Abstract: the recovery of the transmitted signals, propagated through wireless communication channels, is a complicated process due to the distortion and interference impairments in such random channels. However, the channel effect can be compensated by using channel estimation techniques performed at the receiver. Since most of channel estimators are operated in the frequency domain, i.e., inverse modeling, the Ball’s adaptive channel estimation scheme, which was invented by Michael J. Ball, was not considered in the literature as an attractive approach due to its time-based and direct modeling features. However, such features are favorable for time-varying channels. To the best of author’s knowledge, the performance of the Ball’s scheme for channel estimation has not been investigated in the context of OFDM system with fading channels. Therefore, in this paper, we propose to incorporate Ball’s method for channel estimation in OFDM receivers and we also investigate the performance of this scheme using variations of the Recursive Least Squares (RLS)-type algorithms, namely QR Decomposition RLS (QRD-RLS), the Householder RLS (HH-RLS) and the Sliding Window Householder RLS (SWHH-RLS). Our numerical results indicate that QRD-RLS and HH-RLS outperform traditional RLS in terms of bit error rate.

Keywords: Ball’s scheme, adaptive channel estimation, recursive least square, QR-decomposition, householder, sliding window RLS, OFDM.
by allowing the receiver side to obtain information about
the channel in terms of Channel State Information (CSI)
[1], [2]. Typically, the CSI is not is not known at the
receiver and also changing continuously due to the changing
environment. Therefore, adaptive channel estimators are
deployed at the receiver side to track the time-varying
channel via the utilization of adaptive algorithms. In more
details, the CSI is estimated using pilots symbols in the
training phase. Afterward, data symbols can be transmitted
reliably during transmission phase benefiting from the CSI
obtained in the first phase [8], [9].

Traditional channel estimators work by estimating the
Channel Frequency Response (CFR) [10]. First, pilot-based
estimation of CFRs for some training symbols is performed
[11], [12]. Then, CFRs of data symbols are estimated
using the adaptive algorithms and techniques [13], [14],
[15], [16]. For instance the work [15], [16] considers
pilot-assisted estimator which operated with the maximum
likelihood (ML) and the Bayesian minimum mean square
error (MMSE) techniques. Alternatively, with advent of
Machine Learning (ML) algorithms and its applications
in signal processing and communications, several work have
shown the effectiveness of ML-based approaches, like deep
learning [17], [18], [19], [20] and reinforcement learning
[21], in performing adaptive channel estimation.

Along with the increase importance of Multiple-Input
Multiple-Output (MIMO) in achieving higher capacity for
fading channels and also enhancing system throughput,
MIMO-OFDM receivers become standard in many wireless
applications [22], [23]. However, the Single-Input Single-
Output (SISO) channel estimation techniques discussed
above can not be applied directly to MIMO-OFDM systems.
As a results, several work address this problem. The work in
[24] relies on Discrete Fourier Transform (DFT) to derive
the set of training signals which minimize the estimator
MSE. A blind and semi-blind MIMO channel estimators
are proposed in [25] and [26], [27], respectively.

The literature summarized above consider mainly a
frequency-domain channel estimation for OFDM system.
A relevant, but less interesting, line of work have dealt with
the time-domain based channel estimation as in [28] where
a LMMS estimator is proposed with pilot-data multiplexing
and compared to the scheme in [29] which relies on
pilots tones. The work in [30] shows that Kalman filtering-
based channel estimation is suitable to track and estimate
fading channels. A comparison between time-domain and
frequency-domain channel estimation for OFDM system are
discussed in [31], [32]. Even with this line of work, the
channel estimation scheme for multipath communications
proposed in [4], which was invented by Michael J. Ball,
did not receive much attention in the literature. This is
because most of channel estimators are performed in the
frequency domain, i.e., inverse modeling (see Fig. 1-(a)).
On the contrary, the Ball’s channel estimation scheme tries
to equalize the communications channel directly (see Fig. 1-
(b)) as the inverse of the channel frequency response is not
required. This time-based and direct modeling features of
the Ball’s scheme are favorable for modeling, i.e., estimat-
ing multipath and rapidly time-varying channels like intelli-
gent reflecting surface (IRS)-enabled radio links [33], [34].
Unlike previous work, in this paper, we proposes to leverage
Ball’s-based channel estimation scheme in OFDM systems
using different types of RLS algorithm. The standard RLS
algorithm have been applied in several adaptive filtering
applications [35], [4], [8], [36]. Most of these applications
show a performance gain under different channel conditions,
but they incur extra complexity in computations and raise
some stability issues. Therefore, we will investigate the
performance of the proposed scheme under different types
of the RLS algorithm. In details, the major contributions
of this work are:

