Sparse Linear Prediction for Packet Loss Concealment in Networked Music Performances

Leonardo Severi, Cristina Rottondi
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
{leonardo.severi,cristina.rottondi}@polito.it

***Abstract*—We present a sparse linear prediction algorithm for real-time audio packet loss concealment. The method iteratively selects non-contiguous lags to model periodic signals using an Orthogonal Matching Pursuit (OMP)-based approach for lag selection. Implementation on ARM hardware achieves sub-millisecond processing time for model fitting and prediction times of a few tens of nanoseconds per sample, making the algorithm well-suited for networked music performance applications.**

***Index Terms*—Packet Loss Concealment, Networked Music Performance, Linear Prediction, Autoregressive Models, Orthogonal Matching Pursuit**

I. Introduction

Many applications rely on real-time audio streaming over the Internet. As the purpose varies, latency requirements of such applications vary too. Networked Music Performances (NMP) have particularly strict latency constraints: prior studies show that the acceptable end-to-end delay for a remote musical session is generally at most 30 ms [1]. Beyond such threshold, latency significantly impairs ensemble synchronization.

Several factors contribute to end-to-end latency in NMP applications, primarily packet transit time and buffering delays. Packet jitter and loss, being inherently unpredictable, can cause audio gaps when data is unavailable for playback. Receiver buffers can mitigate gaps due to late packets by absorbing timing variations, at the cost of added queuing delay, thus leading to a trade-off between latency and packet loss rate. Moreover, packets discarded by intermediate routers remain unrecoverable, regardless of the receiver buffer size.

To mitigate this issue, various strategies can be employed — one of the most widely used being Packet Loss Concealment (PLC). PLC techniques aim to generate substitute audio samples to mask missing data in a way that is imperceptible to human listeners. While extensive research exists on PLC, most approaches have been tailored for speech applications such as VoIP or videoconferencing and thus hardly applicable in NMP scenarios.

Leonardo Severi's PhD Programme is funded by the European Union in the framework of the Resiliency and Recovery Plan (RRP), within the NextGenerationEU initiative. The authors acknowledge the support of the MUSMET project funded by the EIC Pathfinder Open scheme of the European Commission (grant agreement n. 101184379). This manuscript reflects only the authors' views and opinions and the European Union or the European Innovation Council cannot be considered responsible for them.

To bridge such gap, we introduce an algorithm specifically designed for PLC in the context of NMP applications, addressing the unique challenges of real-time, high-fidelity audio performance over the Internet. The method consists of a Sparse Linear Prediction algorithm that selects a subset of previous noncontiguous lags, enabling modeling of periodic signals while maintaining computational efficiency suitable for real-time constraints. The algorithm relies on a sparse autoregressive model for signal reconstruction, ensuring smooth transitions through alignment-optimized cross-fading. Thanks to its simplicity, the algorithm is lightweight even when implemented on computationally-limited hardware: preliminary tests prove that it can fit the underlying data in some hundreds of microseconds on a Raspberry Pi 4, and predict each missing sample in some tens of nanoseconds on the same hardware.

The remainder of the paper is organized as follows: sec. II presents a brief literature review on PLC techniques, sec. III describes the proposed algorithm and sec. IV provides some implementation details. Preliminary results are provided in sec. V, whereas sec. VI concludes the manuscript.

II. Related Work

A. Early studies

Early packet loss recovery methods can be broadly categorized into sender-based and receiver-based strategies [2]. Sender-based approaches, such as Forward Error Correction (FEC) and packet retransmission, add data redundancy at the cost of increased bandwidth and latency [3]. Using sender-based approaches, packets are usually fully recovered. Conversely, receiver-based methods operate only on successfully received packets and consist in filling the missing audio portion with artificially-generated samples. PLC techniques fall under the latter category.

An historical approach proposed in the ITU-T G.711 standard [4, Appendix I], named pitch waveform replication, uses a pitch detection algorithm to identify periodic segments and replicates them during loss periods. However, it may produce artifacts when the signal characteristics change abruptly, or in the case of non-periodic signals.

B. Linear Prediction Based Methods

Linear Prediction (LP) has been widely adopted for PLC due to its ability to model the spectral envelope of speech

signals efficiently. Gunduzhan and Momtahan [5] proposed an LP-based PLC algorithm for PCM-coded speech that extracts residual signals through linear prediction analysis and uses periodic replication for excitation generation. Kondo and Nakagawa [6] extended this approach by using bidirectional LP, i.e., predicting lost segments based on both preceding and succeeding packets. This enhancement improved reconstruction quality, particularly for longer loss bursts, achieving mean opinion scores similar to those obtained by [4, Appendix I] for packet loss rates up to 10%.

