FFT-based Identification of Gilbert-Elliott Data Loss Models

András Palkó, *Student Member, IEEE,* and László Sujbert, *Senior Member, IEEE*

***Abstract*—Nowadays, radio or Internet-based communication is gaining popularity in various fields, e.g., in signal processing. Due to the not reliable real-time communication, some of the data are lost during the transmission. There are several stochastic models allowing the analysis of this phenomenon. The spectral properties of these models result in specific distortion in the signals' spectra. As the spectrum of the signal can easily be calculated via the fast Fourier transform (FFT), FFT-based identification methods of the data loss can be developed. In this paper a new identification method is proposed for the simpler cases of the Gilbert-Elliott model class. The paper summarizes the mathematical description of data loss, and introduces the Gilbert-Elliott model family. The novelty of the paper is the identification method that is based on the autocorrelation function of the Gilbert-Elliott model. The proposed method is compared to classical procedures based on the Baum-Welch algorithm and to novel ones based on global optimization. The theoretical results are supported by extensive simulations.**

***Index Terms*—autocorrelation, Baum-Welch algorithm, data loss, FFT, Gilbert-Elliott model, global optimization, hidden Markov model, identification**

I. Introduction

In traditional measurement and communication systems reliable, high-precision and fast data transmission is used. Recently, due to the technological development and quickening of the networking process (e.g., sensor networks, Internet of Things [1]), cheaper devices and less reliable data transmission protocols are spreading. This can result in the partial damaging of the message during the transmission, which is commonly called as data loss [2], [3].

Data loss can be viewed as a measurement or communication error caused by the not reliable procedures and equipment. It is a many-faced phenomenon: e.g. packets can be lost over computer networks, radio communication can be impaired by interference, or the success of the measurements can depend on external circumstances. These can lead to invalid or missing samples. Additionally, in a distributed system with multiple clock domains, synchronization issues can lead to artificially repeated or skipped samples.

When some data are lost, it is an obvious idea to retry the measurement or transmission, but there are cases when it is impossible. If a signal is measured in real-time, the lost samples cannot be replaced. To achieve reliable transmission, one needs to implement a complex state machine, which may not be economical. Furthermore, reliable transmission is impossible within a finite time interval, thus in real-time systems data loss must be specially handled.

Spectral estimation is a common measurement task, which is usually accomplished via FFT. As exactly calculating the FFT requires a total block of available samples, the adequate handling of data loss is crucial. One could e.g. pre-process the blocks and use the original FFT algorithm [4], estimate the missing samples then use traditional spectral analysis methods [5], or use algorithms developed for irregular sampling [6].

Since data loss is an error phenomenon, usually stochastic models are used for its description. Some examples are the Gilbert-Elliott model [7], the hierarchical hidden Markov model [8] or a two-level model created for UDP channels [9]. It turned out that each data loss model causes a specific distortion of the spectral estimate [4]. Based on this observation, a frequency domain identification method has been presented previously for the two-state Markov and the random independent models [10]. In [11] we described the properties of the Gilbert-Elliott model family and a simple identification method has been introduced.

In this paper, we derive the autocorrelation function of the Gilbert-Elliott model. Using this result, we present an improved identification method for the simpler models of the Gilbert-Elliott model family. This method utilizes the FFT and can offer higher precision than the one described in [11], while keeping the low computational demands.

One of the aims of this paper is to compare the different possibilities of identifying the Gilbert-Elliott model. Among them, we can emphasize the Baum-Welch algorithm [12], which is a well-known expectation-maximization algorithm usable for identifying hidden Markov models (HMMs) [13], [14], [15]. Moreover, the general algorithm can be easily specialized for the models in question. Another approach is to frame the identification as an optimization problem, and use different global optimization techniques [16], [17]. This paper briefly presents the Baum-Welch algorithm, its specializations, the usage of global optimization for identifying an HMM, and compares the proposed method to them.

The paper, which is an extension of [18], is arranged as follows: section II gives a mathematical description of the data loss, and the Gilbert-Elliott model class is presented. Section III deals with the autocorrelation function and the proposed identification method. Section IV briefly presents the investigated alternative algorithms, discusses their implementation for the studied models, and compares them from a

The authors are with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, 1521 Budapest, Hungary (email:{palko,sujbert}@mit.bme.hu)

The project was funded by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013). The research reported in this paper was supported by the BME-Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC). The research reported in this paper was supported by the BME NC TKP2020 grant of NKFIH Hungary.

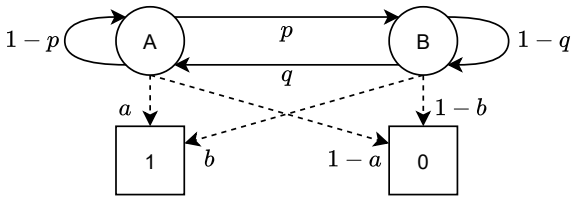

Fig. 1: Gilbert-Elliott data loss model

theoretical point of view. Section V illustrates the procedure with some simulation and measurement examples. The paper concludes in section VI.

II. Preliminaries

A. Basic Definitions

Data loss can be described in discrete time with a so-called K_n availability indicator function [9], [19]:

$$K_n = \begin{cases} 1 & \text{if the sample is available at } n \\ 0 & \text{if the sample is lost at } n \end{cases} \tag{1}$$

With the indicator function we can define the data availability rate μ:

$$\mu = \mathbb{P}\left(K_n = 1\right) \tag{2}$$

where $\mathbb{P}\left(\cdot\right)$ is the probability operator. The R_k autocorrelation function can be defined as follows:

$$R_k = \mathbb{E}\left(K_n K_{n+k}\right) \tag{3}$$

where $\mathbb{E}\left(\cdot\right)$ is the expected value operator.

