On supervised LPC estimation training targets for augmented Kalman filter-based speech enhancement

Sujan Kumar Roy a,*, Aaron Nicolson b, Kuldeep K. Paliwal a

a Signal Processing Laboratory, Griffith University, Nathan 4111, QLd, Australia
b Australian e-Health Research Centre, Commonwealth Scientific and Industrial Research Organisation, Herston 4006, QLd, Australia

A R T I C L E I N F O

Keywords:
Speech enhancement
Augmented Kalman filter
Linear prediction coefficients
Training targets
Temporal convolutional network
Multi-head attention network

A B S T R A C T

The performance of speech coding, speech recognition, and speech enhancement systems that rely on the augmented Kalman filter (AKF) largely depend upon the accuracy of clean speech and noise linear prediction coefficient (LPC) estimation. The formulation of clean speech and noise LPC estimation as a supervised learning task has shown considerable promise as of late. Generally, a deep neural network (DNN) learns to map noisy speech features to a training target that can be used for clean speech and noise LPC estimation. Such training targets fall into four categories: Line spectrum frequency (LSF), LPC power spectrum (LPC-PS), power spectrum (PS), and magnitude spectrum (MS) training targets. The choice of training target can have a significant impact on LPC estimation accuracy. Motivated by this, we perform a comprehensive study of the training targets with the aim of determining which is best for LPC estimation. To this end, we evaluate each training target using a temporal convolutional network (TCN) and a multi-head attention-based network. A large training set constructed from a wide variety of conditions, including real-world non-stationary and coloured noise sources over a range of signal-to-noise ratio (SNR) levels, is used for training. Testing on the NOIZEUS corpus demonstrates that the LPC-PS as the training target produces the lowest clean speech LPC spectral distortion (SD) level. We also construct the augmented Kalman filter (AKF) with the estimated speech and noise LPC parameters of each training target. Subjective AB listening tests and seven objective quality and intelligibility evaluation measures (CSIG, CBAK, COVL, PESQ, STOI, SegSNR, and SI-SDR) revealed that the LPC-PS training target produced enhanced speech at the highest quality and intelligibility amongst the training targets.

1. Introduction

Speech processing applications, such as low-bit rate audio coding, speech enhancement, and speech recognition, rely upon the accuracy of linear prediction coefficient (LPC) estimates of clean speech and noise in practice (Vaseghi, 2006, Chapter 8). For example, inaccurate clean speech and noise LPC estimates impact the quality and intelligibility of enhanced speech produced by an augmented Kalman filter (AKF) (Gibson et al., 1991). To address this, deep learning has been employed to accurately estimate LPCs for the Kalman filter (KF) and AKF. This paper focuses on training targets for supervised LPC estimation for AKF-based speech enhancement.

Paliwal and Basu (1987) introduced the KF for speech enhancement. For the KF, each clean speech frame is represented by an autoregressive (AR) process, whose parameters include the clean speech LPCs and prediction error variance. The LPC parameters and additive noise variance are used to construct the KF recursive equations. Given a frame of noisy speech samples, the KF gives a linear MMSE estimate of the clean speech samples using the recursive equations. Paliwal and Basu (1987) demonstrated that the inaccurate estimates of the LPC parameters and noise variance result in poor quality and intelligibility in the enhanced speech produced by the KF. Later on, Gibson et al. (1991) introduced an AKF for speech enhancement in coloured noise conditions. For the AKF, both the clean speech and additive noise are represented by two AR processes. The speech and noise LPC parameters form an augmented matrix which is used to construct the recursive equations of the AKF. In this speech enhancement algorithm (SEA), the AKF processes the noisy speech iteratively (typically three to four iterations) to eliminate the coloured background noise, yielding the enhanced speech. During this, the LPC parameters for the current frame are computed from the corresponding filtered speech frame of the previous iteration (Gibson et al., 1991). Although iteratively estimating the clean speech and noise LPCs for the AKF improves the signal-to-noise ratio (SNR) of noisy speech, the resultant enhanced speech suffers from musical noise and speech distortion.
Multiple training targets have been investigated for deep learning approaches to speech enhancement. Time–frequency (TF) representations were the first training targets investigated for speech enhancement (Wang and Wang, 2013; Williamson et al., 2016). One example is the ideal binary mask (IBM), whose estimate is applied to the noisy speech magnitude spectrum to completely suppress the noise dominant TF components (Wang and Wang, 2013). Xu et al. (2014) applied a feed-forward neural network (FNN) to map the noisy speech log-power spectrum (LPS) to the clean speech LPS. Han et al. (2015) trained a FNN to learn a mapping from the noisy speech magnitude spectrum (MS) to the clean speech MS. Deep learning has also been investigated for statistical filter-based methods, such as MMSE short-time spectral amplitude estimators (Nicolson and Paliwal, 2019), the KF (Yu et al., 2019), and the AKF (Yu et al., 2020). Recently, Nicolson and Paliwal (2021) demonstrated that the choice of training target for clean speech MS estimation has a significant impact on speech enhancement. It was shown that using the a priori SNR as the training target produced the highest quality enhanced speech, whilst using the gain of an MMSE estimator or the ideal amplitude mask (IAM) produced the highest intelligibility speech and was most suited as a front-end for robust ASR.

It is found in literature that the baseline DNN-based MS or LPS estimators (Xu et al., 2014; Wang et al., 2014) and TF masked-based SEAs (Wang and Wang, 2013; Williamson et al., 2016) have been shown significant speech enhancement performance than classical SEAs (Boll, 1979; Kamath and Loizou, 2002; Ephraim and Malah, 1984; Ephraim and Malah, 1985). In general, the baseline DNN methods estimate the MS or LPS of clean speech, then the time-domain clean speech reconstruction is performed with the noisy speech phase — which has a significant impact on the quality of enhanced speech as addressed in Paliwal et al. (2011). In addition, the DNN-based SEAs applied a compression function to the MS or LPS of clean speech (in decibel) to form a mapped training target. However, a compression function may over compress values above 0 dB, which correspond decibel) to form a mapped training target. However, a compression function may over compress values above 0 dB, which correspond to the ideal amplitude mask (IAM) produced the highest intelligibility speech and was most suited as a front-end for robust ASR.

It is found in literature that the baseline DNN-based MS or LPS estimators (Xu et al., 2014; Wang et al., 2014) and TF masked-based SEAs (Wang and Wang, 2013; Williamson et al., 2016) have been shown significant speech enhancement performance than classical SEAs (Boll, 1979; Kamath and Loizou, 2002; Ephraim and Malah, 1984; Ephraim and Malah, 1985). In general, the baseline DNN methods estimate the MS or LPS of clean speech, then the time-domain clean speech reconstruction is performed with the noisy speech phase — which has a significant impact on the quality of enhanced speech as addressed in Paliwal et al. (2011). In addition, the DNN-based SEAs applied a compression function to the MS or LPS of clean speech (in decibel) to form a mapped training target. However, a compression function may over compress values above 0 dB, which correspond to clean speech formants and a very small proportion of the overall distribution as specified in Nicolson and Paliwal (2021). MS training targets were also found to perform the best at lower SNR levels (−5 and 0 dB). It may result due to the distribution of the clean speech MS does not change with the SNR (Nicolson and Paliwal, 2021). On the other hand, the DNN assisted KF and AKF methods (Roy et al., 2020a,b) address speech enhancement in the time-domain — which is not impacted by the noisy speech phase. Recently, Deep learning assisted KF and AKF methods (Roy et al., 2020a,b) has been shown to outperform that of MS spectral amplitude estimator (Xu et al., 2014; Wang et al., 2014). As a result, deep learning-assisted KF and AKF methods have been found interesting to the researchers in the literature. The next section discusses some state-of-the-art deep learning-assisted KF and AKF methods.

