Influence line extraction by deconvolution in the frequency domain

Gunnstein T. Frøseth*, Anders Rønnquist, Daniel Cantero, Ole Øiseth

Norwegian University of Science and Technology (NTNU). Department of Structural Engineering, Richard Birkelands vei 1A, 7491 Trondheim, Norway

Abstract

This paper proposes a new method for extracting static influence lines from measurements on bridges. The response of a structure is the convolution of the load and the influence line. Previous research has not embraced the fact that convolution is very efficiently handled in the frequency domain. The new method is based on the Fourier transform, which reduces the computational complexity of influence line extraction by several orders of magnitude compared to the conventional matrix method. The method can therefore be used to extract influence lines in near real time when implemented in low-powered devices, high-sensor-count systems, under high sampling rates and/or long signal sizes. It is shown that the inverse approach is ill-posed for certain vehicle configurations. A regularization technique for the ill-posed inverse problem is provided by a stabilizing filter. A numerical example is used to validate the regularization technique. The feasibility of the proposed method on real-world applications is demonstrated by a case study. The method is relevant to applications of and research on B-WIM algorithms, damage detection in structural health monitoring applications as well as model validation and model updating in the model-based evaluation of bridges.

Keywords: Influence lines, Bridge weigh-in-motion, Deconvolution, Traffic monitoring

1. Introduction

Infrastructure is aging worldwide following the rapid expansion of railways and highways at the beginning of the 20th century. Simultaneously, the demands on this infrastructure are increasing as a result of higher axle loads and

*Corresponding author

Email addresses: gunnstein.t.froseth@ntnu.no (Gunnstein T. Frøseth), anders.ronnquist@ntnu.no (Anders Rønnquist), daniel.cantero@ntnu.no (Daniel Cantero), ole.oiseth@ntnu.no (Ole Øiseth)

operational speeds. The combination of older infrastructure reaching and surpassing their design life and operational conditions that were not considered in the initial designs requires that infrastructure owners intensify maintenance and monitoring of their assets. To make the most of limited resources for maintenance, for the renewal and strengthening of infrastructure, it is becoming increasingly important for infrastructure owners to accurately estimate the remaining service life of each component in the network and to monitor critical components to ensure sufficient safety levels and uptime. To achieve this goal, it is necessary to obtain and analyze data from all aspects that affect the service lives of infrastructure components.

Historic and present traffic conditions are among the most important variables in the service life estimation of infrastructure objects, and more precise and accurate service life estimations are possible with improved knowledge of traffic conditions.

Bridge weigh-in-motion systems (B-WIM) utilize measurements on bridges to determine traffic data. The benefits of B-WIM algorithms compared to conventional weigh-in-motion stations are that the installation and service of the measurement station can be performed without disrupting traffic and that it is possible to utilize and combine data extraction intended for other purposes such as structural health monitoring (SHM) and service life estimation. The cost of installing and maintaining B-WIM systems is therefore often lower due to better accessibility and synergy effects with other projects when compared to other traffic monitoring systems.

Current commercial B-WIM algorithms are based on the seminal work by Moses [1]; see Lydon et al. [2] and Yu et al. [3]. These algorithms utilize the fact that the response of a structure to a moving load is proportional to the product of the influence line and the axle load. Moses [1] applies a theoretical influence line based on the static system of the bridge in his work. Theoretical influence lines seldom correspond well with actual or measured influence lines due to boundary and load distribution effects [4]. The accuracy of the B-WIM algorithm is determined by the accuracy of the estimated influence line of a structure. Extracting influence lines from measured responses is therefore a *key topic* in modern B-WIM algorithms [5].

Influence lines are also important in several other fields of research and application. Influence lines have been successfully used to update structural models to facilitate both the better prediction of responses from numerical models and the enhanced assessment of engineering structures [6, 7]. Furthermore, influence lines express structural condition and have been applied to identify and detect structural damage [8–11]. This paper is therefore relevant to researchers and practitioners working with model updating, SHM, damage detection, condition assessment and service life estimation of bridges.

Influence lines can be extracted from response measurements in several ways. McNulty and O'Brien [12] presents a manual step-by-step method of estimating the influence line; however, a common critique of this approach is that the quality depends on the ability of the operator and that it is a manual approach.

O'Brien et al. [13] describes an *inverse* approach to extracting influence lines. The method establishes a matrix from information about a calibration vehicle. The influence line is then determined by inverting the matrix. The solution time becomes substantial for responses obtained for long vehicles or bridge spans due to large matrix dimensions Quilligan [5]. Yamaguchi et al. [14] presents a similar inverse method based on the idea of representing the influence line with a polynomial.

Liljencrantz et al. [15] describes a *forward* approach for determining the influence line as an extension of the work by Quilligan [5]. The method relies on an initial guess of the influence line or a set of candidate influence lines as well as information about the calibration vehicle. The influence line is then found iteratively by minimizing the error between the predicted and measured responses of a train passage. Žnidarič et al. [16] briefly describes an algorithm that is implemented in a commercial B-WIM system. The influence line is modeled by a cubic spline together with constraints imposed from assumptions about the true influence line. The influence line is treated as an unknown and is obtained by minimizing the difference between the measured and predicted responses through a non-linear and non-gradient-based optimization algorithm. The issues with the forward approach are that the influence line may not converge to the correct influence line due to local minima in the objective function and invalid assumptions about the influence line and that the computational cost of extracting influence lines in this manner may be high due to low convergence rates.

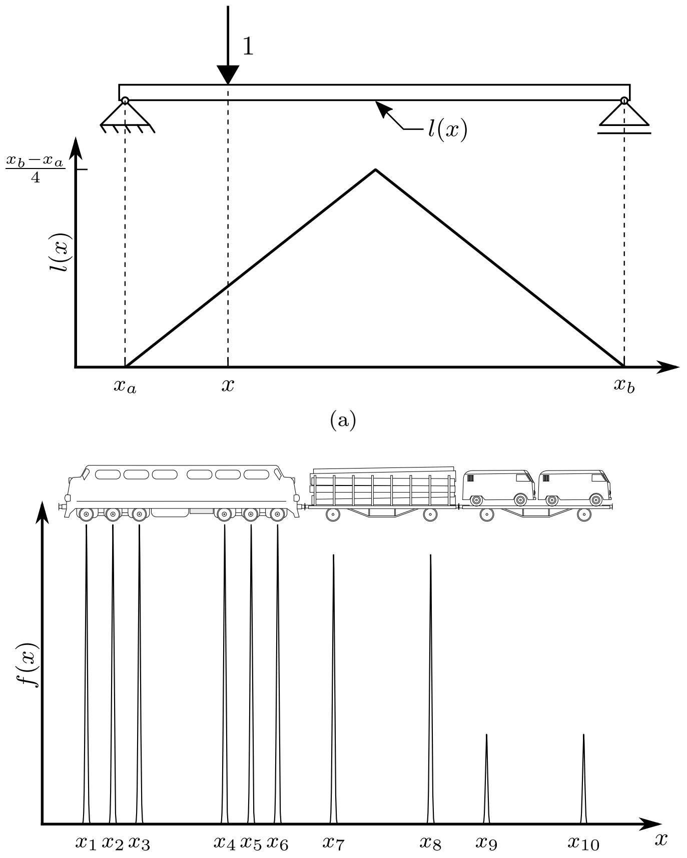
Furthermore, it is the authors' opinion that both the inverse and forward approaches are quite complex to implement in their present formulation.