More recently, Ohidujjaman et al. [7] proposed to replace the traditional autocorrelation method with a modified covariance method for LP coefficient estimation, considering both the forward and backward prediction error in the optimization phase. This approach mitigates the ill-conditioning issues of traditional autocorrelation-based Autoregressive (AR) models.

In the context of NMP, Sacchetto et al. [8] proposed to implement LP leveraging the Burg model, due to its higher numerical stability in comparison to the more widely adopted Yule-Walker model. They demonstrated that AR models can effectively predict missing audio segments with computational requirements suitable for real-time processing, outperforming silence substitution and pattern replication methods commonly used in NMP systems.

The application of AR models to music signals presents unique challenges compared to speech, as music exhibits more complex harmonic structures and longer-term dependencies. The pitch-invariant properties of musical instruments make sparse representations particularly attractive for this domain.

C. Deep Learning Approaches

The advent of deep learning has opened new avenues for PLC. Mohamed et al. [9] provided a comprehensive survey of deep learning methods for speech PLC, highlighting the potential of generative models. Lee et al. [10] proposed a hybrid approach combining generative and predictive models, using a neural vocoder conditioned on features predicted by a separate model.

Deep learning methods typically require significant computational resources and introduce latency that may exceed the strict requirements of NMP applications. The ICASSP 2024 Audio Deep PLC Challenge [11] and the IEEE-IS2 2024 Music PLC Challenge [12] emphasized the importance of computational efficiency alongside reconstruction quality. Research on Deep PLC techniques has led to methods that perform well with a number of parameters in the order of 10^5, as in [13].

III. Proposed method

PLC implementations for NMP applications should process audio frames within a time depending on the size of the dejitter buffer. Such time is typically limited to a few ms, to meet the latency constraints imposed by NMP.

The objective of our proposed algorithm is to leverage the strengths of LP, which inherently supports low-latency operation, while addressing common limitations of AR approaches in long-term predictions. An AR model is a statistical framework that forecasts future values in a sequence based on its past observations. In this sense, the proposed method remains an AR model; however, it introduces two key enhancements:

- enforcing sparsity by using a limited number of parameters, so that predictions rely only on the most informative past values;
- mitigating error accumulation by excluding short-term predictors, which are more prone to compounding prediction errors.

A. Problem Formulation

Given a time-discrete signal $s[t]$, we seek to model it as:

$$s[t] = \sum_{i=1}^{k} \phi_i s[t - d_i] + \epsilon[t] \tag{1}$$

where $\{d_i\}_{i=1}^{k}$ are sparse lags and $\{\phi_i\}_{i=1}^{k}$ are the corresponding coefficients, with $\epsilon[t]$ being an error term. The key idea is the iterative selection of lags based on their correlation with the prediction residual, achieved through heuristics aimed at keeping processing times low while ensuring accurate enough predictions. This method is equivalent to an Orthogonal Matching Pursuit (OMP) where the dictionary of atoms corresponds to the lagged versions of the signal itself.

1) Sparse Representation and OMP: OMP, introduced by Pati et al. [14], has found applications in various signal processing domains. In audio processing, sparse representations have been particularly successful for source separation and compression tasks [15]. OMP evolves from matching pursuit with the key advantage of the re-optimization of all selected coefficients at each iteration, leading to better approximation quality at the cost of increased computational complexity.

Furthermore, under the assumption that the audio signal is weakly stationary, OMP equations can be expressed in terms of autocorrelation values, avoiding the issue of dealing with a potentially large design matrix.

2) Method description: Given the signal $s[t]$, we define:

- $r[d]$ as the autocorrelation value of s at lag d,
- ϕ_d as the coefficient of the model relative to lag d,
- $\tilde{s}[t] = \sum_i \phi_i s[t - d_i]$ as the predicted version of the signal,
- $\epsilon[t] = s[t] - \tilde{s}[t]$ as the error signal,
- $\rho_{\epsilon,s}[d]$ as the cross-correlation between the error signal and the signal itself.