Data loss can easily be modeled as the product of the K_n data availability function and the x_{0n} discrete-time signal subject of data loss:

$$x_n = x_{0n} K_n \tag{4}$$

The data loss is block-based when the data are grouped into fixed size blocks and these blocks are either fully available or fully lost.

B. The Gilbert-Elliott Data Loss Model Family

1) Gilbert-Elliott Model: The Gilbert-Elliott data loss model is a two-state two-output HMM. In this paper, the states will be marked with A and B. The probabilities of a state transition from state A to state B and vice versa will be marked with p and q, respectively. The probabilities of getting an available sample in the two states will be a and b, respectively. Fig. 1 depicts this data loss model.

2) Simpler Models: The Gilbert-Elliott model is practically a model family. In this family simpler models are to be separated:

- **Gilbert-model [20].** When $a \equiv 1$, the sample in state A is always available.
- **Complementary Gilbert-model.** When $b \equiv 0$, the sample in state B is always lost.
- **Two-state Markov model.** When $a \equiv 1$ and $b \equiv 0$, the availability is a deterministic function of the state.
- **Random independent model.** When $a = b$ or $p+q = 1$, the availability of different samples becomes independent. In the former case, the two states behave uniformly, using the same random independent data loss, while in the latter case, the states themselves are independent of each other.

The Markov model is also a special case of the Gilbert and the complementary Gilbert models. Moreover, the random independent model is a special case of all of the mentioned models.

3) Basic Properties: The $\mathbf{P}$ transition probability matrix of the Gilbert-Elliott model is

$$\mathbf{P} = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix} \tag{5}$$

We can get the π stationary state distribution from the $\pi\mathbf{P} = \pi$ equation:

$$\pi = \begin{bmatrix} \pi_{\mathsf{A}} & \pi_{\mathsf{B}} \end{bmatrix} = \begin{bmatrix} \frac{q}{p+q} & \frac{p}{p+q} \end{bmatrix} \tag{6}$$

The data availability rate of the Gilbert-Elliott model is

$$\mu = \pi_{\mathsf{A}} \mathbb{P}\left(1|\mathsf{A}\right) + \pi_{\mathsf{B}} \mathbb{P}\left(1|\mathsf{B}\right) = \frac{aq + bp}{p+q} \tag{7}$$

III. The Proposed Identification Method

Our earlier research has been focused on the spectral description of data loss. It has turned out, that data loss models have specific power spectral density (PSD) function, which implies that frequency domain identification can be worked out [10], [11]. In [11] a simple identification method has been introduced for the Gilbert-Elliott model family. This simple method is based on the sampling of the PSD in three specific points. Due to the inevitable measurement noise this method often fails in practice. The new method proposed in this paper overcomes this problem by the processing of the autocorrelation function of the model. As the autocorrelation function can be easily calculated by the FFT of the model output sequence, it is an FFT-based identification method.

A. Autocorrelation of the Gilbert-Elliott model

The autocorrelation function of the Gilbert-Elliott model (for $k \geq 0$) is

$$R_k = \begin{cases} \frac{Z}{X} & k = 0 \\ \frac{Z^2}{X^2} + \frac{Y}{X^2}\left(1-X\right)^k & k > 0 \end{cases} \tag{8}$$

where $X = p + q$, $Y = pq\left(a-b\right)^2$ and $Z = aq + bp$. For the derivation of the autocorrelation function see appendix A. Using the formula $\mu = \frac{Z}{X}$, we can rewrite the autocorrelation as

$$R_k = \mu^2 + \frac{Y}{X^2}\left(1-X\right)^k + \left(\mu - \mu^2 - \frac{Y}{X^2}\right)\delta_k \tag{9}$$

where δ_k is the Kronecker delta.

We can determine the origins of the different terms by substituting the conditions for the simpler models into (9). The constant term μ^2 simply comes from the nonzero mean of K_n.

If we take the conditions for the random independent model, the exponential term vanishes. When $a = b$ holds, $Y = 0$ is true, and in the $p + q = 1$ case, $\left(1-X\right)^k$ becomes 0. Furthermore, the exponential term appears in all of the other

models of the Gilbert-Elliott family, so we can conclude that the exponential term comes from the underlying Markov chain.

If we calculate the autocorrelation of the function $K_n - \mu$, we get the same result as (9), only without the constant term μ^2. In this case, R_0 equals to the variance of K_n. The variance has two components, this is because there are multiple sources of randomness in the model: the state transitions and the random outputs in each state. The exponential part comes from the state transitions, and if we express the coefficient of δ_k with the model parameters, we get

$$\mu - \mu^2 - \frac{Y}{X^2} = \pi_{\mathsf{A}} a\,(1-a) + \pi_{\mathsf{B}} b\,(1-b) \tag{10}$$

which means that this part of the variance is the sum of the variances in each state weighted by the stationary distribution.

B. Overview of the Proposed Method

The steps of the proposed method are summarized below:

1) Handle the block-based data loss, decimate K_n if needed.
2) Calculate $\hat{\mu} = \text{mean}\,(K_n)$, $K'_n = K_n - \hat{\mu}$.
3) Obtain the $\hat{R}_k$ autocorrelation estimate.
4) Get the LPC (linear prediction coding) coefficients.
5) Examine the derived coefficients, check for random independent model, get $\hat{X}$ and $\hat{Z}$ estimates.
6) Fit the $C(1-\hat{X})^k$ exponential to $\hat{R}_k$, calculate the $\hat{Y}$ estimate, check for two-state Markov model.
7) Calculate the model parameters.

Fig. 2 illustrates the method. Boxes marked with yellow contain the handling of the block-based data loss, red color shows the steps from the method described in [10], while blue color indicates the extension for the Gilbert and complementary Gilbert models.

