Joint Blind Estimation and Equalization Method Based on Deep Learning for Fast Fading Channels

Antoine Siebert[1,2], Guillaume Ferré[1], Bertrand Le Gal[1], Aurélien Fourny[2]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400, Talence, France
[2]THALES, France
Email: forename.name@{ims-bordeaux.fr[1], thalesgroup.com[2]}

***Abstract*—In this paper, we propose a hybrid architecture based on deep learning to perform joint blind channel estimation and equalization on fast fading channels. The architecture uses a combination of Extended Kalman Filters (EKF) and neural network in order to predict the process noise covariance matrix at each symbol time. First, we present the system model with the state-space representation. Next, the proposed architecture, the training method and the dataset used are detailed. Then, the performance between a classical blind EKF equalizer structure and the proposed solution is compared with simulations. We expose the obtain results and show the contribution of deep learning in this context. Our architecture, the Smart Extended Kalman Filter (SEKF) can track fast time-varying channels and equalize at high SNR better than a classical blind EKF equalizer structure in a context of random walk channel impulse response (CIR) evolution.**

***Index Terms*—Channel Estimation, Neural Network, Extended Kalman filter, Fast fading**

I. Introduction

In a high-speed wireless communication, the propagation channel is subject to fast variations. Due to multipath effects, an equalizer is needed to reduce the inter symbol interference (ISI). Traditionally, training sequence pilots are inserted in packets in order to estimate the channel impulse response (CIR) and to help equalization task. The greater the number of pilot sequences used, the more accurate the channel estimation and equalization. In return, spectral efficiency is reduced because a part of the bandwidth is not used to transmit data. To avoid this spectral efficiency loss, blind and semi-blind equalization methods have been proposed [1], [2]. Because of its optimality in term of minimum mean square error (MMSE), the use of Kalman filters to estimate the CIR have been explored [3] and equalization with Kalman filters has also been studied [4]. Works on extended Kalman filters (EKF) with state vector containing both the CIR and the estimated sent symbols for joint blind estimation and equalization have been proposed [5], [6]. Authors of [7], [8] worked on an approach with an Interacting Multiple Model of EKF (IMM-EKF). All these methods allow to keep tracking the CIR and equalize at the same time. However, it is well known that the noise process covariance matrix $\mathbf{Q}$ of the Kalman filter can really affect the performance. As $\mathbf{Q}$ has no physical sense, most of the time it has to be hand-tuned and fixed according to the use case. The higher the coefficients of the $\mathbf{Q}$ matrix, the larger the variations of the predicted state vector. IMM has been developed to improve the performance by combining the different state vectors produced by the EKFs and allow to use multiple $\mathbf{Q}$ matrices, but this method has several drawbacks. Indeed, an IMM can require a lot of computing power and it can be difficult to keep up with real time constraints. With some channels, a delay of s symbol time needs to be introduced and it multiplies by q^s the number of EKFs, with q the size of the constellation. For an IMM with m states and a delay s, it would require the computation of mq^{s+1} EKFs at each symbol time. The alternative idea of using deep learning to predict Kalman parameters was proposed by the authors of [9], [10], they use a Long Short-Term Memory (LSTM) [11] based neural network to predict covariance matrices of EKF in a context of pose regularization or ballistic. The idea of this paper is to draw inspiration from this work to propose a hybrid architecture composed of EKFs and neural network called the Smart Extended Kalman Filter (SEKF). The neural network is used to predict Kalman noise process covariance matrix parameter in a context of digital communications, a kind of generalization of [12], but in a fast-fading context. The paper is organized as follows: the second section of the paper describes the system model. The proposed hybrid architecture and the training method are introduced in the third section. Finally, the achieved performance is presented and compared with the traditional EKF blind equalizer approach in the last section.

In the following, bold letters designate a matrix or a vector. The i^{th} element of $\mathbf{x}$ at time k is noted $x_k(i)$. The operator $(\cdot)^T$ designates the transpose, and $\mathcal{M}_{p,q}(\mathbb{K})$ is the set of matrices of size $p \times q$ with every elements in $\mathbb{K}$, which refers to $\mathbb{R}$ but everything can be extended to $\mathbb{C}$. If $p = q$, we denote $\mathcal{M}_p(\mathbb{K})$.

