AN EFFICIENT ALGORITHM FOR THE COMPUTATION OF THE FIRST PASSAGE TIME MAPS OF A GIVEN SIGNAL

KEVIN THEUNISSEN[1,2], VINCENT DENOËL[1]

ABSTRACT. The First Passage Time (FPT) of a given signal is the time required to reach a level X_f for the first time, when starting from a level X_0. For a given set (X_0, X_f), this time can be obtained by moving along the given signal and clocking the time on when passing through level X_0, then off when passing through level X_f for the first time. For random signals, the FPT is a random variable. Several samples of the first passage time are collected while moving along the given signal to compute the corresponding statistical moments and probability density function. The FPT map associates each statistical moment with every couple (X_0, X_f). A map is constructed for every desired moment. A naive implementation of the signal processing method to compute the FPT map would require to move along the signal as many times as the desired pairs (X_0, X_f). Instead, this paper introduces a new and optimized algorithm to establish the FPT maps of any statistical moment and, optionally, of the whole probability distribution. This algorithm combines different features to produce a computationally effective and memory-efficient method. The algorithm estimates the entire map by iterating only once across the signal. The accuracy of the algorithm has been assessed with particular cases where theoretical solutions exist as well as for lab data. Furthermore the algorithm's performance has also been evaluated in terms of computational burden. A specific attention is given to the computation time scaling for the evaluation of an FPT map, with respect to the signal size.

1. Introduction

Many fields of engineering are concerned with time varying phenomena. The development of cells, the spread of diseases, the progressive accumulation of energy in structural systems are just a few examples of applications where designers need to master the question of the time required to evolve from a known initial configuration, and to reach a target (usually design) configuration. The shortest time required for this transition is usually referred to the First Passage Time. It finds applications in chemistry [1], biology [2], wind engineering [3], sensor design [4], but also in financial applications [5] and the modeling of global warming [6].

In typical real-life applications the data that scientists are dealing with is corrupted by noise. It is therefore essential to develop signal processing techniques that are able to reflect most of the important features of random signals. Standard signal processing techniques include statistical estimators such as the average or standard deviation, the spectral density and autocorrelation [7], filtering and z-transform [8], envelope extraction [9, 10]. Also traditional signal processing techniques are associated with either the time domain, either the frequency domain, or sometimes hybrid time-frequency representations such as wavelet transforms. In this paper, we suggest to characterize the features of a measurement or synthesized random signal by means of its first passage time and present an efficient algorithm for this new signal processing technique.

Key words and phrases. reliability, signal processing, first passage map, level crossing.

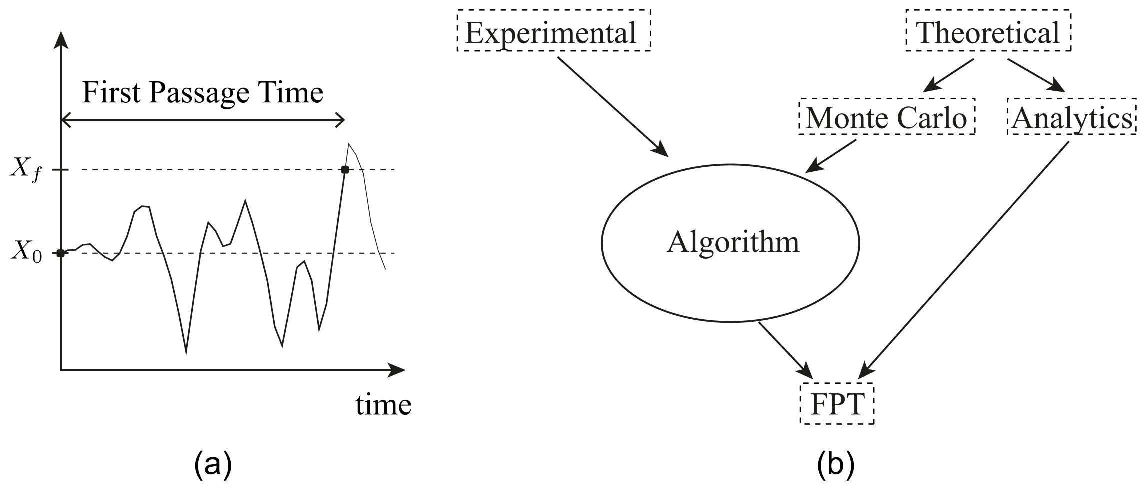

FIGURE 1.1. FPT maps from two different approaches

For a given discrete-time signal, the First Passage Time (FPT) is defined as the time required to reach a target level X_f for the first time after having started at a given level X_0, see Figure 1.1-a. This concept is discussed in detail in Section 2 but from now on it is possible to state that, if the signal is a random process, the FPT is a random variable. Accordingly, it is possible to calculate its mean, its standard deviation and even its Probability Density Function (PDF). In order to obtain a full picture of the specificities of the signal, the FPT maps of the various statistical moments of the FPT can be represented as a function of X_0 and X_f. Examples of maps of averages and standard deviations of FPT are given in [11] and [12], respectively.

In the literature, there exist mainly two very different contexts where FTPs are considered, see Figure 1.1-b. The first one is a theoretical approach in which the first passage times of some specific stochastic processes are studied. In some cases, closed form solutions or their asymptotic behavior are determined [13]; in others, numerical techniques based on the solution of the Fokker-Planck-Kolmogorov equation [14] or path integral methods [15] are used to determine the statistics of the FPT of such classes of processes. An efficient way to validate the results obtained with these semi-analytical techniques is to recourse to Monte Carlo simulations. It consists in generating samples of the considered processes, then for each sample determine the first passage times and proceed to a statistical analysis [16]. In this case it is necessary to have an efficient algorithm at hand in order to perform this statistical analysis and validate the theoretical results. Although some authors do validate their analytical solutions in this manner, there is usually very little information about the way the statistics of FPT from samples are obtained. The second context is the experimental one, where signal processing of measured data, acquired in the lab or in real conditions, is essential to scientific research. In this paper, we suggest to consider the FPT map as a new signal processing technique; it is therefore crucial to have a performant algorithm at hand to enjoy the possibility of this new tool. The proposed algorithm stands out from the current practice, e.g. [3], in the sense that the complete map, for various values of initial X_0 and final X_f values can be established by processing the signal only once.

The algorithm to determine the first passage time, as sketched in Figure 1.1-a, might be seen as pretty similar to the rainflow counting algorithm [17, 18] used for cycle-counting in structural fatigue. However, while in the latter case, it is clear that a drop would just fall from top to bottom along a signal, in the former case, it is in principle necessary to go back and forth in the sampled signal since there exist multiple initial times corresponding to the initial value X_0 and also multiple final times corresponding to the final value X_f. A simple extension of the rainflow counting would therefore actually turn out to be prohibitive.

This paper is organized as follows. The FPT algorithm is described in Section 2. This algorithm is able to compute various cumulants of the FPT, but also its PDF for any combinations (X_0, X_f). In Section 3, the efficiency of the algorithm is assessed by using three stochastic signals: an Ornstein-Uhlenbeck process, a Geometric Brownian Motion as well as a narrowband signal. In Section 4, numerical results obtained with the algorithm are compared with theoretical ones for the Ornstein-Uhlenbeck process. In Section 5, maps and histograms of the FPT computed for vibration data collected on a mechanical experimental setup are compared to those of a numerical model. Conclusions and further work then close this paper.

2. Algorithm

2.1. **Generalities.** Let X_i, $(i = 1, ..., n)$ a discrete-time signal, which can be seen as a realization of a random process. We present an algorithm for computing the First Passage Time maps. These maps represent some selected statistical cumulants of the FPT for various combinations of the initial and final values, X_0 and X_f. The first two cumulants are the average first passage time $(k = 1)$ and its variance $(k = 2)$. In general, the first passage time maps of the first few cumulants $k = 1, ..., p$ are computed. Since the standard error of the estimators increases with p, it is customary to opt for $p = 2$. However, it is not forbidden to use the proposed algorithm for much larger values. The authors have personally applied the algorithm up to $p = 4$ for very long time series.

We seek to compute the maps for N_{map} values of X_0 and X_f which are uniformly spaced between $L_{\min}$ and $L_{\max}$. These values are set to default at the 10^{th} and the 90^{th} percentiles of the given signal, as shown in Figure 2.1-a. The N_{map} equally spaced and ordered levels are noted L_{iL}, with $iL \in \{1, 2, ..., N_{\text{map}}\}$, $L_1 = L_{\min}$ and $L_{N_{\text{map}}} = L_{\max}$.

Beside cumulants, the proposed algorithm also optionally computes the histogram of the FPT for the same combinations of X_0 and X_f, or less if some data storage needs to be saved. Indeed, these histograms are recursively computed as explained next by counting the number of occurrences of first passage times in small predetermined bins $(b = 1, ..., n_{\text{bins}})$ which are allocated at the beginning of the signal processing. These bins can then be aggregated in order to reach some optimality in the density estimator [19, 20].

For cumulants, each of the FPT maps can therefore be seen as an $N_{\text{map}} \times N_{\text{map}}$ array gathering the statistics of the FPT for the different combinations of X_0 and X_f, and for $k = 1, \cdots, p$. The FPT map of the histogram, however, is a $N_{\text{map}} \times N_{\text{map}} \times n_{\text{bins}}$ array.

Since the same crossing levels are used for both X_0 and X_f, elements on the main diagonal of an FPT map are equal to 0 because there $X_0 = X_f$, and it takes no time to reach level $X_f = X_0$, starting from X_0. Also the FPT values above and below this diagonal respectively correspond to $X_f > X_0$ and $X_f < X_0$. The latter case may be less intuitive but is not irrelevant (e.g. [16]); it has practical applications in dissipative systems.