- Conduct numerical experiments and select the suit-
able set of algorithms that best fit the proposed
scheme.
- Investigate the performance of the proposed approach
using, in addition to conventional RLS, the QR De-
composition RLS (ORD-RLS), the Householder RLS
(HH-RLS) and the Sliding Window Householder RLS
(SWHH-RLS). It is the first time here that this set of
RLS algorithms is used in an OFDM receivers as well
as in investigating the performance of the proposed
scheme.
- Design an OFDM system receiver by utilizing the
Ball’s method for channel estimation.

The rest of the paper is organized as follows. The family
of RLS-type algorithms are explained in Section 3 after
presenting the proposed scheme in Section 2. The simula-

tion results are shown in Section 4 Finally, conclusions are
outlined in Section 5.

**Notations:** We use lowercase bold letters to denote vec-
tors while matrices are denoted by uppercase bold letters.
For a matrix \( A \), the notations \( A^{-1}, A^T \) and \( A^H \) denote
its inverse, transpose, and Hermitian transpose respectively.
Scalars are denoted by lowercase letters.

### 2. Ball’s Based Adaptive Channel Estimation

Apart from conventional channel estimation schemes,
the Ball’s scheme brings two unique features: channel
The impulse response is estimated in time domain, and also the ability to perform direct channel equalization since channel frequency response inversion is not required (see Fig. 1). The actual operation of this adaptive scheme is described next.

As illustrated in Fig. 2, the transmitted signal is produced as Pseudo Random (PN) sequence of data. At the receiver, the channel estimator, or adaptive filter, will tune its coefficients based on the Least-Square (LS) of the error, or the difference between output of the filter and the output of the channel. As a reference, the filter uses synchronized and local version of the PN sequence. The optimal weight of the adaptive filter in Fig. 2 is the well-known Wiener solution formulated as

\[ h(n) = F^{-1}(n)c(n), \]  

where \( F \) is the correlation matrix of input data \( U \) as given in

\[ F(n) = \sum_{i=0}^{n} \tau^{n-i}u(i)u(i)^H = U^H(n)U(n), \]

while \( c \) is the vector denoting the cross correlation of desired response \( g \) and input data which is formulated as

\[ c(n) = \sum_{i=0}^{n} \tau^{n-i}g(i)u(i) = U^H(n)g(n), \]

with \( \tau \) being the forgetting parameter. The optimal Wiener filter in (1) is obtained as the solution that minimizes the Mean Square Error (MSE) objective function written as

\[ e(n) = \mathbb{E} \left[ e^2(n) \right] = \mathbb{E} \left[ |g(n) - h^T(n)u(n)|^2 \right]. \]

In every time epoch, denoted as \( n \) in (1)-(4), the components in the weight vector \( h(n) \) are updated according to the adopted algorithm. The set of algorithms considered in this paper are presented in the next section.

3. **Adaptive RLS-Type Algorithms**

In this section, we will give a detailed description for each algorithm considered in this work, namely RLS, QRD-RLS, HH-RLS and SWHH-RLS.

A. **Recursive-Least-Squares (RLS) algorithm**

Despite the fact that RLS algorithm has rapid convergence feature, it is still requires high computational complexity and can experience some stability issues. To proceed, we rewrite correlation matrix \( F \) and the vector of cross correlation \( c \) defined in (2) and (3), respectively, as

\[ F(n) = u(n)u^H(n) + \tau F(n-1) \]

Figure 2. Block diagram of the adaptive Ball’s-based scheme for channel estimation during actual data transmission [4].
and
\[ e(n) = u(n)g(n) + \tau e(n-1). \] (6)

By plugging (5) and (6) in (1), we obtain the following weight update formula

\[ h(n) = h(n-1) + e(n)F^{-1}(n)u(n), \] (7)

where the error vector e(n) and F\(^{-1}\) are calculated as shown in Algorithm 1. This table summarizes the steps required for the RLS algorithm with Γ(n) and Λ(n) being an auxiliary vectors, while τ and ε are forgetting and gain factors, respectively.