This paper overcomes issues related to implementation complexity and computational cost through the realization that the response of the structure is the convolution of the influence line and the loading. From a computational perspective, convolution is very efficiently handled in the frequency domain because the convolution integral transforms into an elementwise multiplication operation. The realization complements current B-WIM theory and facilitates a simpler formulation for both axle load determination and forward and inverse influence line extraction algorithms.

The outline of this paper is as follows: The first part presents the connection between the convolution of the influence line and the load function to the established theory of B-WIM algorithms. The authors have not previously found such a derivation in the B-WIM literature and consider this an important contribution to the field. We then propose an alternative approach to extract static influence lines. The methodology is based on the Fourier transform and the highly efficient fast Fourier transform (FFT) algorithms. The inverse approach is ill-posed for certain types of load configurations, and regularization of the solution by a stabilizing filter is proposed to make the method applicable to the general case. An example is used to demonstrate the regularization technique. The second part discusses the computational advantages of the proposed method versus the conventional methods of extracting influence lines. The final part presents a case study with real field measurements, where we compare both the matrix-based approach and the FFT-based approach and demonstrate the feasibility of the method to real-world problems.

2. Theory

The influence line $l(x)$ is defined as the response of a structure at a certain measurable location to a unit load at location x. Fig. 1a provides an illustration of the influence line of the bending moment at midspan of a simply supported beam to a load moving along the beam-axis.



(a)

(b)

Figure 1: (a) illustrates the influence line for the moment at midspan of a simply supported beam. The coordinate x defines the position of the unit load. (b) shows an example of the load function 1, which represents a train with loads located at x_i, where i corresponds to the axle number of the train.

Assuming that the structure behaves linearly and the dynamic response is negligible compared to the static response, it is possible to utilize the static influence line to generate the response of a structure to an arbitrary load $f(x)$. The load function for a vehicle is presented in the equation below:

$$f(x) = \sum_{i=1}^{N_p} \delta(x - x_i) P_i \tag{1}$$

where N_p corresponds to the number of axles of the vehicle, P_i is the axle load, x_i is the location of an individual axle i, and $\delta(x - x_i)$ is the Dirac delta function; see Fig. 1b.

The response $z(x)$ of the structure at the location where $l(x)$ was obtained to a vehicle, represented by $f(x)$, is the convolution of the two, i.e.,

$$\begin{aligned} z(x) &= (l * f)(x) && \text{(2a)} \\ &= \int_{-\infty}^{\infty} l(s) f(x - s) \, \mathrm{d}s && \text{(2b)} \\ &= \int_{-\infty}^{\infty} l(x - s) f(s) \, \mathrm{d}s && \text{(2c)} \end{aligned}$$

where the commutative property of convolution $(l * f)(x) = (f * l)(x)$ has been used. The discrete representation of the convolution is, in matrix form, given by

$$\begin{aligned} \mathbf{z} &= \mathbf{Lf} && \text{(3a)} \\ &= \mathbf{Fl} && \text{(3b)} \end{aligned}$$

where $\mathbf{z} \in \mathbb{R}^{N_z}$ is the response vector, $\mathbf{l} \in \mathbb{R}^{N_l}$ is the influence line vector, and $\mathbf{f} \in \mathbb{R}^{N_f}$ is the load vector. Note that the dimension of the response vector is related to the dimension of the influence line and the load vector, i.e., $N_z = N_l + N_f - 1$. Furthermore, $\mathbf{L} \in \mathbb{R}^{N_z \times N_f}$ and $\mathbf{F} \in \mathbb{R}^{N_z \times N_l}$ are Toeplitz matrices converted from the zero-padded influence line vector and from the zero-padded load vector, respectively. Note that major numerical libraries, e.g., SciPy, MATLAB, Julia and Octave, have standard functionality to generate Toeplitz matrices from the base vectors, which further simplifies the implementation of B-WIM algorithms from the theory presented herein.

The load function in equation (1) can be represented by the axle load vector $\mathbf{p} = \begin{bmatrix} P_i \end{bmatrix} \in \mathbb{R}^{N_p}$ and the unit impulse matrix $\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \cdots & \mathbf{d}_{N_p} \end{bmatrix} \in \mathbb{R}^{N_f \times N_p}$, where $\mathbf{d}_i \in \mathbb{R}^{N_f}$ is the discrete representation of the Dirac delta function $\delta(x - x_i)$, i.e., the unit impulse vector. The load vector can then be written as

$$\mathbf{f} = \mathbf{Dp} \tag{4}$$

Introducing the alternative formulation of the load vector, equation (4), into the discrete convolution representation of the response, equation (3a), yields

$$\mathbf{z} = \mathbf{LDp} = \mathbf{I}_L \mathbf{p} \tag{5}$$

where $\mathbf{I_L} \in \mathbb{R}^{N_z \times N_p}$ is the influence line matrix first established in the classical work by Moses [1]. The axle loads can be determined by solving the system in equation (5). The problem may be ill-posed for certain types of vehicle configurations, and Tikhonov regularization has successfully been applied to alleviate this issue [17]. Equation (5) is used to generate responses in forward-based influence line and axle determination algorithms [15, 18].

Quilligan [5] and O'Brien et al. [13] start from equation (5) and derive the matrix $\mathbf{W} \in \mathbb{R}^{N_l \times N_l}$ and vector $\varepsilon \in \mathbb{R}^{N_l}$, which relates to the influence line in the following manner:

$$\mathbf{W}\mathbf{l} = \varepsilon \tag{6}$$

Equation (6) is the normal equation to (3b), i.e., $\mathbf{W} = \mathbf{F}^T\mathbf{F}$ and $\varepsilon = \mathbf{F}^T\mathbf{z}$, the first step in the normal equation method to solve least square problems. There are a wide range of algorithms that solve the least square problem in (3b) in addition to the normal equation method, e.g., QR factorization [19]. Regardless of the algorithm used to solve the overdetermined system of equations, the *matrix method* for influence line extraction is herein given by the least square solution of equation (3b):

$$\mathbf{l} = \mathbf{F}^\dagger\mathbf{z} \tag{7}$$

where $\mathbf{F}^\dagger$ denotes the inverse of the matrix $\mathbf{F}$ in the least square sense.

Yamaguchi et al. [14] presents an idea for influence line extraction by assuming that the influence line can be represented by a polynomial, i.e.,

$$l(x) = \sum_{j=0}^{N_q} q_j x^j = \mathbf{x}^T(x)\mathbf{q} \tag{8}$$

where N_q is the order of the selected polynomial, $\mathbf{x}(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^{N_q} \end{bmatrix}^T \in \mathbb{R}^{N_q}$ is the coordinate vector, and $\mathbf{q} \in \mathbb{R}^{N_q}$ is the coefficient vector. Discretizing the polynomial influence line yields

$$\mathbf{l} = \mathbf{X}\mathbf{q} \tag{9}$$

where $\mathbf{X} \in \mathbb{R}^{N_l \times N_q}$ is the Vandermonde matrix. Introducing equation (9) into equation (3b) yields

$$\mathbf{z} = \mathbf{F}\mathbf{X}\mathbf{q} \tag{10}$$

The influence line can then be obtained by solving (10) for the coefficient vector $\mathbf{q}$ and inserting the result back into equation (9). This method of extracting the influence line is herein named the *interpolation method.* Note that details regarding the implementation and performance of the interpolation method are scarce; no studies have been found that addresses selecting the polynomial order or optimal basis functions or that provide a comparison with the well-established matrix method. The interpolation method is therefore not regarded further in this paper.