A naive implementation would entail iterating through the varying initial and final values, leading to reprocessing of the signal for every new combination of X_0 and X_f, and possibly for every other cumulant of interest. Instead, a new algorithm was developed in order to efficiently compute FPT maps of a given signal. The conceptual idea is the same as the algorithm presented in [3], in which the statistics of the FPT are computed for all combinations of X_0 and X_f simultaneously. This algorithm is described by the authors as a variant of the rainflow algorithm. By virtually rotating the time signal by 90°, one can visualise a water droplet falling and its trajectory defines the main envelope, which is stored. Subsequently, a partial envelope can be constructed for every point of the time signal utilising the same process until the water droplet meets the path of the main envelope. Once the main envelope is reached, the partial envelope aligns with it until the end of the time signal. Afterwards, the FPTs are derived from the computed envelopes.

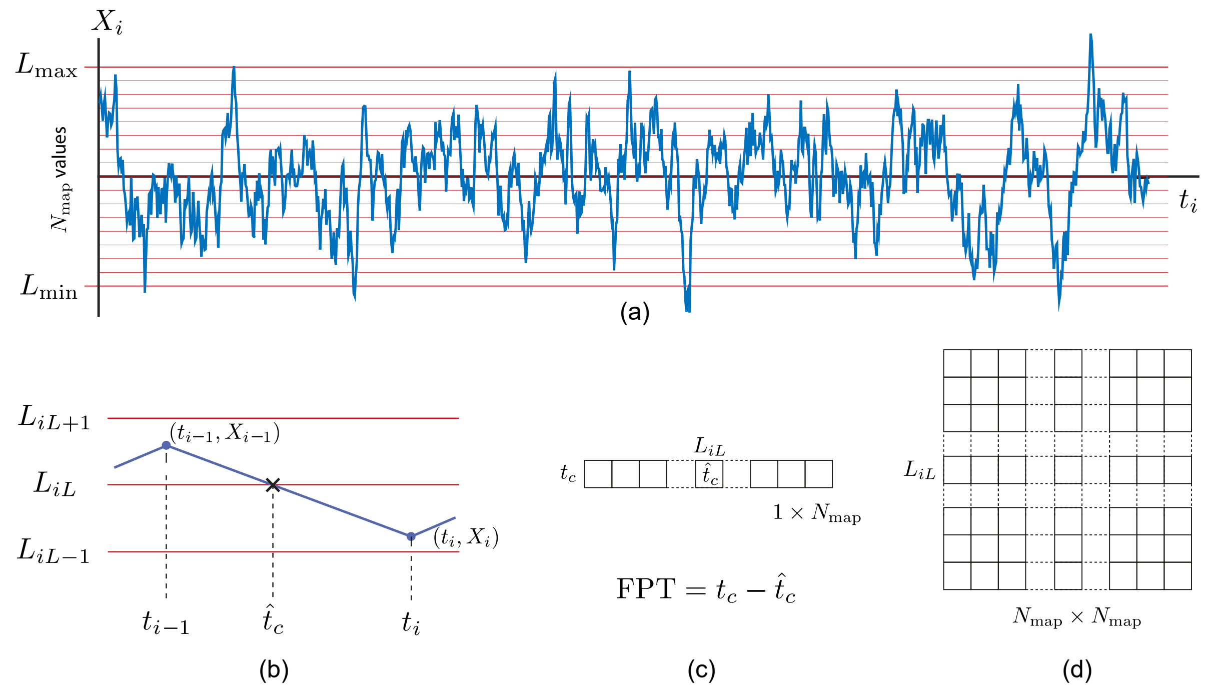

FIGURE 2.1. General overview of the proposed algorithm: (a) definition of levels, spanning a pre-defined value of the minimum and of the maximum, (b) linear interpolation to compute the crossing time $\hat{t}_c$, (c) storage of $\hat{t}_c$ into the vector t_c and FPT calculation (d) storage of FPT into the iL^{th} row of a map of a raw moment

The proposed algorithm differs as the signal is analyzed backward and does not rely on a variant of the rainflow algorithm anymore. As a first advantage, only the last encountered time for each level is required to calculate the FPTs so that, in the reverse analysis process, only those crossing times are temporarily stored. This considerably reduces the memory storage. Also, the specific algorithmic arrangements presented in detail later ensure that the signal is processed only once, which reduces the CPU time for the signal processing.

For a compact view, three pseudo-codes are provided. The main function of the algorithm is 1. Algorithms 2 and 3 correspond to the functions isALevelCrossed() and isAnotherLevelCrossed() used by FPT_map(). They are related to the detection of crossed levels. These three functions are described in more detail later in this document.

In essence, the proposed algorithm is based on two different parts operating in cycle over each time step. The first part is the detection of level crossing and the second part is, if a level has been crossed, the computation of the FPT.

Firstly, the algorithm aims at keeping track of time instants where levels L_{iL} are crossed, while rewinding the signal. The level crossing detection hinges on two main functions aiming at efficiently finding a crossed level: isALevelCrossed() and isAnotherLevelCrossed(). These functions require the sign of the slope of the previous and current crossed levels L_{iL}. The slope is said null if $(X_0 = X_f)$, positive if $X_{i-1} < X_i$ and negative instead. The first crossed level L_{iL} is determined based on the last point of the signal. Once the first crossed level has been determined, a pointer is used to move backward through the signal, point by point.

```
FPT_map()
Input: $t_i$, $X_i$, $\forall i = 1, ..., N$
Output: FPT maps of cumulants and histograms of FPT
Discretisation of $X(t)$ into $N_{map}$ equidistant levels $L$
Determination of the bin width and the number of bins based on the input coef_dt
Initialisation of the last crossed level and its slope based on $X(N)$
for $i \leftarrow N$ to 1
    Save of the slope for the last crossed level $L_{iL}$
    isALevelCrossed()
    while A level is crossed
        Storage of the current time $\hat{t_c}$ for the corresponding crossed level $L_{iL}$ into $t_c$
        $FPT \leftarrow t_c - \hat{t_c}$
        Update $iL^{th}$ row of maps of raw moments
        Update histograms
        isAnotherLevelCrossed()
    end
end
Transformation of raw moments into cumulants
```

Algorithm 1: FPT_map()

The function isALevelCrossed() is always used first in order to check if at least one level is crossed. The efficiency of this function is based on a fast detection of level crossing, and if so, which level(s) has(have) been crossed. Thanks to simple if/else statements, the sign of the current slope and the sign of the slope of the last crossed level, only one level has to be checked in order to detect if a level is crossed. If the checked level is not crossed then no other test has to be performed. In the case of a null slope, the algorithm simply checks if the current level L_{iL} is equal to X_{i-1}. There might be several level crossings over one time step, when the gradient of the signal is large. Therefore, the function isAnotherLevelCrossed() is used. This function is a lighter version of the function isALevelCrossed() which improves the speed of the algorithm. The function isAnotherLevelCrossed() will be called while it detects a level crossing after each FPT calculation.

Secondly, when a level is crossed, say level iL, a crossing time $\hat{t_c}$ is calculated by linear interpolation between t_i and t_{i-1},

$$\hat{t_c} = \frac{L_{iL} - X_{i-1}}{X_i - X_{i-1}} (t_i - t_{i-1}) + t_{i-1} \tag{2.1}$$

to provide a refined estimation, see Figure 2.1-b.

This time $\hat{t_c}$ is stored at the iL^{th} entry of a $1\text{x}N_{\text{map}}$ vector t_c, possibly replacing information previously stored at the same location, to indicate that level L_{iL} has been crossed at that time, see Figure 2.1-c. The vector t_c is used to store the last value $\hat{t_c}$ for each level L_{iL}. In addition, a logical vector $1\text{x}N_{\text{map}}$ called L_{met} is used to track which level has already been encountered since the beginning of the signal processing; $L_{\text{met}}(iL)$ is therefore set to true. We found this vector useful to be able to take advantage of the vectorial programming capabilities of the MATLAB language [21].

Then, the FPT of all encountered levels is simply the difference $t_c - \hat{t_c}$.

```
isALevelCrossed()
Input: $X_i$, $X_{i-1}$, $L$, $slope\_iL$, $iL$, $N_{\mathbf{map}}$
Output: $iL$, $slope\_iL$, $crossing$, $crossing\_diff\_slope$
$crossing \leftarrow$ FALSE
$crossing\_diff\_slope \leftarrow$ FALSE
if $X_i > X_{i-1}$
    if $X_i \geq L_1$ and $X_{i-1} \leq L_{N_{\mathbf{map}}}$
        if $slope\_iL$ is positive
            if $iL - 1 > 0$ and $X_{i-1} \leq L_{iL-1}$
                $iL \leftarrow iL - 1$
                $slope\_iL$ is positive again
                $crossing \leftarrow$ TRUE
            end
        else
            if $iL > 0$ and $X_{i-1} \leq L_{iL}$
                $iL \leftarrow iL - 1$
                $slope\_iL$ is positive now
                $crossing \leftarrow$ TRUE
                $crossing\_diff\_slope \leftarrow$ TRUE
            end
        end
    end
else if $X_i < X_{i-1}$
    if $X_i \leq L_{N_{\mathbf{map}}}$ and $X_{i-1} \geq L_1$
        if $slope\_iL$ is negative
            if $iL + 1 \leq N_{\mathbf{map}}$ and $X_{i-1} \geq L_{iL+1}$
                $iL \leftarrow iL + 1$
                $slope\_iL$ is negative again
                $crossing \leftarrow$ TRUE
            end
        else
            if $iL \leq N_{\mathbf{map}}$ and $X_{i-1} \geq L_{iL}$
                $iL \leftarrow iL - 1$
                $slope\_iL$ is negative now
                $crossing \leftarrow$ TRUE
                $crossing\_diff\_slope \leftarrow$ TRUE
            end
        end
    end
else
    if $X_{i-1} = L_{iL}$
        $slope\_iL$ is horizontal
        $crossing \leftarrow$ TRUE
    end
end
```

Algorithm 2: isALevelCrossed()

```
isAnotherLevelCrossed()
Input: $X_{i-1}$, $L$, $slope\_iL$, $prev\_slope\_iL$, $iL$, $N_{map}$
Output: $iL$, $crossing$

$crossing \leftarrow$ FALSE
if $slope\_iL$ is positive
    if $prev\_slope\_iL$ is positive
        if $iL - 1 > 0$ and $X_{i-1} \leq L_{iL-1}$
            $iL \leftarrow iL - 1$
            $crossing \leftarrow$ TRUE
        end
    else
        if $iL > 0$ and $X_{i-1} \leq L_{iL}$
            $crossing \leftarrow$ TRUE
        end
    end
else if $slope\_iL$ is negative
    if $prev\_slope\_iL$ is negative
        if $iL + 1 \leq N_{map}$ and $X_{i-1} \geq L_{iL+1}$
            $iL \leftarrow iL + 1$
            $crossing \leftarrow$ TRUE
        end
    else
        if $iL \leq N_{map}$ and $X_{i-1} \geq L_{iL}$
            $crossing \leftarrow$ TRUE
        end
    end
end
```

Algorithm 3: isAnotherLevelCrossed()

These values are samples of the random variables corresponding to the first passage times. The integer powers $k = 1, \cdots, p$ of the differences $t_c - \hat{t}_c$ are added to the maps of raw moments which will be used, ultimately, to compute the cumulants of the first passage times, see Figure 2.1-d.