C. The Proposed Method in Detail

Step 1. The case of the block-based data loss is to be handled. In this case, we only need to know the availability of the blocks, so the indicator function can be decimated by the block size. For this step, the block size needs to be known, which is usually known in advance, e.g., from the communication protocol. If for some reason the block size is unknown, it can be estimated as the greatest common divisor of the sequence lengths of the available and the lost samples.

Step 2. The value $\hat{\mu}$ is calculated as the mean of K_n, then subtracted from it:

$$K'_n = K_n - \hat{\mu} \tag{11}$$

The value of $\hat{\mu}$ will be used later for getting the $\hat{Z}$ parameter estimate. The subtraction removes the constant μ^2 term from the autocorrelation as its presence can impair the dynamic model fitting in step 4.

Step 3. The $\hat{R}_k$ autocorrelation of K'_n is calculated. It can be performed efficiently by getting the IFFT of the PSD of K'_n. With this way of estimation the autocorrelation, it is necessary for the autocorrelation to decay in the DFT block. Fortunately, as the decay is exponential, this condition is not critical. For the PSD estimation, overlapping blocks can be used. [21] recommends a maximum of 75% overlap. The DFT length should be chosen with care: in too short DFTs the autocorrelation does not decay, while with too long DFTs the number of DFT blocks will be lower, resulting in worse SNR.

Step 4. The $k > 0$ part of the autocorrelation function is approximated by an autoregressive system. In theory, the Gilbert-Elliott model class gives autocorrelation functions with no zeros and at most one pole. It has no pole if it is random independent and one pole otherwise. The model fitting can be done by using an LPC which determines the coefficients of a forward linear predictor by minimizing the prediction error in the least squares sense [22]. As the result of the LPC estimation we get the a_1, a_2, …, a_L coefficients, where L is the order of the model (based on our experience, $L = 10 \ldots 20$ is appropriate).

Step 5. The LPC coefficients are investigated. If the $a_3, a_4, \ldots$ coefficients are not negligible, the data loss does not belong to the Gilbert-Elliott model family. If the coefficient a_2 is negligible as well, the data loss is random independent with data availability probability $\hat{\mu}$. Else, using (7) for $\hat{Z}$ we can get the parameter estimates

$$\hat{X} = 1 + a_2 \qquad \hat{Z} = \hat{\mu}\hat{X} \tag{12}$$

Step 6. The magnitude of the exponential term in the autocorrelation is estimated. This can be done by least squares fitting the $F_k = C(1-\hat{X})^k = CE_k$ function to $\hat{R}_k$ for $k = 1, \ldots, M$, where M is the length of the autocorrelation. We get the value of constant C as:

$$C = \frac{\sum_{k=1}^{M} E_k \hat{R}_k}{\sum_{k=1}^{M} E_k^2} \tag{13}$$

If $\hat{R}_0 \simeq C$, then the coefficient of the δ_k term is negligible in $\hat{R}_k$, which means that two-state Markov model describes the data loss. As C is the estimate of the magnitude of the exponential term of the autocorrelation, we can get the parameter estimate

$$\hat{Y} = C\hat{X}^2 \tag{14}$$

Step 7. The model parameters are calculated from the $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ estimates. Unfortunately, as the Gilbert-Elliott model has four parameters, the model parameters are ambiguous. However, for the simpler models, this is an unambiguous problem. The Gilbert model parameter estimates can be calculated as

$$\hat{p} = \frac{\hat{X}(\hat{X}-\hat{Z})^2}{(\hat{X}-\hat{Z})^2+\hat{Y}} \quad \hat{q} = \frac{\hat{X}\hat{Y}}{(\hat{X}-\hat{Z})^2+\hat{Y}} \quad \hat{b} = \frac{\hat{Z}(\hat{X}-\hat{Z})-\hat{Y}}{\hat{X}(\hat{X}-\hat{Z})} \tag{15}$$

The complementary Gilbert model parameter estimates can be obtained as

$$\hat{p} = \frac{\hat{X}\hat{Y}}{\hat{Z}^2+\hat{Y}} \quad \hat{q} = \frac{\hat{Z}^2\hat{X}}{\hat{Z}^2+\hat{Y}} \quad \hat{a} = \frac{\hat{Z}^2+\hat{Y}}{\hat{Z}\hat{X}} \tag{16}$$