1.1. Related work

Deep learning has been investigated for LPC estimation — a key parameter for the KF and AKF-based SEAs (Paliwal and Basu, 1987; Gibson et al., 1991). Pickersgill et al. (2018) proposed a deep neural network (DNN)-based LPC estimation method, termed DNN-LPC. In this method, a FNN learns a mapping from each frame of the noisy speech LPS to the log-LPC power spectra of the clean speech. During inference, the estimated log-LPC PS is converted to the LPC-PS, which is followed by an inverse Fourier transform giving the autocorrelation matrix. Next, the Yule–Walker equations are constructed with the estimated autocorrelation matrix, which is solved by the Levinson–Durbin recursion yielding the LPC parameters of the clean speech (Vaseghi, 2006, Section 8.2.2). However, there were methodology limitations, for one, spectral distortion (SD) levels were not reported below 10 dB. Moreover, only six noise recordings were used for training the FNN, indicating that it would struggle to generalise to unobserved conditions.

Yu et al. (2019) proposed a deep learning assisted KF for speech enhancement (FNN-KF). A three-layered FNN was employed to learn a mapping from the noisy speech line spectrum frequencies (LSFs) to the clean speech LSFs (12th order) (Itakura, 1975). The additive noise variance for the KF is computed from the first noisy speech frame with the assumption that the noise is non-stationary and that there is no speech present in the first frame. However, these assumptions do not account for non-stationary noise sources that have time-varying amplitudes. Moreover, the conditions observed by the FNN during training were derived from only four noise recordings and four SNR levels, indicating that it would struggle to generalise to unobserved conditions.

Yu et al. (2020) used a FNN and an long short-term memory (LSTM) network to estimate the clean speech and noise LPCs for coloured KF-based speech enhancement (FNN-CKFS and LSTM-CKFS). The FNN and LSTM network learn a mapping from the noisy speech LSFs to the clean speech and noise LSFs. During inference, the estimated LSFs are converted to the clean speech and noise LPCs. A maximum likelihood (ML) approach (Srinivasan et al., 2006) is employed to estimate the prediction error variances of the speech and noise AR processes. However, FNN-CKFS and LSTM-CKFS demonstrate poor clean speech and noise LPC estimation accuracy in unobserved noise conditions — leading to the use of multi-band spectral subtraction (MB-SS) (Kamath and Loizou, 2002) for post-processing. This could be due to training the FNN and LSTM network with a small dataset (Yu et al., 2019).

Roy et al. (2020a) utilised the DeepMMSE framework (Zhang et al., 2020) to estimate the parameters of the KF for speech enhancement (denoted as Deep Xi-KF, since Deep MMSE uses Deep Xi (Nicolson and Paliwal, 2019)). DeepMMSE utilises a residual network temporal convolutional network (ResNet-TCN) (He et al., 2016; Bai et al., 2018) to estimate the a priori SNR for the MMSE-based noise power spectral density (PSD) estimator. The noise variance for the KF is computed from the noise PSD estimated by DeepMMSE. Roy et al. (2020b) later used DeepMMSE to estimate the noise LPCs for the AKF (Deep Xi-AKF). Roy and Paliwal (2020a) proposed a causal convolutional encoder–decoder (CCED)-based AKF for speech enhancement (denoted as CCED-AKF). In this method, the CCED maps each frame of the noisy speech magnitude spectrum (MS) to the noise magnitude spectrum, from where the noise PSD is computed. Roy and Paliwal (2020b) proposed a residual network (ResNet) assisted AKF for speech enhancement (denoted as ResNet-AKF). This differed by mapping the time-domain samples of a given noisy speech frame to the corresponding noise frame. For Deep Xi-KF, Deep Xi-AKF, CCED-AKF, and ResNet-AKF, a whitening filter is utilised to estimate the clean speech LPCs. The coefficients for the whitening filter are computed from the estimated noise. Each noisy speech frame is then pre-whitened prior to computing the clean speech LPC parameters. However, Roy et al. (2021a) demonstrated that clean speech LPCs estimated in this manner exhibit a high amount of bias.

In order to reduce the amount of bias caused by using the whitening filter, Roy et al. (2021a) proposed the DeepLPC framework, which jointly estimates the clean speech and noise LPC-PS using a ResNet-TCN. During inference, the clean speech and noise LPCs are computed from the corresponding LPC-PS estimates. The DeepLPC produces clean speech LPC estimates with a lower SD level than the aforementioned methods, resulting in higher quality and intelligibility enhanced speech. Recently, Nicolson and Paliwal (2020) demonstrated that the multi-head attention network (MHANet) is able to outperform the ResNet-TCN in terms of speech enhancement performance, citing that the MHANet is better able to model the long-term dependencies of noisy speech. Motivated by this, Roy et al. (2021b) proposed an extension of the DeepLPC framework by replacing ResNet-TCN with MHANet, called DeepLPC-MHANet, to further improve the clean speech and noise LPC estimates for the AKF. DeepLPC-MHANet demonstrates a lower clean speech LPC estimate SD level than DeepLPC-ResNet-TCN in various noise conditions. In addition, the AKF constructed with the clean speech and noise LPC estimates
of DeepLPC-MHANet (DeepLPC-MHANet-AKF) produces higher quality and intelligible enhanced speech than DeepLPC-ResNet-TCN-AKF (Roy et al., 2021a).

This study aims to perform a comprehensive study comparing the LSF, LPC-PS, power spectrum (PS), and magnitude spectrum (MS) training targets for AKF-based speech enhancement. The motivation of this study is to determine which training target produces the most accurate clean speech and noise LPC estimates, as well as which produces AKF-based enhanced speech with the highest quality and intelligibility. Each training target is evaluated using ResNet-TCN and MHANet, where a large training set consisting of a wide variety of conditions is used for training (Roy et al., 2021b). The used test set is the NOIZEUS dataset, which consists of real-world non-stationary and coloured noise conditions over a wide range of SNR levels. We compare the SD level of the clean speech LPC estimates for each training target. We also evaluate the AKF-based speech enhancement performance of each training target using subjective AB listening tests and seven different objective quality and intelligibility measures (CSIG, CBAK, COVL, PESQ, STOI, SegSNR, and SI-SDR).

The structure of this paper is as follows: background knowledge is presented in Section 2, including the signal model and the AKF for speech enhancement. In Section 3, we present the training targets. Following this, Section 4 describes the experimental setup. The experimental results are then presented in Section 5. Finally, Section 6 gives some concluding remarks.

2. Background

2.1. Signal model

The noisy speech \( y(n) \), at discrete-time sample \( n \), is assumed to be given by

\[
y(n) = s(n) + v(n),
\]

where \( s(n) \) is the clean speech and \( v(n) \) is uncorrelated additive coloured noise. A 32 ms rectangular window with 50% overlap is used to convert \( y(n) \) into frames, denoted by \( y(n, l) \):

\[
y(n, l) = s(n, l) + v(n, l),
\]

where \( l \in \{0, 1, \ldots, L - 1\} \) is the frame index, \( L \) is the total number of frames, and \( N \) is the total number of samples within each frame, i.e. \( n \in \{0, 1, \ldots, N - 1\} \).

2.2. AKF for speech enhancement

For simplicity, the frame index is omitted in this Section. Each frame of the clean speech and noise signal in Eq. (2) can be represented with \( p \)th and \( q \)th order AR models, as in Vasgehi (2006, Chapter 8):

\[
\begin{align*}
    s(n) &= -\sum_{i=1}^{p} a_i s(n-i) + u(n), \\
v(n) &= -\sum_{k=1}^{q} b_k v(n-k) + u(n),
\end{align*}
\]

where \( \{a_i; i = 1, 2, \ldots, p\} \) and \( \{b_k; k = 1, 2, \ldots, q\} \) are the LPCs of \( u(n) \) and \( v(n) \) are assumed to be white noise with zero mean and variances \( \sigma_u^2 \) and \( \sigma_v^2 \), respectively.