**Algorithm 1: RLS algorithm**

**Initialization:** 0 \( \ll \tau < 1 \), \( \epsilon \approx 1/\sigma_{\text{in}}^2 \), \( F^{-1}(0) = \epsilon I \)

**for all** \( n \) **do**

1. \( e(n) = g(n) - h^H(n-1)u(n) \)
2. \( Λ(n) = F^{-1}(n-1)u(n); \)
3. \( Γ(n) = \frac{Λ(n)}{\tau + u^H(n)Λ(n)} \)
4. \( F^{-1}(n) = \frac{1}{\tau}[F^{-1}(n-1) - \frac{Γ(n)Λ(n)}{\tau + u^H(n)Λ(n)}] \)
5. \( h(n) = h(n-1) + e(n)Γ(n) \)

**end**

B. QR decomposition (QRD-RLS) algorithm

To avoid the stability problem of RLS procedures mentioned earlier, different approach with enhanced numerical stability is offered by the QRD-RLS algorithm. This implementation relies essentially on QR factorization for the matrix of input data along with back substitution procedures [4], [35]. To proceed, we first note that the matrix \( u(n) \) has dimensions \( (n+1) \times (N+1) \) which dictate that the matrix order depends on the number of iterations. Accordingly, the QRD-RLS convert \( u(n) \) into a triangular matrix \( T(n) \) of order \( (N+1) \times (N+1) \) via an orthogonal matrix \( Θ(n) \) of order \( (n+1) \times (n+1) \) such that [4], [35]

\[ Θ(n)u(n) = \begin{bmatrix} 0 \\ T(n) \end{bmatrix}. \] (8)

where \( Θ_{0 \rightarrow N \times (N+1)} \) denotes an array of zeros with unitary matrix \( Θ(n) \) represents the triangularization steps. Based on (8), we can rewrite \( U(n) \) as

\[ U(n) = Θ^H(n)T(n), \] (9)

where \( Θ^H(n) \) is a unitary matrix partition with size \( (n+N) \times (n+1) \). Thus, we rewrite \( F(n) \) as

\[ F(n) = U^H(n)U(n) = T^H(n)T(n), \] (10)

where \( T(n) \) denotes the Cholesky factor of (10). The rotated error vector \( (n+1) \times 1 \) is expressed as

\[ e_q(n) = Θ(n)e(n) = \begin{bmatrix} e_{q1}(n) \\ e_{q2}(n) \end{bmatrix} = \begin{bmatrix} g_q1(n) \\ g_q2(n) \end{bmatrix} - \begin{bmatrix} 0 \\ T(n) \end{bmatrix}h(n). \] (11)

The weight vector \( h(n) \) is selected so as the quantity in (11) is optimized, i.e., the weighted-square error function is minimized. It follows that by choosing

\[ h(n) = T^{-1}(n)g_q2(n), \] (12)

the term \( T^{-1}(n)g_q2(n) \) in (11) will vanish. The updating steps of QRD-RLS algorithm are illustrated in Algorithm 2.

**Algorithm 2: QRD-RLS algorithm**

**for all** \( n \) **do**

1. Computing \( Θ(n) \) and updating \( T(n) \):
   \[ \Theta(n) = Θ(n) \]
   \[ T(n) = \sqrt{T(n-1)} \]
2. Computing \( γ(n) \): \( γ(n) = \frac{1}{N} \sum_{i=1}^{N} \cos θ_i(n) \)
3. Computing \( e_{q1}(n) \) and \( g_q2(n) \):
   \[ e_{q1}(n) = Θ(n)g_q2(n) \]
   \[ g_q2(n) = \sqrt{T(n-1)} \]
4. Computing \( e_{q1}(n) \):
   \[ e_{q1}(n) = e_{q1}(n)γ(n) \]

**end**

C. Householder (HH-RLS) algorithm

Most of QRD-RLS implementations utilize Gram-Schmidt process and Givens rotation method to orthogonalize the data correlation matrix, but none of these approaches is computationally efficient. Instead, the Householder transformation is a rank-\( n \) update approach with orthogonal matrix transform. Consequently, HH-RLS requires less computations and provides numerical stability even for sparse data structure. The key property of this algorithm is that, at each iteration, the Cholesky factor of \( F \) is updated. Define the square matrix \( Z(n) \) of order \( N+1 \) to be factor for the square root of \( F \), then we can write

\[ F(n) = Z^H(n)Z(n). \] (13)