Finally, the two-state Markov model parameter estimates are

$$\hat{p} = \hat{X} - \hat{Z} \qquad \hat{q} = \hat{Z} \tag{17}$$

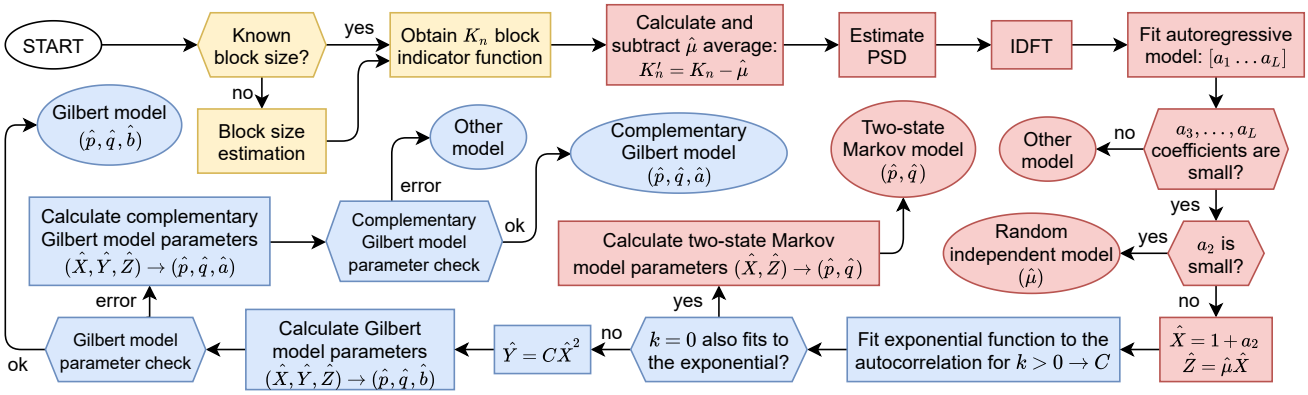

Fig. 2: The proposed data loss model identification method

IV. Alternative Algorithms

The idea of the identification method comes from the observation that data loss distorts the spectra in a specific way. The model for data loss is a kind of state machine, an HMM. In the field of instrumentation and measurement, HMMs are intensively used for the description of various systems that change their states, typically with purpose of classification or detection. A continuous HMM is used in [13] to classify pedestrian activity, while [14] uses multiple HMMs to detect weld lines. Another example is machine health monitoring [15].

The HMMs used in the previous examples are complex: they can have many states, and even continuous outputs. In contrast, the Gilbert-Elliott model class contains some of the simplest HMM structures. This simplicity allows us to use the proposed method.

A. Baum-Welch Algorithm

The training of an HMM is usually performed with the Baum-Welch algorithm [12]. An HMM can be parameterized with $\theta = (\mathbf{P}, \mathbf{O}, \pi)$, where $\mathbf{P}$ is the transition matrix, $\mathbf{O}$ is the output matrix and π is the initial state vector.

The Baum-Welch algorithm finds the

$$\hat{\theta} = \arg\max_{\theta} \mathbb{P}(Y|\theta) \tag{18}$$

maximum likelihood estimate, based only on the observed output sequence Y.

1) Usage for the Gilbert-Elliott Model: Because the Gilbert-Elliott model is one of the simplest HMMs, the general Baum-Welch algorithm is usable for its identification. Two natural ways arise how to modify the general algorithm for the simpler cases of the Gilbert-Elliott model class.

One modification is called the initialized Baum-Welch algorithm. In this case, instead of a randomly chosen output matrix, at least some elements of it are deterministic. Let us denote the initial output probability matrix as

$$\mathbf{O}_0 = \begin{bmatrix} P_{\mathsf{A}1} & P_{\mathsf{A}0} \\ P_{\mathsf{B}1} & P_{\mathsf{B}0} \end{bmatrix} \tag{19}$$

where e.g. $P_{\mathsf{A}0}$ is the probability of getting a lost sample in state A.

According to the definitions of the Gilbert, complementary Gilbert, and two-state Markov models, the following initial output matrices should be used for them, respectively:

$$\mathbf{O}_{0,G} = \begin{bmatrix} 1 & 0 \\ * & * \end{bmatrix} \quad \mathbf{O}_{0,CG} = \begin{bmatrix} * & * \\ 0 & 1 \end{bmatrix} \quad \mathbf{O}_{0,2S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{20}$$

where $*$ marks a randomly chosen positive element (such that the sum of each row is 1).

Another modification could be called the "constrained Baum-Welch algorithm". Now, same constraint matrices as in (20) are introduced. The appropriate one is used for the initial output probability estimate, moreover it is also used as an "estimation mask". This means that during the step of estimating the output probabilities in each iteration, only the ones marked with an $*$ are estimated, the others are used from the constraint matrix.

B. Global Optimization

In essence, (18) describes an optimization problem. The objective function is the likelihood function and the optimization variables are the HMM parameters. Moreover, because the parameters are probabilities, the problem is constrained.

The Baum-Welch algorithm is an expectation-maximization algorithm, which searches for a local optimum. However, we are interested in the global optimum in (18), thus global optimization algorithms can be used for training an HMM. There are several global optimization algorithms, e.g. pattern search [23], genetic algorithm [24] or particle swarm optimization [25].

Traditional optimization algorithms usually use the gradient or higher derivatives to find a local optimum, however this approach does not guarantee a global optimum. Most global optimization algorithms use only the values of the objective function.

Pattern search evaluates the function in a specific pattern around the current point. Based on these values the current point can move to a tested point, the pattern can move or scale.

Genetic algorithms try to model the evolution: there are multiple individuals (test points) in each iteration. The best of them can proceed directly to the next iteration, others can mutate (some variables can randomly change) or two individuals can create a child (with their variables mixed).

There are multiple test points in a particle swarm optimization also, which tries to model the behavior of an insect swarm. In an iteration, the function is evaluated in the test points, then velocities are assigned to the points, finally they move to their new place.

Recently, global optimization algorithms are beginning to be used for HMM training. E.g. [17] uses a particle swarm optimization, while [16] uses a combination of a genetic algorithm and the Baum-Welch method.

1) Usage for the Gilbert-Elliott Model: As the models of the Gilbert-Elliott family can all be described easily using at most four parameters, the optimization problem will contain them directly as optimization variables, instead of the general $\theta = (\mathbf{P}, \mathbf{O}, \pi)$ form. Actually, if the optimization variables are marked with x_i, then the Gilbert-Elliott model parameters can be obviously assigned to them:

$$x_1 = p \quad x_2 = q \quad x_3 = a \quad x_4 = b \tag{21}$$

Similar assignments can be done in the cases of the simpler models. Moreover, the parameters are probabilities, thus they are bounded to the $[0, 1]$ range. The parameters are independent of each other, no further constraints are needed.

C. Theoretical Comparison

The proposed method, the Baum-Welch algorithm and the global optimization approach are compared to each other in Table I.

The table clearly shows that the proposed method was developed exactly for the identification of the simpler models of the Gilbert-Elliott family, while the other two approaches are solutions to more general problems.