Eqs. (2)–(4) are used to form the following augmented state-space model (ASSM) of the AKF, as in Gibson et al. (1991):

\[
\begin{align*}
x(n) &= \Phi x(n-1) + r g(n), \\
y(n) &= c^T x(n).
\end{align*}
\]

In the above ASSM,

1. \( x(n) = [x(n) \ldots x(n-p+1) x(n) \ldots x(n-q+1)]^T \) is a \((p+q) \times 1\) state-vector,

2. \( \Phi = \begin{bmatrix} \Phi_0 & 0 \\ 0 & \Phi_s \end{bmatrix} \) is a \((p+q) \times (p+q)\) state-transition matrix with

\[
\Phi_s = \begin{bmatrix} -a_1 & -a_2 & \ldots & -a_{p-1} & -a_p \\ 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}, \quad \Phi_0 = \begin{bmatrix} -b_1 & -b_2 & \ldots & -b_{q-1} & -b_q \\ 1 & 0 & \ldots & 0 & 0 \end{bmatrix},
\]

3. \( r = \begin{bmatrix} r_s \\ 0 \end{bmatrix}, \quad r_s = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T, \quad r_v = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T, \)

4. \( g(n) = \begin{bmatrix} u(n) \\ v(n) \end{bmatrix}, \)

5. \( c = \begin{bmatrix} c_s^T \\ c_v^T \end{bmatrix}, \quad c_s = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T \) and \( c_v = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T \), respectively.

6. \( y(n) \) is the noisy measurement at sample \( n \).

For each frame, the AKF computes an unbiased linear MMSE estimate \( \hat{x}(n) \) at sample \( n \), given \( y(n) \), by using the following recursive equations (Gibson et al., 1991):

\[
\hat{x}(n|n-1) = \Phi \hat{x}(n|n-1),
\]

\[
\Psi(n|n-1) = \Phi \Psi(n|n-1) \Phi^T + Q r r^T,
\]

\[
K(n) = \Psi(n|n-1) c^T (c^T \Psi(n|n-1) c)^{-1},
\]

\[
\hat{x}(n|n) = \hat{x}(n|n-1) + K(n) (y(n) - c^T \hat{x}(n|n-1)),
\]

\[
\Psi(n|n) = (I - K(n) c^T) \Psi(n|n-1),
\]

where \( Q = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \) is the process noise covariance.

For a noisy speech frame, the error covariances \((\Psi(n|n-1) \) and \( \Psi(n|n) \) corresponding to \( \hat{x}(n|n-1) \) and \( \hat{x}(n|n) \) and the Kalman gain \( K(n) \) are continually updated on a sample-by-sample basis, while \((\{a_i\}, \sigma_u^2) \) and \((\{b_k\}, \sigma_v^2) \) remain constant. At sample \( n \), \( \hat{x}(n|n) \) gives the output of the AKF, \( \hat{x}(n|n) \), where \( h = [1 \ 0 \ 0 \ \ldots \ 0]^T \) is a \((p+q) \times 1\) column vector. As demonstrated in AKF-RMBT (George et al., 2018), \( \hat{x}(n|n) \) is given by

\[
\hat{x}(n|n) = [1 - K_0(n)] \hat{x}(n|n-1) + K_0(n) y(n) - \hat{\epsilon}(n|n-1),
\]

where \( K_0(n) \) is the 1st component of \( K(n) \), given by

\[
K_0(n) = \frac{a^2(n) + \sigma_v^2}{a^2(n) + \sigma_u^2 + \beta^2(n) + \sigma_v^2},
\]

where

\[
a^2(n) = c_s^2 \Phi_s, \quad b^2(n) = c_v^2 \Phi_s, \quad c_s = 1, \quad c_v = 0,
\]

and

\[
\beta^2(n) = c_v^2 \Phi_s, \quad a^2(n) = 0, \quad c_v = 1, \quad c_s = 0.
\]

The measurement of \( a posteriori \) error variances of the clean speech and noise augmented dynamic model from the previous sample, \( n-1 \), respectively (George et al., 2018).

Eq. (15) reveals that \( K_0(n) \) has a significant impact on \( \hat{x}(n) \). In practice, the inaccurate estimates of \((\{a_i\}, \sigma_u^2) \) and \((\{b_k\}, \sigma_v^2) \) introduce bias into \( K_0(n) \), which impacts \( \hat{x}(n) \). In this paper, we determine which training target for supervised learning is best for \((\{a_i\}, \sigma_u^2) \) and \((\{b_k\}, \sigma_v^2) \) estimation accuracy. We also investigate a new training target with the aim of outperforming all previous training targets in terms of \((\{a_i\}, \sigma_u^2) \) and \((\{b_k\}, \sigma_v^2) \) estimation accuracy.
3. Training targets for LPC estimation

The supervised LPC estimation framework is shown in Fig. 1. It can be seen that the framework is fed as input the single-sided noisy speech magnitude spectrum, \(|Y| \in \mathbb{R}^M\), where \(M = 257\) and \(|Y| = \{[Y(l,0)], [Y(l,1)], \ldots, [Y(l,M-1)]\}\), i.e., \(|Y| \in \mathbb{R}^{257 \times M}\). \(Y(l,m)\) is computed from the noisy speech in Eq. (1) using the short-time Fourier transform (STFT):

\[
Y(l,m) = S(l,m) + V(l,m),
\]

where \(Y(l,m), S(l,m),\) and \(V(l,m)\) denote the complex-valued STFT coefficients of the speech signal, \(S\), and noise, \(V\), respectively, for time-frame index \(l\) and discrete-frequency bin \(m\). The Hamming window is used for analysis and synthesis. In this framework, a DNN learns a mapping from \(|Y|\) to the clean speech and noise training targets whose concatenated form is denoted as \(\hat{\zeta}_l\).

In this study, the LSFs, LPC-PS, PS, and MS (Sections 3.1–3.4) of the clean speech and noise are used as training targets for the DNN. A compression function is typically applied to a training target to compress its dynamic range to obtain convergence during stochastic gradient descent. During inference, we split \(\hat{\zeta}_l\) into the clean speech and noise estimates of the training targets, apply the inverse mapping of the compression function, which yields the estimates of the uncompressing training targets. From this, the clean speech and noise LPSs \(\{\hat{\alpha}_l\}, \hat{\delta}_w^2\) and \(\{\hat{\beta}_l\}, \hat{\delta}_z^2\) are then computed. Following Roy et al. (2021a), a clean speech and noise LPC order of \(p = 16\) and \(q = 16\) is used, respectively.

3.1. LSF training target

The LSFs of the clean speech and noise (denoted as \(\{\rho\}_l\) and \(\{\eta\}_l\)) are used as training targets for LPC estimation. First, the clean speech and noise LPC parameters, \(\{\alpha_1, \sigma_w^2\} (p = 16)\) and \(\{\beta_1, \sigma_z^2\} (q = 16)\) are computed from \(s(n,l)\) and \(v(n,l)\) using the autocorrelation method as in Vasighi (2006, Chapter 8). Next, \(\{\rho\}_l\) and \(\{\eta\}_l\) are computed from \(\{\alpha\}_l\) and \(\{\beta\}_l\).