Next, we define

\[ b(n) = \frac{Z^{-T}(n-1)u(n)}{\sqrt{τ}}, \] (14)

where \( τ \) denotes the forgetting parameter. We define the orthogonal Householder matrix \( a(n) \) of order \( (N+2) \times (N+2) \) [4], [35], to formulate

\[ a(n) \begin{bmatrix} 1 \\ b(n) \end{bmatrix} = \begin{bmatrix} -ζ(n) \\ 0 \end{bmatrix}. \] (15)
with $\zeta(n)$ being a positive scalar. It is left to mention that HH-RLS algorithm computes error and weight vectors according to the formulas

$$e(n) = g(n) - h^H(n-1) u(n)$$

(16)

with

$$h(n) = h(n-1) - \frac{e(n)}{\zeta(n)} \nu(n),$$

(17)

respectively, with $\nu(n)$ being the scaled gain vector. The complete description of HH-RLS is given in the following algorithm.

**Algorithm 3:** HH-RLS algorithm

- Initialization: $0 < \tau < 1$, $\epsilon \approx 1/\sigma^2_q$, $h(0) = 0$,
- $Z^{-1}(0) = \sqrt{\epsilon I}$

for all $n$ do

- Computing $b(n)$ using (14)
- Finding $a(n)$ and $\zeta(n)$: $a(n) \begin{bmatrix} 1 \\ b(n) \end{bmatrix} = \begin{bmatrix} -\zeta(n) \\ 0 \end{bmatrix}$
- Computing $Z^{-1}(n-1)$:
- $a(n) T^{-1/2} Z^{-H}(n-1) = \begin{bmatrix} \nu(n) \\ Z^{-H}(n) \end{bmatrix}$
- Computing $e(n)$ using (16) and $h(n)$ using (17)
end

**D. Sliding Window Householder (SWHH-RLS) Algorithm**

For efficient computations of HH-RLS, the complexity can be reduced by applying a sliding window (SW) on the input which organized into blocks of size $Q$. In a mathematical formulation, we write [4], [35]

$$e(n) = \begin{bmatrix} e_n \\ e_{n-1} \\ \vdots \\ e_{n-L+1} \end{bmatrix} = g(n) - U(n) h(n),$$

(18)

where $L$ is the size of the window with $U(n) = [U_n U_{n-1} \ldots U_{n-L+1}]^H$ and $g(n) = [g_n g_{n-1} \ldots g_{n-L+1}]^H$. To this end, We have the following definitions:

$$\bar{U}(n) = \begin{bmatrix} U_{n}^H \\ \vdots \\ U_{n-L}^H \end{bmatrix} = \begin{bmatrix} U_{n-L} \\ U(n-1) \end{bmatrix},$$

(19)

and

$$\bar{g}(n) = \begin{bmatrix} g_{n} \\ \vdots \\ g_{n-L} \end{bmatrix} = \begin{bmatrix} g_{n} \\ g(n-1) \end{bmatrix}.$$  

(20)

The SWHH-RLS requires a two-step procedure for solving LS problem: a weight-update step and a down-date step. To this end, we can write for the weight-update step $h(n-1) \rightarrow \hat{h}(n) \rightarrow h(n)$, with $\hat{h}(n)$ being the solution of the LS problem obtained by replacing (19) and (20) in (18), respectively for $U(n)$ and $g(n)$. For down-date part, we write [4], [35]

$$F(n) = \hat{F}(n) - U_{n-L} U_{n-L}^H.$$  

(21)

Using Cholesky factors, equation (21) reads

$$T(n) T(n) = T(n) T(n) - U_{n-L} U_{n-L}^H.$$  

(22)

The hyper normal matrix $H(n)$ reads

$$H(n) = \begin{bmatrix} U_{n-L}^H \\ T(n) \end{bmatrix} = \begin{bmatrix} O_{Q \times (N+1)} \\ T(n) \end{bmatrix}.$$  

(23)

which is used to calculate $T^{-1}(n)$ from $\hat{T}^{-1}(n)$ using inverse Cholesky factor as [4], [35]

$$H(n) = \begin{bmatrix} O_{Q \times (N+1)} \\ \hat{T}^{-1}(n) \end{bmatrix} = \begin{bmatrix} V^H(n) \\ \hat{T}^{-1}(n) \end{bmatrix},$$  