This concludes the theoretical background for influence line extraction in the B-WIM literature.

2.1. Frequency-domain formulation

Previous works have not embraced the fact that convolution in the spatial domain is very efficiently handled in the frequency domain. The Fourier transform $\mathcal{F}\{\cdot\}$ and the inverse Fourier transform $\mathcal{F}^{-1}\{\cdot\}$ are defined by

$$G(\omega) = \mathcal{F}\{g(x)\} = \int_{-\infty}^{\infty} g(x) \exp\{-\mathrm{i}\omega x\} \,\mathrm{d}x \tag{11a}$$

$$g(x) = \mathcal{F}^{-1}\{G(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} G(\omega) \exp\{\mathrm{i}\omega x\} \,\mathrm{d}\omega \tag{11b}$$

where it is assumed that the Fourier transform of $g(x)$ exists, i.e., it is an integrable function in the spatial domain, and $G(\omega)$ denotes the frequency-domain representation of the same function. Note that $G(\omega)$ is a complex-valued function with a real part $G_R(\omega)$ and imaginary part $G_I(\omega)$, which can be written in polar form with amplitude $|G(\omega)|$ and phase $\phi_G(\omega)$, i.e.,

$$\begin{aligned} G(\omega) &= G_R(\omega) + \mathrm{i}G_I(\omega) \\ &= |G(\omega)| \exp\{\mathrm{i}\phi_G(\omega)\} \end{aligned}$$

The convolution theorem [20] states that if the Fourier transform of the functions $f(x)$ and $l(x)$ exists, then the product of their transforms $F(\omega) = \mathcal{F}\{f(x)\}$ and $L(\omega) = \mathcal{F}\{l(x)\}$ is the transform $Z(\omega) = \mathcal{F}\{z(x)\}$ of the convolution $z(x) = (l * f)(x)$. Taking the Fourier transform of equation (2) therefore yields

$$Z(\omega) = \mathcal{F}\{(l * f)(x)\} = L(\omega)F(\omega) \tag{13}$$

where the convolution integral in the spatial domain has turned into a multiplication in the frequency domain.

Deconvolution is performed by rearranging equation (13) and taking the inverse Fourier transform back into the spatial domain. For influence line extraction, the expression becomes

$$l(x) = \mathcal{F}^{-1}\left\{\frac{Z(\omega)}{F(\omega)}\right\} \tag{14}$$

2.1.1. Addressing noise and ill-posed deconvolution

Direct deconvolution in the frequency domain as presented in (14) is sensitive to noise in the response function. The measured signal $\hat{z}(x)$ is assumed to contain the true signal $z(x)$ and additive noise $n(x)$, i.e., the measured signal is given by

$$\hat{z}(x) = z(x) + n(x)$$

The estimated influence line can then be written as follows:

$$\hat{l}(x) = l(x) + \mathcal{F}^{-1}\left\{\frac{|N(\omega)|}{|F(\omega)|}\exp\left\{\mathrm{i}[\phi_N(\omega) - \phi_F(\omega)]\right\}\right\}$$

Clearly, the estimated signal will have a large error term at frequencies where the noise amplitude is large compared to the amplitude of the load signal, i.e., when the ratio $\frac{|N(\omega)|}{|F(\omega)|}$ is large.

The load function presented in equation (4) can be transformed into the frequency domain to obtain an analytical solution of the load $F(\omega)$; see equation (15).

$$F(\omega) = |F(\omega)|\exp\{\mathrm{i}\phi(\omega)\} \tag{15a}$$

$$\phi(\omega) = -\arctan\frac{\sum_{j=1}^{N_p} P_j \sin x_j\omega}{\sum_{j=1}^{N_p} P_j \cos x_j\omega} \tag{15b}$$

$$|F(\omega)| = \sqrt{\sum_{i=1}^{N_p}\sum_{j=1}^{N_p} P_i P_j \cos(x_i - x_j)\omega} \tag{15c}$$

Deconvolution in the frequency domain, equation (14), becomes ill-posed when the amplitude $|F(\omega)|$ of the frequency response function tends to zero. Considering specific examples, it can be shown that a vehicle with two axles has an amplitude spectrum bounded by

$$|\xi - 1| \leq \left|\frac{F(\omega)|}{P}\right| \leq 1 \tag{16}$$

where $\xi = \frac{P_1}{P_2}$. From this relation, it is clear that the amplitude spectrum becomes zero when the axle loads are identical, i.e., $\xi = 1$. This is obviously a severe limitation because a two-axle vehicle is the rule rather than the exception in the case of highway traffic. Two-axle vehicles are also present in railway networks as well. There are other special circumstances that will render the solution unstable. For instance, a train composed of an even number of axles whereby the axle distances are multiples of each other and each axle is identically loaded is also unstable. It is important to note that the circumstances that make the solution unstable are quite particular and even unusual in practice, especially for trains with multiple axles. The direct approach will work well for the majority of practical configurations, as will be demonstrated in the case study below. Nevertheless, to make the proposed method applicable to the general case, an approach to regularize the solution is given below.

The idea is to create a stabilizing filter $A(\omega)$ that attenuates amplified noise in $L(\omega)$. Tikhonov and Arsenin [21] presents the following general form of the stabilizing filter:

$$A(\omega) = \frac{1}{1 + \alpha\frac{M(\omega)}{|F(\omega)|^2}} \tag{17}$$

where $\alpha \geq 0$ is the regularization parameter and $M(\omega)$ is the regularization function. Fig. 2 shows the frequency response of the general regularizing filter as a function of the regularization coefficients.

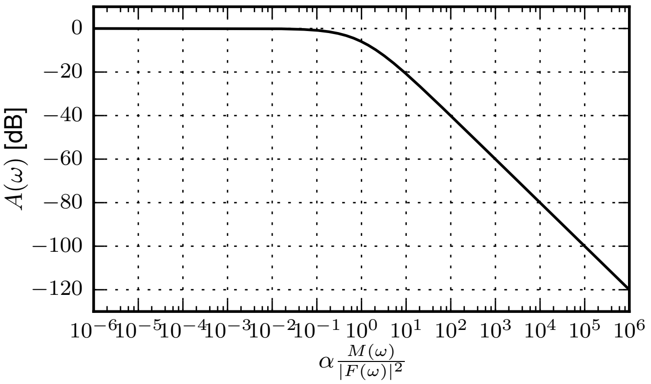


Figure 2: The frequency response of the general form of the stabilizing filter.

The passband (3dB or half power) is limited to frequencies where $\frac{|F(\omega)|^2}{M(\omega)} \geq \alpha$. From this relation, it is also obvious that the regularization parameter must be set at a proper level to ensure that information about the true signal does not become filtered out by the stabilizing filter. We return to the issue of selecting the appropriate regularizing parameter throughout this section.