II. System model

A. State-space representation

Let d_k be the transmitted symbol at time k, uniformly chosen in the symbols constellation $\{u_1, u_2, \ldots, u_q\}$ (q-PSK, q-QAM, etc.) and $\mathbf{h}_k = [h_{k,0}, h_{k,0}, \ldots, h_{k,L-1}]^T$ the CIR of length L. The baseband observation of the transmitted symbol at time k is given by:

$$z_k = \sum_{l=0}^{L-1} h_{k,l} d_{k-l} + n_k \tag{1}$$

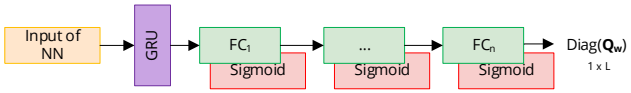

Fig. 1. Neural network architecture used

with n_k an AWGN of variance σ_n^2. By considering

$$\mathbf{D}_k = [d_k, d_{k-1}, \ldots, d_{k-L+1}]^T \tag{2}$$

the L last transmitted symbols, the observation can be written as:

$$z_k = \mathbf{h}_k^T \mathbf{D}_k + n_k \tag{3}$$

As the objective is to jointly estimate the CIR $\mathbf{h}_k$ and the input symbols $\mathbf{D}_k$, let consider the state vector defined by:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{D}_k \\ \mathbf{h}_k \end{pmatrix} \in \mathbb{R}^{2L} \tag{4}$$

In order to simplify the model, we consider independently the I/Q received signals, this is why $\mathbf{D}_k \in \mathbb{R}^L$ and $\mathbf{h}_k \in \mathbb{R}^L$. Given that $\mathbf{D}_{k+1}$ and $\mathbf{h}_{k+1}$ can be written as:

$$\mathbf{D}_{k+1} = \mathbf{F}'\mathbf{D_k} + \mathbf{G}'d_{k+1} \tag{5}$$

$$\mathbf{h}_{k+1} = \mathbf{h}_k + \mathbf{w}_{k+1} \tag{6}$$

with

$$\mathbf{F}' = \begin{pmatrix} \mathbf{0}_{1\times L-1} & 0 \\ \mathbf{I}_{L-1\times L-1} & \mathbf{0}_{L-1\times 1} \end{pmatrix} \in \mathcal{M}_L(\mathbb{R}) \tag{7}$$

$$\mathbf{G}' = \begin{bmatrix} 1 & \mathbf{0}_{1\times L-1} \end{bmatrix}^T \tag{8}$$

and $\mathbf{w}_k$ an AWGN vector of covariance matrix $\mathbf{Q_w}$. The state-space model can now be expressed as:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{v}_{k+1} = \mathbf{F}\mathbf{x}_k + \begin{bmatrix} \mathbf{G}'d_{k+1} \\ \mathbf{w}_{k+1} \end{bmatrix} \\ z_k = h(\mathbf{x}_k) + n_k \end{cases} \tag{9}$$

where

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}' & \mathbf{0}_{L\times L} \\ \mathbf{0}_{L\times L} & \mathbf{I}_L \end{pmatrix} \in \mathcal{M}_{2L}(\mathbb{R}) \tag{10}$$

is the one-step transition matrix,

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}' & \mathbf{0}_{L\times L} \\ \mathbf{0}_{L\times 1} & \mathbf{I}_L \end{pmatrix} \in \mathcal{M}_{2L,L+1}(\mathbb{R}) \tag{11}$$

and $\mathbf{v}_k$ is defined as

$$\mathbf{v}_k = \begin{bmatrix} d_k \\ \mathbf{w}_k \end{bmatrix} \in \mathbb{R}^{L+1} \tag{12}$$

Finally and without loss of generality, considering the constellation of symbols and the CIR are real, $h(\cdot)$ is a non linear function defined as:

$$\begin{array}{rccc} h: & \mathbb{R}^L & \longrightarrow & \mathbb{R} \\ & \mathbf{x} & \longmapsto & \sum_{l=0}^{L-1} x(l)x(L+l) \end{array} \tag{13}$$

An EKF is used in order to jointly estimate the CIR and the transmitted symbol over fast fading channels.