The cumulants are then updated inside matrices. Different cumulants can also be calculated by changing the power k. The histograms are also updated. These FPT histograms are calculated at different locations of a FPT map. The bin width can be determined based on the chosen accuracy and is a multiple (coef_dt) of the time step. A posteriori, the FPT histograms can be post processed using different formulas [19, 20] in order to reduce the number of bins. Before moving the pointer onwards, isAnotherLevelCrossed() checks if another level L is crossed. If this is the case, the FPT calculation is performed once again until the function isAnotherLevelCrossed() returns *false*, i.e that no level crossing has been detected. Once the pointer has reached the first data point in the signal, the processing is complete. The algorithm returns numerical values of the FPT maps which can then be independently graphically represented.

2.2. **Specific features.** In order to optimize the algorithm and its performances, several specific features have been used.

The first one is taking advantage of the vectorization from MATLAB by using the vector L_{met}. This vector is useful to reduce de computational time by avoiding redundant loops. Moreover, the vectorization of MATLAB has also been used in the computation of the FPT histograms. Indeed by determining all cells of the 3D matrix that need to be updated, the increment of these cells is performed directly without any explicit loop.

A second feature of the proposed algorithm concerns the windowing of signals with missing data. If a part of a signal has been corrupted or is missing, the user can discard that part. Indeed, since the algorithm also returns the occurrences of the first passage time, by sending one fraction of the signal at a time to the algorithm, the user can get the matrices with the data that is already processed for the cumulants calculation. A simple averaging of these matrices is possible and allows to concatenate information coming from several fragments of a corrupted signal with missing pieces.

The last feature concerns the handling of signal with low digitalization or rounded values. Indeed, in case of low digitalization, there are two major issues : (i) identical consecutive values resulting in a slope equal to zero and (ii) data values which correspond exactly to a level L_{iL}. While in a signal with significant digitalization, these two cases are marginal, they occur much more frequently in case of low digitalization. In order to limit the bias, the first issue is easily solved by taking into account the special case of the null slope in the functions isALevelCrossed() and isAnotherLevelCrossed() while the second issue required more attention. In the original version of our algorithm if a data point was equal to a level L_{iL} this point was taken into account twice. To face this issue it has been decided that the FPT calculation is skipped if $L_{iL} = X_i$ except for the point $X(n)$, which is the first point encountered by the algorithm. This removes the double counting.

3. Efficiency of the proposed algorithm

In this section, the efficiency of the algorithm is discussed.

As a first comparison, we were able to re-use the data utilized in [3]. Both algorithms have processed this data. The algorithm featured in [3] took 2600 seconds to complete its calculations whereas, the proposed algorithm only required 6 seconds, resulting in a calculation time 400 times shorter.

As a second comparison, we conducted a parametric study to demonstrate the effectiveness of the proposed algorithm in addressing various aspects of a problem. In this case three types of signals with very different spectra have been used: an Ornstein-Uhlenbeck (O-U) process, a Geometric Brownian Motion (GBM) and the displacement response of a single oscillator (1-DOF) with mechanical properties as per [22] (m =28.138 to, k =22495 kN/m, ξ =4%) and subjected to narrowband white noise excitation in the range [4; 5] Hz.

The first two processes are well-known theoretical stochastic processes. They have been generated numerically by using the Euler scheme and the definition of the Wiener process [16]. The third sample has been obtained by solving the motion equation of the single degree-of-freedom oscillator using the Newmark scheme under a loading history sampled from a narrow banded white noise around the first bending mode (4.5 Hz) of the corresponding specimen. Figure 3.1 shows two samples each of these processes. A close-up view in the range $t \in [1000; 1020]$ s offers a better picture of the processes at stake. Also, while the geometric Brownian motion is an unbounded process, the other two considered processes are Gaussian, as indicated by the histograms of the sampled time series. A reference case is obtained by computing the FPT map of samples of these three random processes. They are sampled at 500 Hz, for a duration of 10 s (i.e. contain 5000 data points) and the FPT map is established for $N_{\text{map}} \times N_{\text{map}} = 400$ values. For the three processes (O-U, GBM and 1-DOF), the mean execution times, averaged over 150 repetitions, are respectively 0.0175, 0.0105 and 0.0144 seconds. Unlike more traditional processing techniques such as the Fourier transform, the computational burden associated with the establishment of an FPT map depends on the specific nature of the signal. In particular, the computational time associated with the geometric Brownian motion is significantly shorter than for the other two processes. As detailed in the following, this is a consequence of the drift associated with a geometric Brownian motion.

A parametric study has been carried out in order to determine the influence on the runtime (to compute the whole FPT map) of three major parameters that have been identified as having a possible influence on the quality of the resulting FPT map:

(1) the total length of the signal,
(2) the sampling frequency of the signal,
(3) the number of level discretization N_{map}.

The results of this analysis are reported in Figure 3.2 in terms of a wall-clock time, i.e. the elapsed time as displayed on a chronometer. This measure has been chosen over CPU-time as the latter is dependent on the number of processing cores utilized. The wall-clock time is represented for each studied parameter. Moreover, for enhanced analysis of the influence of the different parameters on the total execution time, two distinct times, namely t_{cal} and t_{level}, have been utilized. The time t_{cal} is the time required to calculate FPTs and to store them into matrices while t_{level} is the time needed to detect if one level is crossed. Together, t_{cal} and t_{level} constitute the total execution time. The order of magnitude of the Matlab time accuracy is roughly 0.1s. The number of repetitions is therefore chosen as 150 in order to better discriminate between the different cases. Based on the number of measurements and the time accuracy of Matlab, a threshold value equal to $0.1/\sqrt{150} = 8.10^{-3}$s has been estimated. Below this value, the estimation of the wall-clock time is inaccurate and the corresponding zone is colored in light gray because the calculated time may be affected by other CPU tasks, for example. These performance tests have been carried out on a computer with the following specifications: CPU: Intel(R) Core(TM) i7-10510U CPU 2.30GHz, RAM: 16Go 2667MHz.

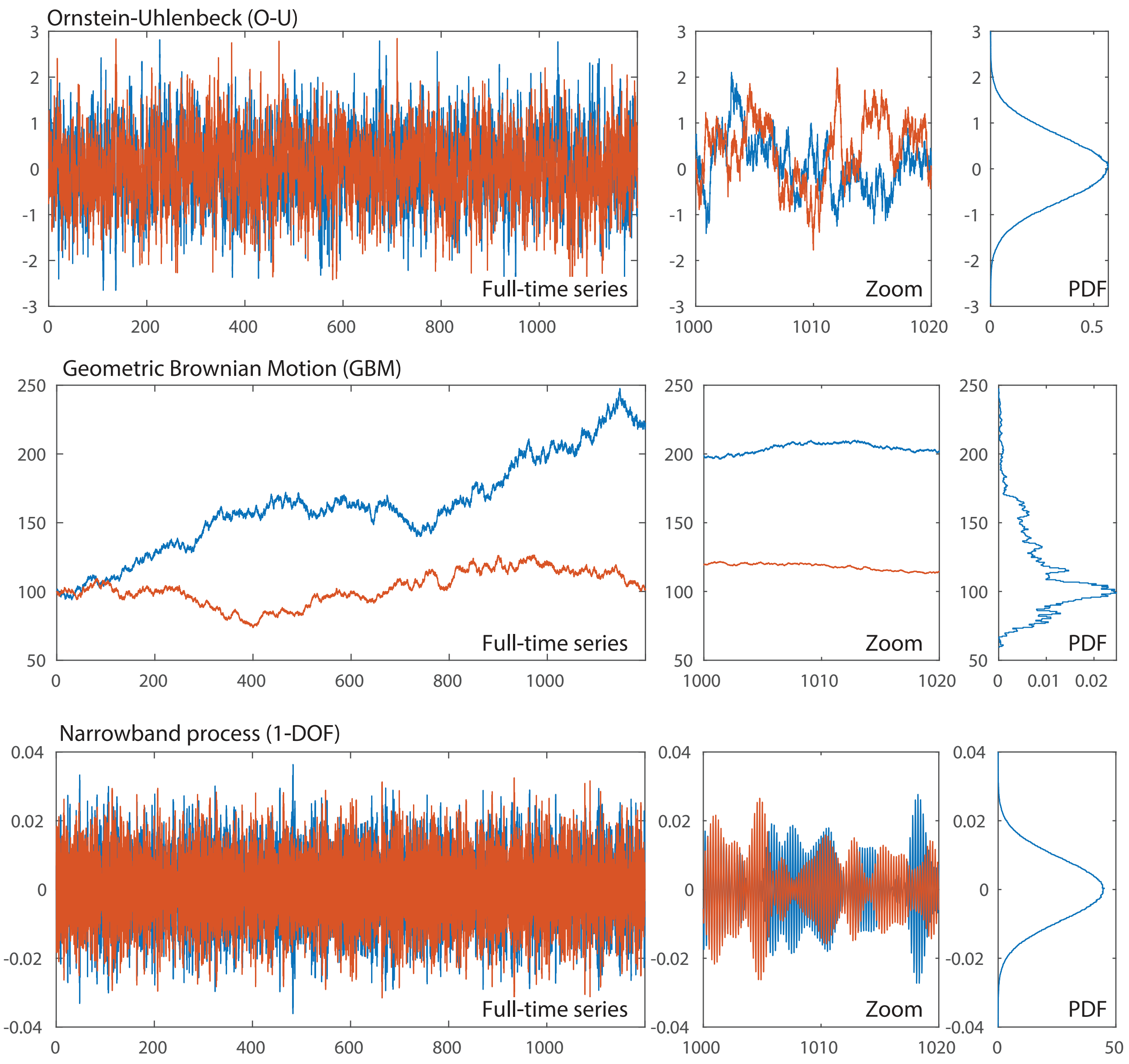

FIGURE 3.1. Generated samples of the three considered stochastic processes: the Ornstein-Uhlenbeck process, the Geometric Brownian Motion and a narrowband process. Two samples are shown for each process.