The regularizing function can be selected based on the available prior information. If both the spectral density of the noise $|N(\omega)|^2$ and the spectral density of the influence line $|L(\omega)|^2$ are known, the Wiener filter can be used. This minimizes the mean square error between the true and extracted influence lines [22] and therefore represents the optimal filter for deconvolution. The Wiener filter uses the regularization function $M(\omega) = \frac{|N(\omega)|^2}{|L(\omega)|^2}$, which leads to the signal-to-noise ratio $\mathrm{SNR}(\omega) = \frac{|Z(\omega)|^2}{|N(\omega)|^2} = \frac{|L(\omega)|^2|F(\omega)|^2}{|N(\omega)|^2}$ being directly related to the definition of the passband, i.e., the passband is determined by

$$\mathrm{SNR}(\omega) \geq \alpha \tag{18}$$

In the case of the Wiener filter, the regularization parameter can be set to $\alpha = 1$ directly, which means that signal components where noise is contributing more than the true signal will be filtered out.

Unfortunately, the spectral density of the influence line may be unknown due to a lack of previous measurement and, in the case of SHM, where the influence line has changed due to structural damage, and the bias introduced by any assumption about the influence line might be unwanted. Similarly, the noise spectral density may be challenging to determine due to transient noise effects induced by electrical vehicles or excessive structural vibrations. We briefly mention that, if concrete information on either the influence line or the noise is available, such information can be included in the stabilizing filter to achieve enhanced performance.

If no *a priori* information on the influence line or the noise is available, $M(\omega) = 1$ is a simple and practical choice that works well in stabilizing the solution. The filter response and passband are only dependent on the power in the load function, i.e., the passband is defined by

$$|F(\omega)|^2 \geq \alpha \tag{19}$$

which means that all information for which the spectral density of the load is below the regularization parameter will be filtered out. This is unbiased in the sense that no assumptions are made on either the influence line or the noise. $M(\omega) = 1$ is therefore recommended in cases where changes to the influence line or noise are expected due to structural damage or transient noise effects induced by passing vehicles. It is infeasible to give a general recommendation for the regularization parameter in this case. Other diagnostics must be used to select the regularization parameter, as will be discussed below.

There are several methods to determine the optimal regularization parameter α in cases where it cannot be determined by prior knowledge [23], e.g., by minimizing the error between the measured and predicted response history [24] and estimating the signal-to-noise-ratio regularization coefficients [25] from a series of measurements. The L-curve is a simple and intuitive tool that can also be used to select the optimal regularization parameter for the present regularization problem, as we will demonstrate in a numerical example below.

The L-curve is used to determine the optimal regularization parameter by balancing the quality of fit of the model to the data and the amount of damping provided by the regularization term [26]. The fit of the model is quantified by the *residual norm* $\|\mathcal{F}^{-1}\{F(\omega)L_\alpha(\omega)\} - \hat{z}(x)\|_2$, and the amount of damping by the regularization term is quantified by the *solution norm* $\|l_\alpha(x)\|_2$, where $L_\alpha(\omega)$ and $l_\alpha(x)$ denote the regularized extracted influence line. The optimal balance between the two quantities is located in the characteristic corner produced in a log-log graph $(\|\mathcal{F}^{-1}\{F(\omega)L_\alpha(\omega)\} - \hat{z}(x)\|_2,\ \|l_\alpha(x)\|_2)$ for selected values of the regularization parameter α. The p-norm is for a general vector $\mathbf{v} \in \mathbb{R}^{N_v}$ defined by equation (20)

$$\|\mathbf{v}\|_p = \left(\sum_{j=1}^{N_v} |v_j|^p\right)^{1/p} \tag{20}$$

2.2. Fourier-domain influence line extraction method

The final form of the proposed Fourier-domain method (FD method) for influence line extraction is obtained by applying the regularizing filter in equation (17) to the equation for direct deconvolution, equation (14):

$$l_\alpha(x) = \mathcal{F}^{-1}\left\{\frac{Z(\omega)F^*(\omega)}{|F(\omega)|^2 + \alpha M(\omega)}\right\} \tag{21}$$

where $F^*(\omega)$ is the complex conjugate of $F(\omega)$.

It should be made clear that an FFT algorithm, e.g., [27], should be used to obtain the forward and inverse Fourier transforms. There are two important details to consider when addressing FFT algorithms.

1. For strictly real input, the Fourier transform results in a signal for which the components around $\omega = 0$ are complex conjugates of each other, i.e., $Z(\omega_i) = Z^*(\omega_{-i})$, which means that half the signal does not contain any additional information. This can be exploited to further reduce hardware requirements and increase computational efficiency [28]. FFT algorithms exploiting this property are called *real* FFT algorithms and should be used for increased computational speed in calculating the Fourier transform for real signals such as the response, influence line and load function used throughout this paper.
2. The length of the input vector $\mathbf{g}$ to a specific FFT algorithm is an important aspect in achieving optimum computational efficiency [28]. Zero padding should be utilized to obtain the optimal vector lengths and maximize the computational efficiency of the FD method.

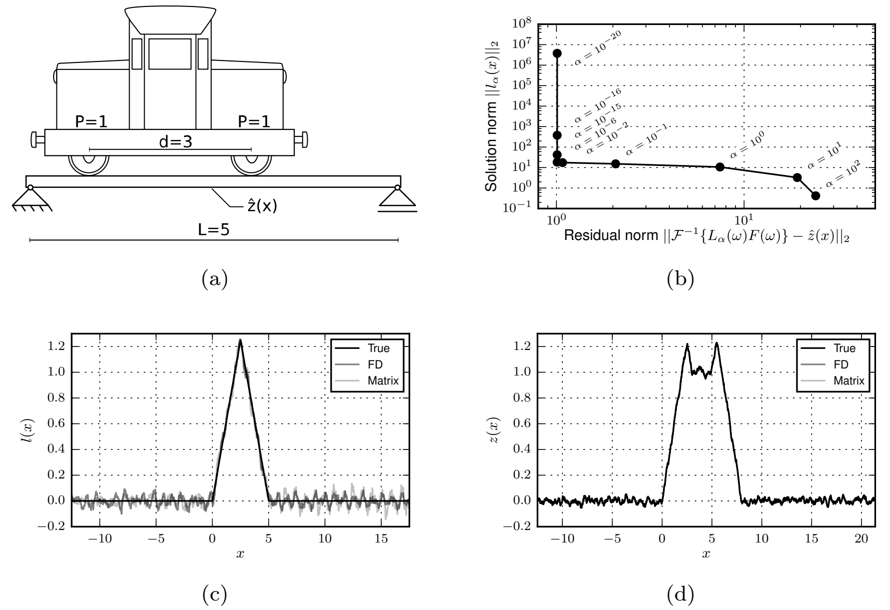
The analytical expression for the frequency-domain load function, provided in equation (15), can be used in the final implementation. It may, however, for a vehicle with a large number of axles be slower than establishing the load vector $\mathbf{f}$ in the spatial domain and converting it to the frequency domain via an FFT algorithm because the analytical implementation involves the outer product of the frequency and the load position vector and subsequently matrix multiplication between the resulting matrix and the load magnitude vector.

2.3. Example: Two-axle vehicle passing a simply supported bridge

The purpose of this example is to demonstrate that the proposed unbiased stabilizing filter $M(\omega) = 1$ can be applied to regularize the solution with the FD method and to show that the well-known L-curve methodology can be utilized to find the correct regularization parameter α.