(24)

where $V(n)$ has dimensions $(N + 1) \times Q$. Let us define the vector $\psi(n) = -\hat{T}^{-1}(n) U_{n-L}$, then a hyper normal matrix $H(n)$ is obtained in the form

$$H(n) = \begin{bmatrix} \bar{A}(n) \\ \psi(n) \end{bmatrix} = \begin{bmatrix} \bar{\Lambda}(n) \\ O_{(N+1) \times Q} \end{bmatrix},$$  

(25)

with $\bar{\Lambda}(n)$ is a square matrix and we defined $T(n), \psi(n)$ and $\Lambda(n)$ as in (19). Finally, the weight update is calculated recursively according to [4], [35]

$$h(n) = \hat{h}(n) + V(n) \bar{\Lambda}^{-1}(n) (\bar{g}_{n-L} - U_{n-L}^H \hat{h}(n)).$$  

(26)

The complete steps of SWHH-RLS algorithm is summarized in Algorithm 4. Since the algorithm applies a window of size $L$ on the input data and also the fact that the two-step procedures in Algorithm 4 are executed for $n > L$, it is therefore required to run the weight-update initialization $L$ times and afterward the algorithm will transit to the two-step method discussed above.

**4. Numerical Results**

In this section, we present simulation results to assess the validity of the suggested Ball’s based OFDM receiver, as shown in Fig. 3, using the four algorithms discussed in Sec. 3-A through Sec. 3-D. To this end, we consider time varying multipath Rayleigh fading channel with four paths specified with gains $G = [0 - 3 - 7 - 9]$ dB and delays $D = [0; 0.75; 1.66; 2.94] \mu s$ for LOS and the three NLOS respectively [37]. The FFT size is 64 with $CP = 16$ samples. The carrier frequency is $f_c = 800$ MHz and the sampling rate is $f_s = 6.78$ MHz at channel bandwidth $BW = 20$ MHz. The comb pilot is based on Superimposed Training Sequence (STS) technique used in [9]. We select the length of RLS filter as 80 and also Doppler shift as $f_d = 60$ Hz.
Figure 3. OFDM system using the proposed channel estimation scheme with CP and STS pilot. The blocks labeled as P/S and S/P denote parallel to serial and serial to parallel conversion, respectively.

Algorithm 4: SWHH-RLS algorithm

for all \( n > L \) do

Step 1: Weight-update
Computing: \( \psi(n) = -T^{-H}(n-1)U_n \)
Computing \( \Theta(n) \):
\[
\begin{bmatrix}
\Theta(n)
\end{bmatrix}
= \begin{bmatrix}
I_Q
\end{bmatrix}

\begin{bmatrix}
\Delta(n)
\end{bmatrix}
= \begin{bmatrix}
O_{Q(N+1)\times Q}
\end{bmatrix}

T^{-H}(n-1)
= \begin{bmatrix}
E_Q(n)
\end{bmatrix}

\begin{bmatrix}
T^{-1}(n-1)
\end{bmatrix}

Updating: \( \Theta(n) \):
Computing:
\[
\tilde{h}(n) = h(n-1) - E(n) \tilde{A}^{-H}(n)(g^*_n - U^H(n)h(n-1))
\]
Step 2: Down-date
Computing \( \tilde{\psi}(n) \):
\[
\tilde{\psi}(n) = -T^{-H}(n)U_{n-L}
\]
Computing \( H(n) \) using (24)
Down-dating \( T^{-1}(n) \):
\[
H(n) \begin{bmatrix}
O_{Q(N+1)}
\end{bmatrix}
T^{-H}(n)
= \begin{bmatrix}
V^H(n)
\end{bmatrix}
T^{-H}(n)
\]
Updating:
\[
h(n) = h(n) - V(n) \tilde{A}^{-H}(n) (g^*_n - U^H(n)\tilde{h}(n))
\]
end

Unless otherwise stated, we set the all initial coefficients (including \( h(0) = 0 \)) to zeros.