At each iteration, we try to select an informative lagged version of the signal $s[t - d_i]$ by choosing d_i such that:

$$d_i = \arg\max_d \rho_{e,s}[d] \tag{2}$$

We initialize the model by considering $\tilde{s}[t] = 0$, thus $\epsilon[t] = s[t]$, and $\rho_{e,s}[d] = r[d]$. The assumption of stationarity permits to simplify computations. The ordinary least squares (OLS) optimizer for simple linear regression is expressed as per eq. 3:

$$\boldsymbol{\phi} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{3}$$

Algorithm 1 OMP-based Sparse AR

1: **Input:** array of last w samples x, max iterations K, max lag L
2: Initialize sets of lags and coefficients: $\mathcal{D} = \emptyset$, $\phi = \emptyset$
3: Build array of autocorrelations $\boldsymbol{c}$ of length L (up to lag $L-1$)
4: **for** $j = 1$ to K **do**
5: Compute the L-length array $\boldsymbol{\rho}$ according to eq. 5
6: Find $d_j = constrained\_argmax(\boldsymbol{\rho})$
7: **if** stopping criterion met **then**
8: **break**
9: **end if**
10: Update set of lags: $\mathcal{D} = \mathcal{D} \cup \{d_j\}$
11: Construct $\mathbf{R}_{ij} = \mathbf{c}[|d_i - d_j|]$
12: Construct $\mathbf{r}_i = \mathbf{c}[d_i]$
13: Solve $\mathbf{R}\boldsymbol{\phi} = \mathbf{r}$
14: **end for**
15: **Output:** Lags $\mathcal{D}$, coefficients $\boldsymbol{\phi}$

where $\boldsymbol{X}$ is the design matrix, with each row being a sample of the input regressors space, $\boldsymbol{y}$ is the array of scalar responses, and $\boldsymbol{\phi}$ is the array of parameters. There exist a trivial parallelism between eq. 3 and a modified version of the Yule-Walker eq. 4:

$$\boldsymbol{\phi} = \boldsymbol{R}^{-1}\boldsymbol{r} \tag{4}$$

In eq. 4, $\boldsymbol{R}$ is a matrix built such that $\boldsymbol{R}_{i,j} = r(|i-j|)$ and $\boldsymbol{r}$ is a column vector such that $\boldsymbol{r}_i = r(i)$. It can be proven that the two formulations are equivalent for weakly stationary signals. In our case, we define $\boldsymbol{R}$ such that $\boldsymbol{R}_{i,j} = r(|d_i - d_j|)$ and $\boldsymbol{r}_i = r(d_i)$. Under the same assumption, $\rho_{\epsilon,s}$ can be computed without the need of explicitly calculating $\epsilon[t]$, as it can be easily expressed in terms of autocorrelations by means of eq. 5:

$$\rho_{\epsilon,s}[d] = r[d] - \sum_i \phi_i r[|d - d_i|] \tag{5}$$

This finding provides an efficient way to go through the iterations of the OMP algorithm, where each iteration consists of a lag selection according to eq. 2 and eq. 5, and model update according to eq. 4. Algorithm 1 presents a simple implementation of the proposed method.

IV. Method Analysis and Implementation Considerations

A. Model fitting constraints and suggestions

The model inputs are: (i) an array of samples, whose size can be fixed or dynamically determined based on signal conditions; (ii) the maximum number of iterations, which determines the maximum number of parameters in the model; and (iii) the highest lag to consider, which determines the number of autocorrelations to compute.

An effective implementation should incorporate several criteria and hyperparameters to mitigate the possible drawbacks of the model as-it-is:

1) The number of recent samples to be considered for model fitting should be limited to a portion of the signal that exhibits stationary properties. Events such as note onsets may alter the local properties of the signal, and signal portions immediately before or after such events should be excluded.
2) The *constrained_argmax* function (line 6 of Alg. 1) can replace the standard argmax with additional policies, such as:
   - Excluding a given number of initial lags (more exclusions reduce the risk of error compounding in long predictions).
   - Excluding or penalizing lags that are too close to those already in $\mathcal{D}$.
   - Stopping if the correlation associated to the last selected lag is too high or too low.
3) When constructing $\boldsymbol{R}$, regularization can be applied. For instance, Tikhonov regularization can be implemented by solving the system using $\boldsymbol{R}' = \boldsymbol{R} + \lambda\boldsymbol{I}$ instead of $\boldsymbol{R}$ at line 13. The computed coefficients can be further refined using more complex objective functions (e.g., LASSO regularization [16]), which may be effective for larger values of K.
4) The algorithm may terminate before K iterations, based on appropriate stopping criteria 7.

B. Continuity at boundaries

The proposed method can predict multiple missing audio samples without depending on previously predicted ones. However, since the first predicted samples are independent of the last samples in the input window, discontinuities may appear where prediction starts, potentially causing audible glitches that propagate to successive portions of generated audio. These issues can be avoided by either predicting the first samples with a model that guarantees continuity, or using the proposed method to predict the few last present samples. In either case, two versions of the same audio portion are available, enabling cross-fading to mitigate discontinuity. The same approach can be applied at the end of the predicted audio portion, where real samples become available again.