In Figure 3.2-a, the influence of the signal length is shown. The same sampling frequency is set at $f_s = 500$ Hz, and the parametric analysis concentrates on the signal size, which ranges from a total duration $T = 0.2$ s for $n = 10^2$ to $T = 2000$ s for $n = 10^6$ points in the signal. In the worst case, the wall-clock time grows proportionally to the number of points. The geometric Brownian motion is the most advantageous case due to the peculiar occurrence of crossing times of this random process. In Figures 3.2-d, 3.2-g and 3.2-j, t_{cal} and t_{level} are represented for each process. For the O-U and 1 DDL processes, t_{cal} always exceeds t_{level} in value by at least one order of magnitude. This is not the case for the GBM process for which t_{level} is getting closer to t_{cal} and even becomes larger as the signal size increases. This deviation from the other two processes can be explained by the exponential global behavior of the GBM process. Eventually, this process stops trailing through the lower levels after a certain time. Therefore, for the same number of points, fewer FPTs are calculated for the GBM process than for the other two processes. As the signal size increases, the algorithm spends proportionally more time checking if a level is crossed than computing FPTs.

Figure 3.2-b illustrates the influence of the sampling frequency. A total duration of $T =$ 10 s is maintained and the sampling frequency is adjusted so that using $n = 10^2$ data points corresponds to a sampling frequency $f_s = 10$ Hz and using $n = 10^6$ data points corresponds to a sampling frequency $f_s = 100{,}000$ Hz. In Figures 3.2-e and 3.2-h, it can be observed that t_{level} is increasing faster than t_{cal}. Indeed, doubling the sampling frequency f_s does not necessarily double the number of calculated FPTs; in practice, this number usually decreases. However, the number of data points doubles which results in a larger increment of t_{cal} compared to t_{level}. In Figure 3.2-k, the behavior of t_{cal} is different. At a sampling frequency $f_s \approx 100$ Hz, t_{cal} is constant, which can be attributed to the numerical scheme. When the time step is small enough, i.e. the sampling frequency f_s is sufficiently high compared to the range of the narrowband white noise used as an input, the Newmark scheme under the same loading history will compute identical response signals, corresponding to the same number of calculated FPTs. Finally, it can be concluded that t_{cal} becomes dominant over t_{level} for each signal in this section beyond a certain high sampling frequency f_s.

In Figure 3.2-c, the influence of N_{map} is shown. The proposed algorithm displays uniform behavior across all processes: the wall-clock time mainly depends on the number of calculated FPTs, i.e. is proportional to N_{map}^2. This trend is also observable in Figures 3.2(f), 3.2(i) and 3.2(l) where t_{cal} is at least one order of magnitude higher than t_{level}. For the same signal duration and sampling frequency, the O-U process exhibits more threshold crossings, thereby contributing to its longer runtime when compared to the 1DOF and GBM processes.

Memory efficiency is a crucial aspect of this algorithm. Furthermore, this algorithm has the added benefit of memory usage only relying on the parameter N_{map} and the number of histograms as well as their bin width. This added perk further legitimises the algorithm's efficacy as it requires minimal storage for computing statistical moments of FPT.

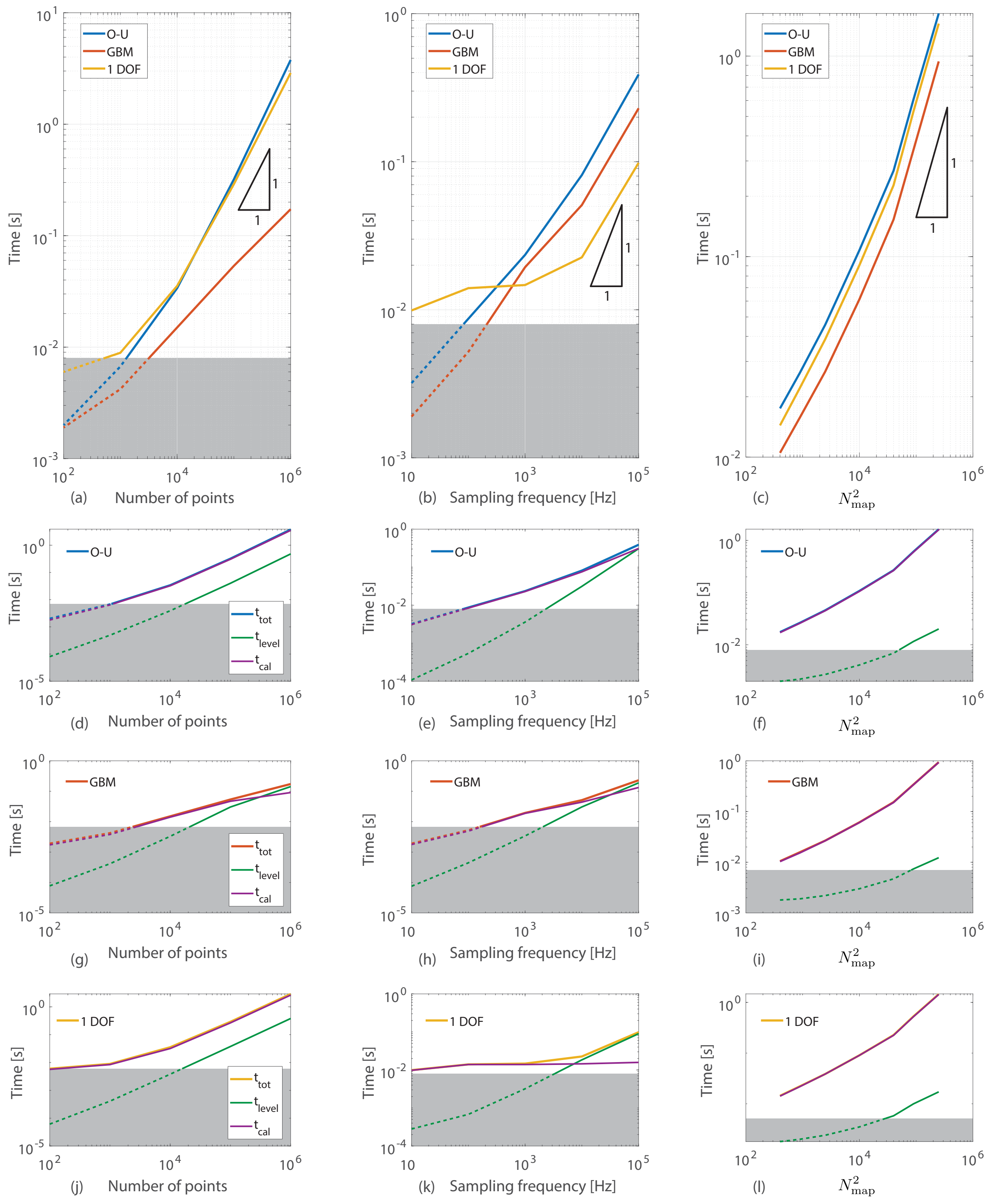

FIGURE 3.2. Execution time of the algorithm based on the influences of the total length (a,d,g,j), of the frequency (b,e,h,k) and of the parameter N_{map} (c,f,i,l) for the Ornstein-Uhlenbeck process (d,e,f), the Geometric Brownian Motion (g,h,i) and a narrowband process (j,k,l)

4. Verification

In this section, the algorithm's results are compared with theoretical values to validate the accuracy of the algorithm. The well-known Ornstein-Uhlenbeck process has been chosen, which is the solution of the stochastic equation (7.1). Please refer to Appendix A for further details. The Appendix also contains information about the average (7.2) and standard deviation (7.3) of the first passage time. The mathematical development of the Probability Density Function (PDF) has been summarized in the Appendix based on [23]. The numerical values chosen for the parameters are $\alpha = 0$ and $\beta = \sigma = 1$, and the stochastic equation is solved numerically with an Euler-Maruyama scheme [24]. Following the same Monte Carlo procedure as in the previous section, samples of this O-U process are generated and the maps of average FPT and standard deviation of FPT are computed. Histograms of the first passage time are also computed for some selected combinations of (X_0, X_f).

The stochastic differential equation has been integrated with three different time steps, corresponding to frequencies 500 Hz, 5000 Hz and 50000 Hz, which allows appreciating the impact of this simulation parameter on the accuracy of FPT maps. Moreover, several signals have been used and the results of each signal have been merged in order to encounter at least 10^6 FPTs for each combination of (X_0, X_f). The chosen number minimises uncertainties on the FPT statistics. In Figure 4.1-a, the average FPT map is represented. The numerical results properly match the theoretical results summarized in the Appendix. As expected, greater sampling frequencies improve accuracy. Figures 4.1-(b,c,d) show absolute errors at sampling frequencies of 500 Hz, 5000 Hz and 50000 Hz. The Table 1 displays the highest relative errors for the average FPT at various sampling frequencies. These values are located near the main diagonal of the map where X_0 and X_f are close to each other. They correspond to very short passage times and are affected by the discretization.