The spectral estimation is the most computationally demanding step in the case of the proposed method. From this we get $\mathcal{O}(N \log N_{\text{DFT}})$ complexity, where N is the number of samples and N_{DFT} is the DFT length.

The complexity of the Baum-Welch algorithms comes from the forward-backward algorithm, which is used in the expectation step [12]. This requires $\mathcal{O}(N)$ operations in our case. This calculation is performed in every iteration, yielding the complexity of $\mathcal{O}(IN)$, where I is the maximal number of allowed iterations.

The forward-backward algorithm is also used at the global optimization algorithms for calculating the likelihood function (used as the objective function). Regardless of the actual optimization algorithm, the objective function is evaluated at a set of points in each iteration (upper bounds for the number of points can be expressed for each algorithm) [23], [24], [25]. In our case, the evaluation of the objective function is the dominant term, which makes the complexity $\mathcal{O}(IN)$.

While the computational complexity of all the methods is linear in the number of samples, the proposed method shows significantly lower computational times. Moreover, this linearity guarantees that this observation stays valid at any N.

The optimization cell of the proposed method bears clarification: the least squares fitting steps can be considered as simple optimization problems. Their solution is known in closed form and can be computed at once, this results in lower computational complexity and the ability to find the global optimum.

TABLE I: Comparison of the identification methods

Criterion	Proposed Method	Baum-Welch Algorithm	Global Optimization
Computational complexity	Low	High	
Identifiable models	Simpler models of the Gilbert-Elliott family	General HMM	
Optimization type	Finds global optimum (of simple problems)	Searches for local optimum	Searches for global optimum
Model selection and parameter estimation	Series of simple model selection and parameter estimation steps (see Fig. 2)	Model selection in advance, only parameter estimation	

V. Examples

A. Simulations

1) The Simulation Environment: A simulation environment was created in MATLAB to test the proposed method, and to compare it to the Baum-Welch algorithm and three global optimization algorithms: pattern search, particle swarm optimization and the genetic algorithm. A series of simulations were conducted with the Gilbert model using the model parameters p, q, b and the number of samples N as input variables. They could take the following values:

- $(p, q) \in \{0.01, 0.02, \ldots, 0.1\}^2$
- $b \in \{0.1, 0.2, \ldots, 0.9\}$
- $N \in \{1000, 2000, 5000, 10000\}$

and each parameter configuration has been tested with the proposed method and the Baum-Welch algorithms. A less dense grid was used for the global optimization algorithms.

The samples of the indicator function were generated artificially, starting from stationary state distribution.

In the proposed method, the order of the resulted LPC model was 10. The negligibility of the got a_3, a_4, ...coefficients was decided by comparing their Euclidean norm with 0.1. After calculation of the $\hat{p}$, $\hat{q}$, $\hat{b}$ parameter estimates, they were checked for being in the $[0, 1]$ range. If any of the parameters were outside this range or the a_3, a_4, ...coefficients were not negligible, the parameter estimates were excluded from further processing.

The original and initialized Baum-Welch algorithms were the ones provided by the `hmmtrain` function of MATLAB [26]. It was the base of our implementation of the constrained Baum-Welch algorithm. If the algorithms did not converge, their results were excluded from further processing.

The used global optimization algorithms were provided by the Global Optimization MATLAB Toolbox [27]. The log likelihood was chosen for the objective function instead of directly using the likelihood as in (18). It was calculated using the `hmmdecode` MATLAB function [26]. If the algorithms did not converge, their results were excluded from further processing.

$M = 20$ simulations were carried out for each variable configuration. The mean and standard deviation of the parameter estimation errors were calculated. All the simulations were

TABLE II: Computation times for the different methods by the number of samples. BW: Baum-Welch.

Method	Mean (seconds)			
	$N=1000$	$N=2000$	$N=5000$	$N=10000$
Proposed	0.0030	0.0035	0.0040	0.0049
Original BW	0.0804	0.1726	0.4777	1.0368
Initialized BW	0.0614	0.1212	0.3058	0.5987
Constrained BW	0.0620	0.1202	0.3033	0.6077
Pattern search	0.7596	1.4692	3.5490	7.1673
Particle swarm	2.8282	5.4849	13.393	26.664
Genetic alg.	7.2845	14.236	35.84	72.323

conducted on the same computer to be able to compare the computation times. The mean of the required computation time was also calculated.

The simulations described in [18] were focusing on testing the behavior of the proposed method. In contrast, these simulations depict a situation when there is a measurement record with N samples of the indicator function, which needs to be identified and now we seek the best identification method.

2) Computation Time: The mean of the required computation times for the different methods and number of samples is given in Table II. The actual values of the different times in seconds do not carry much information in themselves, as they depend heavily on the hardware. However, their relations are usable for comparison.

The table can be divided into three sections: the proposed method requires the least time, the Baum-Welch based algorithms significantly more, while the global optimization ones have even higher demands. It is clear that the iterative methods require multiple orders of magnitude more computation time than the proposed one.

The time measurement, especially for the small time values could be significantly influenced by the measurement overhead or the operating system. However, even considering these effects the former comparisons remain valid.

The results clearly show that with the exception of the proposed method, all the other algorithms have computational time proportional to the number of samples. As the algorithm is fast, the overhead was the main component of the elapsed time for the proposed method, at higher number of samples the proportionality is clearly observable.

3) Bias and Standard Deviation: Fig. 3 shows the relative bias and relative standard deviation of the parameter estimates given by the different methods. Each mark corresponds to a parameter configuration. Green crosses mean simulations with $N < 10000$, while the $N = 10000$ cases are drawn with black dots.

