised learning) the normal behavior and some induced faults. Based on peak and frequency displacement of the FFT, the ANN would react and categorize the cases in which there was a fault trigger for the system. To implement this training, it was chosen the *Back Propagation Algorithm,* as described in [7, 8].

Back-propagation (BP) is a technique in which the network is trained with a set of inputs and expected outputs. For each iteration of the algorithm, weights are auto-adjusted in order to minimize the error (expected output - real output), this adjustment is calculated from the output to input in backwards direction, for a neuron in the output layer, this adjustment is:

$$W_{kj}(k+1) = \eta\delta_{pk}y_{pj} + W_{kj}(k)$$

Where η is the learning rate, δ_{pk} for a linear neuron is the difference between the expected output and the actual output of the net, and y_{pj} corresponds to the output for the j hidden neuron.

For the case of the hidden layer weights, hence the name of back-propagation, the delta (δ_{pj}) requires the delta value from the output layer δ_{pk}, and can be calculated as:

$$\delta_{pj} = y_{pj}(1-y_{pj})\sum_{k}\delta_{pk}W_{kj}$$

Where the first term arises from the derivative behavior of the sigmoidal activation function. The final weight adjustment for hidden layer neurons follows the procedure described for the output, which is described by:

$$W_{ji}(k+1) = \eta\delta_{pj}x_i + W_{ji}(k)$$

BP algorithm, described above, was implemented in a stand-alone C routine, and the weights obtained for each neuron in each layer were recorded and reproduced in the final implementation of the neural network.

III. Results

Data acquisition, for ANN training, using Intel Galileo, was performed at a sample rate of 200 Hz, filling a data buffer of 1000 samples (every 5 seconds). This process was repeated for about 30 minutes.

FFTW was applied, every 5 seconds to the updated data buffer and the result was written on a file located in the SD card of the board, in order to obtain enough information for training and validation of the neural network. The Figure 6 shows the FFT spectrum of the buffer at an arbitrary time, as shown, the maximum in terms of amplitude occurs at 12.8 Hz approximate. This value was an average of the most common values between the entire data set. For the purposes of this paper, it represents one of the natural frequencies of the bridge oscillation.

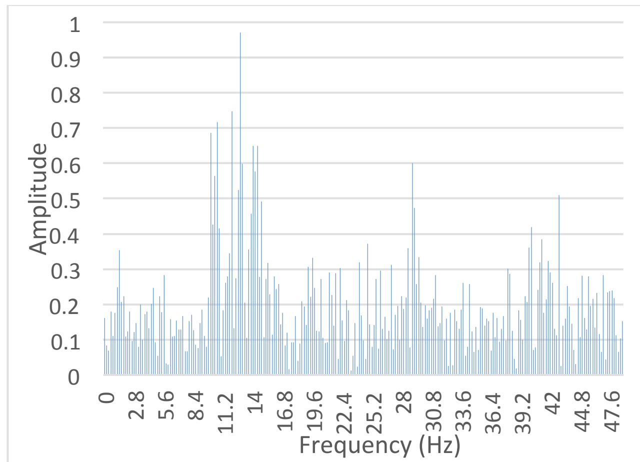

Figure 6. FFT data from the bridge vibration signal

The implemented artificial neural network, as described previously, was trained with 192 samples of frequencies, corresponding to 192000 data points from the vibration signal collected on the bridge. Since data comes from a real structure, faults could not be forced in the bridge, this means it was necessary to simulate the possible faults by assuming a decrease in the natural frequency as proposed by R. Salgado Estrada study. Doing so, the ANN could learn from real and simulated data the behavior of the fault and natural vibration frequencies. Expected output of the ANN was mapped in the -1 to 1 range, where -1 represents a fault and 1 represents a normal behavior.

As a classification method, the values of the actual output closer to -1 (the negative ones) were cataloged as a possible fault and the ones closer to 1 (or positive values) were catalog as normal behavior. Table 1 contains data information from ANN validation process. Random frequencies from the entire data set (in first column) were introduced in the network with its respective expected output.

The final output and the final fault prediction, based on the sign of the output, are also shown. For the case of these 34 random cases, the net responded correctly in 100% of the cases, determining in which cases there was a simulated fault induced to the system.