Sampling frequency	500 Hz	5000 Hz	50000 Hz
Maximum error on average of FPT	36%	11%	4%
Maximum error on STD of FPT	19%	6%	3%

Table 1. Maximum errors for the average and the STD of FPT for different sampling frequencies

In Figure 4.2-a, the STD of FPT is represented. The curves overlap significantly, similar to the average FPT map. In Figures 4.2-(b,c,d), relative errors are plotted for frequencies of 500 Hz, 5000 Hz and 50000 Hz. The maximum values are listed in Table 1. It can be seen that the maximum relative errors on the map of the STD of FPT are lower than the maximum errors of the average FPT map for the same frequency. The largest error is still located near the main diagonal. The error reduces to below 1% when considering combinations of (X_0, X_f) that satisfy the constraint $|X_0 - X_f| \gtrsim 0.2$. The algorithm can precisely compute the FPT cumulants. However, the values around the main diagonal of the map are largely impacted by the sampling frequency. This limitation is more attributable to the simulation process rather than the algorithm itself.

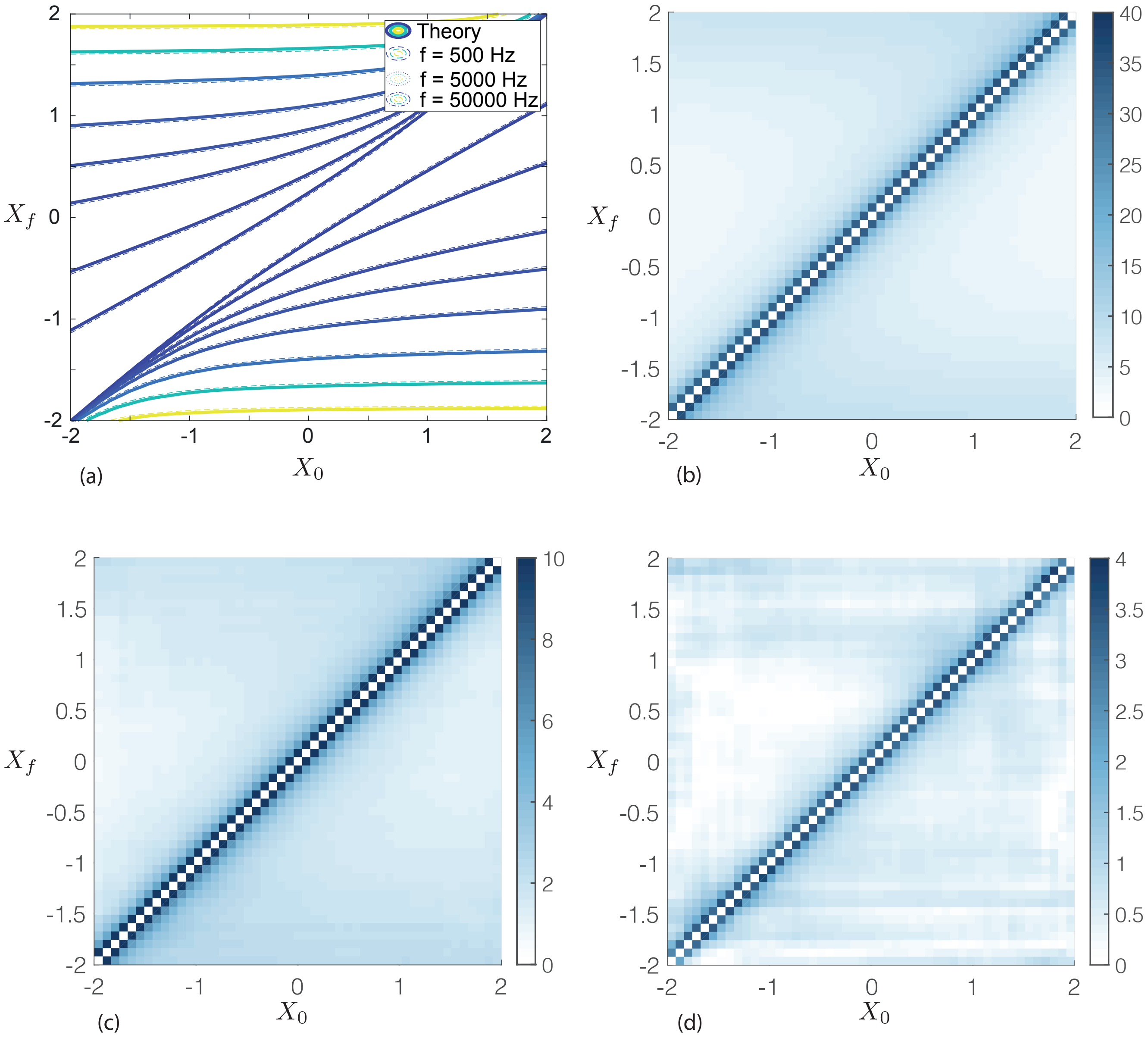

FIGURE 4.1. (a) Average FPT map, (b,c,d) Error map [in %] of the average FPT for a frequency of (b) 500Hz, (c) 5000Hz and (d) 50000Hz.

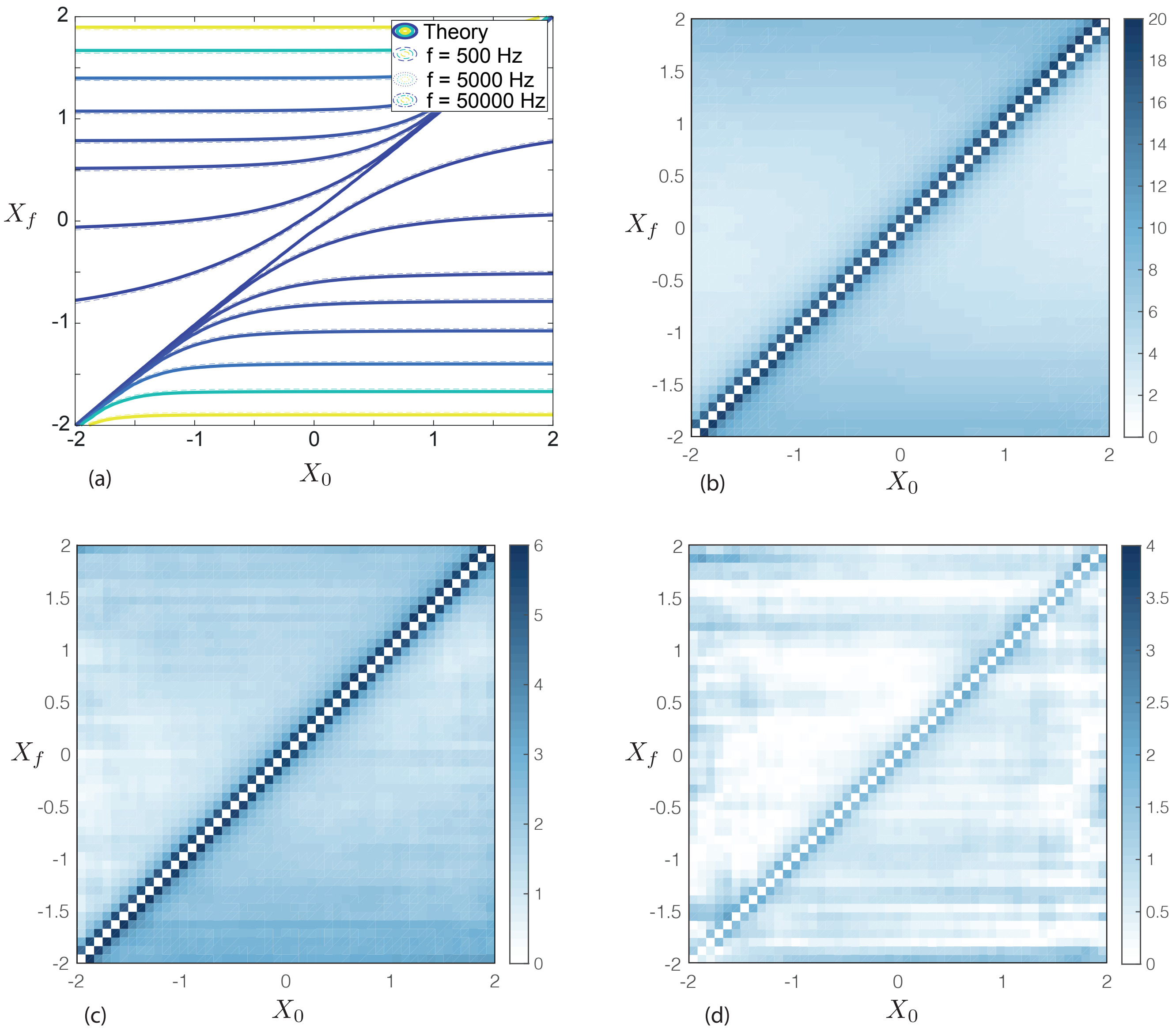

FIGURE 4.2. (a) STD of FPT map, (b,c,d) Error map [in %] of the STD of FPT for a frequency of (b) 500Hz, (c) 5000Hz and (d) 50000Hz.

It has been decided to draw the histograms of the FPT for 9 combinations of (X_0, X_f) located at different places of the FPT map, in order to get a better understanding of the distribution of the FPT for a given set (X_0, X_f). These empirical PDFs are shown in Figure 4.3 together with the analytical solution, given as a reference. The location of the combination (X_0, X_f) is indicated by a red square on the pictogram. In Figures 4.3-(c,e,g), corresponding to combinations located close to the diagonal of the map, the order of magnitude of the average FPT is the same as the time step or even below. It means that FPTs are over-estimated due to the large time step resulting in a higher mean value. This explains the large relative errors computed in that area. By increasing the sampling frequency, the time step is reduced and for 50000 Hz, the shape of the PDF is well represented. For Figures 4.3-(a,b,d,f,h,i), corresponding to combinations (X_0, X_f) lying out of the main diagonal of the map, two observations can be made:

(1) The shape of the PDF matches the theoretical one. However, it can be observed that the PDF is slightly overestimated for 500 Hz and 5000 Hz for higher FPT values. This is a well-know issue [25, 26] that directly comes from the frequency choice of the numerical signal simulation. By using a smaller time step (= a higher sampling frequency), the probability of obtaining extreme FPT values is reduced. Hence, it follows that the results obtained using a sampling frequency of 500 and 5000 Hz slightly overestimate the probability of large FPT values.