Fig. 3a gives the example problem. The bending moment at midspan of a simply supported beam with $L = 5$ is the measured quantity. The response $z(x)$ is generated at a spatial sampling rate of $f_x = 100$ sampling points per unit length, and Gaussian white noise $n(x)$ with a signal-to-noise ratio $\left(\text{SNR} = \left(\frac{\|z(x)\|_2}{\|n(x)\|_2}\right)^2\right)$ equal to 20 is added to the response signal. The simply supported beam is loaded by a two-axle vehicle with identical axle loads, $P_1 = P_2 = 1$, with an axle distance of 3. The two-axle vehicle with identical axle load renders the direct solution unstable, as shown by equation (16).

The L-curve for the present example is found in Fig. 3b, which confirms that the solution is unstable for $\alpha = 0$. Furthermore, it is clear that the characteristic corner indicates that the optimal regularization parameter is $\alpha \approx 10^{-6}$. The figure also shows that the solution is not sensitive to the regularization parameter; $\alpha = 10^{-15}$ to 10^{-2} yields similar results to the optimal value.



(a)

(b)

(c)

(d)

Figure 3: Influence line extraction from the response of a two-axle vehicle with identical axle loads passing a simply supported bridge with regularization function $M(\omega) = 1$. The characteristic L-curve for the problem is depicted in (b). The extracted influence lines in (c) and the responses in (d) are filtered with a moving average filter order of 25 to remove random noise. The FD solution is regularized with $\alpha = 10^{-6}$.

Fig. 3c shows that, for the extracted influence line by the proposed FD method with $\alpha = 10^{-6}$, the results compare well with both the true underlying influence line and the solution provided by the matrix method. Note that the difference between the extracted influence lines and the true influence line before and after the simply supported beam in Fig. 3c is due to the added noise and should not be mistaken for free vibration. Fig. 3d confirms that the influence lines obtained with the proposed regularized FD method and the matrix method correspond well. The difference in the influence lines extracted by the two methods is, in practice, negligible. The unstable solution obtained by the direct method has effectively been regularized by the stabilizing filter.

3. Computational complexity of influence line extraction

This section discusses the computational advantage of using the FD method in comparison to the conventional matrix method. In this discussion, $O(\cdot)$ ("Big O") denotes the number of operations necessary to implement an algorithm in the asymptotic sense. This helps assess the computational complexity of an algorithm as the dimensions of the vectors and matrices associated with the problem become large; see Golub and Van Loan [19].

The discussion assumes two extremes in regard to the implementation of the algorithms: 1) implementation does not consider the underlying structure of the matrices involved in influence line extraction, and 2) the sparsity, type and structure of the matrices and vectors are considered in the implementation of each method.

3.1. Structure of matrices and available algorithms for implementation

In regard to the type, sparsity and structure of the matrices and vectors associated with the underlying theory of methods, note the following:

- The load vector $\mathbf{f} \in \mathbb{R}^{N_f}$ is a sparse vector with N_p non-zero elements corresponding to the number of axles in the vehicle. $N_p << N_f$ for all practical cases.
- $\mathbf{F} \in \mathbb{R}^{N_z \times N_l}$ is a sparse Toeplitz matrix, each column $\mathbf{f_i} \in \mathbb{R}^{N_z}$ of which consists of the vector $\mathbf{f}$ with zero padding. Each column therefore consist of N_p non-zero elements.

Furthermore, the discussion relies on implementation with well-established algorithms available in typical numerical libraries. Specifically, the following algorithms are assumed to be available for implementation:

- A forward and inverse FFT algorithm, which are $O(N_z \log_2 N_z)$ [27, 28].
- A least square solver for general-type matrices, i.e., methods that ignore the structure of the systems. These are invariably $O(N_m N_n^2)$ algorithms, where $N_m > N_n$ [19].
- A sparse matrix-vector multiplication algorithm, e.g., compressed sparse row (CSR)[29]. The matrix-vector multiplication $\mathbf{F}^T\mathbf{f_i}$ is then $O(N_p N_l) \approx O(N_l)$. It follows that the sparse matrix-matrix multiplication $\mathbf{F}^T\mathbf{F}$ is $O(N_l^2)$.
- The Levinson-Durbin algorithm [19] for solving Toeplitz matrices. Solving the system with the normal matrix $\mathbf{F}^T\mathbf{F}$ then has complexity $O(N_l^2)$.

Note that previous work has not presented similar analysis of matrix structures in B-WIM applications or in the literature, and a typical implementation should be regarded as involving the general least square solver presented above.

3.2. Analysis of algorithms

3.2.1. Frequency-domain method

The proposed FD method relies on an FFT algorithm for computing the Fourier transforms of the load and response vector as well as the inverse FFT to obtain the influence line. The FD method therefore has complexity $O(N_z \log_2 N_z)$.

3.2.2. Matrix method

The matrix method can be solved by general least square solvers in $O(N_z N_l^2)$ by solving equation (3b). However, the matrix $\mathbf{F}^T\mathbf{F}$ can be established in $O(N_l^2)$ by a sparse matrix-vector algorithm, and the resulting system can then be solved by the Levinson-Durbin algorithm [19] in $O(N_l^2)$.

The matrix method therefore has computational complexity $O(N_z N_l^2)$ in its simplest and slowest form and $O(N_l^2)$ in the fastest implementation.

The computational complexity of the two methods is summarized in table 1.

Table 1: Comparison of the computational complexity of the inverse methods available for influence line extraction.

Method	Complexity
FD	$O(N_z \log_2 N_z)$
Matrix	$O(N_l^2)$ – $O(N_z N_l^2)$

3.3. Numerical experiment: Complexity ratio in practice

The actual time it takes for a computer to solve the equations depends on the hardware, implementation of each algorithm, programming language and operating system. This means that the same algorithm will have different execution times between two computers. Regardless, a numerical experiment was performed to augment the discussion on computational complexity and validate the theoretical analysis of the algorithms in the previous section.

The simply supported beam in the previous example was utilized as a test case. A vehicle with identical axle loads and four axles spaced at $x = \{0, 3, 16, 30\}$ was used to make the length of the influence line and load vectors equal, i.e., $N_l = N_f$. The size of the vectors was varied by altering the spatial resolution f_x.

The analysis was performed on a laptop running a 64-bit Linux operating system with a 2.8 GHz quad-core processor and 16 GB of RAM. All algorithms were implemented in 64-bit Python version 2.7.12 using the functionality of the scientific library SciPy [30] v0.17.0. Note that SciPy is a frontend to the BLAS and LAPACK libraries [31], which are the basis for several free and commercial scientific computing languages. Each algorithm was executed 25 times to obtain the average execution time.

The results from the numerical experiment are presented in Fig. 4.

3.4. Dimensions of the influence line extraction problem

To estimate the signal size in BWIM applications, it is first necessary to discuss the desired resolution in the spatial domain.