B. Weighted Gaussian Sum approximation

First of all, it is known that if the process noise $\mathbf{G}\mathbf{v}_k$ is not Gaussian, the use of the EKF can lead to poor performance. In our model system, due to d_k which is uniformly distributed in the alphabet of the symbols $\{u_1, \ldots, u_q\}$, $\mathbf{v}_k$ is not Gaussian. This issue can be solved by assuming each $\{d_k\}_{1\leq k\leq q}$ has a conditional power density function, which is a sum of q Gaussian distributions if $d_k \in \mathbb{R}$ and $2q$ if $d_k \in \mathbb{C}$. Each distribution is centered on a symbol of the constellation and has a small variance λ_k^2. Thus in a real symbol context, we have:

$$\forall j \in \{1, \ldots, q\}, p(d_j) = \sum_{i=1}^{q} \frac{1}{q} \mathcal{N}(u_i, \lambda_i^2) \tag{14}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution of mean μ and variance σ^2. The variances $\{\lambda_i^2\}_{i=1,\ldots,q}$ can be chosen at the same value λ^2, small enough so that all the Gaussian distributions are located around the symbols of the constellation. With this approximation, the noise model $\mathbf{G}\mathbf{v}_k$ is Gaussian, zero-mean and with covariance matrix given by:

$$\mathbf{Q} = \begin{pmatrix} \lambda^2 \mathbf{G}'\mathbf{G}'^T & \mathbf{0}_{L\times L} \\ \mathbf{0}_{L\times L} & \mathbf{Q_w} \end{pmatrix} \in \mathcal{M}_{2L}(\mathbb{R}) \tag{15}$$

Authors of [13] chose to consider the conditional power density function as one centered Gaussian with a variance equal to that of the constellation. It then allows to reduce the number of EKFs needed at each symbol time, but in counterpart the spectral efficiency is reduced because they need 40% of pilot symbols per packet. For a larger spectral efficiency, we assume the need of q EKFs at each symbol time. The associated architecture is detailed in the next section.

III. Presentation of the Hyrbid Architecture

A. Hybrid Architecture

In the following, $\hat{\mathbf{x}}_{k|j}^i$ denotes the state vector computed at time k by the EKF with the assumption $d_k = u_i$, and by taking into account the observations up to time j. The SEKF architecture is illustrated in Fig. 2 in a general q symbols constellation context. After receiving the observation at time k, i.e. z_k, the q EKFs prediction equations can be computed, and the vectors $\{\hat{\mathbf{x}}_{k|k-1}^i\}_{i=1,\ldots,q}$ are obtained. Then the neural network inference is done for each EKF in order to predict the $\mathbf{Q_w}$ diagonal matrix. After the update part of the different EKFs the final vectors $\{\hat{\mathbf{x}}_{k|k}^i\}_{i=1,\ldots,q}$ with the covariance error matrices $\{\mathbf{P}_k^i\}_{i=1,\ldots,q}$ and the likelihoods $\{\beta^i\}_{i=1,\ldots,q}$ are obtained. The symbol decision at time k is made by taking $\hat{d}_k = u_j$ with $j = \arg\max_i \beta^i$, and then $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_k$ are set to $\hat{\mathbf{x}}_{k|k}^j$ and $\mathbf{P}_k^j$, respectively. The initialization of the state vector, which is an important point when an EKF is used, is made with a Least Square (LS) estimation of the channel $\hat{\mathbf{h}}_{\text{LS}}$, and with the L last pilot symbols. Explicitly, the first state vector is defined as:

$$\mathbf{x}_{N_p-1|N_p-1} = [\mathbf{D}_{N_p-1}, \hat{\mathbf{h}}_{\text{LS}}] \in \mathbb{R}^{2L} \tag{16}$$