The Bit Error Rate (BER) versus the signal-to-noise ratio, in terms of \( E_b/N_0 \), is plotted in Fig. 4 for BPSK modulation scheme. It is seen that QRD-RLS and HH-RLS algorithms provide better performance compared to traditional RLS. For instance, QRD-RLS provides performance gains of 15 dB and 8 dB compared to SWHH-RLS and RLS, respectively, at BER = 0.1. This improvement can be attributed to the fact that QRD-RLS and HH-RLS eliminate more errors compared to RLS due to the numerical stability and the reliable computation of the recursive least square problem.

Fig. 5 illustrates the Mean Square Error (MSE) versus the number of iterations required for the same algorithms outlined in Fig. 4. We first observe that the QRD-RLS algorithm, which provided the best BER performance in Fig. 4, have slower convergence rate that requires about 2000 iterations to reach MSE less than 0.2, compared to the conventional RLS algorithm. This is a direct result from the fact that this algorithm is stable and yield better performance at the cost of having more computational
complexity. The figure also confirms that RLS and HH-RLS reach typical convergence with less than 500 iterations.

Figure 5. MSE versus iteration number for BPSK modulation.

Figure 6. Numerical comparison of MSE of the four algorithms considered in Fig. 5 using Bar graph. The figure shows MSE after 1, 100 and 500 iterations with BPSK modulation.

To provide more insight about the convergence behavior of the proposed scheme, we present in Fig. 6 a numerical comparison of the MSE, which is computed numerically from (4), for the considered algorithms after 1, 100 and 500 iterations. Inline with the results from Fig. 5, this figure confirms the observation that QRD-RLS yields the worst MSE performance compared to the other algorithm which reaches moderate MSE ($\leq 0.4$) after 500 iterations. In addition, RLS and HH-RLS maintain the least MSE performance during iterations. The latter case is attributed to the fact that the rapid convergence comes at the cost of having higher BER performance as seen in Fig. 4.

Finally, in order to shed light on the effect of the modulation type on the suggested scheme, Fig. 7 adopt a 16-QAM modulation scheme for the same set-up studied in Fig. 4. Numerically, it is noted that the performance of QRD-RLS outperforms HH-RLS and RLS by 2 dB and 10 dB, respectively, at BER = 0.2. This improvement is attributed to the same reasons discussed in the comments on Fig. 4.

Figure 7. BER versus SNR for 16-QAM modulation.

5. CONCLUDING REMARKS

This paper considers the evaluation of a novel OFDM receiver, that incorporate Ball’s adaptive channel estimation scheme, over fading channels. The performance evaluation, as measured in terms of BER and MSE learning curve, illustrates that RLS and HH-RLS algorithms show faster convergence rate and low miss-adjustment at steady states compared to other RLS-type algorithms, while QRD-RLS algorithm achieves the best BER performance. As a final remark, the HH-RLS strikes a good balance in providing moderate BER performance gain while maintaining acceptable convergence properties. Among the future directions for this study are the incorporation of the proposed scheme in massive MIMO OFDM systems.

REFERENCES


Ali Al-Shuwaili Ali Al-Shuwaili gained a Ph.D. degree in electrical engineering from New Jersey Institute of Technology (NJIT), Newark 07012, NJ, USA. Before that, he received the M.Sc. and B.Sc. degrees in communication engineering from Electrical and Electronic Engineering (EEE) department at University of Technology, Baghdad, Iraq, in 2005 and 2008, respectively. He worked in telecommunication industry with ZTE and mSolutions companies and currently he holds an active research position and teaching position at UOITC. His research interests are in wireless communication, optimization and machine learning.

Mustafa Dh. Hassib Mustafa Dh. Hassib was born in Baghdad, Iraq. He received his B.Eng. degree in Electronics and Communications engineering from the University of Technology Iraq, in 1991 and his M.Sc. degree in Communications engineering from the Military engineering college at Baghdad, Iraq in 2004 and the Ph.D. degree in mobile wireless communications from the National University of Malaysia (UKM), in 2015. He has held various positions in the Department of communications at the University of Technology Iraq. His research interests take account of mobile wireless communication, Optical communication and coding theory.

Thamer M. Jamel Thamer M. Jamel graduated from the University of Technology with a Bachelor’s degree in electronics engineering. He received a Master’s and a Doctoral degree in communication engineering in 1997. His scientific degree is Professor and currently, he is one of the staff of the communication engineering department at University of Technology, Baghdad, Iraq.

His Research Interests are Adaptive Digital Signal Processing (Algorithms and Applications) for modern and future Communications system.