The plots of the figure have common bounds for easier comparison. There were results outside these bounds, their amount is shown in Table III. The original Baum-Welch algorithm was not plotted, because it gave so large errors that only a fraction of the results would be visible in the plots.

We can conclude that there were a significant proportion of "outliers" only at $N < 10000$: for the parameters p, q with the particle swarm and the genetic algorithm.

Comparing the figures, at first glance we can see that the different algorithms produced estimates of similar quality. It can be generally observed that while the iterative methods require multiple orders of magnitude more calculations, they do not offer multiple orders of magnitude better results.

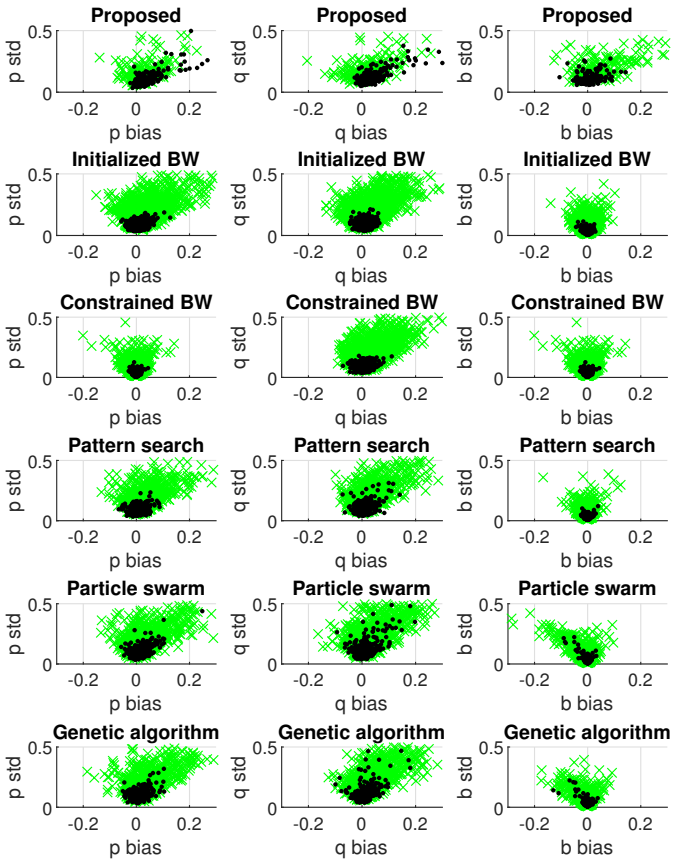

Fig. 3: Relative bias and relative standard deviation of the estimates. BW: Baum-Welch. Green crosses: $N < 10000$, black dots: $N = 10000$.

TABLE III: Amount of data points outside the plot borders in Fig. 3

Method	N	Parameter		
		p	q	b
Proposed	< 10000	1.59%	1.58%	6.35%
	10000	0.56%	1.69%	0.0%
Initialized Baum-Welch	< 10000	2.46%	1.73%	0.17%
	10000	0.0%	0.0%	0.0%
Constrained Baum-Welch	< 10000	0.06%	2.08%	0.06%
	10000	0.0%	0.0%	0.0%
Pattern search	< 10000	4.53%	5.31%	0.0%
	10000	0.0%	0.0%	0.0%
Particle swarm	< 10000	15.43%	18.82%	0.1%
	10000	3.09%	4.32%	0.0%
Genetic algorithm	< 10000	12.90%	16.99%	0.0%
	10000	2.56%	4.15%	0.0%

It must be noted, that when more samples are available for identification, the proposed method is able to give estimates of better quality. As an example, in the $N = 10^6$ case (discounting the outliers) the relative bias is in ± 0.03, while the relative standard deviation is below 0.04. It is reasonably fast to apply the proposed method for a large number of samples (less than 0.2 seconds), while the other methods would require much more time (e.g. the genetic algorithm would require about two hours).

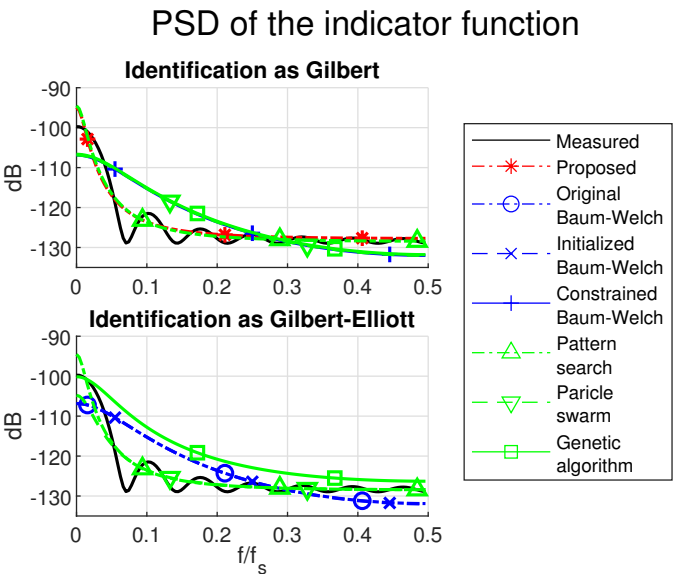

Fig. 4: Measurements – accuracy

B. Measurements

The performance of the proposed method is demonstrated by the measurement data used in [11]. These data were acquired by periodically sending UDP packets between two computers. The indicator function was obtained from the identifiers of the received packets. Several measurements were performed with varying packet size, sending interval, means of connection, parallel network activity, etc. Since now the data loss model type is also unknown, the quality of the estimate could be determined based on the measured PSD and the PSD of the estimated model.

1) Accuracy: Fig. 4 illustrates the accuracy of the proposed method using the data of a measurement from the previously described set. This measurement incorporated a shared Internet connection of a mobile phone, the packets were sent between computers physically about 250 km apart.