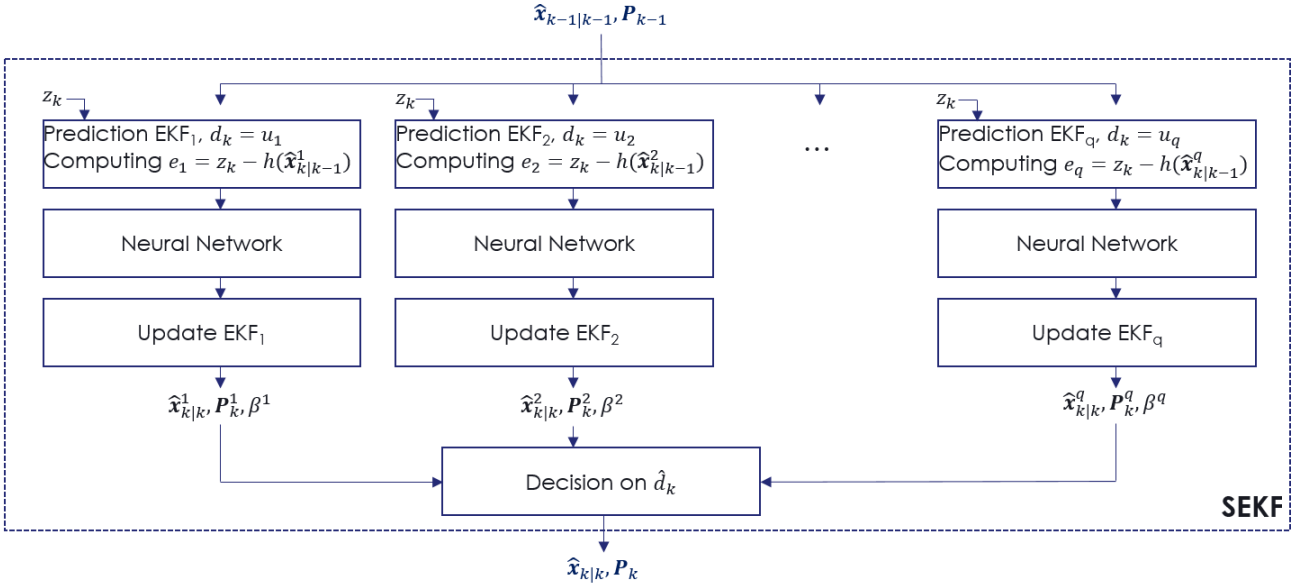

Fig. 2. Illustration of the SEKF processing at time k, using a constellation of q symbols.

B. Neural Network

In order to predict $\mathbf{Q_w}$, lots of neural network configurations were tested. The proposed architecture is described in Fig. 1. It is composed of a Gated Recurrent Unit (GRU) [14] layer with a hidden layer of size h, followed by a set of n fully connected layers activated with sigmoïd function. Note that LSTM based neural network has been tested instead of GRU layer, but performance was the same and for computational complexity reason, GRU layer has been kept. The proposed neural network gives as output the diagonal of the $\mathbf{Q_w}$ matrix at each symbol time. Different configurations were tested with:

1) Different values of GRU hidden layer size $h \in \{1, 4, 8, 16, 32, 64, 128\}$
2) Different number of fully connected layers $n \in \{1, 3, 5\}$

Additionally, different inputs in a BPSK modulation context were experimented:

1) The last observations $[z_k, z_{k-1}, \ldots]$
2) The difference between two consecutive CIR estimations
3) The errors made by supposing the last transmitted symbol e_1, e_2
4) A concatenation of 2) and 3) in a vector of size $L + 2$

In the following, a choice of $n = 3$ fully connected layers of size $32/16/L$ was empirically made, with $L = 3$. Indeed, this configuration process better than others. The impact of the hidden layer size h has been studied, and shows that the loss obtained is not proportional to the size of h. As no gain is obtained by increasing the size of the hidden layer too much, $h = 64$ has been retained. Finally, the 4^{th} input has been selected, which is more explicitly for $L = 3$ at time $k + 1$ the vector $[\hat{x}_{k|k}(0) - \hat{x}_{k-1|k-1}(0), \hat{x}_{k|k}(1) - \hat{x}_{k-1|k-1}(1), \hat{x}_{k|k}(2) - \hat{x}_{k-1|k-1}(2), e_1, e_2]$. The dataset and the training method are presented in the next section.

C. Dataset construction and training method

In order to train the neural network, random walk datasets have been built. Datasets are composed of N packets of size N_s BPSK symbols, each one containing N_p pilot symbols at the beginning of the packet. The channel evolves according to:

$$\mathbf{h}_{k+1} = \mathbf{h}_k + \mathbf{w}_{k+1} \tag{17}$$

with $\mathbf{h}_0 = [1, 0.2, 0.5]^T$ and $\mathbf{w}_k \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}_3)$. The dataset is denoted "RW_σ_w^2_N_N_s_N_p". An illustration of the CIR evolution is given in Fig. 3. As we can observe, the three coefficients of the CIR are set to their initial value $[1, 0.2, 0.5]$ every new packet i.e. every 100 symbols. Inside a