Cross-fading can be used in fixed-sized regions at the predicted section boundaries. However, it is advisable to identify sub-regions that maximize similarity between overlapping portions and apply cross-fading there, following the overlap-add procedure of the WSOLA algorithm [17].

C. Stability

Without any regularization, the output of the model is often unstable. Taking the z-transform of eq. 1, where $S(z) = Z\{s[t]\}$ and $E(z) = Z\{\epsilon[t]\}$ we obtain:

$$S(z) = \sum_{i=1}^{k} \phi_i z^{-d_i} S(z) + E(z) \tag{6}$$

which can be rewritten as:

$$S(z)[1 - \sum_{i=1}^{k} \phi_i z^{-d_i}] = E(z) \tag{7}$$

Therefore:

$$S(z) = E(z)/\Phi(z)$$
$$\text{where } \Phi(z) = 1 - \sum_{i=1}^{k} \phi_i z^{-d_i} \tag{8}$$

This represents an all-pole infinite impulse response filter. If any pole of $1/\Phi(z)$ lies outside the unit circle, the model is unstable. Empirically, low-order models seem less prone to exhibit early instability in their predictions. In any case, the prediction output should be monitored for unstable resonances or exponential growth.

D. Relationship to Waveform Replication Method

The stopping criteria at line 7 can be implemented according to several strategies. When it is based on high correlation between predicted and original signals, strongly periodic signals (e.g., sustained notes) may cause the algorithm to terminate with a single lag. Setting the associated coefficient to 1 maintains constant signal power, making the method equivalent to pitch waveform replication [4].

E. Computational Complexity

The algorithm has been specifically designed to have a very low complexity. Once the autocorrelation values are computed, no more operations are performed on the historical samples. Additionally, the number of selected lags, which is directly linked to the number of iterations of the algorithm, is usually very low. The overall complexity of the model fit phase can be computed as follows:

- **Autocorrelation computation:** its complexity is $\Theta(Nlog(N))$ using a Fast Fourier Transform (FFT) - based approach.
- **Cross-correlation computation:** Each iteration k requires $(k-1)L$ operations to compute the error-signal cross-correlation array, resulting in an overall complexity of $O(k^2L)$. The additional lag selection phase is a linear search on the array ($O(L)$), which makes it negligible w.r.t. the array computation.
- **System solution:** Solving each linear system has a cubic complexity, thus iteration k requires k^3 operations. Note that, contrarily to AR models, the $\boldsymbol{R}$ matrix is symmetric but it is not Toeplitz, so, unfortunately, the Levinson-Durbin recursion is not applicable. Overall, this phase has a $O(k^4)$ complexity.

It follows that the total complexity is $O(k^2L+k^4+Nlog(N))$. With k small and constant and $L \approx N$, this reduces to $O(Nlog(N))$, dominated by the FFT-based autocorrelation computation.

V. Results

The proposed method has been implemented in C++. FFT operations and linear algebra operations are performed respectively via the FFTW [18] and the Eigen3 [19] libraries. Furthermore, the code is executed under FIFO scheduling on a PREEMPT_RT Linux kernel to better reflect its impact on a NMP application. Results are obtained by running the implementation on a Raspberry Pi 4, on the same dataset (29 files with heterogeneous musical content, each of them corrupted by 100 lost portions of 128 samples each, randomly inserted with uniform distribution) of our baseline (i.e., the Burg-based AR model implemented in [20][1]). The prediction quality is assessed in terms of the coefficient of determination R^2 and reported in Fig.1, which highlights that the two methods achieve comparable results.

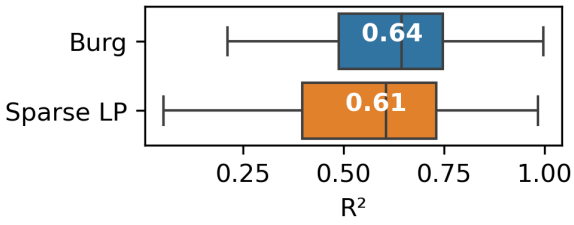

Fig. 1. R^2 comparison between Burg-based AR (model order $p = 128$) and the proposed Sparse LP method

The execution time does not exhibit a strong dependency on model order, which has been limited to 3. In the proposed method, continuity at the left boundary of the gap is guaranteed by a lightweight Burg AR model (with order $p \leq 8$), implemented based on [21] and exploiting the autocorrelation values already obtained with the FFT method. Samples generated by the Burg AR model are cross-faded with the LP-predicted ones. The median time to fit the whole model based on a history of 2048 samples is 256 μs, the minimum 234 μs, the maximum 571 μs and first and third quartiles are respectively 244 μs and 289 μs. This reduces by a factor of three the time required to fit the Burg method with hybrid denominator optimization, used as baseline, which requires $\approx$ 807 μs on the same hardware. Prediction time is between some tens and some hundreds of nanoseconds per sample, i.e., at least one order of magnitude lower than the playout time per sample, making it negligible in the majority of real application scenarios.