High resolution is especially important in the inverse problem due to numerical stability and discretization errors: the higher the resolution, the more accurate the axles can be positioned and axle loads determined. The most

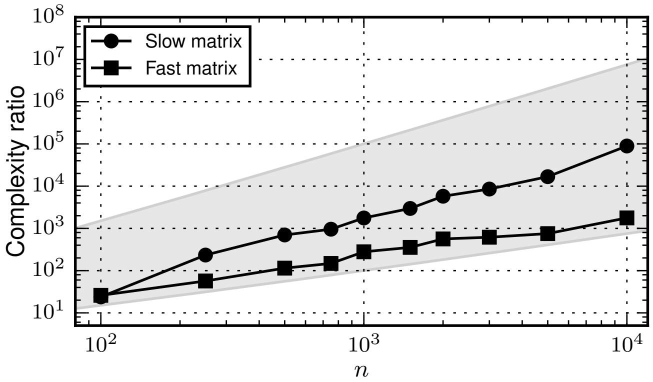


Figure 4: The gray area indicates the range of the ratio between the asymptotic computational complexity of the conventional methods for influence line extraction and that of the FD method. The slow and fast matrix lines indicate the ratio of the execution time between the respective matrix implementations and the FD method obtained in the numerical experiment. Note that $n = N_l = N_f \approx {}^{N_z}/_2$

.

closely spaced axles are located 1 m to 2 m apart, depending on whether the B-WIM system is installed on a railway or highway bridge; a meter resolution is therefore too coarse. Methods for identifying axles and calculating axle distances are on the order of 10cm [32]. At vehicle speeds of $v = 90\,\text{km/h}$, this corresponds to a temporal sampling frequency of 250 Hz, which is at the lower end for current B-WIM systems [33].

Considering practical B-WIM applications, Zhao et al. [34] uses sampling rates of 512 Hz, O'Brien et al. [13] uses 512 Hz and 1024 Hz, and Marques et al. [18] presents sampling rates of 1000 Hz. Note that the sampling rate is not just determined by influence line extraction on systems that serve other purposes as well.

Based on the above discussion, a spatial sampling rate of $f_x = 10/\text{m}$ is assumed, which should be considered conservative compared to what current B-WIM systems use.

The length of the path from which the influence line is obtained, i.e., the length of the bridge, is typically from 5 m to 100 m for railways, whereas it is from 5 m to 1000 m for highway applications. Conversely, the length of the loading is from 10 m to 1000 m for trains and from 2 m to 20 m for cars and trucks. Under the assumed spatial resolution, this means that, for railway applications, N_l ranges between 10^1 to 10^3, and N_f ranges between 10^2 to 10^4. For highway applications, the vector sizes are on the same order but opposite for the respective vector sizes.

3.5. Comparison of methods

Fig. 4 depicts the *complexity ratio*, i.e., the ratio between the approximate number of operations of the conventional methods and that of the FD method.

The gray area indicates the range in which the theoretical complexity ratio between the matrix method and the FD method can be expected to lie for different hardware and programming languages. The complexity ratio for the matrix implementation and that for the FD method obtained from the numerical experiment are shown as marked lines. The growth rate for both the fastest and slowest implementation corresponds well to the theoretical bounds. The lower bound for the theoretical complexity is in good agreement with the fastest implementation of the matrix method, whereas the upper bound for the theoretical complexity is too conservative in comparison to the numerical experiment. The theoretical computational complexity does not reveal constant and lower order computational costs associated with each algorithm, and it is to be expected that the theoretical complexity will deviate from the numerical experiment. Further discussion therefore focuses on the results from the numerical experiment, as these are expected to represent reality for most implementations in the presented range of signal lengths.

With signal lengths of $n = 1000$, the FD method is approximately 1000-times faster than the general implementation of the matrix method and 100-times more efficient than the fastest matrix implementation. The execution time on the computer system presented in section 3.3 is approximately 1 second for the slowest matrix implementation. This may not be a significant difference on either current laptop and desktop computers or low-count sensor setups, but it may be important on embedded or low-power systems installed off the power grid. The difference may also be important on larger sensor networks, where influence line extraction is conducted on a range of concurrent signals. Another case would be SHM, where prompt information about the state of the structure is critical, often on a computer system working on other tasks in parallel.

The difference between the proposed FD method and the matrix method becomes more significant at longer signal lengths. It can be argued that the slowest implementation becomes impractical for $n > 1000$ on modern personal computers, as the time ratio rapidly reaches and exceeds 10000 in comparison to the FD method. In regard to the fastest implementation of the matrix method, the FD method is in general more efficient. For modern computers and low-count sensor systems, the difference only becomes significant at the higher end of the presented range for signal lengths.

In any case, the proposed FD method is in general at least one order of magnitude more efficient than the matrix method and should be considered when faced with issues related to computational load.

4. Case study – Hell railway bridge

4.1. Description of the Hell railway bridge, field measurements and preprocessing of data

The Hell railway bridge (Fig. 5) crosses the river Stjørdalselva and lies on Nordlandsbanen 29 km north of Trondheim, Norway. Construction of the bridge, which is an open-deck Parker pony truss bridge with 5 identical spans, each 35 m in length, was completed in 1902.

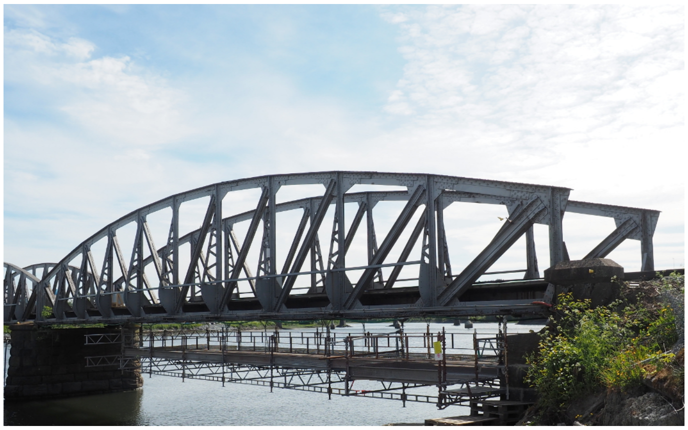


Figure 5: Hell railway bridge.

Strain data from the lower flange of the quarter span of the first stringer onto the bridge and the lower flange in the midspan of the fourth crossgirder were utilized to demonstrate the feasibility of the proposed method; see Fig. 6.

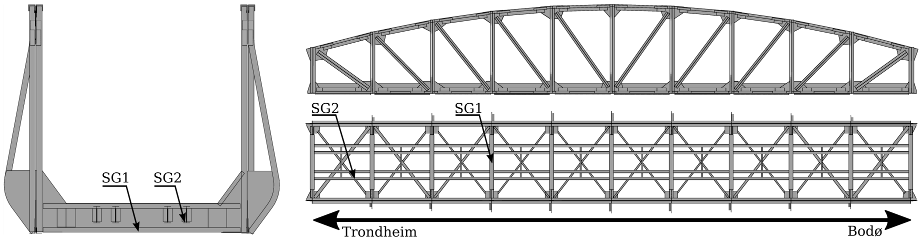


Figure 6: Placement of straingages on crossgirder (SG1) and stringer (SG2).

The stringers are of rolled-type INP36 with nominal length 3.5 m, and the crossgirder is a built up riveted I-section with nominal length 4.5 m. The strain measurements were taken with 120 Ω quarter bridge strain gages together with a data acquisition unit from National Instruments. The temporal sampling rate was $f_s = 794\,\mathrm{Hz}$.

A train with 8 axles consisting of a track vehicle, one empty flatbed trailer and one loaded ballast hopper was used to calibrate the measurement system and provide data for the influence line extraction. The train had known axle loads in the range 3 t to 19 t and axle spacing between 1.8 m and 6.7 m. The train passed the bridge 8 times at speeds between 10 km/h to 40 km/h, i.e., between crawl speed and the maximum allowed at the site. An estimate of the actual speed at the site was obtained by the phase correlation method described in Liljencrantz et al. [15].