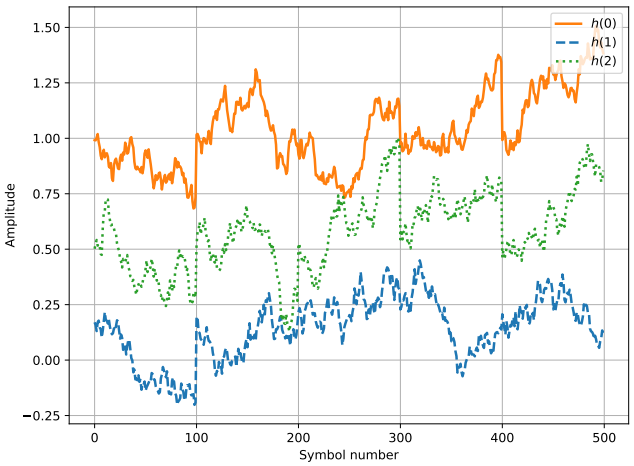

Fig. 3. Example of random walk CIR evolution with $\sigma_w^2 = 10^{-3}$.

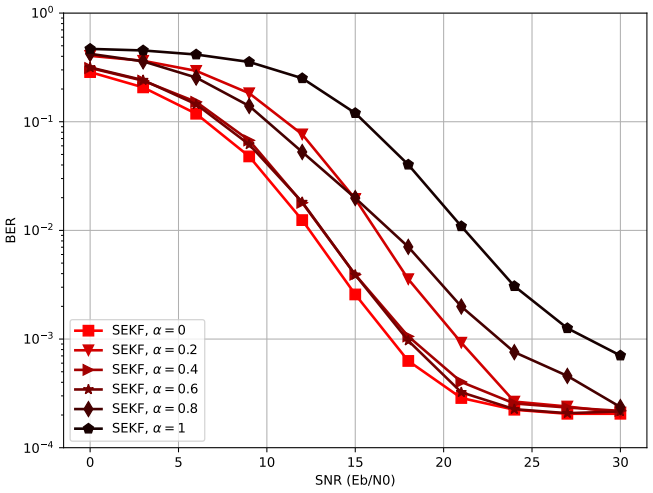

Fig. 4. BER comparison performance computed on RW_10^{-3}_4000_100_5 for different α values during at SNR = 30 dB the training of the neural network.

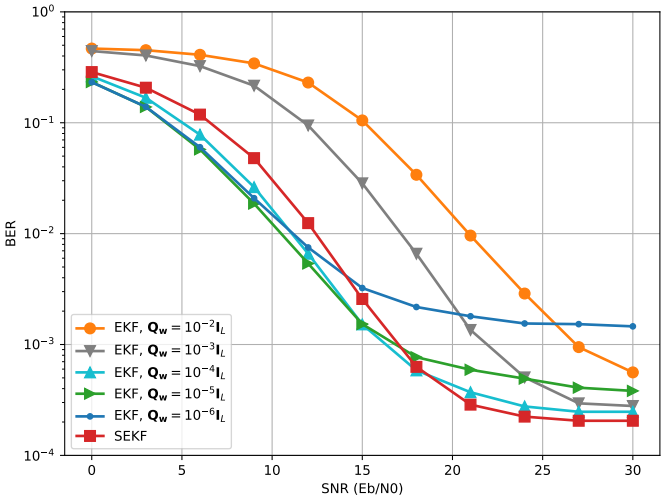

Fig. 5. BER comparison performance computed on RW_10^{-3}_4000_100_5 between standard EKF with constant $\mathbf{Q_w}$ matrix versus the proposed SEKF trained at high SNR.

packet, the CIR evolves basically like a random walk. Many trainings were computed with many different parameters. The loss function applied during the training process is:

$$L(\hat{\mathbf{h}}, \mathbf{h}, \hat{\mathbf{b}}, \mathbf{b}) = (1-\alpha)\text{MSE}(\hat{\mathbf{h}}, \mathbf{h}) + \alpha\text{BCE}(\hat{\mathbf{b}}, \mathbf{b}) \quad (18)$$

with $\mathbf{b}$ and $\hat{\mathbf{b}}$ the received bits and the sent ones, respectively. The parameter α allows to give more importance to the BER than the CIR estimation during the training. The BCE is the binary cross entropy. The optimizer used was Adam with a learning rate of 10^{-3}. The obtained results are presented in the next section.