(2) The O-U process is symmetrical and therefore, the PDFs of FPT for (X_0, X_f) and $(-X_0, -X_f)$ are identical. As a result, the maps and PDFs enjoy a central symmetry about $(X_0, X_f) = (0, 0)$. The proposed algorithm can adequately model this symmetrical behavior.

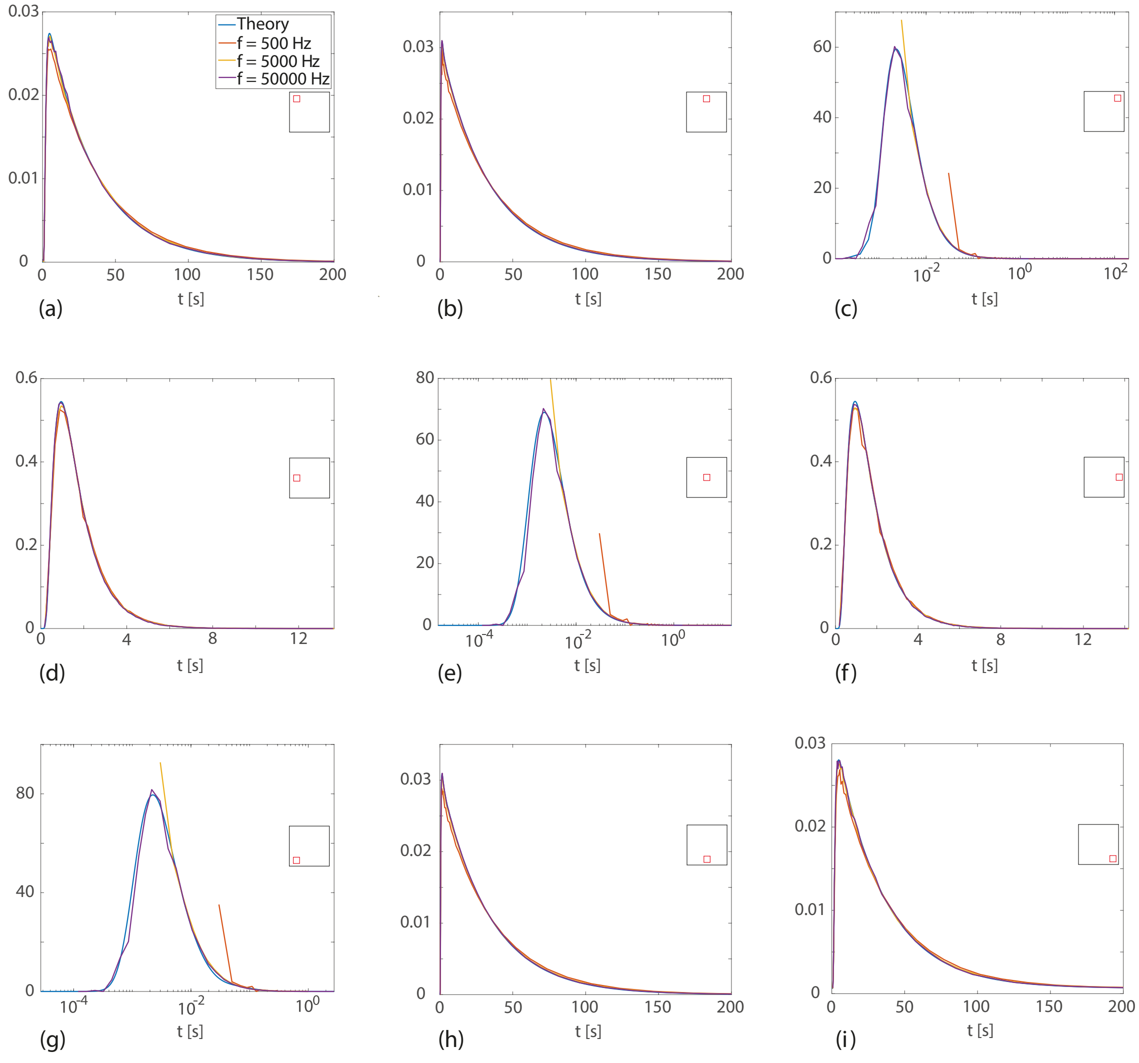

FIGURE 4.3. PDFs of FPT: (X_0, X_f) = (a) $(-1.84, 1.84)$, (b) $(0.04, 1.84)$, (c) $(1.76, 1.84)$, (d) $(-1.84, 0.04)$, (e) $(-0.04, 0.04)$, (f) $(1.84, -0.04)$, (g) $(-1.84, -1.76)$, (h) $(-0.04, -1.84)$, (i) $(1.84, -1.84)$

5. Illustration of the computation of FPT maps for experimental data

In this section, the proposed algorithm has been used in order to compute the FPT maps and PDFs of velocities measured on a steel strip. The experimental setup is similar to the one used in [27, 28]. It has been later reproduced in [29] and it is shown in Figure 5.1. The setup comprises a vertical steel strip and a horizontal shaker located at its bottom end, with pretension added by attaching a mass. Both ends of the steel strip are clamped. A limited-band white noise signal has been generated by the shaker. The frequency of this signal is located in the range of [35;43]Hz in order to contain only the second bending mode frequency of the strip, which is equal to 39.25Hz. The velocity of the steel strip has been measured by a laser located at point P. This point corresponds to the antinode of the second bending mode.

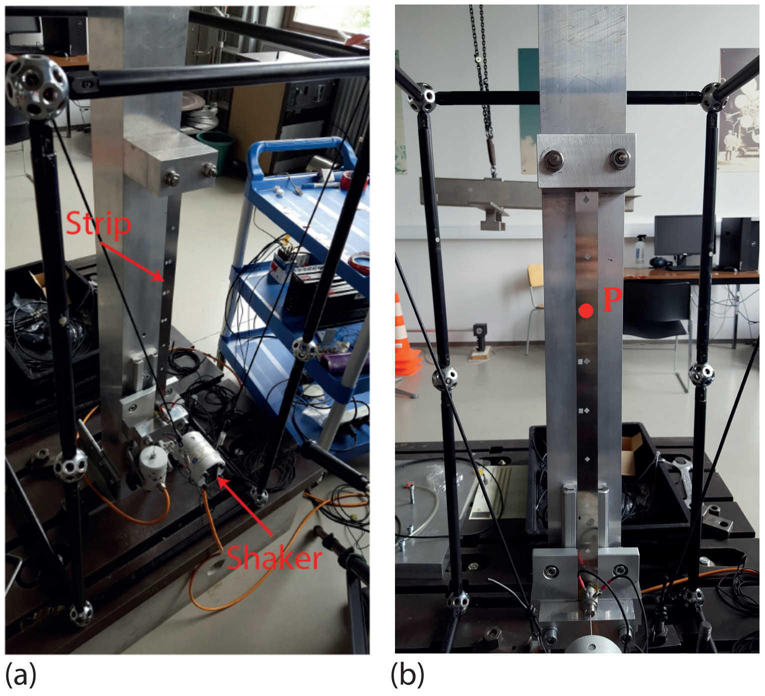

Figure 5.1. Experimental setup of the steel strip: (a) global view and (b) front view

In parallel, a twin model has been realised and updated. This model employs 2-D beam finite elements because our attention is focused only on flexion modes, especially on the second bending mode. The model parameters are based on the physical and mechanical properties of the steel strip and the shaker. The parameters values of the bond between the shaker and the steel strip as well as the values of the rigidity of the clamped supports have been adjusted by means of a Bayesian updating technique. Based on measured modal information (frequencies & mode shapes), the Metropolis-Hasting (MH) algorithm has been used to determine the distribution and most probable values of model parameters. Then, starting from the most probable values, a non linear least squares fitting was conducted to obtain the best suited parameters. This methodology gives a good confidence in the obtained values and the accuracy of the updated numerical model. More details can be found in [30]. The frequencies and the Modal Assurance Criterion (MAC) values can be found in Appendix B. Based on the aforementioned results, it is deemed that the numerical model is an accurate twin of the experimental setup.

In Figure 5.2 (a) and (b), the average FPT map and the STD of FPT map are drawn respectively. The results indicate a good agreement between the statistical moments of FPT of the numerical model and the experimental setup. However, a slight difference appears in Figure 5.2 (b) around $X_f = 0$ for large values of $|X_0|$.