Before analyzing the data further, each signal was resampled to a spatial sampling frequency of $f_x = 10/\mathrm{m}$ after applying an 8th-order Bessel anti-aliasing

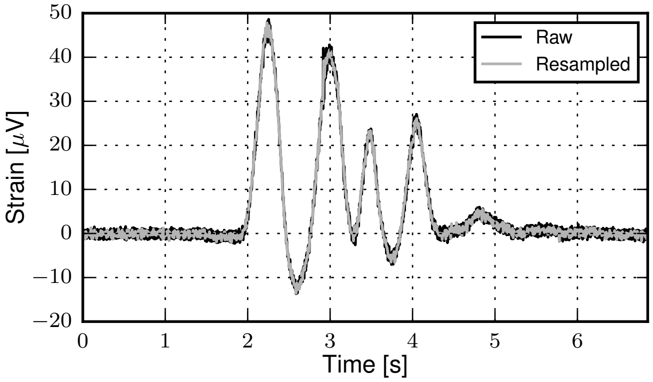


Figure 7: Raw and resampled strain obtained from measurements on the stringer (SG2) during passage of the calibration train towards Bodø at a speed of 21.6 km/h.

filter, see Fig. 7 for an example of a raw and resampled signal.

4.2. Influence line extraction by matrix method and deconvolution

This section compares influence lines extracted by both the matrix and FD methods to evaluate the performance of the proposed method applied to real data. Influence line extraction with the matrix method is performed by solving equation (7) with a general least square solver. The FD method is described in section 2.2. No regularization for either method was used in extracting the influence line.

The raw influence lines were aligned with the maximum value at the location of the sensor in the length direction of the bridge. A 3rd-order spline was then used to collocate points along the two influence lines. The aligned influence lines contained the signal 3 m before and after the bridge. A mean influence line was produced for each of the two methods for comparison. The final step before comparing the influence lines from either method was to filter random noise by applying a 5th-order moving average filter.

The infinite norm $\|\cdot\|_\infty$ and the 2nd norm $\|\cdot\|_2$ were calculated on the residual vector $\hat{\mathbf{l}}_{\mathrm{MAT}} - \hat{\mathbf{l}}_{\mathrm{FD}}$ to quantitatively compare the influence lines extracted by the two methods.

The results from the 8 passages and the mean influence line are presented in table 2.

The infinite norm $\|\cdot\|_\infty$ expresses the maximum point difference between the influence lines, and the 2nd norm $\|\cdot\|_2$ expresses the difference in an overall sense between the influence lines extracted by the two methods. Table 2 shows that the largest infinite norm and 2nd norm are found for the influence line extracted from passages 3 and 2 for the data obtained from the crossgirder and the stringer, respectively. The influence lines corresponding to the largest deviation in regard to the infinite and 2nd norms are also presented in Figs. 8a-8d to provide a qualitative comparison of the two methods. The maximum

Table 2: Comparison of the influence lines extracted by the matrix method and by the proposed FD method. The speed is given in units of km/h, and the units of the norms are µV/t. Negative speed is defined as in the direction toward Trondheim. Bold characters indicate the worst comparison between the influence lines for each of the norms.

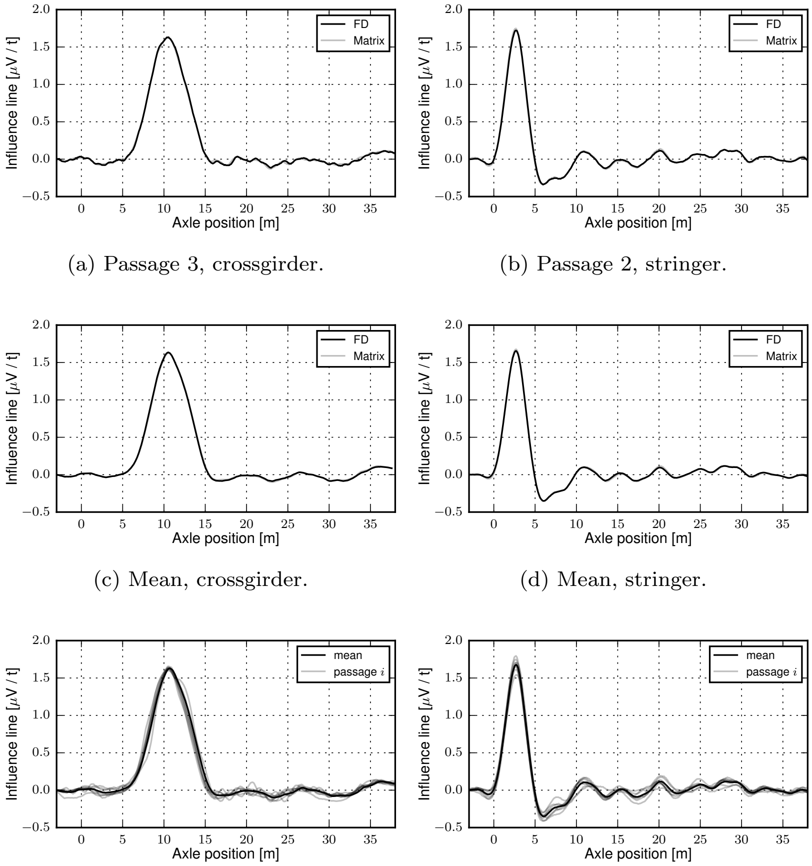
		Crossgirder		Stringer	
Passage	Speed	$\lVert\cdot\rVert_\infty$	$\lVert\cdot\rVert_2$	$\lVert\cdot\rVert_\infty$	$\lVert\cdot\rVert_2$
1	10.7	0.03	0.25	0.05	0.28
2	21.6	0.03	0.22	**0.05**	**0.30**
3	32.2	**0.04**	**0.26**	0.03	0.22
4	41.8	0.03	0.26	0.03	0.21
5	-10.7	0.02	0.16	0.03	0.20
6	-20.0	0.03	0.21	0.03	0.20
7	-30.7	0.03	0.21	0.03	0.21
8	-40.7	0.03	0.25	0.03	0.19
Mean	–	0.03	0.19	0.03	0.20

point difference between the influence line extracted by the two methods is 0.04 µV/t for the crossgirder and 0.05 µV/t for the stringer. This corresponds to approximately 3 % of the maximum amplitude (≈ 1.5 µV/t) of the influence line.

Figs. 8e and 8f show the variation in the influence lines obtained for different passages by the matrix method. The *smallest* maximum point difference, i.e., the best comparison, between the influence line and the mean influence line is 0.04 µV/t for the crossgirder and 0.07 µV/t for the stringer. It is clear by inspection of Fig. 8 that the difference between the influence lines extracted for the same passage by the two methods is less than the variation to be expected between two passages extracted by the matrix method alone.

5. Conclusion

This paper presented a new method for extracting static influence lines from bridge structures subject to traversing loads. The method is based on the realization that the response of a structure is the convolution of the influence line and the load function. The method utilizes the Fourier transform and, more specifically, the FFT algorithm to achieve a very efficient method for extracting influence lines. The proposed method is ill-posed for certain types of vehicle configurations, and a regularization technique based on a stabilizing filter was presented. Two different stabilizing filters were suggested based on the a priori information available on the influence line and noise spectrum. A numerical example was used to demonstrate that the regularization technique stabilizes the solution and shows how the regularization parameter can be chosen based on the L-curve.