Following this, the LPSs can be converted to LSFs, which we briefly describe (McLoughlin, 2008). Each time-domain sample \(s(n,l)\) under the linear prediction analysis model can be generated as the output of a finite impulse response filter, \(A(z)\). Thus, the clean speech LPCs \(\{\alpha\}_l\), computed from \(s(n,l)\) are used to generate \(A(z)\), as in McLoughlin (2008):

\[
A(z) = 1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_pz^{-p}
\]

To compute LSFs, \(A(z)\) is decomposed into both symmetrical and anti-symmetrical parts, represented by the polynomials, \(P(z)\) and \(Q(z)\), as in McLoughlin (2008):

\[
P(z) = A(z) + z^{-p+1}A(z^{-1}),
\]

\[
Q(z) = A(z) - z^{-p+1}A(z^{-1}).
\]

The clean speech LSFs \(\{\rho\}_l\) are expressed as the zeros (or complex roots denoted by \(\{\theta\}_l\)) of \(P(z)\) and \(Q(z)\) in terms of angular frequency. Then \(\{\rho\}_l\) are computed as (McLoughlin, 2008):

\[
\{\rho\}_l = \tan^{-1}\left(\frac{\text{Re}(\theta)}{\text{Im}(\theta)}\right), \quad i = 1, 2, \ldots, p.
\]

where \(\{\rho\}_l\) are expressed in radians (between \([0, \pi]\)). Using Eqs. (20)–(23), the noise LSFs, \(\{\eta\}_l\) are computed from \(\{\beta\}_l\).

To improve the rate of convergence during stochastic gradient descent, the dynamic range of \(\{\rho\}_l\) and \(\{\eta\}_l\) are compressed to the interval \([0, 1]\) as follows: \(\bar{\rho} = (\frac{\rho}{\rho_{\text{max}}}, \frac{\rho_{\text{min}}}{\rho_{\text{max}}}, \ldots, \frac{\rho_l}{\rho_{\text{max}}})\) and \(\bar{\eta} = (\frac{\eta}{\eta_{\text{max}}}, \frac{\eta_{\text{min}}}{\eta_{\text{max}}}, \ldots, \frac{\eta_l}{\eta_{\text{max}}})\).

In Yu et al. (2020), the prediction error variances \(\bar{\delta}_w^2\) and \(\bar{\delta}_z^2\) are estimated using an ML approach (Srinivasan et al., 2006) using the estimated \(\{\bar{\alpha}\}_l\) and \(\{\bar{\beta}\}_l\). In this study, we jointly estimate \(\{\hat{\alpha}_l\}, \hat{\delta}_w^2\) and \(\{\hat{\beta}_l\}, \hat{\delta}_z^2\). Hence, \(\zeta_l\) for the LPC estimation framework in Fig. 1 becomes:

\[
\zeta_l = \{\bar{\rho}, \bar{\eta}, \bar{\delta}_w^2, \bar{\delta}_z^2\},
\]

where \(\bar{\rho}, \bar{\eta}, \bar{\delta}_w^2, \text{ and } \bar{\delta}_z^2\) are split into \(\tilde{\rho}, \tilde{\eta}, \tilde{\delta}_w^2, \text{ and } \tilde{\delta}_z^2\). Then, \(\tilde{\rho}, \tilde{\eta}\) are multiplied by \(\kappa\) (the inverse mapping of the compression function), which yield \(\hat{\rho}\) and \(\hat{\eta}\). Finally, \(\hat{\rho}\) and \(\hat{\eta}\) are converted into \(\{\hat{\alpha}\}_l\) and \(\{\hat{\beta}\}_l\) using the LSF to LPC conversion method, as in McLoughlin (2008).

3.2. LPC-PS training target

The LPC-PS of clean speech and noise, \(P_l(m, l)\) and \(P_l(m, l)\) were used as the training targets in Roy et al. (2021a,b). During training, \(P_l(m, l)\) and \(P_l(m, l)\) are computed as in Vasighi (2006, Chapter 9):

\[
P_l(m, l) = \frac{\sigma_w^2}{1 + \sum_{k=1}^{p} a_k e^{-2\pi k l/M}}.
\]

\[
P_l(m, l) = \frac{\sigma_z^2}{1 + \sum_{k=1}^{q} b_k e^{-2\pi k l/M}}.
\]

where \(m, 0, 1, \ldots, M - 1\) \((M = 257)\).

The dynamic range of \(P_l(m, l)\) and \(P_l(m, l)\) are compressed to the interval \([0, 1]\) through utilising the cumulative distribution function (CDF) of \(P_l(m, l)|_{\text{max}}\) and \(P_l(m, l)|_{\text{min}}\), where \(P_l(m, l)|_{\text{max}} = 10 \log_{10}(P_l(m, l))\) and \(P_l(m, l)|_{\text{min}} = 10 \log_{10}(P_l(m, l))\) (Roy et al., 2021a). It can be seen from Figs. 2 (a) and (c) that \(P_l(m, l)|_{\text{max}}\) and \(P_l(m, l)|_{\text{min}}\) follow a Gaussian distribution. Hence, it is assumed that \(P_l(m, l)|_{\text{max}}\) and \(P_l(m, l)|_{\text{min}}\) are distributed normally with mean, \(\mu_{w}\) and \(\mu_{z}\), and variance \(\sigma_{w}^2\) and \(\sigma_{z}^2\), respectively \((P_l(m, l)|_{\text{max}} \sim N(\mu_{w}, \sigma_{w}^2)\) and \(P_l(m, l)|_{\text{min}} \sim N(\mu_{z}, \sigma_{z}^2)\)).

The statistics of \(P_l(m, l)|_{\text{max}}\) and \(P_l(m, l)|_{\text{min}}\), i.e., \((\mu_{w}, \sigma_{w}^2)\) and \((\mu_{z}, \sigma_{z}^2)\) for each frequency bin \(m\) were found over a sample of the training dataset. The resultant CDFs used to compress the dynamic range of \(P_l(m, l)|_{\text{max}}\) and \(P_l(m, l)|_{\text{min}}\) are shown in Figs. 2(b) and (d), respectively, and are applied as follows (Roy et al., 2021a):

\[
P_l(m, l) = \frac{1}{2} \left[1 + \text{erf} \left(\frac{P_l(m, l)|_{\text{max}} - \mu_{w}}{\sigma_{w}}\sqrt{2}\right)\right],
\]

where \(2500\) randomly selected clean speech recordings were mixed with \(2500\) randomly selected noise recordings from the training set (Section 4.2) with SNR levels: \(-10\) dB to \(+20\) dB in 1 dB increments, giving \(2500\) noisy speech signals. For each frequency bin, \(m\), the sample mean and variances, \(\mu_{w}\) and \(\sigma_{w}^2\) and \(\mu_{z}\) and \(\sigma_{z}^2\) were computed from \(2500\) concatenated clean speech and scaled noise recordings, respectively.
Speech Communication 142 (2022) 49–60
53

During inference, Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding (Equations (26) and (27)), we construct the Yule–Walker equations of (b) over the sample of the training set 1.

\[ \hat{a}_i = \sum \left( \frac{P_{\ell}(l, m)_{\text{dB}} - \mu_i}{\sigma_i \sqrt{2}} \right) \]

The training target for the LPC estimation framework in Fig. 1 is formed by concatenating \( \hat{P}_s(l, m) \) and \( \hat{P}_r(l, m) \):

\[ \xi_l = \{ \hat{P}_s(l, 0), \hat{P}_s(l, 1), \ldots, \hat{P}_s(l, M - 1), \hat{P}_r(l, 0), \hat{P}_r(l, 1), \ldots, \hat{P}_r(l, M - 1) \}. \]

During inference, \( \xi_l \) is first split into \( \hat{P}_s(l, m) \) and \( \hat{P}_r(l, m) \). Following this, the inverse mapping of the CDFs compute the estimated LPC-PS:

\[ \hat{P}_s(l, m) = 10^{(\mu_i - 10 \log_{10}(|\hat{P}_s(l, m)_{\text{dB}}|)) \gamma(\mu_i, \sigma_i \sqrt{2})}, \]

\[ \hat{P}_r(l, m) = 10^{(\mu_i - 10 \log_{10}(|\hat{P}_r(l, m)_{\text{dB}}|)) \gamma(\mu_i, \sigma_i \sqrt{2})}. \]

The IDFT of \( \hat{P}_s(l, m) \) and \( \hat{P}_r(l, m) \) gives an estimate of the autocorrelation matrices, \( \hat{R}_m(\tau) \) and \( \hat{R}_m(\tau) \), respectively. Using Roy et al. (2021a, Equations (26) and (27)), we construct the Yule–Walker equations using \( \hat{R}_m(\tau) \) and \( \hat{R}_m(\tau) \). We solve the Yule–Walker equations using the Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding (\( \hat{a}_i \), \( \hat{a}_i^2 \)) (\( p = 16 \)) and (\( \hat{b}_i \), \( \hat{a}_i^2 \)) (\( q = 16 \)).