VI. Conclusion

We presented a novel packet loss concealment technique based on Sparse Linear Prediction, which allows for the reconstruction of complex periodic signals while maintaining computational efficiency. Our implementation demonstrates a median model fitting time of 256 μs and per-sample prediction time of tens of nanoseconds on a Raspberry Pi 4, thus meeting the stringent latency requirements of networked music performance applications while providing comparable reconstruction quality compared to classic autoregressive models on short predictions. Future work will focus on enhancing the method's adaptability to varying network conditions and signal characteristics, conducting comprehensive perceptual evaluations against existing PLC techniques, and addressing stability issues to enable reliable longer-term predictions.

[1]Dataset and code are available at https://github.com/matteosacchetto/burg-implementation-experiments

REFERENCES

[1] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An Overview on Networked Music Performance Technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.

[2] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, pp. 40–48, Sept. 1998.

[3] B. Wah, Xiao Su, and Dong Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," in *Proceedings International Symposium on Multimedia Software Engineering*, (Taipei, Taiwan), pp. 17–24, IEEE Comput. Soc, 2000.

[4] "A high quality low-complexity algorithm for packet loss concealment with g.711," Sep. 1999. ITU-T Rec. G.711 Appendix I.

[5] E. Gunduzhan and K. Momtahan, "Linear prediction based packet loss concealment algorithm for PCM coded speech," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 778–785, Nov. 2001.

[6] K. Kondo and K. Nakagawa, "A packet loss concealment method using recursive linear prediction.," in *INTERSPEECH*, pp. 2633–2636, 2004.

[7] Ohidujjaman, N. Yasui, Y. Sugiura, T. Shimamura, and H. Makinae, "Packet Loss Compensation for VoIP through Bone-Conducted Speech Using Modified Linear Prediction," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 18, pp. 1781–1790, Nov. 2023.

[8] M. Sacchetto, Y. Huang, A. Bianco, and C. Rottondi, "Using Autoregressive Models for Real-Time Packet Loss Concealment in Networked Music Performance Applications," in *AudioMostly 2022*, (St. Pölten Austria), pp. 203–210, ACM, Sept. 2022.

[9] M. M. Mohamed and B. Schuller, "On deep speech packet loss concealment: A mini-survey," *arXiv preprint arXiv:2005.07794*, 2020.

[10] J.-M. Valin, A. Mustafa, C. Montgomery, T. B. Terriberry, M. Klingbeil, P. Smaragdis, and A. Krishnaswamy, "Real-Time Packet Loss Concealment With Mixed Generative and Predictive Model," 2022.

[11] L. Diener, S. Branets, A. Saabas, and R. Cutler, "The icassp 2024 audio deep packet loss concealment grand challenge," *IEEE Open Journal of Signal Processing*, 2024.

[12] A. I. Mezza and A. Bernardini, "The ieee-is2 2024 music packet loss concealment challenge," 2024.

[13] A. I. Mezza, M. Amerena, A. Bernardini, and A. Sarti, "Hybrid Packet Loss Concealment for Real-Time Networked Music Applications," *IEEE Open Journal of Signal Processing*, vol. 5, pp. 266–273, 2024.

[14] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA, USA), pp. 40–44, IEEE Comput. Soc. Press, 1993.

[15] S. Schulze and E. J. King, "Sparse pursuit and dictionary learning for blind source separation in polyphonic music recordings," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, p. 6, Jan. 2021.

[16] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, pp. 267–288, Jan. 1996.

[17] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in *IEEE International Conference on Acoustics Speech and Signal Processing*, (Minneapolis, MN, USA), IEEE, 1993.

[18] M. Frigo and S. G. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[19] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3," 2010.

[20] M. Sacchetto, C. Rottondi, and A. Bianco, "Implementation and optimization of Burg's method for real-time packet loss concealment in networked music performance applications," *Personal and Ubiquitous Computing*, vol. 28, pp. 727–743, Oct. 2024.

[21] K. Vos, "A fast implementation of burg's method," 2013. OPUS codec.