The measurement data were identified with all the previously described methods. The proposed method recognized it as a Gilbert model and gave a good fit.

The original and the initialized Baum-Welch algorithms both identified a Gilbert-Elliott model. They gave identical, poor results. The constrained Baum-Welch algorithm identifed a Gilbert model, but its estimate was also poor.

The global optimization algorithms were run for both the Gilbert and the Gilbert-Elliott models. For the Gilbert model, the pattern search gave a good estimate, while the particle swarm and the genetic algorithms yielded poor results. In the case of the Gilbert-Elliott model, the genetic algorithm gave a poor fit, while the other two resulted in acceptable, similar estimates.

2) Other Model: Fig. 5 shows the PSD of the indicator function from another measurement (wired Internet connection, 250 km distance, sending many packets in a short time), and of the identified models.

All the methods except the proposed one forced the identification of a predetermined model, thus they gave an estimate. This is a poor fit disregarding the two bumps in the PSD at higher frequencies. It is interesting that all the estimates coincide.

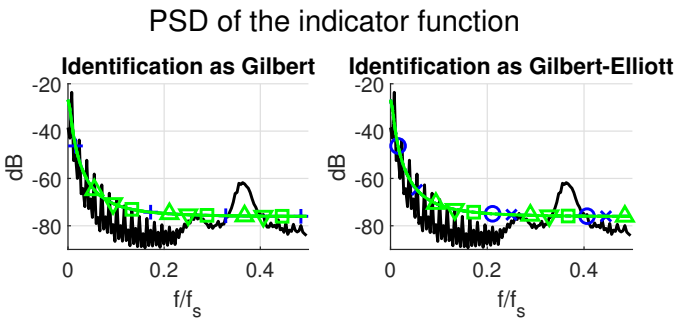

Fig. 5: Measurements – other model. The legend is the same as in Fig. 4.

On the contrary, the proposed method recognized that this is not a Gilbert-Elliott type model: during the 5th step in the identification the higher LPC coefficients were not negligible, thus this measurement cannot be described with a Gilbert-Elliott model. This model type validation is an advantage of the proposed method over the other ones.

VI. Conclusion

In this paper a new identification method for the simpler cases of the Gilbert-Elliott model class was presented. The paper started with the mathematical description of the data loss, then the Gilbert-Elliott model class was introduced. Next, the main contribution of the paper was presented. This is the proposed identification method which estimates the autocorrelation, then fits a dynamic model. The autocorrelation function of the Gilbert-Elliott model is of key importance. Up to the knowledge of the authors it has not been shown before. Examining the coefficients it can be determined which model of the Gilbert-Elliott family is a good hypothesis and the model parameters are calculated. The usage of the Baum-Welch algorithm, its modifications and the global optimization for the task was also discussed. The proposed method has been extensively tested using simulations and measurements, accompanied by comparisons to the other methods. Generally, the proposed FFT-based method offers similar identification results to the Baum-Welch algorithm and the global optimization, along with significantly less computational demand.

Appendix

A. Derivation of the Autocorrelation Function

The derivation needs the expression of the powers of the transition probability matrix. Based on [28] and [29] they can be expressed as

$$\mathbf{P}^k = \frac{\left(\lambda_2\lambda_1^k - \lambda_1\lambda_2^k\right)\mathbf{I} + \left(\lambda_2^k - \lambda_1^k\right)\mathbf{P}}{\lambda_2 - \lambda_1} \tag{22}$$

where $\lambda_1 = 1 - p - q = y$ and $\lambda_2 = 1$ are the eigenvalues of $\mathbf{P}$. With some calculation, we can get

$$\mathbf{P}^k = \frac{1}{p+q}\begin{bmatrix} q + py^k & p - py^k \\ q - qy^k & p + qy^k \end{bmatrix} \tag{23}$$

Now the autocorrelation function can be expressed by definition:

$$R_k = \mathbb{E}\left(K_n K_{n+k}\right) = \mathbb{E}\left(K_0 K_k\right) \tag{24}$$

The latter equation holds because of the time invariance of the Gilbert-Elliott model. Considering that $K_n \in \{0,1\}$ we can write

$$R_k = \sum_{\mathsf{S}_1,\mathsf{S}_2\in\{\mathsf{A},\mathsf{B}\}} \pi_{\mathsf{S}_1}\mathbb{P}\left(1|\mathsf{S}_1\right)\mathbb{P}\left(\mathsf{S}_1 \overset{k}{\to} \mathsf{S}_2\right)\mathbb{P}\left(1|\mathsf{S}_2\right) \tag{25}$$

where $\mathbb{P}\left(1|\mathsf{S}_i\right)$ is the probability of getting a 1 output in state S_i and $\mathbb{P}\left(\mathsf{S}_i \overset{k}{\to} \mathsf{S}_j\right)$ is the probability of transitioning in k steps from state S_i to S_j. Noticing that the former probabilities are the a and the b parameters of the Gilbert-Elliott model and the latter probabilities are the elements of $\mathbf{P}^k$, the autocorrelation function can be expressed as:

$$\begin{aligned} R_k &= \pi_{\mathsf{A}} a \mathbf{P}^k_{\mathsf{AA}} a + \pi_{\mathsf{A}} a \mathbf{P}^k_{\mathsf{AB}} b + \pi_{\mathsf{B}} b \mathbf{P}^k_{\mathsf{BA}} a + \pi_{\mathsf{B}} b \mathbf{P}^k_{\mathsf{BB}} b = \\ &= \frac{(aq+bp)^2 + pq\,(a-b)^2\, y^k}{(p+q)^2} \end{aligned} \tag{26}$$