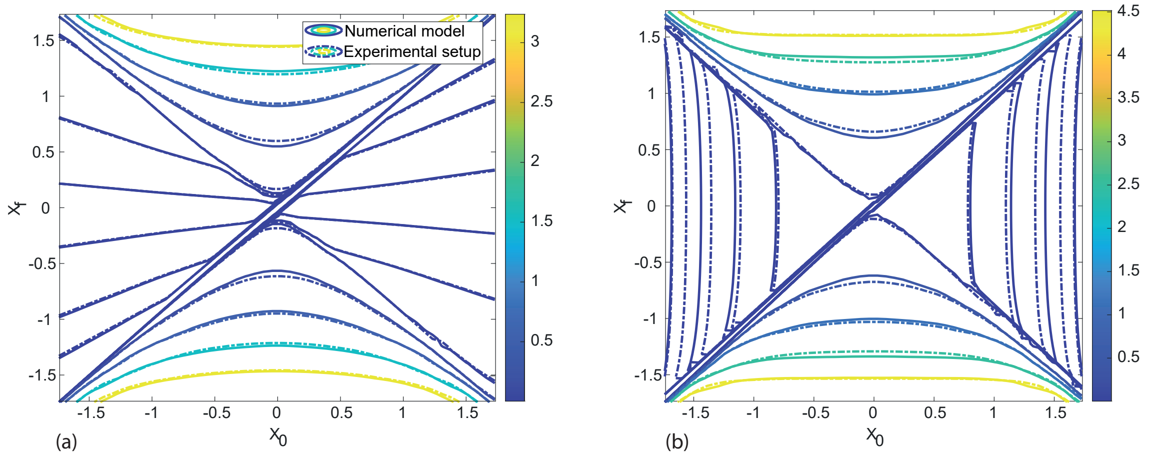

FIGURE 5.2. Comparison of FPT maps between the experimental and numerical results: (a) Average FPT map and (b) STD of FPT map

In Figure 5.3, nine different combinations of (X_0, X_f) have been selected to plot the corresponding FPT histograms. First of all, the measured and simulated signals are both symmetrical, which can be seen by comparing Figures 5.3 (a) and (i), Figures 5.3 (b) and (h) and Figures 5.3 (d) and (f). These histograms tend to be identical. Then, by comparing the experimental and numerical results, it can be observed that the empirical PDFs are similar, which supports the fact that the updated numerical model is accurate enough to reproduce the measured data. However, the difference that can be observed in Figure 5.2 (b) can be explained by comparing the numerical and experimental PDFs of FPT in Figures 5.3 (d) and (f). Indeed, it can be seen that the experimental PDFs values are roughly higher around 7.10^{-3}s than the numerical PDFs values. It results in lower STDs of FPT for the experimental results than the numerical results. This observation is also supported by the fact that the STD of FPT is slightly increasing as $|X_0| \to 0$, when $X_f \approx 0$, which is shown in Figure 5.2 (b). For $X_f \approx 0$, the experimental contour lines are situated to the right of the numerical ones when $X_0 < 0$ and the opposite when $X_0 > 0$. Consequently, this indicates the slightly larger STD in the numerical model.

As a summary, the good agreement between the numerical and experimental results shows the new prospects offered by the proposed algorithm to compute FPT statistics and histograms based on a measured signal, which comes either from an experimental setup or from a numerical model. Moreover, each 1-hour signal contains more than 7 million points, with a sample frequency of 2048Hz. The proposed algorithm computed all FPT maps in 30 seconds using the same computer as described in Section 3.

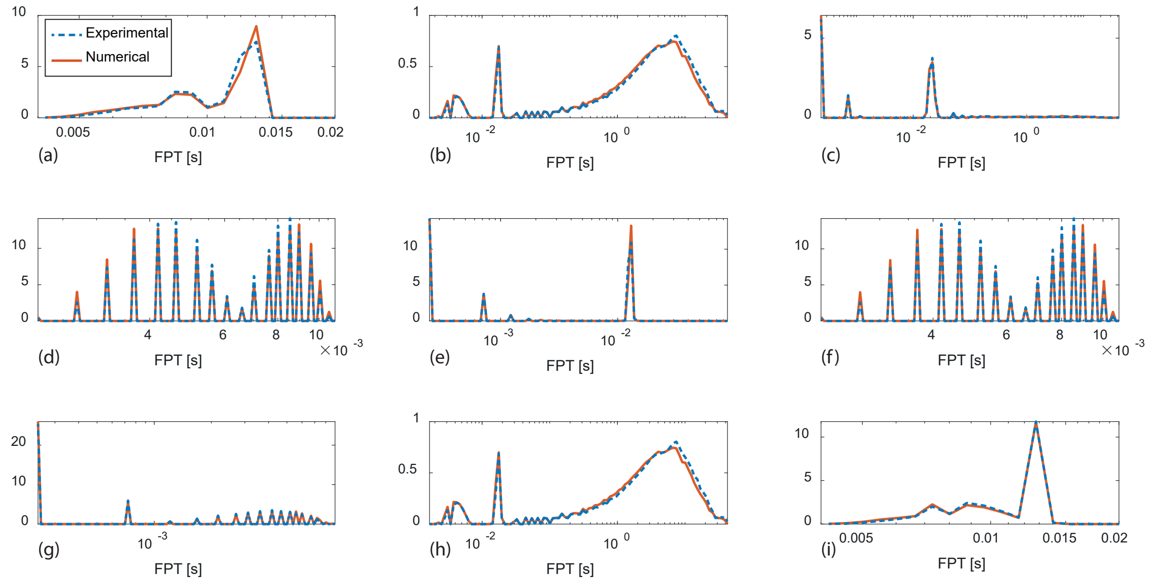

FIGURE 5.3. Histograms of FPT for the numerical (plain line) and experimental (dashed line) results: (X_0, X_f) = (a) $(-1.59, 1.59)$, (b) $(0.04, 1.59)$, (c) $(1.52, 1.59)$, (d) $(-1.59, 0.04)$, (e) $(-0.04, 0.04)$, (f) $(1.59, -0.04)$, (g) $(-1.59, -1.52)$, (h) $(-0.04, -1.59)$, (i) $(1.59, -1.59)$

6. CONCLUSION

We have presented a novel and optimized algorithm to use as a signal processing tool for discrete-time signals. With the proposed algorithm, a sample of a stochastic signal is associated with FPT maps. These maps represent statistical information about the FPT of any order, such as the average and the standard deviation. Moreover, the histograms of the FPTs can be obtained for different sets of $(X_0; X_f)$.

The performance of the algorithm was initially evaluated by comparing it to the algorithm detailed in [3]. As a result, the proposed algorithm decreased the calculation time by 400. Additionally, a parametric study was conducted, varying three major parameters for three different stochastic processes: the Ornstein-Uhlenbeck process, the Geometric Brownian Motion and a narrowband process. This algorithm enjoys fast execution times and offers interesting features such as the possibility to estimate the distribution of FPT. The accuracy of the results has also been studied through the Ornstein-Uhlenbeck process. Moreover, the proposed algorithm results have been tested by comparing FPT maps and histograms from an experimental setup and a numerical model. A good agreement has been shown between the computed FPTs. This comparison enhances the use of this algorithm as it enables the computation of FPT statistical moments directly from both measured and simulated signals, rather than relying solely on simulated data.

Concerning further work, recent studies have shown that FPT maps are more affected by slight modifications in structural systems than more traditional approaches, such as natural frequencies [29]. Hence, this algorithm is seen as a promising technique for improving damage detection strategies [31, 32]. This would be based on comparing FPT maps or PDFs of FPT in healthy (reference) and damaged (current) states.

7. ACKNOWLEDGEMENT

K. Theunissen has been supported by the Belgian Fund for Scientific Research.

Appendix A

Ornstein-Uhlenbeck process. The Ornstein-Uhlenbeck process is defined as

$$\mathrm{d}X_t = \beta(\alpha - X_t)\mathrm{d}t + \sigma \mathrm{d}W_t \tag{7.1}$$

where α is a constant drift, $\beta > 0$, $\sigma > 0$ and W_t is a Wiener process.

The average FPT and STD of FPT values are given in [33]:

$$\text{average FPT}\left(X_f|X_0\right) = t_1\left(X_f|X_0\right) \tag{7.2}$$

$$\text{STD of FPT}\left(X_f|X_0\right) = \sqrt{t_2\left(X_f|X_0\right) - t_1^2\left(X_f|X_0\right)} \tag{7.3}$$

where

$$t_1\left(X_f|X_0\right) = \theta\left\{\sqrt{\pi}\left[\phi_1\left(\frac{X_f}{\sigma\sqrt{\theta}}\right) - \phi_1\left(\frac{X_0}{\sigma\sqrt{\theta}}\right)\right] + \psi_1\left(\frac{X_f}{\sigma\sqrt{\theta}}\right) - \psi_1\left(\frac{X_0}{\sigma\sqrt{\theta}}\right)\right\} \tag{7.4}$$

$$\begin{aligned} t_2\left(X_f|X_0\right) = {} & 2\theta t_1\left(X_f|X_0\right)\left[\sqrt{\pi}\phi_1\left(\frac{X_f}{\sigma\sqrt{\theta}}\right) + \psi_1\left(\frac{X_f}{\sigma\sqrt{\theta}}\right)\right] \\ & + 2\theta^2\left\{\sqrt{\pi}\ln(2)\left[\phi_1\left(\frac{X_f}{\sigma\sqrt{\theta}}\right) - \phi_1\left(\frac{X_0}{\sigma\sqrt{\theta}}\right)\right]\right. \\ & \left. -\sqrt{\pi}\left[\phi_2\left(\frac{X_f}{\sigma\sqrt{\theta}}\right) - \phi_2\left(\frac{X_0}{\sigma\sqrt{\theta}}\right)\right] - \psi_2\left(\frac{X_f}{\sigma\sqrt{\theta}}\right) + \psi_2\left(\frac{X_0}{\sigma\sqrt{\theta}}\right)\right\} \end{aligned} \tag{7.5}$$

$$\phi_1(z) = \mathrm{Erfi}(z) = \int_0^z \exp(t^2)\mathrm{d}t \tag{7.6}$$

$$\phi_2(z) = \sum_{n=0}^{\infty} \frac{z^{2n+3}}{(n+1)!\,(2n+3)} \sum_{k=0}^{n} \frac{1}{2k+1} \tag{7.7}$$

$$\psi_1(z) = \sum_{n=0}^{\infty} \frac{2^n}{(n+1)\,(2n+1)!!} z^{2n+2} \tag{7.8}$$

$$\psi_2(z) = \sum_{n=0}^{\infty} \frac{2^n z^{2n+4}}{(2n+3)!!\,(n+2)} \sum_{k=0}^{n} \frac{1}{k+1} \tag{7.9}$$

with $\theta = \frac{1}{\beta}$.

The theoretical values for the PDF of FPT are given in [23] by solving the backward problem. Firstly Equation (7.1) is rewritten as follows:

$$\mathrm{d}\bar{X}_{\bar{t}} = -\bar{X}_{\bar{t}}\mathrm{d}\bar{t} + \mathrm{d}W_{\bar{t}} \tag{7.10}$$

where $\bar{t} = \beta t$, $\bar{X} = \frac{\sqrt{\beta}}{\sigma}(X - \alpha)$, $\bar{X}_0 = \frac{\sqrt{\beta}}{\sigma}(X_0 - \alpha)$ and $\bar{X}_f = \frac{\sqrt{\beta}}{\sigma}(X_f - \alpha)$.