3.3. PS training target

The PS of the clean speech and noise (denoted as \( \hat{\lambda}_s(l, m) \) and \( \hat{\lambda}_n(l, m) \)) can also be used as the training targets for supervised LPC estimation. \( \hat{\lambda}_s(l, m) \) and \( \hat{\lambda}_n(l, m) \) are computed directly from the squared magnitude of the single-sided clean speech and noise spectrum, respectively: \( \hat{\lambda}_s(l, m) = |S(l, m)|^2 \) and \( \hat{\lambda}_n(l, m) = |N(l, m)|^2 \). As in Nicolson and Palival (2021), we utilise the CDF of the training targets (in dB), \( \hat{\lambda}(l, m)_{\text{dB}} \) and \( \hat{\lambda}(l, m)_{\text{dB}} \), to compress their dynamic range to the interval \([0, 1]\), where \( \hat{\lambda}(l, m)_{\text{dB}} = 10 \log_{10}(\hat{\lambda}(l, m)) \) and \( \hat{\lambda}(l, m)_{\text{dB}} = 10 \log_{10}(\hat{\lambda}(l, m)) \).

We observe in Figs. 3(a) and (c) that \( \hat{\lambda}(l, 64)_{\text{dB}} \) and \( \hat{\lambda}(l, 64)_{\text{dB}} \) follow a Gaussian distribution. Therefore, we assume that \( \hat{\lambda}(l, m)_{\text{dB}} \) and \( \hat{\lambda}(l, m)_{\text{dB}} \) are also distributed normally, with mean \( \mu_i \) and \( \mu_i \), and variance \( \sigma_i^2 \) and \( \sigma_i^2 \), respectively (\( \hat{\lambda}(l, m)_{\text{dB}} \sim N(\mu_i, \sigma_i^2) \) and \( \hat{\lambda}(l, m)_{\text{dB}} \sim N(\mu_i, \sigma_i^2) \)). The statistics of \( \hat{\lambda}(l, m)_{\text{dB}} \) and \( \hat{\lambda}(l, m)_{\text{dB}} \), i.e., (\( \mu_i, \sigma_i^2 \)) and (\( \mu_i, \sigma_i^2 \)) for each frequency bin \( m \) were found over a sample of the training set 1. The resultant CDFs used to compress the dynamic range of \( \hat{\lambda}(l, m)_{\text{dB}} \) and \( \hat{\lambda}(l, m)_{\text{dB}} \) are shown in Fig. 3(b) and (d), respectively, and are applied as follows:

\[ \hat{\lambda}(l, m) = \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{\hat{\lambda}(l, m)_{\text{dB}} - \mu_i}{\sigma_i \sqrt{2}} \right) \right]. \]
The training target for the LPC estimation framework in Fig. 1 is formed by concatenating \(\hat{x}_i(l, m)\) and \(\hat{x}_i(l, m)\):

\[
\zeta_i = [\hat{x}_i(l, 0), \hat{x}_i(l, 1), \ldots, \hat{x}_i(l, M - 1), \hat{x}_i(l, 0), \\
\hat{x}_i(l, 1), \ldots, \hat{x}_i(l, M - 1)].
\]

During inference, \(\zeta_i\) is first split into \(\hat{x}_i(l, m)\) and \(\hat{x}_i(l, m)\). Following this, the inverse mapping of the CDFs are used to obtain \(\hat{\lambda}_i\) and \(\hat{\lambda}_i\):

\[
\hat{\lambda}_i(l, m) = 10^{\phi_i}(\sqrt{\text{erf}^{-1}(2\hat{\lambda}_i(l, m)-1)+\hat{\mu}_i})/10,
\]

\[
\hat{\lambda}_i(l, m) = 10^{\phi_i}(\sqrt{\text{erf}^{-1}(2\hat{\lambda}_i(l, m)-1)+\hat{\mu}_i})/10.
\]

The Yule–Walker equations are then solved using the Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding (\(\hat{\lambda}_i\), \(\hat{\sigma}_i\)) (\(p = 16\)) and (\(\hat{b}_i\), \(\hat{\delta}_i\)) (\(q = 16\)).

### 3.4. MS training target

The magnitude of the single-sided clean speech and noise spectrum (denoted as \(C_i(l, m)\) and \(C_i(l, m)\)) can also be used as the training targets for supervised LPC estimation (Roy and Paliwal, 2020a). \(C_i(l, m)\) and \(C_i(l, m)\) are computed directly from the magnitude of the clean speech and noise spectral components: \(C_i(l, m) = |S(l, m)|^2\) and \(C_i(l, m) = |V(l, m)|^2\). Min–max normalisation (Han et al., 2011, section 3.5.2) is then employed to compress the dynamic range of \(C_i(l, m)\) and \(C_i(l, m)\), as in Roy and Paliwal (2020a):

\[
\tilde{C}_i(l, m) = \frac{C_i(l, m) - S_{\text{max}}(l)}{S_{\text{max}}(l) - S_{\text{min}}(l)},
\]

\[
\tilde{C}_i(l, m) = \frac{C_i(l, m) - V_{\text{max}}(l)}{V_{\text{max}}(l) - V_{\text{min}}(l)},
\]

where \(S_{\text{max}}(l), S_{\text{min}}(l)\) and \(V_{\text{max}}(l), V_{\text{min}}(l)\) are the minimum and maximum values for each frequency bin \(m\) of the clean speech and noise MS, respectively, over all frames in the aforementioned sample 1. The training target for the LPC estimation framework in Fig. 1 is formed by concatenating \(\tilde{C}_i(l, m)\) and \(\tilde{C}_i(l, m)\):

\[
\zeta_i = [\tilde{C}_i(l, 0), \tilde{C}_i(l, 1), \ldots, \tilde{C}_i(l, M - 1), \tilde{C}_i(l, 0), \\
\tilde{C}_i(l, 1), \ldots, \tilde{C}_i(l, M - 1)].
\]

During inference, \(\zeta_i\) is first split into \(\tilde{C}_i(l, m)\) and \(\tilde{C}_i(l, m)\). Next, the clean speech and noise magnitude spectra are computed from \(\tilde{C}_i(l, m)\) and \(\tilde{C}_i(l, m)\) using inverse min–max normalisation (Han et al., 2011, section 3.5.2):

\[
\hat{C}_i(l, m) = S_{\text{min}}(l) + \tilde{C}_i(l, m)(S_{\text{max}}(l) - S_{\text{min}}(l)),
\]

\[
\hat{C}_i(l, m) = V_{\text{min}}(l) + \tilde{C}_i(l, m)(V_{\text{max}}(l) - V_{\text{min}}(l)).
\]

Taking the square of \(\hat{C}_i(l, m)\) and \(\hat{C}_i(l, m)\) followed by the [IDFT] gives the autocorrelation matrices, \(\hat{R}_{\text{sv}}(\tau)\) and \(\hat{R}_{\text{sw}}(\tau)\). Using Roy et al. (2021a, Equations (26) and(27)), we construct the Yule–Walker equations using \(\hat{R}_{\text{sv}}(\tau)\) and \(\hat{R}_{\text{sw}}(\tau)\). The Yule–Walker equations are then solved using the Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding ((\(\hat{\lambda}_i\), \(\hat{\sigma}_i\)) (\(p = 16\)) and ((\(\hat{b}_i\), \(\hat{\delta}_i\)) (\(q = 16\)).