Table 1. Validation data for artificial neural network

Frequency (Hz)	Fault	Expected Output	Final Output	Fault Prediction
13,6	0	1	0,158297	0
14,0	0	1	0,193457	0
13,6	0	1	0,158297	0
14,2	0	1	0,210621	0
17,0	0	1	0,423674	0
9,0	1	-1	-0,332459	1
11,4	1	-1	-0,055948	1
13,6	0	1	0,158297	0
13,6	0	1	0,158297	0
29,0	0	1	0,938729	0
13,6	0	1	0,158297	0
32,8	0	1	1,024909	0
17,8	0	1	0,475995	0
13,6	0	1	0,158297	0
11,4	1	-1	-0,055948	1
11,6	1	-1	-0,03496	1
9,0	1	-1	-0,332459	1
11,8	1	-1	-0,014281	1
16,4	0	1	0,382073	0
13,6	0	1	0,158297	0
9,0	1	-1	-0,332459	1
13,6	0	1	0,158297	0
11,2	1	-1	-0,077247	1
21,2	0	1	0,663058	0
14,0	0	1	0,193457	0
20,0	0	1	0,603023	0
14,0	0	1	0,193457	0
2,0	1	-1	-1,379501	1
32,0	0	1	1,00856	0
32,0	0	1	1,00856	0
32,2	0	1	1,012728	0
27,6	0	1	0,900674	0
26,0	0	1	0,852009	0
1,8	1	-1	-1,413221	1

IV. Conclusions and future Work

One of the most relevant aspects of an embedded system is its portability. The results presented show the viability of an embedded online bridge fault detection system, using the Intel Galileo Board or some high performance embedded system. Also, Yocto Project for customized Linux distribution creation takes advantage of device's computational power by adding powerful tools and libraries, in the case of this experiment the FFTW, which could not be done in regular microcontrollers based systems.

Online FFT-ANN approach is a significant analysis method, because it allows the implementation of real-time fault detection system.

Through measurements of acceleration is possible to determine the natural vibration frequency of a structure and from changes shifts in this frequency it is feasible to determine whether the bridge suffered variations in stiffness due to damage on any structural element.

Fault prediction by the artificial neural network, was 100% accurate for the classification of induced faults and it demonstrates the fully functionality of the network to detect the performance changes in the bridge natural vibration frequencies.

In the future this approach will be implemented in other points of the Virilla's river bridge, and also will be scaled up to others bridges in Costa Rica.

V. References

[1] P. Alvarado, *Procesamiento Digital de Señales. Notas de Clase*. 2011.

[2] T. Zanussi y R. Purdie, "Yocto Project Board Support Package Developer's Guide". [Online]. Available: https://www.yoctoproject.org/docs/current/bsp-guide/bsp-guide.html#bsp. [Accessed: 08-apr-2014]

[3] F. J. Carrión Viramontes, A. Lozano Guzmán, M. Fabela Gallegos, D. Vázquez Vega, and J. A. Romero Navarrete, "Evaluación de Puentes Mediante el Análisis de Vibraciones", *Publicación Técnica*, n.º 132, 1999.

[4] R. Salgado Estrada, "Damage detection methods in bridges through vibration monitoring : evaluation and application", *Métodos de detecção de dano em pontes mediante a utilização de técnicas de monitorização dinâmica : avaliação e aplicação*, 2008.

[5] M. S. Molina Menache, R. Salgado Estrada, S. A. Zamora Castro and E. G. Lagunes Lagunes, "Detección de Daño e Puentes Mediante un Modelo Experimental", 2012.

[6] M. Frigo and S. G. Johnson, "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.

[7] D. Reby, S. Lek, I. Dimopoulos, J. Joachim, J. Lauga, and S. Aulagnier, "Artificial neural networks as a classification method in the behavioural sciences," *Behavioural Processes*, vol. 40, no. 1, pp. 35–43, Apr. 1997

[8] V. Reyes and M. Antonio, "Algoritmo backpropagation para redes neuronales : conceptos y aplicaciones," Instituto Politécnico Nacional. Centro de Investigación en Computación, Technical Report, 2007