(a) Passage 3, crossgirder.

(b) Passage 2, stringer.

(c) Mean, crossgirder.

(d) Mean, stringer.

(e) Matrix influence lines, crossgirder.

(f) Matrix influence lines, stringer.

Figure 8: (a) and (b) show the worst influence lines according to the 2nd and infinite norms, (c) and (d) show the mean influence lines extracted by the FD and matrix methods, and (e) and (f) show the mean influence line (solid black) together with the individual influence lines obtained by the matrix method.

Analysis of algorithms and a numerical experiment showed that the FD method is faster than the well-established matrix method for influence line extraction. The practical difference between the two methods is not significant on modern personal computer systems for small signal sizes $n < 10^3$. Proper implementation of the matrix method becomes important to avoid significantly longer run times than the proposed methodology at signal sizes of $n = 10^3$. At even larger signal sizes, proper implementation of the matrix method is necessary to obtain results in a reasonable computing time. For multiple-sensor setups, low-power systems and limited-resource systems, the computational advantage of the proposed methodology is considered to be relevant at all signal sizes.

A case study on the Hell railway bridge was presented to demonstrate the feasibility of the FD method. Influence lines were extracted from strain measurements on one stringer and crossgirder. The results obtained by the FD method were virtually identical to those obtained by the well-established matrix method.

Acknowledgments

This research is funded by the Norwegian national rail administration (Jernbaneverket). The authors acknowledge the funding and all assistance provided during installation of the measurement system on the Hell railway bridge as well as provisioning of the train for the calibration of the measurement system.

References

[1] Moses F. Weigh-in-motion system using instrumented bridges. J Transp Eng 1979;105(3):233–49.

[2] Lydon M, Taylor SE, Robinson D, Mufti A, O'Brien EJ. Recent developments in bridge weigh in motion (B-WIM). J Civ Struct Health Monit 2016;6(1):69–81.

[3] Yu Y, Cai CS, Deng L. State-of-the-art review on bridge weigh-in-motion technology. Adv Struct Eng 2016;19(9):1514–30.

[4] Grundy P. Fatigue life of Australian railroad bridges. In: IABSE Symposium (Washington DC): Maintenance, Repair and Rehabilitation of Bridges. 1982, p. 77–82.

[5] Quilligan MJ. Bridge weigh in motion. Licenciate; Royal Institute of Technology (KTH); Stockholm, Sweden; 2003.

[6] Strauss A, Wendner R, Frangopol D, Bergmeister K. Influence line- model correction approach for the assessment of engineering structures using novel monitoring techniques. Smart Struct Syst 2012;9(1):1–20.

[7] Xiao X, Xu YL, Zhu Q. Multiscale Modeling and Model Updating of a Cable-Stayed Bridge . II : Model Updating Using Modal Frequencies and Influence Lines. Journal of Bridge Engineering 2014;20(10):1–12.

[8] Zaurin R, Catbas FN. Integration of computer imaging and sensor data for structural health monitoring of bridges. Smart Mater Struct 2010;19:1–15.

[9] Cavadas F, Smith IFC, Figueiras J. Damage Detection using data-driven methods applied to moving-load responses. Mech Syst Signal Process 2013;39:409–25.

[10] Chen ZW, Zhu S, Xu YL, Li Q, Cai QL. Damage detection in long suspension bridges using stress influence lines. J Bridge Eng 2014;20(3):1–11.

[11] Cantero D, Karoumi R, González A. The virtual axle concept for detection of localised damage using bridge weigh-in-motion data. Eng Struct 2015;89:26–36.

[12] McNulty P, O'Brien EJ. Testing of bridge weigh-in-motion system in a sub-Arctic climate. J Test Eval 2003;31(6):1–10.

[13] O'Brien EJ, Quilligan MJ, Karoumi R. Calculating an influence line from direct measurements. Proc Inst Civ Eng Bridge Eng 2006;159(BE1):31–4.

[14] Yamaguchi E, Kawamura S, Matuso K, Matsuki Y, Naito Y. Bridge-weigh-in-motion by two-span continuous bridge with skew and heavy-truck flow in Fukuoka area, Japan. Adv Struct Eng 2009;12(1):115–25.

[15] Liljencrantz A, Karoumi R, Olofsson P. Implementing bridge weigh-in-motion for railway traffic. Comput Struct 2007;85(1-2):80–8.

[16] Žnidarič A, Kalin J, Kreslin M, Favai P, Kolakowski P. Railway bridge weigh-in-motion system. Transp Res Procedia 2016;14:4010–9.

[17] O'Brien EJ, Rowley CW, González A, Green MF. A regularised solution to the bridge weigh-in-motion equations. Int J Heavy Veh Syst 2009;16(3):310–27.

[18] Marques F, Moutinho C, Hu WH, Cunha Á, Caetano E. Weigh-in-motion implementation in an old metallic railway bridge. Eng Struct 2016;123:15–29.

[19] Golub GH, Van Loan CF. Matrix computations. 4th ed. Baltimore, MD: The Johns Hopkins University Press; 2013.

[20] Kreysig E. Advanced engineering mathematics. 8th ed. Singapore: John Wiley & Sons; 1999.

[21] Tikhonov AN, Arsenin VY. Solutions of ill-posed problems. New York, NY: Hallsted Press; 1977.

[22] Pratt WK. Processing digital image processing; vol. 5. 3rd ed. New York, NY: John Wiley & Sons; 2001.

[23] Hansen PC. Discrete inverse problems: Insight and algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2010.

[24] Hunt BR. The application of constrained least squares estimation to image restoration by digital computer. IEEE Trans Comput 1973;100(9):805–12.

[25] Zhou Z, Gao F, Zhao H, Zhang L. Application of Fourier-wavelet regularized deconvolution for improving image quality of free space propagation x-ray phase contrast imaging. Phys Med Biol 2012;57(22):7459–79.

[26] Hansen PC. The L-curve and its use in the numerical treatment of inverse problems. In: Johnston P, editor. Advances in computational bioengineering; vol. 3. Southhampton, UK: WIT Press; 2001, p. 119–42.

[27] Cooley JW, Tukey JW. An algorithm for the machine calculation of complex fourier series. Math Comput 1965;19:297–301.

[28] Smith SW. The scientist and engineer's guide to digital signal processing. San Diego, CA: California Technical Publishing; 1997.

[29] Saad Y. Iterative methods for sparse linear systems. 2nd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM); 2003.

[30] Jones E, Oliphant T, Peterson P. SciPy: Open source scientific tools for Python. `http://www.scipy.org`; 2016. [accessed 20.12.16].

[31] Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, et al. LAPACK users' guide. 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1999.

[32] Chatterjee P, O'Brien EJ, Li Y, González A. Wavelet domain analysis for identification of vehicle axles from bridge measurements. Comput Struct 2006;84(28):1792–801.

[33] O'Brien EJ, González A, Dowling J. A filtered measured influence line approach to bridge weigh-in-motion. In: Bridge Maintenance, Safety, Management and Life-Cycle Optimization - Proceedings of the 5th International Conference on Bridge Maintenance, Safety and Management. 2010, p. 965–72.

[34] Zhao H, Uddin N, Shao X, Zhu P, Tan C. Field-calibrated influence lines for improved axle weight identification with a bridge weigh-in-motion system. Struct Infrastruct Eng 2015;11(6):721–43.