### 4. Experimental setup

#### 4.1. Deep neural networks

The DNNs used in this study – ResNet-TCN and MHANet – are briefly described below. Each is trained to map \(|Y_i|\) to \(\zeta_i\), where \(\zeta_i\) is given by Eqs. (24), (29), (34), and (39). During inference, each DNN computes \(\hat{\zeta}_i\). The training strategy for each DNN is detailed in Section 4.3.

![Fig. 4](Colour online) ResNet-TCN. The kernel size, output size, and dilation rate for each convolutional unit is denoted as (kernel size, output size, dilation rate).

#### 4.1.1. ResNet-TCN

The ResNet-TCN used for the DeepLPC framework (Roy et al., 2021a) is used in this study to estimate \(\zeta_i\) (Eqs. (24), (29), (34), and (39)) from \(|Y_i|\). The ResNet-TCN is shown in Fig. 4. For each training target (Section 3), the input, \(|Y_i|\) is first passed through FC, a fully-connected layer of size \(d_{model}\), followed by layer normalisation (LN) (Ba et al., 2016) and the rectified linear unit (ReLU) activation function (He et al., 2015). FC is followed by B bottleneck residual blocks, where \(j\in\{1, 2, \ldots, B\}\) is the block index. Each block comprise of three one-dimensional causal convolutional units. Each convolutional unit (CU) is pre-activated by LN (Ba et al., 2016) followed by the ReLU activation function (He et al., 2015). The kernel size, output size, and dilation rate for each convolutional unit is denoted as (kernel size, output size, dilation rate).

The first and third CUs in each block have a kernel size of one, whilst the second convolutional unit has a kernel size of \(k\). The output size of the first and second CU is \(d_j\), while the third one is \(d_{model}\). A dilation rate of one is set for the first and the third CU, which is \(d\) for the second CU. The second CU provides a contextual field over previous time steps. The dilation rate, \(d\) is cycled as the block index \(j\) increases as: \(d = 2^j - 1 \mod \log_2(m_{Dilation})\), where \(m_{Dilation}\) is the mod operation, and \(D\) is the maximum dilation rate. The last residual block is followed by the output layer, \(O\), which is a fully-connected layer with sigmoidal units. The O layer gives an estimate of \(\zeta_i\). For the LPC-PS, PS, and MS training targets, the hyperparameters used in DeepLPC (Roy et al., 2021a) were used: \(d_{model} = 256, d_j = 64, B = 40, k_j = 3,\) and \(D = 16\). With this set of hyperparameters, ResNet-TCN exhibits approximately 2.1 million parameters. For the LSF training target, all the above hyperparameters were used except \(d_j = 34\), giving around 1.91 million parameters.

#### 4.1.2. MHANet

MHANet is an attention-based architecture for speech enhancement that is based on the encoder of the Transformer (Nicolson and Paliwal, 2020). Along with ResNet-TCN, we use it to compare LPC estimation training targets. MHANet is briefly summarised in this Section. The
The simplest form of MHANet is shown in Fig. 5. The processing steps of the MHANet from input to output are described as follows. The first layer in MHANet is used to project the input to a size of \( \mathbf{d}_{\text{model}} \). As in Nicolson and Paliwal (2019), the first layer is formed as: \( \text{max}(0, \text{LN}(|\mathbf{X}|\mathbf{W}^{f} + \mathbf{b}^{f})) \), where LN is frame-wise layer normalisation (Ba et al., 2016), and \( \mathbf{W}^{f} \in \mathbb{R}^{\mathbf{M}\times\mathbf{d}_{\text{model}}} \) and \( \mathbf{b}^{f} \in \mathbb{R}^{\mathbf{d}_{\text{model}}} \) are the learnable weights and biases of the first layer, respectively. Next, the positional encoding from Nicolson and Paliwal (2021, Section A.2) is added after the first layer, where the time-frame index indicates the position. The positional encoding is learned using weight matrix \( \mathbf{W}_{p} \), with a maximum length of 2048 time-frames (i.e. \( \mathbf{W}_{p} \in \mathbb{R}^{2048 \times 256} \)). This is followed by \( B \) cascading blocks.

Each block includes an MHA module, a two-layer feed-forward neural network (FNN), residual connections (He et al., 2016), and frame-shifts (i.e. \( \mathbf{d}_{\text{f}} = 1024, \mathbf{d}_{\text{model}} = 256, H = 8, P_{\text{drop}} = 0.0, \) and \( \Gamma = 40000 \)). With this set of hyperparameters, MHANet exhibits approximately 4.27 million parameters.

### 4.2. Training and validation set

The noisy speech for the training and validation sets are formed from clean speech and noise recordings. For the clean speech recordings, the train-clean-100 set of the LibriSpeech corpus (Panayotov et al., 2015) (28 539), the CSTR VCTK corpus (Veaux et al., 2019) (42 015), and the \( si^{+} \) and \( sx^{+} \) training sets of the TIMIT corpus (Garofolo et al., 1993) (3696) were used, giving a total 74 250 clean speech recordings. To form the validation set, 5% of the clean speech recordings (3713) are randomly selected. Thus, 70 537 of the clean speech recordings are used for the training set. For the noise recordings, the QUT-NOISE dataset (Dean et al., 2010), the Nonspeech dataset (Hu, 2004), the Environmental Background Noise dataset (Saki et al., 2016; Saki and Kehtarnavaz, 2016), the noise set from the MUSAN corpus (Snyder et al., 2015), multiple FreeSound packs (https://freesound.org/),\(^2\) and coloured noise recordings (with an \( \alpha \) value ranging from 2 to 2 in increments of 0.25) were used, giving a total of 16 243 noise recordings. For the validation set, 5% of the noise recordings (813) are randomly selected. The remaining 15 430 noise recordings are used for the training set. All the clean speech and noise recordings are single-channel with a sampling frequency of 16 kHz. To create the noisy speech for the validation set, each of the 3713 clean speech recordings is corrupted by a random section of a randomly selected noise recording (from the set of 813 noise recordings) at a randomly selected SNR level (−10 to +20 dB, in 1 dB increments). The noisy speech for the training set was created using the method described in Section 4.3.

### 4.3. Training strategy

The following training strategy was employed for training ResNet-TCN and MHANet:

- Mean squared error is used as the loss function.
- The Adam optimiser (Kingma and Ba, 2014) is used for stochastic gradient descent optimisation when training ResNet-TCN and MHANet. For ResNet-TCN, the default hyperparameters were used. For MHANet, \( \beta_{1} = 0.9, \beta_{2} = 0.98, \) and \( \epsilon = 10^{-9} \) were used, where the learning rate, \( \alpha_{r} \), depends on the training step (Vaswani et al., 2017):
  \[
  \alpha_{r} = \frac{\gamma^{0.5}}{\text{model}} \cdot \min(\gamma^{0.5}, \Gamma^{-1.5}),
  \]
  where \( \gamma \) is the total number of training steps and \( \Gamma \) is the number of warmup steps.
- Gradient norms that exceed \([-1, 1]\) are clipped (Pascu et al., 2013).
- The number of training examples in an epoch is equal to the number of clean speech recordings used in the training set, i.e., \( 70 537 \).
- A mini-batch size of eight training examples is used. To generate the mini-batch, at first, we select the shortest length of the corrupted utterance among the 8 utterances in the mini-batch. Then we pad the samples of the other utterances in the mini-batch longer than the shortest length utterance.
- The noisy speech signals are generated on the fly as follows: each clean speech recording is randomly selected and corrupted with a random section of a randomly selected noise recording at a randomly selected SNR level (−10 to +20 dB, in 1 dB increments).\(^3\)
- A total of 150 epochs are used to train both ResNet-TCN and MHANet.