IV. Results

The different simulation results are presented in this section. Trainings of the neural network were computed at 30 dB, with $R = 10^{-3}$ on MW_10^{-3}_100_100_5 dataset. The impact of the α parameter was studied with different trainings with α in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and BER comparison is presented in Fig. 4. First of all, we observe at high SNR a floor around 2×10^{-4}. This floor appears because with a random walk CIR evolution of variance 10^{-3}, sometimes $h(0)$ is close to 0, and without any delay it is not possible to estimate the correct symbol with certainty. The neural network could theoretically be trained with some delay to avoid this issue, but we focused on an instantaneous decision. We observe in general the neural networks trained with a low α value provide better performance than those trained with a higher α value. Particularly, the neural network trained with $\alpha = 0$ provides better performance than the others at low SNR, and is very close to the best ones at very high SNR. For this reason, this neural network has been kept in the following. However, taking into account the BER and the MSE on the CIR during the training was natural in view of the state vector, and maybe with other neural network architecture a non-zero α value would be preferable. The BER comparison between the SEKF and standard EKFs is show in Fig. 5. We selected few natural standard $\mathbf{Q_w}$ values typically hand tuned to do the comparison. We observe that the proposed SEKF offers the best performance over all conventional EKFs from 18 dB upwards. The performance at low SNR does not reach the one provides by some standard EKF because the training was done at high SNR. Some trainings were made at a lower SNR of 10 dB but the loss did not converge and the neural network did not learn effectively.

V. Conclusion and Perspectives

In this paper, we proposed a hybrid architecture composed of EKFs and a GRU based neural network. The EKFs allow to jointly estimate and equalize the channel impulse response, and the neural network tries to predict the best model noise covariance matrix at each symbol time. Different neural network configurations were tested with different inputs, and with different hyper parameters. The neural network did well learn on the selected dataset, and the obtained architecture provides better performance from 18 dB forwards than any classical blind EKF equalizer structure. In future works, it could be interesting to add some delay during the training to adapt the work for larger CIR evolution models, such as Rayleigh fading channels. In addition, work on neural network compression is underway to efficiently implement such architectures on embedded systems for real-time applications.

References

[1] S. Abrar and A. K. Nandi, "Blind equalization of square-qam signals: a multimodulus approach," *IEEE Transactions on Communications*, vol. 58, no. 6, pp. 1674–1685, 2010.

[2] M. Paskov, D. Lavery, and S. J. Savory, "Blind equalization of receiver in-phase/quadrature skew in the presence of nyquist filtering," *IEEE Photonics Technology Letters*, vol. 25, no. 24, pp. 2446–2449, 2013.

[3] P. Banelli, R. C. Cannizzaro, and L. Rugini, "Data-aided kalman tracking for channel estimation in doppler-affected ofdm systems," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 3, 2007, pp. III–133–III–136.

[4] K. Murakami, K. Ueda, M. Takano, and T. Fujino, "Design of an adaptive kalman equalizer and its performance over fading multipath channels," in *40th IEEE Conference on Vehicular Technology*, 1990, pp. 564–701.

[5] B.-S. Chen and J.-F. Liao, "Spc06-1: Robust blind equalizer over fast time-varying isi fading channels," pp. 1–6, 2006.

[6] R. Amara and S. Marcos, "A blind network of extended kalman filters for nonstationary channel equalization," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 4, 2001, pp. 2117–2120 vol.4.

[7] M. Ben Mabrouk, "PA efficiency enhancement using digital linearization techniques in uplink cognitive radio systems," Theses, Université de Bordeaux, Dec. 2015. [Online]. Available: https://theses.hal.science/tel-01290393

[8] Z. Kamran, T. Kirubarajan, and A. Gershman, "Blind estimation and equalization of time-varying channels using the interacting multiple model estimator," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 5, 2004, pp. V–V.

[9] H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari, "Long short-term memory kalman filters: Recurrent neural estimators for pose regularization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5525–5533.

[10] C. Chen, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, "Dynanet: Neural kalman dynamical model for motion estimation and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5479–5491, 2021.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Siebert, G. Ferré, B. Le Gal, and A. Fourny, "The smart kalman filter: A deep learning-based approach for time-varying channel estimation," in *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2023, pp. 1–6.

[13] X. Li and T. F. Wong, "Turbo equalization with nonlinear kalman filtering for time-varying frequency-selective fading channels," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 691–700, 2007.

[14] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014. [Online]. Available: https://arxiv.org/abs/1409.1259