The above expression is correct for $k > 0$, but incorrect for $k = 0$. For $k = 0$ the availability probability was counted twice, hence the result is incorrect. The correct result can be obtained directly from the definition:

$$R\left(0\right) = \mathbb{E}\left(K_0^2\right) = \mathbb{E}\left(K_0\right) = \mu \tag{27}$$

as $K_n \in \{0,1\}$. Putting together, the autocorrelation function is

$$R_k = \begin{cases} \frac{aq+bp}{p+q} & k = 0 \\ \frac{(aq+bp)^2+pq(a-b)^2y^k}{(p+q)^2} & k > 0 \end{cases} \tag{28}$$

With the substitutions $p+q \mapsto X$, $pq\,(a-b)^2 \mapsto Y$ and $aq+bp \mapsto Z$, the correlation function is

$$R_k = \begin{cases} \frac{Z}{X} & k = 0 \\ \frac{Z^2}{X^2} + \frac{Y}{X^2}\left(1-X\right)^k & k > 0 \end{cases} \tag{29}$$

References

[1] H. Kopetz, *Real-Time Systems,*, 2nd ed. Springer, London, U.K., 2011, p. 378.

[2] L. Kong, M. Xia, X. Liu, G. Chen, Y. Gu, M. Wu, and X. Liu, "Data loss and reconstruction in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2818–2828, 2014.

[3] M. Mathiesen, G. Thonet, and N. Aakvaag, "Wireless ad-hoc networks for industrial automation: Current trends and future prospects," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 89–100, 2005, 16th IFAC World Congress.

[4] L. Sujbert and G. Orosz, "FFT-based spectrum analysis in the case of data loss," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 5, pp. 968–976, 2016.

[5] P. M. T. Broersen, "Five separate bias contributions in time series models for equidistantly resampled irregular data," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1370–1379, 2009.

[6] S. Ferraz-Mello, "Estimation of periods from unequally spaced observations," *The Astronomical Journal*, vol. 86, p. 619, 03 1981.

[7] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," in *The Bell System Technical Journal*, vol. 42, Sept. 1963.

[8] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, Jul 1998.

[9] M. Ellis, D. P. Pezaros, T. Kypraios, and C. Perkins, "A two-level markov model for packet loss in udp/ip-based real-time video applications targeting residential users," *Computer Networks*, vol. 70, pp. 384 – 399, 2014.

[10] L. Sujbert and G. Orosz, "Frequency domain identification of data loss models," in *Acta IMEKO*, vol. 6, no. 4, Dec. 2017, pp. 61–67.

[11] A. Palkó and L. Sujbert, "Time and frequency domain description of Gilbert-Elliott data loss models," in *2019 20th Int. Carpathian Control Conf. (ICCC)*, May 2019, pp. 1–6.

[12] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[13] G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Händel, "Continuous hidden markov model for pedestrian activity classification and gait analysis," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 5, pp. 1073–1083, 2013.

[14] L. Zhang, Q. Ye, W. Yang, and J. Jiao, "Weld line detection and tracking via spatial-temporal cascaded hidden markov models and cross structured light," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 4, pp. 742–753, 2014.

[15] J. Yu, "Health condition monitoring of machines based on hidden markov model and contribution analysis," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 8, pp. 2200–2211, 2012.

[16] S. Celorrio, F. Py, D. Khemani, and K. Rajan, "Optimizing hidden markov models for ocean feature detection." in *Proc. of the 24th Int. Florida Artif. Intell. Res. Soc., FLAIRS - 24*, 01 2011.

[17] L. Chang, Y. Ouzrout, A. Nongaillard, and A. Bouras, "Optimized hidden markov model based on constrained particle swarm optimization," 11 2018.

[18] A. Palkó and L. Sujbert, "FFT-based identification of Gilbert-Elliott data loss models," in *2020 IEEE Int. Instrum. and Meas. Technol. Conf. (I2MTC)*, Dubrovnik, Croatia, May 2020, pp. 1–6.

[19] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 1, pp. 1453 – 1464, 10 2004.

[20] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[21] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, Jan 1978.

[22] L. Jackson, *Digital Filters and Signal Processing 2nd ed.* Kluwer Academic Publishers, 1989.

[23] C. Audet and J. E. Dennis, "Analysis of generalized pattern searches," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 889–903, 2002. [Online]. Available: https://doi.org/10.1137/S1052623400378742

[24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, ser. Artificial Intelligence. Addison-Wesley, 1989.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of ICNN'95 - Int. Conf. on Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[26] Statistics and machine learning toolbox. Accessed: 2020-09-29. [Online]. Available: https://www.mathworks.com/help/stats/index.html

[27] Global optimization toolbox. Accessed: 2020-09-29. [Online]. Available: https://www.mathworks.com/help/gads/index.html

[28] Formula for matrix raised to power n. Accessed: 2018-10-18. [Online]. Available: https://www.freemathhelp.com/forum/threads/55028-Formula-for-matrix-raised-to-power-n?p=227129&viewfull=1#post227129

[29] Using Cayley-Hamilton theorem to calculate matrix powers. Accessed: 2018-10-18. [Online]. Available: https://www.physicsforums.com/threads/using-cayley-hamilton-theorem-to-calculate-matrix-powers.671687/

András Palkó received the M.Sc. degree in electrical engineering from the Budapest University of Technology, Budapest, Hungary, in 2020. He is currently a Ph.D. student at the Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest. His Ph.D. topic is signal processing in the case of data loss.

László Sujbert received the M.Sc. and Ph.D. degrees in electrical engineering from the Budapest University of Technology, Budapest, Hungary, in 1992 and 1998, respectively. Since 1992, he has been with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest. His research interest includes measurement technics, digital signal processing, system identification, active noise control, and embedded systems.