### 4.4. Test set

For the objective experiments, 30 phonetically balanced IEEE utterances belonging to six speakers (three male and three female) are taken from the NOIZEUS corpus (Loizou, 2013, Chapter 12). In this experiment, filtering is not performed to the clean speech utterances as in the original NOIZEUS corpus (Loizou, 2013, Chapter 12). The noisy speech for the test set is formed by mixing the clean speech with real-world non-stationary (voice babble, street, restaurant, and shopping mall) and coloured (factory1, factory2, fchannel, and f16) noise recordings selected from (Saki et al., 2016; Saki and Kehtarnavaz, 2016; Pearce and Hirsch, 2000; Varga and Steeneken, 1993) at multiple SNR levels.

\(^2\) FreeSound packs that were used: 147, 199, 247, 379, 622, 643, 1133, 1563, 1840, 2432, 4366, 4439, 15046, 15598, 21558.

\(^3\) For clean speech recordings longer than the noise recordings, we simply append the noise recording until it becomes larger than or equal to the clean speech recording. Then, the noise recording is clipped to the length of the clean speech recording. The same applies when generating the validation set.
varying from −5 dB to +15 dB, in 5 dB increments. This provides 30 examples per condition with 40 total conditions, yielding 1200 examples. All the clean speech and noise recordings are single-channel with a sampling frequency of 16 kHz. Note that the speech and the noise recordings in the test set are different from those used in the training and validation sets.

4.5. SD level evaluation

The frame-wise spectral distortion (SD) (dB) (Gray and Markel, 1976) is used to evaluate the accuracy of LPC estimates obtained using ResNet-TCN and MHANet for the training targets; LSF, LPC-PS, PS, and MS. Specifically, the estimated clean speech LPCCs are evaluated. The SD for the kth frame, denoted by \( D_k \) (in dB) is defined as the root-mean-square-difference between the LPC-PS estimate in dB \( \tilde{P}_k(l,m)_{dB} \), and the oracle case in dB \( P_k(l,m)_{dB} \), as in Gray and Markel (1976):

\[
D_k = \sqrt{\frac{1}{M} \sum_{m=0}^{M-1} [P_k(l,m)_{dB} - \tilde{P}_k(l,m)_{dB}]^2}.
\]

4.6. Speech enhancement methods

We also evaluate the speech enhancement performance of each training target (as described in “3. AKF” below). We also compare the performance of each LPC estimation target to other SEAs in the literature:

1. Noisy: speech corrupted with additive noise.
2. Oracle-AKF: AKF, where \( (a_i, \sigma_i^2) \) and \( (b_i, \sigma_i^2) \) are computed from the clean speech and noise, respectively, where \( p = 16 \), \( q = 16 \), \( w_f = 32 \) ms, \( s_f = 16 \) ms, and a rectangular window is used for framing.
3. AKF constructed from the speech and noise LPC estimates derived from the training targets — LSF, LPC-PS, PS, and MS estimated using ResNet-TCN and MHANet. Thus, there are eight AKF methods, where \( p = 16 \), \( q = 16 \), window length=32 ms, frame shift=16 ms, and a rectangular window is used for framing.
4. LSTM-CKFS (Yu et al., 2020): AKF constructed using \( (a_i, \sigma_i^2) \) and \( (b_i, \sigma_i^2) \) are computed using an LSTM and maximum-likelihood (ML)-based approaches (Srinivasan et al., 2006), followed by post subtraction using the multi-band spectral subtraction (MB-SS) method (Kamath and Loizou, 2002), where \( p = 12 \), \( q = 12 \), \( w_f = 20 \) ms, \( s_f = 0 \) ms, and a rectangular window is used for framing.
5. IAM-IFD (Zheng and Zhang, 2019): Phase-aware DNN for speech enhancement, where \( w_f = 20 \) ms, \( s_f = 5 \) ms, and the Hamming window is used for analysis and synthesis.
6. ResNet20-AKF (Roy and Paliwal, 2020b): AKF-based SEA, where \( (b_i, \sigma_i^2) \) is estimated using the ResNet20-based method and \( (a_i, \sigma_i^2) \) are computed from pre-whitened speech corresponding to each noisy speech frame, where \( p = 16 \), \( q = 16 \), \( w_f = 32 \) ms, \( s_f = 16 \) ms, and a rectangular window is used for framing.

4.7. Objective quality and intelligibility measures

Objective measures are used to evaluate the quality and intelligibility of the enhanced speech with respect to the corresponding clean speech. Table 1 shows the objective quality and intelligibility measures used in this study.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Assesses</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSIG (Hu and Loizou, 2008)</td>
<td>Quality</td>
<td>[1,5]</td>
</tr>
<tr>
<td>CRK (Hu and Loizou, 2008)</td>
<td>Quality</td>
<td>[1,5]</td>
</tr>
<tr>
<td>COVL (Hu and Loizou, 2008)</td>
<td>Quality</td>
<td>[1,5]</td>
</tr>
<tr>
<td>PESQ (Rix et al., 2001)</td>
<td>Quality</td>
<td>[−0.5,4.5]</td>
</tr>
<tr>
<td>STOI (Taal et al., 2011)</td>
<td>Intelligibility</td>
<td>[0,100]%</td>
</tr>
<tr>
<td>SI-SDR (Roux et al., 2019)</td>
<td>Quality</td>
<td>[−∞,∞]</td>
</tr>
<tr>
<td>SegSNR (Mermelstein, 1979)</td>
<td>Quality</td>
<td>[−∞,∞]</td>
</tr>
</tbody>
</table>

4.8. Subjective evaluation for speech enhancement

The subjective evaluation was carried out through a series of blind AB listening tests (Paliwal et al., 2010, Section 3.3.4). To perform the tests, we generated a set of stimuli by corrupting six IEEE utterances sp01, sp05, sp10, sp15, sp26, and sp27 from the NOIZEUS corpus (Loizou, 2013, Chapter 12). The reference transcript for recording sp01 is: “The birch canoe slid on the smooth planks”, and is corrupted with hfcannel at 0 dB. The reference transcript for recording sp05 is: “Wipe the grease off his dirty face”, and is corrupted with f16 at 5 dB. The reference transcript for recording sp10 is: “The sky that morning was clear and bright blue”, and is corrupted with voice babble at 10 dB. The reference transcript for recording sp15 is: “The clothes dried on a thin wooden rack”, and is corrupted with shopping mall at 0 dB. The reference transcript for recording sp26 is: “She has a smart way of wearing clothes”, and is corrupted with street at 0 dB. The reference transcript for recording sp27 is: “Bring your best compass to the third class”, and is corrupted with factory2 at 10 dB. Utterances sp01, sp05, and sp10 were uttered by male and utterances sp15, sp26, and sp27 were uttered by female, respectively. In this test, the enhanced speech produced by eight SEAs as well as the corresponding clean speech and noisy speech signals were played as stimuli pairs to the listeners. Specifically, the test is performed on a total of 540 stimuli pairs (90 for each utterance) played in a random order to each listener, excluding the comparisons between the same method. Each listener’s perceptual preference for the first or second stimuli was recorded. Pairwise scoring was used, with 100% award is given to the preferred method, 0% to the other, and 50% to both if there was no preference. The participants could re-listen to the stimuli if required. Ten English speaking listeners participate in the blind AB listening tests. The average of the preference scores given by the listeners termed as mean subjective preference score (%), is used to subjectively compare the SEAs.