The backward equation for the cumulative hitting probability $G\left(\bar{t}, \bar{X}_0\right)$ reads

$$G_t\left(\bar{t},\bar{X}_0\right) = -\bar{X}_0 G_{\bar{X}_0}\left(\bar{t},\bar{X}_0\right) + \frac{1}{2} G_{\bar{X}_0\bar{X}_0}\left(\bar{t},\bar{X}_0\right) \tag{7.11}$$

with $G\left(0,\bar{X}_0\right) = 0$ and $G\left(\bar{t},\bar{X}_f\right) = 1$.

Finally, the first hitting density is given by $g\left(\bar{t},\bar{X}_0\right) = G_t\left(\bar{t},\bar{X}_0\right)$.

Appendix B

In Table 2, the experimental frequencies are compared to the numerical frequencies of the updated twin model. The differences between the first six bending mode frequencies are smaller than 1%.

	Experimental frequencies [Hz]	Numerical frequencies [Hz]	Difference [%]
1st Bending mode	18.35	18.26	0.52
2nd Bending mode	39.25	39.16	0.24
3rd Bending mode	64.74	65.25	0.79
4th Bending mode	97.30	97.17	0.13
5th Bending mode	133.73	134.21	0.35
6th Bending mode	171.93	171.72	0.12

Table 2. Comparison of experimental and numerical frequencies of the first six bending modes

In Figure 7.1, the MAC values between the numerical and the experimental eigenmodes of the first six bending modes are plotted. A really good agreement can be observed between the numerical and experimental identified modes because the diagonal values are close to one while the off-diagonal values are near zero.

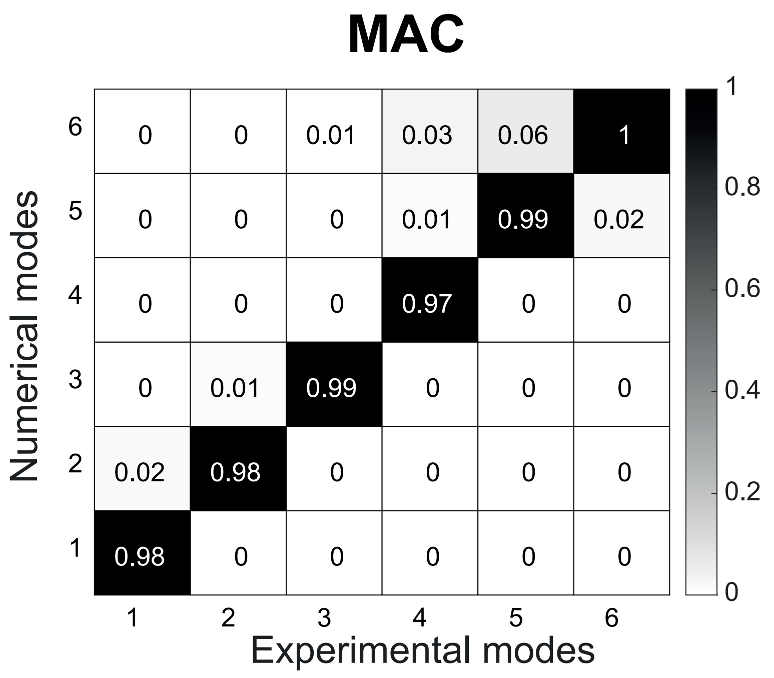

Figure 7.1. MAC values between the numerical and the experimental eigenmodes of the first six bending modes

References

[1] Shoon Kyung Kim. Mean first passage time for a random walker and its application to chemical kinetics. *The Journal of Chemical Physics*, 28(6):1057–1067, 1958.

[2] Zhonghan Hu, Liwen Cheng, and BJ Berne. First passage time distribution in stochastic processes with moving and static absorbing boundaries with application to biological rupture experiments. *The Journal of chemical physics*, 133(3):034105, 2010.

[3] H. Vanvinckenroye, T. Andrianne, and V. Denoël. First passage time as an analysis tool in experimental wind engineering. *Journal of Wind Engineering and Industrial Aerodynamics*, 177:366–375, 6 2018.

[4] Yan Qiao, Wei Xu, Jiaojiao Sun, and Hongxia Zhang. Reliability of electrostatically actuated mems resonators to random mass disturbance. *Mechanical Systems and Signal Processing*, 121:711–724, 2019.

[5] Pierre Patie. *On some first passage time problems motivated by financial applications.* PhD thesis, Universität Zürich, 2004.

[6] P. C. Chu. First passage time analysis for climate prediction. In T. N. Palmer and P. Williams, editors, *Stochastic Physics and Climate Modelling*, chapter 6, pages 157–190. Cambridge University Press, Cambridge, 2009.

[7] William A Gardner. Introduction to random processes with applications to signals and systems. *New York, MacMillan Co., 1986, 447*, 1986.

[8] Sophocles J Orfanidis. *Introduction to signal processing*. Prentice-Hall, Inc., 1995.

[9] S.R. Qin and Y.M. Zhong. A new envelope algorithm of hilbert–huang transform. *Mechanical systems and signal processing*, 20(8):1941–1952, 2006.

[10] M. Feldman. Hilbert transform in vibration analysis. *Mechanical systems and signal processing*, 25(3):735–802, 2011.

[11] H. Vanvinckenroye and V. Denoël. Average first-passage time of a quasi-hamiltonian mathieu oscillator with parametric and forcing excitations. *Journal of Sound and Vibration*, 406:328–345, 2017.

[12] H. Vanvinckenroye and V. Denoël. Second-order moment of the first passage time of a quasi-hamiltonian oscillator with stochastic parametric and forcing excitations. *Journal of Sound and Vibration*, 427:178–187, 2018.

[13] Zeev Schuss. *Theory and applications of stochastic processes: an analytical approach*, volume 170. Springer Science & Business Media, 2009.

[14] HP Langtangen. A general numerical solution method for fokker-planck equations with applications to structural reliability. *Probabilistic Engineering Mechanics*, 6(1):33–48, 1991.

[15] Ioannis A Kougioumtzoglou and Pol D Spanos. Response and first-passage statistics of nonlinear oscillators via a numerical path integral approach. *Journal of Engineering Mechanics*, 139(9):1207–1217, 2013.

[16] Primožič T. Estimating expected first passage times using multilevel monte carlo algorithm, 2011.

[17] M. Matsuichi and T. Endo. Fatigue of metals subjected to varying stress. *Proceedings of the Kyushu Branch of Japan Society of Mechanics Engineering*, 102, 01 1968.

[18] Mahera Musallam and C. Mark Johnson. An efficient implementation of the rainflow counting algorithm for life consumption estimation. *IEEE Transactions on Reliability*, 61(4):978–986, 2012.

[19] Herbert A Sturges. The choice of a class interval. *Journal of the american statistical association*, 21(153):65–66, 1926.

[20] David Freedman and Persi Diaconis. On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476, 1981.

[21] MathWorks. Vectorization. https://www.mathworks.com/help/matlab/matlab_prog-/vectorization.html, 2022-04-01.

[22] Kebig T, Nguyen V-A, Bender M, Schäfer M, and Maas F. Repeatability and precision of different static deflection measurements on a real bridge-part under outdoor conditions in view of damage detection. *Proceedings of the 10th International Conference on Structural Health Monitoring of Intelligent Infrastructure*, SHMII 10, 2021.

[23] Alexander Lipton and Vadim Kaushansky. On the first hitting time density of an ornstein-uhlenbeck process. *arXiv preprint arXiv:1810.02390*, 2018.

[24] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*, volume 23. Springer Berlin, Heidelberg, 1992.

[25] Peter E Kloeden and Eckhard Platen. Higher-order implicit strong numerical schemes for stochastic differential equations. *Journal of statistical physics*, 66(1):283–314, 1992.

[26] Scott D Brown, Roger Ratcliff, and Philip L Smith. Evaluating methods for approximating stochastic differential equations. *Journal of mathematical psychology*, 50(4):402–410, 2006.

[27] Elise Delhez, Hélène Vanvinckenroye, Jean-Claude Golinval, and Vincent Denoël. Numerical and experimental study of first passage time of a steel strip subjected to forced and parametric excitations. In *Proceedings of the 28th International Conference on Noise and Vibration Engineering*, September 2018.

[28] E. Delhez, H. Vanvinckenroye, V. Denoël, and J.-C. Golinval. Experimental and numerical study of the second order moment of the first passage time of a steel strip subjected to forced and parametric excitations. In Nikolaos Dervilis, editor, *Special Topics in Structural Dynamics & Experimental Techniques, Volume 5*, pages 39–42, Cham, 2020. Springer International Publishing.

[29] Kevin Theunissen, Edouard Verstraelen, Jean-Claude Golinval, and Vincent Denoël. Influence of structural damages on first passage time maps. In *ISMA-USD Noise and Vibration Engineering Conference 2022*, 2022.

[30] Kevin Theunissen, Edouard Verstraelen, Jean-Claude Golinval, and Vincent Denoël. Localization of structural damage based on first passage times for a pre-stressed steel strip. In *IMAC-XLI*, 2023.

[31] Dan Xu, Shao-Bo Sui, Wei Zhang, Mengli Xing, Ying Chen, and Rui Kang. Rul prediction of electronic controller based on multiscale characteristic analysis. *Mechanical Systems and Signal Processing*, 113:253–270, 2018.

[32] Tianmei Li, Xiaosheng Si, Hong Pei, and Li Sun. Data-model interactive prognosis for multi-sensor monitored stochastic degrading devices. *Mechanical Systems and Signal Processing*, 167:108526, 2022.

[33] AG Nobile, LM Ricciardi, and L Sacerdote. Exponential trends of ornstein–uhlenbeck first-passage-time densities. *Journal of Applied Probability*, 22(2):360–369, 1985.

[1] Structural & Stochastic Dynamics, University of Liège, Belgium, [2] F.R.S.-FNRS, National Fund for Scientific Research, Belgium