5. Results and discussion

5.1. SD level comparison

The average clean speech LPC estimation SD attained by each of the training targets are shown in Fig. 6. The SD levels for noisy speech indicate the upper bounds of the SD level. It can be seen that the LPC-PS is able to produce the lowest SD levels for both real-world non-stationary as well as coloured noise conditions. PS produced the next lowest SD level. This indicates that the LPC-PS as the training target produces the most accurate clean speech LPC estimates. The low SD levels attained by LPC-PS will be of benefit to the AKF for speech enhancement.

The AB listening tests were conducted with approval from Griffith University’s Human Research Ethics Committee: database protocol number 2018/671.
Table 2
Mean objective scores on NOIZEUS corpus in terms of CSIG, CBAK, COVL, PESQ, STOI, SegSNR, and SI-SDR. Apart from Oracle-AKF, the highest score amongst the competing methods for each measure is given in boldface.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CSIG</th>
<th>CBAK</th>
<th>COVL</th>
<th>PESQ</th>
<th>STOI</th>
<th>SegSNR</th>
<th>SI-SDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy speech</td>
<td>2.33</td>
<td>2.21</td>
<td>2.02</td>
<td>1.36</td>
<td>52.74</td>
<td>1.13</td>
<td>5.87</td>
</tr>
<tr>
<td>LSTM-CKFS</td>
<td>2.86</td>
<td>2.44</td>
<td>2.35</td>
<td>1.87</td>
<td>77.59</td>
<td>7.12</td>
<td>11.89</td>
</tr>
<tr>
<td>ResNet-TCN-LSF-AKF</td>
<td>2.94</td>
<td>2.52</td>
<td>2.43</td>
<td>1.96</td>
<td>77.86</td>
<td>7.21</td>
<td>12.12</td>
</tr>
<tr>
<td>MHANet-LSF-AKF</td>
<td>3.03</td>
<td>2.68</td>
<td>2.51</td>
<td>2.04</td>
<td>78.33</td>
<td>7.28</td>
<td>12.31</td>
</tr>
<tr>
<td>IAM-IFD</td>
<td>3.10</td>
<td>2.74</td>
<td>2.58</td>
<td>2.12</td>
<td>78.46</td>
<td>7.37</td>
<td>12.49</td>
</tr>
<tr>
<td>ResNet20-AKF</td>
<td>3.18</td>
<td>2.81</td>
<td>2.64</td>
<td>2.21</td>
<td>79.78</td>
<td>7.45</td>
<td>12.68</td>
</tr>
<tr>
<td>ResNet-TCN-MS-AKF</td>
<td>3.26</td>
<td>2.87</td>
<td>2.72</td>
<td>2.27</td>
<td>80.67</td>
<td>7.58</td>
<td>12.91</td>
</tr>
<tr>
<td>ResNet-TCN-P5-AKF</td>
<td>3.34</td>
<td>2.93</td>
<td>2.80</td>
<td>2.33</td>
<td>81.47</td>
<td>7.62</td>
<td>13.11</td>
</tr>
<tr>
<td>Deep Xi-ResNet-TCN-MMSE-LSA</td>
<td>3.39</td>
<td>3.04</td>
<td>2.89</td>
<td>2.39</td>
<td>82.66</td>
<td>7.75</td>
<td>13.47</td>
</tr>
<tr>
<td>MHANet-MS-AKF</td>
<td>3.41</td>
<td>3.11</td>
<td>3.02</td>
<td>2.43</td>
<td>83.79</td>
<td>7.96</td>
<td>13.69</td>
</tr>
<tr>
<td>MHANet-PS-AKF</td>
<td>3.47</td>
<td>3.19</td>
<td>3.11</td>
<td>2.52</td>
<td>85.35</td>
<td>8.78</td>
<td>14.14</td>
</tr>
<tr>
<td>Deep Xi-MHANet-MMSE-LSA</td>
<td>3.66</td>
<td>3.36</td>
<td>3.37</td>
<td>2.67</td>
<td>88.66</td>
<td>10.05</td>
<td>15.52</td>
</tr>
<tr>
<td>MHANet-LPC-PS-AKF</td>
<td>3.74</td>
<td>3.47</td>
<td>3.33</td>
<td>2.76</td>
<td>89.12</td>
<td>9.97</td>
<td>16.01</td>
</tr>
<tr>
<td>Oracle-AKF</td>
<td>4.36</td>
<td>4.17</td>
<td>4.03</td>
<td>2.81</td>
<td>95.88</td>
<td>11.04</td>
<td>17.02</td>
</tr>
</tbody>
</table>

5.2. Objective evaluation

In this section, we analyse the speech enhancement performance of the AKF constructed using the clean speech and noise LPC estimates given by each training target. The objective measures are described in Table 1. We also compare each LPC estimation training target to other deep learning approaches to speech enhancement described in Section 4.6.

The mean objective scores on the NOIZEUS corpus are shown in Tables 2. It can be seen that Oracle-AKF produces the highest objective
scores amongst all methods, which is the upper boundary of speech enhancement performance for the AKF. Noisy speech produced the lowest objective scores amongst all methods, indicating the lower boundary of performance. LPC-PS produced the best objective scores amongst the LPC estimation training targets. This is likely due to the fact that LPC-PS as the training target exhibits the least amount of bias. Amongst the SEAs, MHANet-LPC-PS-AKF performed best, attaining the highest CSIG, CBAK, PESQ, STOI, and SI-SDR scores (except for COVL, SegSNR). Deep Xi-MHANet-MMSE-LSA was the next best performing SEA, producing the highest COVL and SegSNR scores.

Figs. 7 and 8 show the PESQ and STOI scores, respectively, of each SEA for multiple conditions. MHANet-LPC-PS-AKF produced the highest PESQ and STOI scores for each condition. Following MHANet-LPC-PS-AKF, Deep Xi-MHANet-MMSE-LSA (Nicolson and Paliwal, 2020) attained the next highest objective scores for each condition. This indicates that LPC-PS as the training target enables the AKF to objectively outperform the MMSE-LSA estimator with the \textit{a priori} SNR as the training target.

5.3. Subjective evaluation by AB listening test

The mean subjective preference score (%) for each SEA is shown in Fig. 9. For this study, we selected the eight SEAs from Section 5.2 that achieved the highest objective quality and intelligibility scores. It can be seen that MHANet-LPC-PS-AKF is preferred (73.43%) by the listeners, apart from the clean speech (100%) and the Oracle-AKF method (82.86%). Deep Xi-MHANet-MMSE-LSA was the next most preferred (70.21%), followed by ResNet-TCN-LPC-PS-AKF (64.70%), MHANet-PS-AKF (60.71%), ResNet20-AKF (58.57%), IAM-IFD (49.50%) and then MHANet-LSF-AKF (44.71%). This indicates that the enhanced speech produced by MHANet-LPC-PS-AKF exhibits the highest perceived quality amongst all tested SEAs.

6. Conclusion

This paper conducts a comprehensive study on LPC estimation training targets, namely LSF, LPC-PS, PS, and MS training targets. Experiments on the NOIZEUS dataset demonstrate that LPC-PS produces the lowest clean speech LPC estimation SD levels amongst all of the training targets. Objective and subjective scores also indicate that the AKF produces the highest quality and intelligibility enhanced speech when constructed with the clean speech and noise LPC estimates derived from the LPC-PS training target. Moreover, we find that pairing the LPC-PS training with the AKF produces higher quality and intelligibility enhanced speech than pairing the \textit{a priori} SNR as the training target with the MMSE-LSA estimator. We also show that MHANet is able to outperform the ResNet-TCN in terms of objective and subjective quality and intelligibility scores, as well as clean speech LPC estimation SD levels.

CRediT authorship contribution statement

Sujan Kumar Roy: Conceptualization, Methodology, Software, Data curation, Writing – review & editing, Investigation, Visualization. Aaron Nicolson: Writing – review & editing. Kuldip K. Paliwal: Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.
Acknowledgements

We gratefully thank the reviewers and editors for their effort in the improvement of this work.

References


